



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

**H.263**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

(03/96)

**TRANSMISSION DE SIGNAUX  
NON TÉLÉPHONIQUES**

---

**CODAGE VIDÉO POUR COMMUNICATIONS  
À FAIBLE DÉBIT**

**Recommandation UIT-T H.263**

(Antérieurement «Recommandation du CCITT»)

---

## AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1<sup>er</sup>-12 mars 1993).

La Recommandation UIT-T H.263, que l'on doit à la Commission d'études 15 (1993-1996) de l'UIT-T, a été approuvée le 19 mars 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

---

### NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

## TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application.....	1
2	Références.....	1
3	Brève spécification.....	1
3.1	Entrée et sortie vidéo.....	1
3.2	Sortie et entrée numérique.....	1
3.3	Fréquence d'échantillonnage.....	1
3.4	Algorithme de codage de la source.....	2
3.5	Débit.....	3
3.6	Stockage temporaire des données.....	3
3.7	Symétrie de transmission.....	3
3.8	Traitement des erreurs.....	3
3.9	Fonctionnement multipoint.....	4
4	Codeur de source.....	4
4.1	Format de la source.....	4
4.2	Algorithme de codage de la source vidéo.....	5
4.3	Contrôle du débit des données.....	8
4.4	Rafraîchissement forcé.....	8
4.5	Alignement par les octets des codes de déclenchement.....	8
5	Syntaxe et sémantique.....	8
5.1	Couche des images.....	9
5.2	Couche des groupes de blocs.....	12
5.3	Couche des macroblocs.....	13
5.4	Couche des blocs.....	18
6	Processus de décodage.....	18
6.1	Compensation de mouvement.....	18
6.2	Décodage des coefficients.....	24
6.3	Reconstruction des blocs.....	25
	Annexe A – Spécification de la précision de la transformée inverse.....	26
	Annexe B – Décodeur fictif de référence.....	27
	Annexe C – Considérations relatives aux communications multipoints.....	28
	C.1 Demande de gel d'image.....	28
	C.2 Demande de rafraîchissement rapide.....	28
	C.3 Fin de gel d'image.....	28
	C.4 Communication multipoint à présence continue (CPM) (continuous presence multipoint) (mode non utilisé pour la Recommandation H.324).....	29
	Annexe D – Considérations relatives aux communications multipoints.....	29
	D.1 Vecteurs cinétiques pointant hors des limites d'image.....	29
	D.2 Extension de la gamme des vecteurs cinétiques.....	30
	Annexe E – Mode de codage arithmétique syntaxique (SAC).....	30
	E.1 Introduction.....	30
	E.2 Spécification du codeur en mode SAC.....	31
	E.3 Spécification du décodeur en mode SAC.....	31
	E.4 Syntaxe.....	32

	<i>Page</i>
E.5	Registre PSC_FIFO ..... 33
E.6	Symboles de longueur fixe ..... 33
E.7	Symboles de longueur variable..... 33
E.8	Modèles de codage SAC..... 34
Annexe F	– Mode de codage arithmétique syntaxique (SAC) ..... 36
F.1	Introduction ..... 36
F.2	Quatre vecteurs cinétiques par macrobloc ..... 36
F.3	Compensation de mouvement par superposition pour la luminance ..... 38
Annexe G	– Mode de codage par trames PB..... 40
G.1	Introduction ..... 40
G.2	Trames PB et blocs INTRA ..... 40
G.3	Couche des blocs ..... 41
G.4	Calcul des vecteurs pour l'image B dans une trame PB ..... 41
G.5	Prédiction d'un bloc de type B dans une trame PB ..... 41
Annexe H	– Correction d'erreur directe pour signal vidéo codé ..... 43
H.1	Introduction ..... 43
H.2	Verrouillage de trame de correction d'erreur ..... 43
H.3	Code de correction d'erreur..... 43
H.4	Temps de rétablissement pour le verrouillage de trame de correction d'erreur ..... 44

## RÉSUMÉ

La présente Recommandation spécifie une représentation codée que l'on peut utiliser pour comprimer la composante d'images animées de services audiovisuels transmis à faible débit. La configuration de base de l'algorithme de source vidéo s'appuie sur la Recommandation H.261; elle repose conjointement sur la prédiction interimage afin d'utiliser la redondance temporelle et sur le codage par transformée du signal résiduel afin de réduire la redondance spatiale. Le codeur de la source peut traiter cinq formats d'image normalisés: sub-QCIF, QCIF, CIF, 4CIF et 16CIF.

Le décodeur est doté de la capacité de compensation de mouvement, ce qui permet, le cas échéant, d'incorporer cette technique au codeur. La compensation de mouvement utilise une précision d'un demi-pixel, contrairement à la Recommandation H.261 qui utilise une précision d'un pixel entier ainsi qu'un filtre en boucle. La transmission des symboles utilise un codage de longueur variable.

Outre l'algorithme de base pour le codage de la source vidéo, cette Recommandation décrit quatre options de codage négociables, destinées à améliorer la qualité de fonctionnement: vecteurs cinétiques non restreints, codage arithmétique syntaxique, prédiction avancée et trames PB. Toutes ces options peuvent être retenues conjointement ou séparément.



## CODAGE VIDEO POUR COMMUNICATIONS À FAIBLE DEBIT

(Genève, 1996)

### 1 Domaine d'application

La présente Recommandation spécifie une représentation codée qui peut être utilisée pour comprimer les composantes d'images animées de services audiovisuels à faibles débits. La configuration de base de l'algorithme de codage vidéo à la source est fondée sur la Recommandation H.261. Quatre options de codage, négociables, sont présentées afin d'améliorer les performances.

### 2 Références

Les Recommandations et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute Recommandation ou autre référence est sujette à révision; tous les utilisateurs de la présente Recommandation sont donc invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références indiquées ci-après. Une liste des Recommandations UIT-T en vigueur est publiée régulièrement.

- [1] Recommandation UIT-T H.223 (1996), *Protocole de multiplexage pour communications multimédias à faible débit binaire.*
- [2] Recommandation UIT-T H.242 (1996), *Procédures permettant des communications entre trois terminaux audiovisuels à l'aide de canaux numériques dont le débit peut aller jusqu'à 2 Mbit/s.*
- [3] Recommandation UIT-T H.245 (1996), *Protocole de commande pour communications multimédias.*
- [4] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à  $p \times 64$  kbit/s.*
- [5] Rec. UIT-T H.262 (1995) | ISO/CEI 13818-2:1995, *Technologies de l'information – Codage générique des images animées et du son associé: données vidéo.*
- [6] Recommandation UIT-T H.320 (1996), *Systèmes et équipements terminaux à bande étroite incluant la visioconférence et la visiophonie.*
- [7] Recommandation UIT-T H.324 (1996), *Terminaux pour communications multimédias à faible débit binaire.*

### 3 Brève spécification

Un diagramme synoptique du codec est représenté à la Figure 1.

#### 3.1 Entrée et sortie vidéo

De façon qu'une seule Recommandation puisse s'appliquer aux régions utilisant les normes de télévision à 625 et à 525 lignes, le codeur de source travaille sur des images conformes au format intermédiaire commun (CIF) (*common intermediate format*). Ne sont pas soumises à la présente Recommandation les normes des signaux de télévision à l'entrée et à la sortie (ces signaux pouvant être, par exemple, composites ou en composantes, analogiques ou numériques) ainsi que les méthodes permettant d'effectuer l'éventuelle conversion à destination/en provenance du format de codage source.

#### 3.2 Sortie et entrée numérique

Le codeur vidéo fournit un flux binaire numérique autonome qui peut être combiné avec d'autres signaux multiservice (comme ceux qui sont définis dans la Recommandation H.223). Le décodeur vidéo exécute l'opération inverse.

#### 3.3 Fréquence d'échantillonnage

Les images sont échantillonnées suivant un multiple entier de la fréquence de ligne vidéo. Cette fréquence d'échantillonnage n'est pas verrouillée en synchronisme avec l'horloge du réseau numérique.

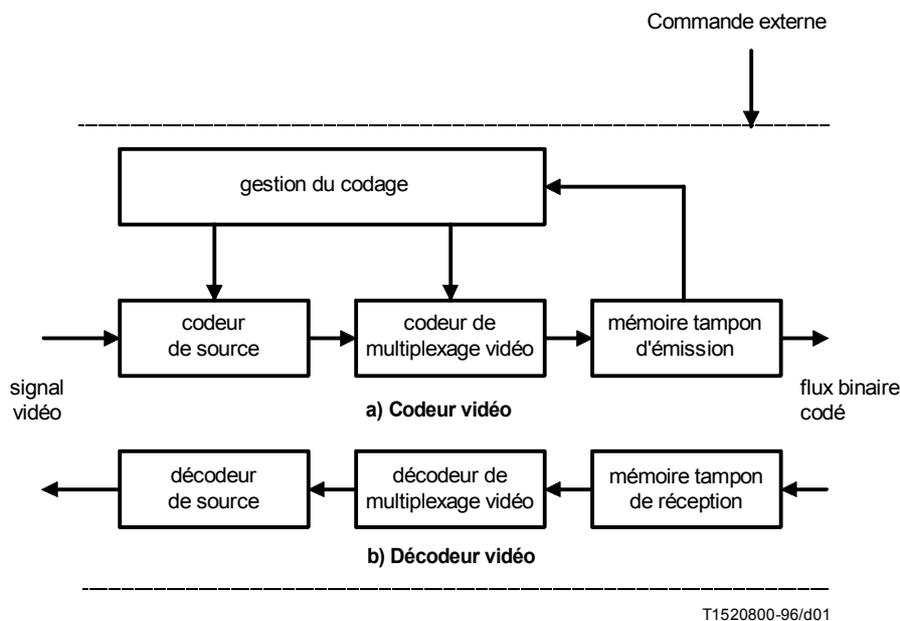


FIGURE 1/H.263

### Diagramme synoptique du codec vidéo

#### 3.4 Algorithme de codage de la source

On adopte un algorithme hybride de prédiction interimage qui utilise la redondance temporelle et le codage par transformée du signal restant afin de réduire la redondance spatiale. Le décodeur est capable de compenser les mouvements, ce qui permet d'introduire cette technique à titre d'option dans le codeur. Une précision au demi-pixel est utilisée pour la compensation des mouvements, contrairement à la Recommandation H.261, qui fait appel à une précision au pixel près et à un filtre en boucle. On utilise le codage à longueur variable pour les symboles à transmettre.

En plus de l'algorithme de codage principal de la présente Recommandation, celle-ci comprend quatre options de codage négociables, qui seront décrites dans les paragraphes suivants. Toutes ces options peuvent être utilisées ensemble ou séparément.

##### 3.4.1 Mode de codage par vecteurs cinétiques non restreints

Dans ce mode facultatif, les vecteurs cinétiques sont autorisés à pointer hors de l'image. Les pixels périphériques servent à prédire les pixels «inexistants». Ce mode permet d'obtenir un gain notable en cas de mouvement au-delà des bords de l'image, en particulier pour les petits formats d'image (voir également l'Annexe D). De plus, ce mode implique une extension de la gamme des vecteurs cinétiques, de manière à pouvoir utiliser des vecteurs cinétiques plus importants, ce qui est particulièrement utile dans le cas d'un mouvement de caméra.

##### 3.4.2 Mode de codage arithmétique syntaxique

Dans ce mode facultatif, le codage à longueur variable est remplacé par le codage arithmétique. Le rapport signal sur bruit (SNR) (*signal-to-noise ratio*) et les images reconstituées seront identiques mais nécessiteront beaucoup moins d'éléments binaires (voir aussi l'Annexe E).

##### 3.4.3 Mode de prédiction avancé

Ce mode facultatif utilise la compensation de mouvement par superposition de blocs (OBMC) (*overlapped block motion compensation*) pour la composante de luminance des images de type P (voir également l'Annexe F). Pour certains des macroblocs de l'image, on utilise quatre vecteurs à  $8 \times 8$  éléments d'image au lieu d'un seul vecteur à  $16 \times 16$  éléments. Le codeur doit déterminer le type de vecteurs à utiliser. Quatre vecteurs utilisent un plus grand nombre de bits mais donnent une meilleure prédiction. Ce mode apporte en général une amélioration considérable. Il apporte en particulier un gain subjectif car la compensation OBMC provoque moins de phénomènes de sautellement.

### 3.4.4 Mode de codage par trames PB

Une trame de type PB se compose de deux images codées ensemble en tant qu'unité. Les lettres PB correspondent aux désignations d'images P et d'images B dans la Recommandation H.262. Une trame PB se compose donc d'une image P qui est prédite à partir de la précédente image P décodée et d'une image B qui est prédite à partir d'une partie de la précédente image P décodée et d'autre part de l'image P en cours de décodage. Le terme image B a été retenu parce que certaines de ses parties peuvent être prédites bilatéralement à partir des images passées et futures. Cette option de codage permet d'augmenter considérablement la fréquence d'image, sans augmenter beaucoup le débit.

### 3.5 Débit

Le rythme de transmission est apporté de l'extérieur. Le débit vidéo peut être variable. Dans la présente Recommandation, aucune contrainte n'est imposée au débit vidéo; les contraintes seront apportées par le terminal ou par le réseau.

### 3.6 Stockage temporaire des données

Le codeur doit régler son flux binaire de sortie en fonction des prescriptions applicables au décodeur fictif de référence défini dans l'Annexe B. Les données vidéo doivent être présentées à chaque cycle d'horloge valide, ce qui peut être assuré par un bourrage de type MCBPC [type de macrobloc et structure de bloc codée pour la chrominance (*macroblock type and coded block pattern for chrominance*)] (voir les Tableaux 4 et 5); ou, si l'on utilise la correction d'erreur directe (FEC) (*forward error correction*), au moyen de trames de bourrage de type FEC (voir l'Annexe H).

Le nombre d'éléments binaires créés par le codage d'une image quelconque ne doit pas dépasser une valeur maximale spécifiée par le paramètre BPPmaxKb qui est mesuré en unités de 1024 bits. La valeur minimale admissible du paramètre BPPmaxKb dépend du plus grand format d'image source qui a été négocié pour utilisation dans le flux binaire (voir le Tableau 1). Un codeur peut utiliser pour le paramètre BPPmaxKb une valeur plus grande que celle qui est spécifiée dans le Tableau 1, à condition que cette plus grande valeur soit d'abord négociée par des moyens extérieurs, par exemple selon la Recommandation H.245.

TABLEAU 1/H.263

Valeur du paramètre BPPmaxKb pour chacun des formats d'image source

format d'image source	paramètre BPPmaxKb
SQCIF	64
QCIF	64
CIF	256
4CIF	512
16CIF	1024

### 3.7 Symétrie de transmission

On peut se servir du codec pour une communication audiovisuelle bidirectionnelle ou unidirectionnelle.

### 3.8 Traitement des erreurs

Il y a lieu d'assurer le traitement des erreurs par des moyens externes (par exemple selon la Recommandation H.223). S'ils ne sont pas fournis par des moyens externes (par exemple selon la Recommandation H.221), les codes facultatifs de correction d'erreur et de verrouillage de trame décrits dans l'Annexe H peuvent être utilisés.

Un décodeur peut envoyer un ordre pour coder un ou plusieurs groupes de blocs (GOB) (*group of blocks*) de sa prochaine image en mode INTRA, avec des paramètres de codage permettant d'éviter un dépassement des mémoires tampons. Un décodeur peut également envoyer l'ordre de ne transmettre que des en-têtes de groupe de blocs non vides. La méthode de transmission de ces signaux relève de moyens externes (par exemple selon la Recommandation H.245).

### 3.9 Fonctionnement multipoint

L'Annexe C indique les caractéristiques nécessaires pour assurer le fonctionnement multipoint en mode commuté.

## 4 Codeur de source

### 4.1 Format de la source

Le codeur de source agit sur des images non entrelacées apparaissant à raison de 30 000/1001 (soit environ 29,97) fois par seconde. La tolérance sur la fréquence d'image est de  $\pm 50 \times 10^{-6}$ .

Les images sont codées sous forme d'une composante de luminance et de deux composantes de différence de couleur ( $Y$ ,  $C_B$  et  $C_R$ ). Ces composantes, ainsi que les codes représentant leurs valeurs échantillonnées, sont définis dans la Recommandation UIT-R.

- noir = 16;
- blanc = 235;
- différence de couleur nulle = 128;
- différences de couleur de crête = 16 et 240.

Ces valeurs sont nominales et l'algorithme de codage fonctionne avec des valeurs d'entrée allant de 1 à 254.

Il existe cinq formats d'image normalisés: sub-QCIF, QCIF, CIF, 4CIF et 16CIF. Dans chacun de ces formats d'image, la structure d'échantillonnage de la luminance est de  $dx$  pixels par ligne et de  $dy$  lignes par image à affichage rectangulaire. L'échantillonnage de chacune des deux composantes de différence de couleur est effectué tous les  $dx/2$  pixels par ligne et tous les  $dy/2$  lignes par image, en affichage rectangulaire. Les valeurs des coefficients  $dx$ ,  $dy$ ,  $dx/2$  et  $dy/2$  sont indiquées dans le Tableau 2 pour chacun des formats d'image.

TABLEAU 2/H.263

**Nombre de pixels par ligne et nombre de lignes pour chacun des formats d'image H.263**

format d'image	nombre de pixels par code de luminance ( $dx$ )	nombre de lignes par code de luminance ( $dy$ )	nombre de pixels par code de chrominance ( $dx/2$ )	nombre de lignes par code de chrominance ( $dy/2$ )
sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

Pour chacun des formats d'image, les échantillons de différence chromatique sont situés de telle façon que leurs limites de bloc coïncident avec celles des blocs de luminance indiquées sur la Figure 2. Le rapport hauteur/largeur (aspect) en pixels est le même pour chacun de ces formats d'image et correspond à celui qui est défini pour les formats QCIF et CIF dans la Recommandation H.261:  $(4/3) * (288/352)$ . L'aire d'image couverte par tous les formats d'image (sauf le format sub-QCIF) a un rapport d'aspect de 4:3.

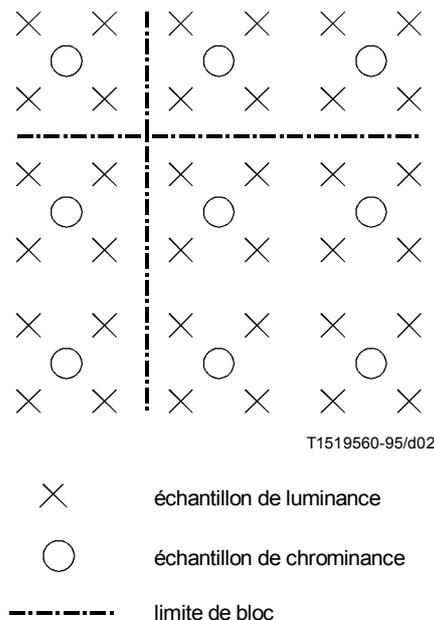


FIGURE 2/H.263

### Positionnement des échantillons de luminance et de chrominance

Tous les décodeurs doivent être capables de fonctionner en format sub-QCIF. Tous les décodeurs doivent également être en mesure de fonctionner en format QCIF. Certains décodeurs peuvent aussi fonctionner en CIF, 4CIF ou 16CIF. Les codeurs doivent être capables de fonctionner avec un des formats suivants: sub-QCIF ou QCIF. Les codeurs déterminent lequel de ces deux formats doit être utilisé; il n'est pas requis qu'ils soient en mesure de fonctionner avec les deux formats. Certains codeurs peuvent également fonctionner en CIF, 4CIF ou 16CIF. Des moyens externes signalent les formats qui peuvent être assurés par le décodeur, par exemple selon la Recommandation H.245. Pour avoir une vue d'ensemble complète sur les formats d'image et algorithmes de codage vidéo possibles, se reporter à la description du terminal, par exemple selon la Recommandation H.324.

NOTE – Pour le format CIF, le nombre de pixels par ligne est compatible avec l'échantillonnage des portions actives des signaux de luminance et de différence de couleur issus de sources à 525 ou à 625 lignes, à 6,75 MHz ou à 3,375 MHz selon le cas. Ces fréquences se rapportent directement à celles qui sont indiquées dans la Recommandation UIT-R 601.

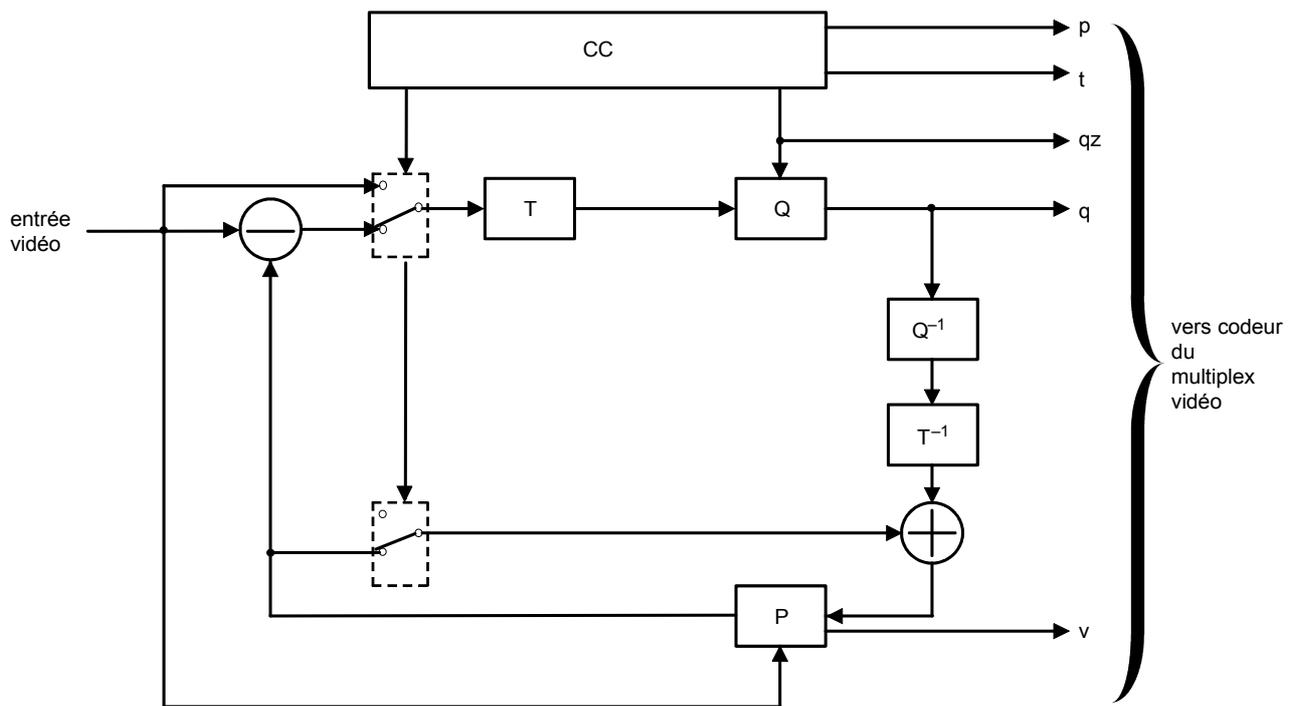
Il doit être possible de limiter la fréquence d'image maximale des codeurs, en minimisant le nombre d'images non transmises entre les images transmises. La sélection de ce nombre minimal doit être effectuée par des moyens externes (par exemple selon la Recommandation H.245). Pour le calcul du nombre minimal d'images non transmises en mode de trames PB, les images P et les images B des trames PB doivent être considérées comme formant deux images différentes.

## 4.2 Algorithme de codage de la source vidéo

Le codeur de source est représenté sous forme synoptique à la Figure 3. Les principaux éléments sont: la prédiction, la transformation en blocs et la quantification.

### 4.2.1 Groupes de blocs, macroblocs et blocs

Chaque image est subdivisée en groupes de blocs (GOB) (*group of blocks*) comprenant  $k * 16$  lignes,  $k$  dépendant du format d'image ( $k = 1$  pour sub-QCIF, QCIF et CIF;  $k = 2$  pour 4CIF;  $k = 4$  pour 16CIF). Le nombre de groupes de blocs par image est de 6 pour sub-QCIF, 9 pour QCIF et 18 pour CIF, 4CIF et 16CIF. La numérotation des groupes GOB est faite par analyse verticale des groupes de blocs en commençant par le groupe supérieur (numéro 0) et en finissant par le groupe inférieur. La Figure 4 montre un exemple de disposition de groupes GOB dans une image pour le format d'image CIF. Les données relatives aux groupes GOB sont transmises groupe par groupe, dans l'ordre croissant de leurs numéros.



T1519570-95/d03

- T transformation
- Q quantificateur
- P mémoire d'image avec retard variable de compensation de mouvement
- CC gestion du codage
- p fanion pour INTRA/INTER
- t fanion de transmission ou non d'un bloc
- qz indication du quantificateur
- q index de quantification pour les coefficients de la transformée
- v vecteur de mouvement

FIGURE 3/H.263

### Codeur de source

Chaque groupe GOB est divisé en macroblocs. Un macrobloc correspond à 16 pixels par 16 lignes de composante Y et, spatialement, aux 8 pixels par 8 lignes des composantes  $C_B$  et  $C_R$ . De plus, un macrobloc se compose de quatre blocs de luminance et des deux blocs spatialement correspondants de différence de couleur, comme indiqué sur la Figure 5. Chaque bloc de luminance ou de chrominance correspond à 8 pixels par 8 lignes de composante Y,  $C_B$  ou  $C_R$ . Un groupe GOB comprend une seule rangée de macroblocs pour les formats sub-QCIF, QCIF et CIF, deux rangées de macroblocs pour le format 4CIF et de quatre rangées de macroblocs pour le format 16CIF.

La numérotation des macroblocs est assurée par un balayage horizontal des rangées de macroblocs de gauche à droite à partir de la rangée supérieure de macroblocs jusqu'à la rangée inférieure de macroblocs. Les données des macroblocs sont transmises macrobloc par macrobloc, en ordre de numérotation croissant. Les données pour les blocs sont transmises bloc par bloc, en ordre de numérotation croissant (voir la Figure 5).

Les critères de choix du mode et de transmission d'un bloc ne font pas l'objet de recommandations. On peut les faire varier en tant que paramètres dynamiques intervenant dans la stratégie d'optimisation du codage. Les blocs transmis sont transformés et les coefficients résultants sont quantifiés et codés statistiquement.

#### 4.2.2 Prédiction

La prédiction est du type à codage interimage. Elle peut être améliorée par compensation de mouvement (voir 4.2.3). Le mode de codage dans lequel la prédiction est appliquée est appelé INTER; si aucune prédiction n'est appliquée, le mode de codage est appelé INTRA. Ce dernier mode peut être signalé au niveau des images (INTRA pour les images I ou INTER pour les images P). Dans les trames facultatives de type PB, les images de type B sont toujours codées en mode INTER. Les images B sont en partie prédites bilatéralement (voir l'Annexe G).

					0					
					1					
					2					
					3					
					4					
					5					
					6					
					7					
					8					
					9					
					10					
					11					
					12					
					13					
					14					
					15					
					16					
					17					

FIGURE 4/H.263

**Disposition d'un groupe de blocs dans une image en format CIF**

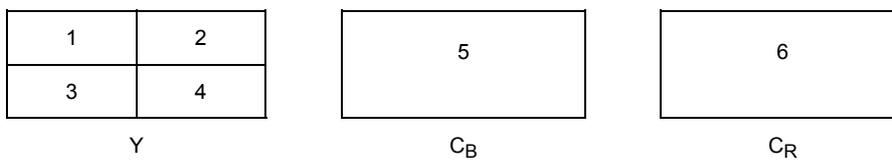


FIGURE 5/H.263

**Disposition des blocs dans un macrobloc**

**4.2.3 Compensation de mouvement**

Le décodeur acceptera un seul vecteur par macrobloc ou, si l'on utilise le mode de prédiction avancé, le décodeur acceptera un ou quatre vecteurs par macrobloc (voir l'Annexe F). Si l'on utilise le mode par trames PB, un vecteur delta supplémentaire peut être transmis dans chaque macrobloc pour adapter les vecteurs cinétiques à la prédiction des macroblocs de type B.

Les composantes aussi bien horizontale que verticale des vecteurs cinétiques ont des valeurs entières ou semi-entières. Dans le mode de prédiction par défaut, ces valeurs sont limitées à l'intervalle  $[-16, 15,5]$  (ce qui est également valide pour les composantes des vecteurs cinétiques anticipés et différés pour images B). Dans le mode de codage par vecteurs cinétiques non restreints, l'étendue maximale des composantes vectorielles est cependant  $[-31,5, 31,5]$ , avec la restriction que seules pourront être atteintes les valeurs comprises dans l'intervalle  $[-16, 15,5]$  autour du prédicteur de chaque composante vectorielle cinétique, si ce prédicteur est compris dans l'intervalle  $[-15,5, 16]$ . Si le prédicteur est extérieur à l'intervalle  $[-15,5, 16]$ , toutes les valeurs comprises dans l'intervalle  $[-31,5, 31,5]$ , ayant le même signe que le prédicteur plus valeur zéro, pourront être atteintes (voir également l'Annexe D).

Une valeur positive de la composante horizontale ou verticale du vecteur de mouvement indique que la prédiction est formée à partir de pixels contenus dans l'image référencée, spatialement placés à droite ou au-dessous des pixels en cours de prédiction.

Les vecteurs cinétiques sont bornés de telle sorte que tous les pixels auxquels ils font référence sont dans la zone d'image codée, sauf lorsque l'on utilise le mode de codage par vecteurs cinétiques non restreints et/ou le mode de prédiction avancé (voir les Annexes D et F).

#### 4.2.4 Quantification

Le nombre de quantificateurs est de 1 pour le premier coefficient des blocs de type INTRA et de 31 pour tous les autres coefficients. A l'intérieur d'un macrobloc, on utilise le même quantificateur pour tous les coefficients à l'exception du premier des blocs INTRA. Les niveaux de décision ne sont pas définis. Le premier coefficient des blocs INTRA est, nominalement, la valeur du coefficient DC de la transformée, quantifiée à l'échelon 8. Chacun des 31 autres quantificateurs utilise des niveaux de reconstruction équidistants, avec une zone morte centrée sur zéro et avec un échelon de valeur paire comprise entre 2 et 62. Pour les formules exactes, voir 6.2.

NOTE – Pour les échelons de quantification très faibles, on ne peut pas représenter toute la dynamique des coefficients résultant de la transformation.

### 4.3 Contrôle du débit des données

Divers paramètres peuvent être modifiés afin de gérer le débit de production des données vidéo codées. Ces paramètres peuvent porter sur le traitement en amont du codeur de source, sur le quantificateur, sur le critère de sélection des blocs et sur la fréquence de sous-échantillonnage temporel. L'importance relative de ces actions dans la stratégie globale de régulation ne fait l'objet d'aucune recommandation particulière.

Lorsqu'il est mis en œuvre, le sous-échantillonnage temporel se fait par élimination d'images entières.

Un décodeur peut signaler sa préférence pour un certain compromis entre résolutions spatiale et temporelle du signal vidéo. Le codeur doit signaler son compromis par défaut au début de la communication. Il doit également indiquer s'il est capable de répondre aux requêtes du décodeur en vue de modifier son compromis. La méthode de transmission de ces signaux est d'ordre externe (par exemple selon la Recommandation H.245).

### 4.4 Rafraîchissement forcé

On applique cette fonction en forçant l'algorithme de codage à revenir au mode INTRA. La séquence de rafraîchissement n'est pas définie. Pour limiter l'accumulation d'erreurs d'adaptation lors de la transformation inverse, chaque macrobloc doit être codé en mode INTRA au moins une fois toutes les 132 fois où des coefficients sont transmis pour ce macrobloc dans les images P.

### 4.5 Alignement par les octets des codes de déclenchement

L'alignement par les octets des codes de déclenchement consiste à insérer un mot codé en bourrage se composant de moins de 8 zéros binaires avant le code de déclenchement, de manière que le premier bit du code de déclenchement soit le premier bit (de poids fort) d'un octet. Un code de déclenchement est donc aligné par les octets si la position de son bit de poids fort est un multiple de 8 bits à partir du premier bit contenu dans le flux binaire H.263. Tous les codes de déclenchement d'image doivent être alignés en octets. Les codes de déclenchement de groupe GOB et de fin de séquence EOS peuvent être alignés en octets.

#### NOTES

- 1 Le nombre de bits consacrés à une image donnée est variable mais toujours un multiple de 8 bits.
- 2 La Recommandation H.324 prescrit que les codeurs H.263 alignent leurs codes de déclenchement d'image sur le début des unités logiques d'information transmises à la couche d'adaptation (unités AL\_SDU).

## 5 Syntaxe et sémantique

Le multiplex de données vidéo est disposé selon une structure hiérarchique à quatre couches. Celles-ci sont, de haut en bas, les suivantes:

- image;
- groupe de blocs;
- macrobloc;
- bloc.

Le diagramme syntaxique est représenté sur la Figure 7. Les abréviations et la sémantique sont définies dans les paragraphes suivants.

Sauf indication contraire, le bit de poids fort est transmis en premier. Il s'agit du bit 1 et c'est l'élément binaire placé le plus à gauche dans les tables de code contenues dans la présente Recommandation. Sauf indication contraire, tous les bits inutilisés ou tenus en réserve sont forcés à «1». Les bits tenus en réserve ne doivent pas être utilisés avant que leur fonction ait été spécifiée par l'UIT.

## 5.1 Couche des images

Pour chaque image, les données comportent un en-tête d'image suivi des données de groupe de blocs, puis finalement d'un code de fin de séquence (EOS) et de bits de bourrage. Cette structure est représentée sur la Figure 6. L'indicateur de sous-flux binaire (PSBI) (*picture sub-bitstream indicator*) n'est présent que s'il est indiqué par une communication multipoint à présence continue (CPM) (*continuous presence multipoint*). Les codes de référence temporelle pour les trames B (TR<sub>B</sub>) (*temporal reference for B-frames*) et d'information de quantification pour images B (DBQUANT) (*quantization information for B-pictures*) ne sont présents que si le mot de code PTYPE (information de type) indique «trame PB». Aucune combinaison des codes PSPARE et PEI ne peut être présente. Le mot de code EOS peut être absent, tandis que ESTUF ne peut être présent que si EOS l'est également. Les en-têtes des images éliminées ne sont pas transmis.

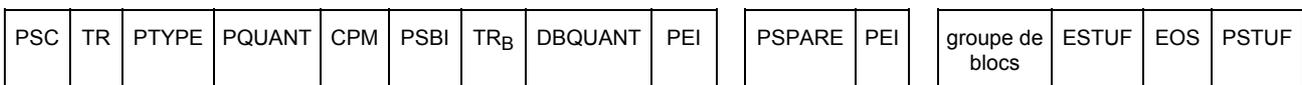


FIGURE 6/H.263

### Structure de la couche des images

#### 5.1.1 Code de déclenchement d'image (PSC) (*picture start code*) (22 bits)

Le code PSC est un mot de 22 éléments binaires. Sa valeur est 0000 0000 0000 0000 1 00000. Tous les codes de déclenchement d'image doivent être alignés en octets, ce qui doit être effectué par insertion du mot PSTUF avant le code de déclenchement, de manière que le premier bit de ce code soit le premier bit (de poids fort) d'un octet.

#### 5.1.2 Référence temporelle (TR) (*temporal reference*) (8 bits)

Ce nombre de 8 bits peut avoir 256 valeurs. On le forme en ajoutant, à sa valeur contenue dans le précédent en-tête d'image transmis, 1 plus le nombre d'images non transmises (à 29,97 Hz) depuis la dernière image transmise. Les opérations arithmétiques sont effectuées sur les huit bits de poids faible seulement. En mode facultatif de trames PB, la référence temporelle ne s'applique qu'aux images P. Pour la référence temporelle aux images B, voir 5.1.7.

#### 5.1.3 Informations de type (PTYPE) (*type information*) (13 bits)

Informations sur l'image complète:

- bit 1: toujours «1», afin d'éviter une émulation du code de déclenchement;
- bit 2: toujours «0», par opposition à la Recommandation H.261;
- bit 3: indicateur de partage d'écran: «0» inactivé, «1» activé;
- bit 4: indicateur de caméra document: «0» inactivé, «1» activé;
- bit 5: fin de gel d'image: «0» inactivé, «1» activé;
- bits 6-8: format de source: «000» interdit, «001» SQCIF, «010» QCIF, «011» CIF, «100» 4CIF, «101» 16CIF, «110» réservé, «111» réservé;
- bit 9: type de codage d'image: «0» INTRA (image de type I), «1» INTER (image de type P);
- bit 10: mode facultatif de vecteur cinétique non restreint: «0» inactivé, «1» activé;
- bit 11: mode facultatif de codage arithmétique syntaxique: «0» inactivé, «1» activé;
- bit 12: mode facultatif de prédiction avancé: «0» inactivé, «1» activé;
- bit 13: mode facultatif de trames PB: «0» image normale de type I ou P, «1» trame PB.

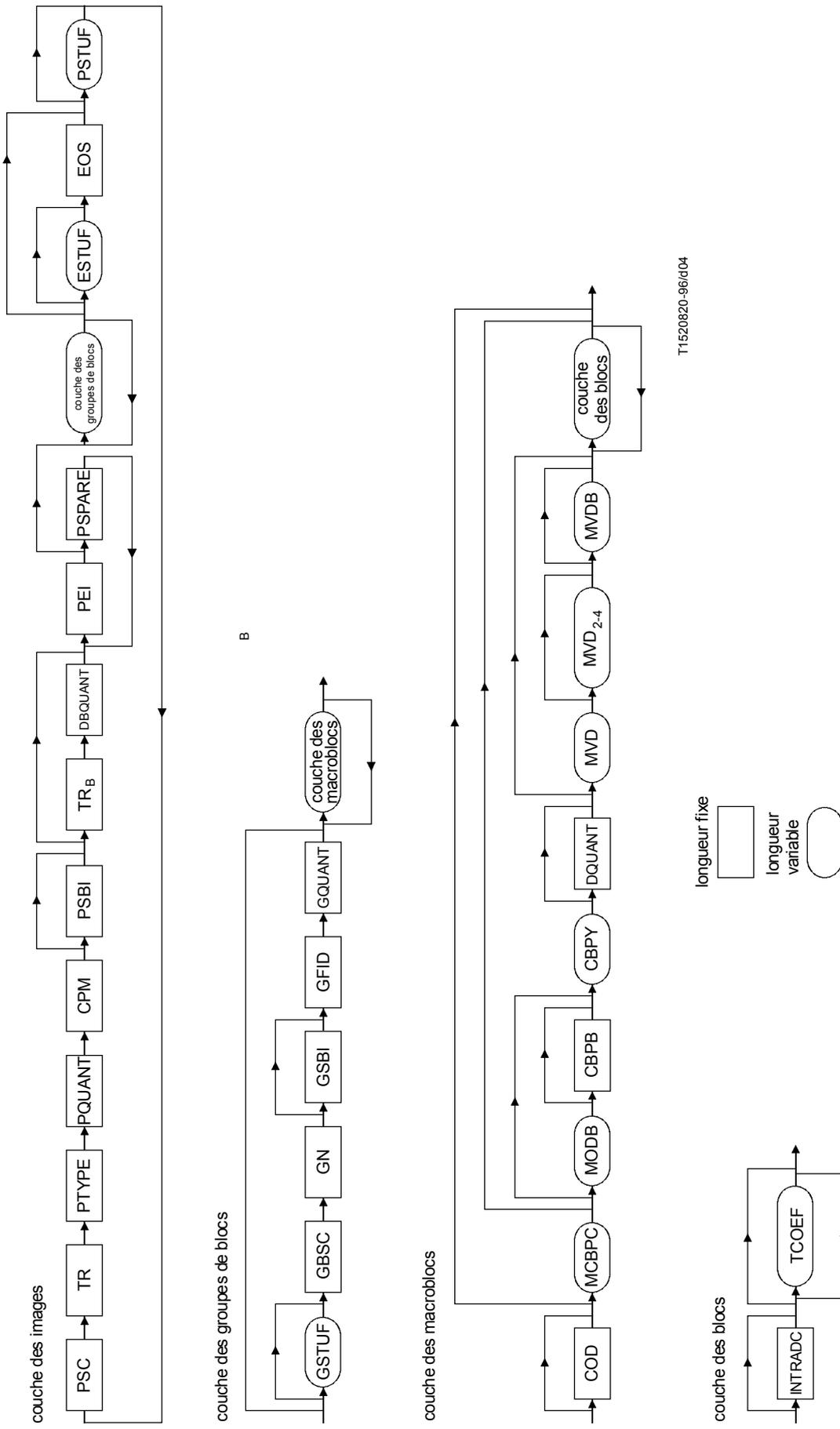


FIGURE 7/H.263

Document communiqué sous le régime de l'accès à l'information

L'indicateur de partage d'écran est un signal qui indique que les moitiés supérieure et inférieure de l'image décodée peuvent être affichées côte à côte. Ce bit n'a pas d'incidence directe sur le codage ou sur le décodage de l'image.

La fin de gel d'image est un signal issu d'un codeur qui répond à une demande de retransmission en mode paquet (s'il n'y a pas eu d'accusé de réception) ou à une demande de rafraîchissement rapide (voir également l'Annexe C). Ce signal permet à un décodeur de sortir de son mode de gel d'image et d'afficher normalement l'image décodée.

Si les bits 6-8 indiquent un format de source différent de celui qui est indiqué dans le précédent en-tête d'image, l'image actuelle doit être de type I.

Les bits 10-13 se rapportent aux modes facultatifs qui ne sont utilisés qu'après négociation entre codeur et décodeur (voir également les Annexes D, E, F et G, respectivement). Si le bit 9 est mis à «0», le bit 13 doit aussi être mis à «0».

**5.1.4 Information sur le quantificateur (PQUANT) (*quantizer information*) (5 bits)**

Mot de code de longueur fixe (5 bits) qui indique le quantificateur QUANT à utiliser pour l'image jusqu'à ce qu'il soit mis à jour par un mot GQUANT ou DQUANT ultérieur. Ce mot de code est la représentation binaire naturelle des valeurs de quantification (QUANT) qui, leur pas étant moitié de l'échelon de quantification, vont de 1 à 31.

**5.1.5 Communication multipoint à présence continue (CPM) (*continuous presence multipoint*) (1 bit)**

Fanion (1 bit) qui signale l'utilisation du mode facultatif de communication multipoint à présence continue (CPM); «0» indique que le mode est inactivé, «1» qu'il est activé. Pour l'utilisation du bit CPM, voir l'Annexe C.

**5.1.6 Indicateur de sous-flux binaire (PSBI) (*picture sub-bit stream indicator*) (2 bits)**

Mot de code de longueur fixe (2 bits) qui n'est présent que si le mode de communication multipoint à présence continue est indiqué par le fanion CPM. Ce mot de code est la représentation binaire naturelle du numéro de sous-flux binaire pour l'en-tête d'image et toutes les informations qui le suivent, jusqu'au prochain code de déclenchement d'image ou de groupe de blocs (voir également l'Annexe C).

**5.1.7 Référence temporelle pour les images B (TR<sub>B</sub>) (*temporal reference for B-pictures*) (3 bits)**

La référence TR<sub>B</sub> est présente si le mot PTYPE indique «trame PB» (voir également l'Annexe G) ainsi que le nombre d'images non transmises (à 29,97 Hz) depuis la dernière image P ou I et avant l'image B. Ce mot de code est la représentation binaire naturelle du nombre d'images non transmises plus un. Le nombre maximal d'images non transmises est de 6.

**5.1.8 Information de quantification pour images B (DBQUANT) (*quantization information for B-pictures*) (2 bits)**

Le code DBQUANT est présent si le mot PTYPE indique «trame PB» (voir également l'Annexe C). Dans le processus de décodage, on obtient un paramètre de quantification QUANT pour chaque macrobloc. Avec des trames PB, le mot QUANT est utilisé pour les blocs de type P, tandis que pour les blocs de type B, on utilise un paramètre de quantification différent: BQUANT. Le mot QUANT peut prendre une valeur comprise entre 1 et 31. Le code DBQUANT indique la relation entre QUANT et BQUANT selon la définition du Tableau 3. Dans ce tableau, la barre oblique (/) indique une division par troncature. Les valeurs de BQUANT vont de 1 à 31; si la valeur de BQUANT, issue du Tableau 3, est supérieure à 31, elle est arrondie à 31.

TABLEAU 3/H.263

**Codes DBQUANT et relations entre codes QUANT et BQUANT**

DBQUANT	BQUANT
00	$(5 \times \text{QUANT})/4$
01	$(6 \times \text{QUANT})/4$
10	$(7 \times \text{QUANT})/4$
11	$(8 \times \text{QUANT})/4$

### 5.1.9 Informations supplémentaires d'insertion (PEI) (*extra insertion information*) (1 bit)

Fanion qui, lorsqu'il est mis à «1», signale la présence du champ de données facultatif suivant.

### 5.1.10 Informations réservées (PSPARE) (*spare information*) (0/8/16... bits)

Si le fanion PEI est mis à «1», il est suivi de 9 bits comprenant 8 bits de données réservées (PSPARE) puis d'un autre fanion PEI pour indiquer si un nouveau champ de 9 bits suit, et ainsi de suite. Les codeurs ne doivent pas insérer le mot PSPARE avant que celui-ci ait été spécifié par l'UIT. Les décodeurs doivent être conçus de façon à rejeter le champ PSPARE si le fanion PEI est mis à «1». Cela permettra à l'UIT de spécifier, dans le champ PSPARE, de futurs compléments rétrocompatibles. Si le mot PSPARE est suivi d'un fanion PEI = 0, la valeur PSPARE = xx000000 est interdite afin d'éviter toute émulation de code de déclenchement (x = valeur indifférente, de sorte que 4 valeurs sur 256 sont interdites).

### 5.1.11 Bourrage (ESTUF) (*stuffing*) (longueur variable)

Mot de code de longueur variable, se composant de moins de 8 zéros binaires. Les codeurs peuvent insérer ce mot de code directement avant un mot de code EOS. Si le mot ESTUF est présent, son dernier bit doit être le dernier bit (de poids faible) d'un octet, de manière que le début du mot de code EOS soit aligné en termes d'octets. Les décodeurs doivent être conçus de façon à rejeter le mot ESTUF.

### 5.1.12 Fin de séquence (EOS) (*end of sequence*) (22 bits)

Mot de code de 22 bits dont la valeur est 0000 0000 0000 0000 1 1111. Il appartient au codeur d'insérer ou de ne pas insérer ce mot de code, qui peut être aligné en octets. Pour cela, on insère le mot ESTUF avant le code de déclenchement de manière que le premier bit du code de déclenchement soit le premier bit (de poids fort) d'un octet.

### 5.1.13 Bourrage (PSTUF) (*stuffing*) (longueur variable)

Mot de code de longueur variable composé de moins de 8 zéros binaires. Les codeurs doivent insérer ce mot de code pour l'alignement en octets du code PSC suivant. Le dernier bit du mot PSTUF doit être le dernier bit (de poids faible) d'un octet, de manière que le flux binaire vidéo, comprenant le mot PSTUF, soit un multiple de 8 bits à partir du premier bit du flux binaire H.263. Les décodeurs doivent être conçus de façon à rejeter le mot PSTUF.

Si, pour une raison ou une autre, le codeur arrête de coder les images pendant un certain temps et reprend le codage ensuite, le mot PSTUF doit être transmis avant l'arrêt du codeur afin de s'assurer que les 7 derniers éléments binaires (au plus) de l'image précédente seront envoyés avant la reprise du codage.

## 5.2 Couche des groupes de blocs

Les données de chaque groupe de blocs (GOB) se composent d'un en-tête de groupe suivi des données correspondant aux macroblocs, selon la structure représentée à la Figure 8. Chaque groupe de blocs contient une ou plusieurs rangées de macroblocs. Pour le premier groupe de blocs de chaque image (de numéro 0), aucun en-tête de groupe ne doit être transmis. L'en-tête de tous les autres groupes GOB peut être vide, selon la stratégie du codeur. Un décodeur peut signaler au codeur distant, par des moyens externes comme la Recommandation H.245, qu'il ne doit émettre que des en-tête GOB non vides. Le mot GSTUF peut être présent lorsque le mot GBSC est présent. Les mots GN, GFID et GFQUANT sont présents lorsque le mot GBSC est présent. Le mot GSBI est présent lorsque le mode CPM est activé, comme indiqué dans l'en-tête d'image.

GSTUF	GBSC	GN	GSBI	GFID	GQUANT	données de macrobloc
-------	------	----	------	------	--------	----------------------

FIGURE 8/H.263

### Structure de la couche des groupes de blocs

### 5.2.1 Bourrage (GSTUF) (*stuffing*) (longueur variable)

Mot de longueur variable, se composant de moins de 8 zéros binaires. Les codeurs peuvent insérer ce mot de code directement avant un mot GBSC. Si le mot GSTUF est présent, son dernier bit doit être le dernier bit (de poids faible) d'un octet, de manière que le début du mot GBSC soit aligné en octets. Les décodeurs doivent être conçus de façon à rejeter le mot GSTUF.

### 5.2.2 Code de déclenchement d'un groupe de blocs (GBSC) (*group of block start code*) (17 bits)

Mot de 17 bits dont la valeur est 0000 0000 0000 0000 1. Les codes de déclenchement des groupes de blocs peuvent être alignés en octets. Pour cela, on insère le mot GSTUF avant le code de déclenchement de manière que le premier bit du code de déclenchement soit le premier bit (de poids fort) d'un octet.

### 5.2.3 Numéro de groupe (GN) (*group number*) (5 bits)

Mot de longueur fixe, codé sur 5 bits qui sont la représentation binaire du numéro de groupe de blocs. Pour le groupe de numéro 0, l'en-tête GOB (y compris les codes GSTUF, GBSC, GN, GSBI, GFID et GQUANT) est vide. Le groupe numéro 0 est utilisé dans le code de déclenchement d'image (PSC). Le groupe numéro 31 est utilisé dans la fin de séquence et les valeurs de 18 à 30 sont réservées pour utilisation future par l'UIT.

### 5.2.4 Indicateur de sous-flux binaire d'un groupe de blocs (GSBI) (*GOB sub-bit stream indicator*) (2 bits)

Mot de longueur fixe, codé sur 2 bits, qui n'est présent que si le mode de communication multipoint à présence continue est indiqué par le mot CPM. Ce mot de code est la représentation binaire naturelle du numéro de sous-flux binaire pour l'en-tête de groupe GOB et pour toutes les informations suivantes jusqu'au prochain code de déclenchement d'image ou de groupe de blocs (voir également l'Annexe C).

### 5.2.5 Identificateur de trame GOB (GFID) (*GOB frame ID*) (2 bits)

Mot de longueur fixe, codé sur 2 bits, qui doit avoir la même valeur dans chaque en-tête de groupe GOB d'une image donnée. Par ailleurs, si le mot PTYPE indiqué dans un en-tête d'image est le même que pour l'image précédemment émise, l'identificateur GFID doit avoir la même valeur que dans cette image précédente. Si cependant le mot PTYPE diffère, dans un certain en-tête d'image, du mot PTYPE contenu dans l'en-tête de la précédente image transmise, la valeur du mot GFID dans cette image doit différer de la valeur contenue dans l'image précédente.

### 5.2.6 Information sur le quantificateur (GQUANT) (*quantizer information*) (5 bits)

Mot de code de longueur fixe (5 bits) qui indique le quantificateur QUANT à utiliser pour la partie restante de l'image jusqu'à ce qu'il soit mis à jour par un mot GQUANT ou DQUANT ultérieur. Ce mot de code est la représentation binaire naturelle des valeurs de quantification (QUANT) qui, leur pas étant moitié de l'échelon de quantification, vont de 1 à 31.

## 5.3 Couche des macroblocs

Les données de chaque macrobloc comprennent un en-tête de macrobloc suivi des données correspondant aux blocs. La structure de cette couche est représentée sur la Figure 9. L'indication de macrobloc codé (COD) (*coded macroblock indication*) n'est présente que dans les images pour lesquelles PTYPE indique «INTER», pour chaque macrobloc contenu dans ces images. Le code de type de macrobloc et de structure de bloc codée pour la chrominance (MCBPC) est présent lorsqu'il est indiqué par le code COD ou lorsque PTYPE indique «INTRA». Le code de mode de macrobloc pour blocs d'images B (MODB) est présent pour les macroblocs de type 0-4 si PTYPE indique «trame PB». Les codes CBPY, DQUANT, MVD et MVD<sub>2-4</sub> sont présents s'ils sont indiqués par le code MCBPC. Les codes CBPB et MVDB ne sont présents que s'ils sont indiqués par le mot MODB. Les données de bloc sont présentes lorsqu'elles sont indiquées par les mots MCBPC et CBPY. Les mots MVD<sub>2-4</sub> ne sont présents qu'en mode de prédiction avancé (voir l'Annexe F). Les codes MODB, CBPB et MVDB ne sont présents qu'en mode de trames PB (voir l'Annexe G). Pour le codage des symboles en mode arithmétique syntaxique, voir l'Annexe E.

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD <sub>2</sub>	MVD <sub>3</sub>	MVD <sub>4</sub>	MVDB	données de bloc
-----	-------	------	------	------	--------	-----	------------------	------------------	------------------	------	-----------------

FIGURE 9/H.263

### Structure de la couche des macroblocs

### 5.3.1 Indication de macrobloc codé (COD) (*coded macroblock indication*) (1 bit)

Fanion qui, lorsqu'il est mis à «0», signale que le macrobloc est codé. S'il est mis à «1», aucune autre information n'est transmise pour ce macrobloc; dans ce cas, le décodeur doit traiter le macrobloc comme un macrobloc de type INTER avec un vecteur cinétique, pour l'ensemble du macrobloc, de valeur nulle et sans données de coefficient. Le fanion COD n'est présent que dans les images pour lesquelles PTYPE indique «INTER», dans chaque macrobloc de ces images.

NOTE – En mode de prédiction avancé, la compensation de mouvement par superposition est également effectuée si le fanion COD est mis à «1».

### 5.3.2 Type de macrobloc et structure de bloc codée pour la chrominance (MCBPC) (*macroblock type & coded block pattern for chrominance*) (longueur variable)

La structure de bloc codée pour la chrominance (CBPC) indique qu'il s'agit de blocs  $C_B$  et/ou  $C_R$  lorsque au moins un coefficient non-INTRADC (autre que le coefficient DC de transformée de blocs INTRA, voir 5.4.1) est transmis. Si un quelconque coefficient non-INTRADC est présent pour le bloc N, on a  $CBPC_N = 1$ ; sinon,  $CBPC_N = 0$ , pour N = 5 et 6 dans la structure de bloc codée. La numérotation des blocs est indiquée dans la Figure 5. Lorsque le mot MCBPC a la valeur «bourrage», la partie restante de la couche des macroblocs est sautée. Dans ce cas, la valeur COD = 0 précédente ne vise aucun macrobloc particulier, codé ou non codé; le numéro de macrobloc n'est donc pas augmenté. Pour les images P, les bourrages multiples sont effectués par des séries multiples de valeurs COD = 0 et MCBPC = bourrage. Voir les Tableaux 4 et 5.

La structure de bloc codée pour la chrominance (CBPC) indique qu'il s'agit de blocs  $C_B$  et/ou  $C_R$  lorsque au moins un coefficient non-INTRADC (autre que le coefficient DC de transformée de blocs INTRA) est transmis. Si un quelconque coefficient non-INTRADC est présent pour le bloc N, on a  $CBPC_N = 1$ ; sinon,  $CBPC_N = 0$ , pour N = 5 et 6 dans la structure de bloc codée. La numérotation des blocs est indiquée dans la Figure 5. Lorsque le mot MCBPC a la valeur «bourrage», la partie restante de la couche des macroblocs est sautée. Dans ce cas, la valeur COD = 0 précédente ne vise aucun macrobloc particulier, codé ou non codé; le numéro de macrobloc n'est donc pas augmenté. Pour les images codées en mode INTER, les bourrages multiples sont effectués par des séries multiples de valeurs COD = 0 et MCBPC = bourrage. Voir les Tableaux 6 et 7.

TABLEAU 4/H.263

Table des codes de longueur variable (VLC) pour le mot MCBPC (images de type I)

index	type de macrobloc	CBPC (56)	nombre de bits	code
0	3	00	1	1
1	3	01	3	001
2	3	10	3	010
3	3	11	3	011
4	4	00	4	0001
5	4	01	6	0000 01
6	4	10	6	0000 10
7	4	11	6	0000 11
8	bourrage	–	9	0000 0000 1

### 5.3.3 Mode de macrobloc pour blocs d'images B (MODB) (*macroblock mode for B-blocks*) (longueur variable)

Le code MODB est présent pour les macroblocs de type 0-4 si PTYPE indique «trame PB» et est un mot de code de longueur variable indiquant si le mot CBPB est présent (c'est-à-dire que des coefficients d'image B sont transmis pour ce macrobloc) et/ou si le mot MVDB est présent. Le Tableau 8 définit les mots de code pour les modes MODB.

### 5.3.4 Structure de bloc codée pour blocs d'images B (CBPB) (*coded block pattern for B-blocks*) (6 bits)

Le code de structure CBPB n'est présent qu'en mode de trames PB, s'il a été indiqué par le code MODB.  $CBPB_N = 1$  si un coefficient quelconque est présent pour le bloc N d'images B. Sinon  $CBPB_N = 0$ , pour chaque bit du mot CBPB dans la structure de bloc codée. La numérotation des blocs est indiquée dans la Figure 5, le bit le plus à gauche du mot CBPB correspondant au bloc numéro 1.

TABLEAU 5/H.263

Table des codes de longueur variable (VLC) pour le mot MCBPC (images de type P)

index	type de macrobloc	CBPC (56)	nombre de bits	code
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	bourrage	–	9	0000 0000 1

TABLEAU 6/H.263

Types de macroblocs et éléments de données inclus pour images normales

type d'image	type de macrobloc	nom	COD	MCBPC	CBPY	DQUANT	MVD	MVD <sub>2-4</sub>
INTER	non codé	–	X					
INTER	0	INTER	X	X	X		X	
INTER	1	INTER + Q	X	X	X	X	X	
INTER	2	INTER4V	X	X	X		X	X
INTER	3	INTRA	X	X	X			
INTER	4	INTRA + Q	X	X	X	X		
INTER	bourrage	–	X	X				
INTRA	3	INTRA		X	X			
INTRA	4	INTRA + Q		X	X	X		
INTRA	bourrage	–		X				

NOTE – La lettre «X» indique que l'élément est présent dans le macrobloc.

TABLEAU 7/H.263

## Types de macroblochs et éléments de données inclus pour trames PB

type d'image	type de macro-bloc	nom	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD <sub>2-4</sub>
INTER	non codé	-	X								
INTER	0	INTER	X	X	X	X	(X)		X	(X)	
INTER	1	INTER + Q	X	X	X	X	(X)	X	X	(X)	
INTER	2	INTER 4V	X	X	X	X	(X)		X	(X)	X
INTER	3	INTRA	X	X	X	X	(X)		X	(X)	
INTER	4	INTRA + Q	X	X	X	X	(X)	X	X	(X)	
INTER	bourrage	-	X	X							

NOTES

- 1 La lettre «X» indique que l'élément est présent dans le macrobloc.
- 2 Les codes CBPB et MVDB ne sont présents que s'ils sont indiqués par MODB.
- 3 Les blocs de type B sont toujours codés en mode INTER, même si le type MB du macrobloc de trames PB indique INTRA.

TABLEAU 8/H.263

## Table des codes de longueur variable (VLC) pour le mot MODB

index	CBPB	MVDB	nombre de bits	code
0			1	0
1		X	2	10
2	X	X	2	11

NOTE – La lettre «X» indique que l'élément est présent dans le macrobloc.

### 5.3.5 Structure de bloc codée pour la luminance (CBPY) (*coded block pattern for luminance*) (longueur variable)

Mot de code de longueur variable contenant un numéro de structure désignant les blocs de luminance contenus dans le macrobloc pour lequel au moins 1 coefficient de transformée non INTRADC (autre que le coefficient DC de transformée de blocs INTRA, voir 5.4.1) est transmis.

$CBPY_N = 1$  si un quelconque coefficient non-INTRADC est présent pour le bloc N. Sinon  $CBPY_N = 0$ , pour chaque bit du mot  $CBPY_N$  dans la structure de bloc codée. La numérotation des blocs est indiquée dans la Figure 5, le bit le plus à gauche du mot  $CBPY_N$  correspondant au bloc numéro 1. Pour une certaine structure  $CBPY_N$ , différents mots de code sont utilisés pour les macroblochs INTER et INTRA, comme indiqué dans le Tableau 10.

### 5.3.6 Informations sur le quantificateur (DQUANT) (*quantizer information*) (2 bits)

Mot codé sur 2 bits pour définir un changement de l'information QUANT. Le Tableau 9 indique les valeurs différentielles pour les différents mots de code. Les valeurs de QUANT vont de 1 à 31; si la valeur de QUANT est inférieure à 1 ou supérieure à 31 après addition de la valeur différentielle, elle est arrondie à 1 ou à 31 selon le cas.

TABLEAU 9/H.263

**Codes DQUANT et valeurs différentielles pour QUANT**

index	valeur différentielle	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

TABLEAU 10/H.263

**Table des codes de longueur variable pour CBPY**

index	CBPY(INTRA) (12 34)	CBPY(INTER) (12 34)	nombre de bits	code
0	00 00	11 11	4	0011
1	00 01	11 10	5	0010 1
2	00 10	11 01	5	0010 0
3	00 11	11 00	4	1001
4	01 00	10 11	5	0001 1
5	01 01	10 10	4	0111
6	01 10	10 01	6	0000 10
7	01 11	10 00	4	1011
8	10 00	01 11	5	0001 0
9	10 01	01 10	6	0000 11
10	10 10	01 01	4	0101
11	10 11	01 00	4	1010
12	11 00	00 11	4	0100
13	11 01	00 10	4	1000
14	11 10	00 01	4	0110
15	11 11	00 00	2	11

### 5.3.7 Données de vecteur cinétique (MVD) (*motion vector data*) (longueur variable)

Le mot MVD est inclus pour tous les macroblocs INTER (ainsi que pour les macroblocs INTRA en mode «trames PB»). Il se compose d'une séquence codée de longueur variable pour la composante horizontale, suivie d'une séquence codée de longueur variable pour la composante verticale. Les séquences codées de longueur variable sont indiquées dans le Tableau 11.

### 5.3.8 Données de vecteur cinétique (MVD<sub>2-4</sub>) (*motion vector data*) (longueur variable)

Les trois mots de code MVD<sub>2-4</sub> sont inclus si cela est indiqué par PTYPE et par MCBPC. Ils consistent chacun en une séquence codée de longueur variable pour la composante horizontale, suivie d'une séquence codée de longueur variable pour la composante verticale de chaque vecteur. Les séquences codées de longueur variable sont indiquées dans le Tableau 11. Les mots MVD<sub>2-4</sub> ne sont présents que dans le mode de prédiction avancé (voir l'Annexe F).

### 5.3.9 Données de vecteur cinétique pour macrobloc de type B (MVDB) (*motion vector data for B-macroblock*) (longueur variable)

Le mot de code MVDB n'est présent qu'en mode de trames PB si cela est indiqué par MODB. Il consiste en une séquence codée de longueur variable pour la composante horizontale, suivie d'une séquence codée de longueur variable pour la composante verticale de chaque vecteur. Les séquences codées de longueur variable sont indiquées dans le Tableau 11. Pour l'utilisation des mots MVDB, voir l'Annexe G.

## 5.4 Couche des blocs

En dehors du mode de trames PB, un macrobloc se compose de quatre blocs de luminance et d'un des deux blocs de différence de couleur (voir la Figure 5). La structure de la couche des blocs est représentée sur la Figure 10. Le mot INTRADC est présent pour chaque bloc du macrobloc si MCBPC indique que le macrobloc est du type 3 ou 4 (voir les Tableaux 4 et 5). Le mot TCOEF est présent s'il a été indiqué par MCBPC ou par CBPY.

En mode de trames PB, un macrobloc se compose de douze blocs. Les données pour les six blocs P sont transmises d'abord, comme en mode H.263 par défaut, suivies des données pour les six blocs B. Le mot INTRADC est présent pour chaque bloc P du macrobloc si le mot MCBPC indique que le macrobloc est du type 3 ou 4 (voir les Tableaux 4 et 5). Le mot INTRADC n'est pas présent pour les blocs B. Le mot TCOEF est présent pour les blocs B s'il a été indiqué par CBPB.

Pour le codage des symboles en mode arithmétique syntaxique, voir l'Annexe E.

### 5.4.1 Coefficient DC pour les blocs INTRA (INTRADC) (*DC coefficient for INTRA blocks*) (8 bits)

Mot codé sur 8 bits. La séquence 0000 0000 n'est pas utilisée. La séquence 1000 0000 n'est pas utilisée, le niveau de reconstruction de 1024 étant codé par 1111 1111 (voir le Tableau 12).

### 5.4.2 Coefficient de transformation (TCOEF) (*transform coefficient*) (longueur variable)

Les événements (EVENT) les plus fréquents sont codés avec les séquences de longueur variable indiquées dans le Tableau 13. Le dernier bit «s» indique le signe de l'amplitude: «0» pour positif et «1» pour négatif.

Un événement (EVENT) est une combinaison d'une dernière indication de coefficient non nul (LAST; «0»: il y a encore des coefficients non nuls dans ce bloc; «1»: ce coefficient non nul est le dernier dans ce bloc), du nombre de zéros successifs qui précèdent le coefficient codé (RUN) et de la valeur non nulle du coefficient codé (LEVEL).

Les autres combinaisons des éléments (LAST, RUN, LEVEL) sont codées sur un mot de 22 bits composé de 7 bits d'échappement (ESCAPE), 1 bit LAST, 6 bits RUN et 8 bits LEVEL. L'utilisation de ce mot de 22 bits pour le codage des combinaisons énumérées dans le Tableau 13 n'est pas interdite. Les séquences 0000 0000 et 1000 0000 ne sont pas utilisées dans le mot de 8 bits exprimant la grandeur LEVEL. Le Tableau 14 indique les codes pour RUN et LEVEL.

## 6 Processus de décodage

### 6.1 Compensation de mouvement

Cet article décrit la compensation de mouvement pour le mode de prédiction par défaut de la présente Recommandation. On trouvera à l'Annexe D une description de la compensation de mouvement en mode vectoriel cinétique non restreint. On trouvera à l'Annexe F une description de la compensation de mouvement en mode de prédiction avancé.

TABLEAU 11/H.263

Table des codes de longueur variable pour MVD

index	différences vectorielles		nombre de bits	codes
0	-16	16	13	0000 0000 0010 1
1	-15,5	16,5	13	0000 0000 0011 1
2	-15	17	12	0000 0000 0101
3	-14,5	17,5	12	0000 0000 0111
4	-14	18	12	0000 0000 1001
5	-13,5	18,5	12	0000 0000 1011
6	-13	19	12	0000 0000 1101
7	-12,5	19,5	12	0000 0000 1111
8	-12	20	11	0000 0001 001
9	-11,5	20,5	11	0000 0001 011
10	-11	21	11	0000 0001 101
11	-10,5	21,5	11	0000 0001 111
12	-10	22	11	0000 0010 001
13	-9,5	22,5	11	0000 0010 011
14	-9	23	11	0000 0010 101
15	-8,5	23,5	11	0000 0010 111
16	-8	24	11	0000 0011 001
17	-7,5	24,5	11	0000 0011 011
18	-7	25	11	0000 0011 101
19	-6,5	25,5	11	0000 0011 111
20	-6	26	11	0000 0100 001
21	-5,5	26,5	11	0000 0100 011
22	-5	27	10	0000 0100 11
23	-4,5	27,5	10	0000 0101 01
24	-4	28	10	0000 0101 11
25	-3,5	28,5	8	0000 0111
26	-3	29	8	0000 1001
27	-2,5	29,5	8	0000 1011
28	-2	30	7	0000 111
29	-1,5	30,5	5	0001 1
30	-1	31	4	0011
31	-0,5	31,5	3	011
32	0		1	1
33	0,5	-31,5	3	010
34	1	-31	4	0010
35	1,5	-30,5	5	0001 0
36	2	-30	7	0000 110
37	2,5	-29,5	8	0000 1010
38	3	-29	8	0000 1000
39	3,5	-28,5	8	0000 0110
40	4	-28	10	0000 0101 10
41	4,5	-27,5	10	0000 0101 00
42	5	-27	10	0000 0100 10
43	5,5	-26,5	11	0000 0100 010
44	6	-26	11	0000 0100 000
45	6,5	-25,5	11	0000 0011 110
46	7	-25	11	0000 0011 100
47	7,5	-24,5	11	0000 0011 010
48	8	-24	11	0000 0011 000
49	8,5	-23,5	11	0000 0010 110
50	9	-23	11	0000 0010 100
51	9,5	-22,5	11	0000 0010 010
52	10	-22	11	0000 0010 000
53	10,5	-21,5	11	0000 0001 110
54	11	-21	11	0000 0001 100
55	11,5	-20,5	11	0000 0001 010
56	12	-20	11	0000 0001 000
57	12,5	-19,5	12	0000 0000 1110
58	13	-19	12	0000 0000 1100
59	13,5	-18,5	12	0000 0000 1010
60	14	-18	12	0000 0000 1000
61	14,5	-17,5	12	0000 0000 0110
62	15	-17	12	0000 0000 0100
63	15,5	-16,5	13	0000 0000 0011 0

INTRADC	TCOEF
---------	-------

FIGURE 10/H.263

**Structure de la couche des blocs**

TABLEAU 12/H.263

**Niveaux de reconstruction pour coefficient DC en mode INTRA**

index	code de longueur fixe		niveau de reconstruction avant transformée inverse
0	0000 0001	(1)	8
1	0000 0010	(2)	16
2	0000 0011	(3)	24
.	.	.	.
.	.	.	.
126	0111 1111	(127)	1016
127	1111 1111	(255)	1024
128	1000 0001	(129)	1032
.	.	.	.
.	.	.	.
252	1111 1101	(253)	2024
253	1111 1110	(254)	2032

**6.1.1 Vecteurs cinétiques différentiels**

Le vecteur de macrobloc s'obtient par addition des prédicteurs aux différences vectorielles indiquées par le mot MVD (voir le Tableau 11). On trouvera à l'Annexe F une description du codage différentiel avec quatre vecteurs par macrobloc. Dans le cas de vecteur unique par macrobloc, les prédicteurs pouvant être pris en compte pour le codage différentiel sont extraits de trois macroblocs adjacents, comme indiqué sur la Figure 11. Les prédicteurs sont calculés séparément pour les composantes horizontale et verticale.

Dans les cas particuliers des bords du groupe de blocs ou de l'image en cours, les règles décisionnelles suivantes sont appliquées, en ordre croissant:

- 1) le prédicteur candidat est mis à zéro lorsque le macrobloc correspondant a été codé en mode INTRA (s'il n'est pas déjà en mode de trames PB) ou n'a pas été codé (COD = 1);
- 2) le prédicteur candidat MV1 est mis à zéro si le macrobloc correspondant est extérieur à l'image (du côté gauche);
- 3) puis les prédicteurs candidats MV2 et MV3 sont mis à la valeur de MV1 si les macroblocs correspondants sont à l'extérieur (du haut) de l'image ou à l'extérieur du groupe GOB (en haut) si l'en-tête du groupe de blocs en cours n'est pas vide;
- 4) puis le prédicteur candidat MV3 est mis à zéro si le macrobloc correspondant est à l'extérieur (à droite) de l'image.

Pour chaque composante, le prédicteur est la valeur médiane des trois prédicteurs candidats pour cette composante.

On tire parti du fait que l'étendue des valeurs des composantes vectorielles cinétiques est bornée. Chaque mot de code à longueur variable pour données MVD représente une paire de valeurs différentielles. Une seule de ces paires fournira une composante vectorielle de macrobloc s'inscrivant dans l'intervalle permis [-16, 15,5]. Une valeur positive de la composante horizontale ou verticale du vecteur cinétique signifie que la prédiction est formée à partir de pixels de l'image précédente, spatialement situés à droite ou au-dessous des pixels en cours de prédiction.

TABLEAU 13/H.263

Table des codes de longueur variable pour le mot TCOEF

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
0	0	0	1	3	10s
1	0	0	2	5	1111s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110s
19	0	2	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0101 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s
31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100s
45	0	14	1	9	0001 1011s
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	0000 1111 0s
50	0	19	1	10	0000 1110 1s

TABLEAU 13/H.263 (fin)

Table des codes de longueur variable pour le mot TCOEF

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s
57	0	26	1	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	4	1	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	14	1	9	0001 0101s
76	1	15	1	9	0001 0100s
77	1	16	1	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s
89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ÉCHAPPEMENT			7	0000 011

TABLEAU 14/H.263

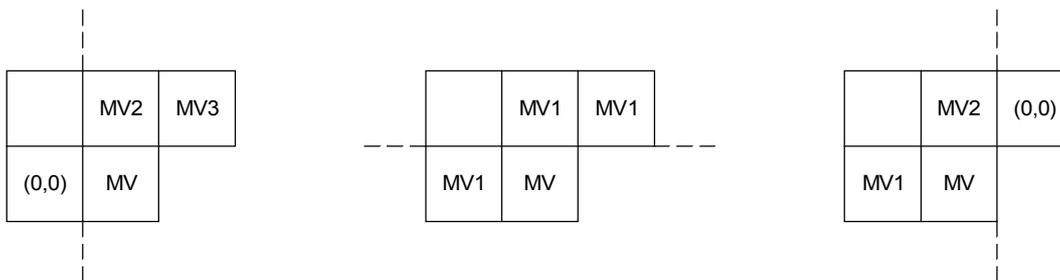
Table des codes de longueur fixe pour les longueurs (RUN) et les amplitudes (LEVEL)

Index	Run	Code
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Index	Level	Code
–	–128	INTERDIT
0	–127	1000 0001
.	.	.
125	–2	1111 1110
126	–1	1111 1111
–	0	INTERDIT
127	1	0000 0001
128	2	0000 0010
.	.	.
253	127	0111 1111

Le vecteur cinétique est utilisé pour tous les pixels des quatre blocs de luminance contenus dans le macrobloc. Pour calculer les vecteurs cinétiques des deux blocs de chrominance, on divise par deux les valeurs constituantes du vecteur de macrobloc, en raison du format inférieur de la chrominance. Les valeurs constituantes des vecteurs résultants, de résolution au quart de pixel près, sont modifiées pour s'approcher de la plus proche position médiane entre deux pixels, comme indiqué dans le Tableau 15.

	MV2	MV3
MV1	MV	



T1519590-95/d05

- limite d'image ou de groupe GOB
- MV vecteur cinétique en cours (*current motion vector*)
- MV1 vecteur cinétique précédent (*previous motion vector*)
- MV2 vecteur cinétique supérieur (*above motion vector*)
- MV3 vecteur cinétique supérieur droit (*above right motion vector*)

FIGURE 11/H.263  
Prédiction des vecteurs cinétiques

6.1.2 Interpolation pour la prédiction en fraction de pixel

On détermine les valeurs des demi-pixels en utilisant une interpolation bilinéaire comme décrit dans la Figure 12. Une barre oblique («/») indique une division par troncature.

TABLEAU 15/H.263

**Modification des composantes vectorielles de chrominance en résolution au quart de pixel**

position au quart de pixel près	0	1/4	1/2	3/4	1
position résultante	0	1/2	1/2	1/2	1

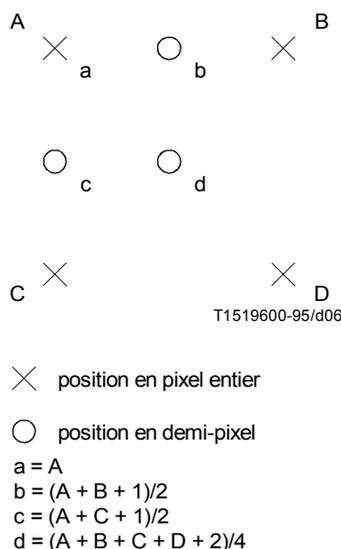


FIGURE 12/H.263

**Prédiction au demi-pixel près par interpolation bilinéaire**

**6.2 Décodage des coefficients**

**6.2.1 Quantification inverse**

Si LEVEL = «0», le niveau de reconstruction REC = «0». Le niveau de reconstruction du coefficient INTRADC est donné par le Tableau 12. Les niveaux de reconstruction de tous les coefficients non nuls autres que le coefficient INTRADC sont donnés par les formules suivantes:

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) \quad \text{si QUANT} = \text{«impair»}$$

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) - 1 \quad \text{si QUANT} = \text{«pair»}$$

On notera que ce processus élimine les nombres de valeur paire. On a constaté que cela évitait l'accumulation d'erreurs dues à une désadaptation de la transformée inverse (IDCT) (*inverse discrete cosine transform*). Après calcul de la valeur |REC|, on ajoute son signe pour obtenir la grandeur REC, soit:  $REC = \text{sign}(\text{LEVEL}) \cdot |REC|$

Le terme  $\text{sign}(\text{LEVEL})$  est donné par le dernier bit du code de coefficient TCOEF (voir le Tableau 13) ou par le Tableau 14.

**6.2.2 Ecrêtage des niveaux de reconstruction**

Après quantification inverse, les niveaux de reconstruction de tous les coefficients autres que le coefficient INTRADC sont écrêtés jusqu'à l'étendue de -2048 à 2047.

**6.2.3 Positionnement en zigzag**

Les coefficients quantifiés de la transformée sont rangés en blocs de  $8 \times 8$  éléments, conformément à la séquence indiquée dans la Figure 13. Le coefficient 1 est le coefficient dc.

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

FIGURE 13/H.263

**Positionnement en zigzag des coefficients quantifiés de la transformée**

**6.2.4 Transformation inverse**

Après quantification inverse et positionnement en zigzag des coefficients, on applique aux blocs de  $8 \times 8$  éléments résultants une transformation inverse en cosinus discrets à deux dimensions séparables, à  $8 \times 8$  coefficients. Le signal de sortie de la transformation inverse va de  $-256$  à  $+255$  après écrêtage pour être codé avec 9 bits. La fonction de transfert de la transformée inverse est donnée par l'équation suivante:

$$f(x, y) = 1/4 \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v)\cos [\pi(2x + 1)u/16] \cos [\pi(2y + 1)v/16]$$

avec  $u, v, x, y = 0, 1, 2, \dots, 7$

où:

$x, y$  sont les coordonnées spatiales dans le domaine des pixels;

$u, v$  sont les coordonnées dans le domaine de la transformée;

$C(u) = 1/\sqrt{2}$  pour  $u = 0$ , sinon  $C(u) = 1$ ;

$C(v) = 1/\sqrt{2}$  pour  $v = 0$ , sinon  $C(v) = 1$ .

NOTE – A l'intérieur du bloc en cours de transformation, les coordonnées  $x = 0$  et  $y = 0$  se rapportent au pixel situé, respectivement, le plus près des bords gauche et supérieur de l'image.

Les procédures arithmétiques permettant de calculer la transformée inverse ne sont pas définies mais il convient qu'elles satisfassent à la tolérance d'erreur spécifiée dans l'Annexe A.

**6.3 Reconstruction des blocs**

**6.3.1 Sommation**

Après compensation du mouvement et décodage des coefficients (y compris la transformation inverse), un processus de reconstruction est appliqué à chaque bloc de luminance et de chrominance. Pour les blocs INTRA, la reconstruction équivaut au résultat de la transformation inverse. Pour les blocs de type INTER, la reconstruction est effectuée par sommation de la prédiction et du résultat de la transformation inverse. Cette sommation est effectuée au niveau des pixels.

**6.3.2 Ecrêtage**

Pour éviter que la distorsion d'amplitude des coefficients de la transformée, due à la quantification, ne provoque un débordement arithmétique dans les circuits en boucle du codeur et du décodeur, des fonctions d'écrêtage y sont insérées. L'écrêteur intervient après la sommation de la prédiction et de l'erreur de prédiction après reconstruction. Il prend en compte des valeurs résultantes de pixel inférieures à 0 ou supérieures à 255, qui sont respectivement ramenées à 0 et à 255.

## Annexe A

### Spécification de la précision de la transformée inverse

(Cette annexe fait partie intégrante de la présente Recommandation)

**A.1** Générer des valeurs de pixels entières aléatoires allant de  $-L$  à  $+H$ , selon le générateur de nombres aléatoires ci-dessous (version C). Disposer en blocs de  $8 \times 8$ . Des ensembles de données de 10 000 blocs chacun doivent être générés pour  $(L = 256, H = 255)$ ,  $(L = H = 5)$  et  $(L = H = 300)$ .

**A.2** Pour chaque bloc de  $8 \times 8$ , effectuer une transformée en cosinus discrète directe, orthonormale séparable, avec multiplication matricielle, en virgule flottante sur au moins 64 bits conformément à la fonction de transfert suivante:

$$F(u, v) = 1/4 C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos [\pi(2x + 1)u/16] \cos [\pi(2y + 1)v/16]$$

avec  $u, v, x, y = 0, 1, 2, \dots, 7$

où:

$x, y$  sont les coordonnées spatiales dans le domaine des pixels;

$u, v$  sont les coordonnées dans le domaine de la transformée;

$C(u) = 1/\sqrt{2}$  pour  $u = 0$ ; sinon  $C(u) = 1$ ;

$C(v) = 1/\sqrt{2}$  pour  $v = 0$ ; sinon  $C(v) = 1$ .

**A.3** Pour chaque bloc, arrondir les 64 coefficients de la transformée ainsi obtenus à la valeur entière la plus proche. Tronquer les valeurs à l'intervalle  $[-2048$  à  $+2047]$ . On obtient ainsi des données d'entrée à 12 bits pour la transformée inverse.

**A.4** Pour chaque bloc de  $8 \times 8$  données à 12 bits produit en A.3, effectuer une transformée discrète inverse (IDCT) (*inverse discrete transform*) en cosinus, orthonormale, séparable, avec multiplication matricielle, en virgule flottante sur au moins 64 bits. Arrondir les valeurs ainsi obtenues au nombre entier le plus proche, et les tronquer à l'intervalle  $[-256$  à  $+255]$ . Ces blocs de  $8 \times 8$  éléments constituent les données de sortie de référence de la transformée en cosinus discrète inverse.

**A.5** Pour chaque bloc de  $8 \times 8$  données produit en A.3, utiliser le dispositif à l'essai de transformation en cosinus discrète inverse, en tronquant la sortie à l'intervalle  $[-256$  à  $+255]$ . Ces blocs de  $8 \times 8$  pixels constituent les données de sortie d'essai de la transformée en cosinus discrète inverse.

**A.6** Pour chacun des 64 pixels de sortie de la transformée en cosinus discrète inverse, et pour chacun des ensembles de 10 000 blocs produits, mesurer l'erreur maximale, l'erreur moyenne et l'erreur quadratique moyenne entre les données de référence et les données d'essai.

- A.7**
- Pour tout pixel, l'erreur absolue maximale ne doit pas être supérieure à 1.
  - Pour tout pixel, l'erreur quadratique moyenne ne doit pas dépasser 0,06.
  - Sur l'ensemble, l'erreur quadratique moyenne ne doit pas dépasser 0,02.
  - Pour tout pixel, l'erreur absolue moyenne ne doit pas être supérieure à 0,015.
  - Sur l'ensemble, l'erreur absolue moyenne ne doit pas être supérieure à 0,0015.

**A.8** Tous les zéros à l'entrée doivent donner des zéros à la sortie.

**A.9** Procéder à une seconde série de mesures en utilisant exactement les mêmes valeurs de données qu'en A.1, mais en inversant le signe de chaque pixel.

*Programme «C» de génération de nombres aléatoires*

*/\* L et H sont des entiers longs de 32 bits \*/*

```
long rand (L,H)
long      L,H;
{
```

```

static long randx = 1;          /* long est à 32 bits */
static double z = (double) 0x7fffffff;

long      i,j;
double    x;                  /* double est à 64 bits */

randx = (randx * 1103515245) + 12345;
i = randx & 0x7fffffff;      /* conserver 30 bits */
x = ( (double)i ) / z;       /* valeur de 0 à 0,99999 ... */
x *= (L+H+1);                /* valeur de 0 à < L+H+1 */
j = x;                       /* partie entière */
return(j - L);               /* valeur de -L à H */
}

```

## Annexe B

### Décodeur fictif de référence

(Cette annexe fait partie intégrante de la présente Recommandation)

Le décodeur fictif de référence (HRD) (*hypothetical reference decoder*) est défini comme suit.

**B.1** Travaillant en synchronisme, le décodeur HRD et le codeur doivent avoir la même fréquence d'horloge ainsi que le même débit de format CIF.

**B.2** Le niveau d'utilisation du tampon récepteur du décodeur HRD est égal à  $(B + BPP_{\max}Kb * 1024 \text{ bits})$  où le produit  $(BPP_{\max}Kb * 1024)$  est le nombre maximal d'éléments binaires par image qui a été négocié pour utilisation dans le flux binaire (voir 3.6). La valeur de  $B$  est définie comme suit:

$$B = 4 \cdot R_{\max} / 29,97$$

où 29,97 est exprimé en hertz et où  $R_{\max}$  est le débit binaire vidéo pendant la connexion, en bits par seconde. Cette valeur de  $B$  est minimale. Un codeur peut utiliser une plus grande valeur de  $B$ , à condition que celle-ci soit préalablement négociée par des moyens externes, par exemple la Recommandation H.245.

La valeur de  $R_{\max}$  dépend de la configuration du système (par exemple RTC ou RNIS, liaison simple ou multiple). Elle peut être égale au débit binaire maximal qui est supporté par la liaison physique. La négociation de  $R_{\max}$  est effectuée par des moyens externes, comme la Recommandation H.245.

**B.3** Le décodeur HRD est initialement vide.

**B.4** La mémoire tampon du décodeur HRD est examinée aux intervalles du format CIF (1000/29,97 ms). Si au moins une image codée complète est dans le tampon, toutes les données de l'image la plus récente sont instantanément supprimées (par exemple à l'instant  $t_{n+1}$  dans la Figure B.1). Immédiatement après la suppression de ces données, le niveau d'utilisation du tampon doit être inférieur à  $B$ . Cette prescription est applicable au flux binaire à la sortie du codeur (y compris les données codées d'image et le type MCBPC et le mot de bourrage STUF) mais non aux bits de verrouillage pour la correction d'erreur, à l'indicateur de remplissage (Fi), aux bits de remplissage ou aux informations de parité pour la correction d'erreur, décrits dans l'Annexe H.

Pour les besoins de la présente définition, une image complète codée est une image I ou P normale ou une trame PB.

Pour répondre à cette prescription, le nombre de bits pour la  $(n+1)$  ième image,  $d_{n+1}$ , doit satisfaire à la relation suivante:

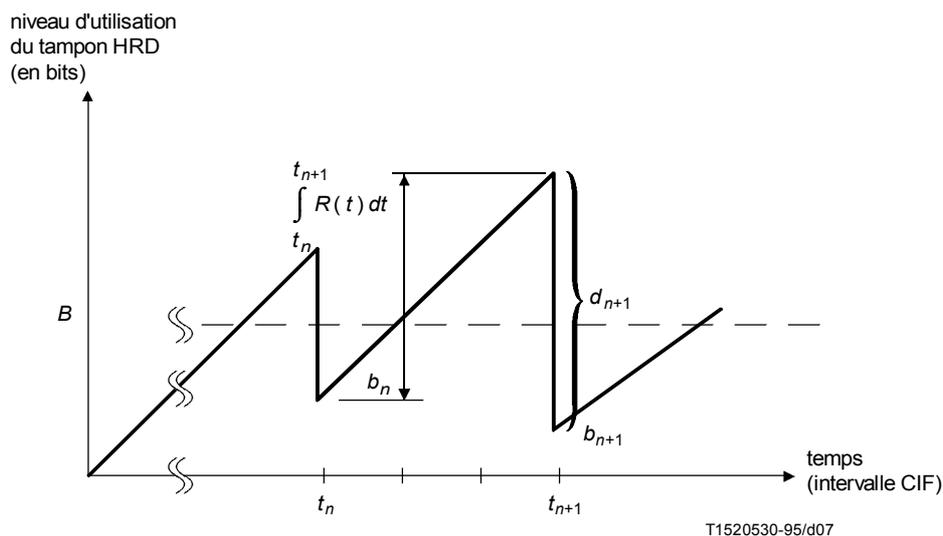
$$d_{n+1} \geq b_n + \int_{t_n}^{t_{n+1}} R(t)dt - B$$

où:

$b_n$  est le niveau d'utilisation de la mémoire tampon immédiatement après l'instant  $t_n$ ;

$t_n$  est l'instant où la  $n$  ième image codée est supprimée du tampon du décodeur HRD;

$R(t)$  est le débit vidéo à l'instant  $t$ .



NOTE – La durée  $(t_{n+1} - t_n)$  est un nombre entier de périodes d'image au format CIF (1/29,97, 2/29,97, 3/29,97, ...).

FIGURE B.1/H.263  
Niveau d'utilisation du tampon de décodeur HRD

## Annexe C

### Considérations relatives aux communications multipoints

(Cette annexe fait partie intégrante de la présente Recommandation)

Les facilités suivantes sont offertes à l'appui du fonctionnement multipoint en mode commuté.

#### C.1 Demande de gel d'image

Cette demande fait que le décodeur gèle son image affichée jusqu'à la réception d'un signal de fin de gel d'image ou jusqu'à l'expiration d'une période de temporisation d'au moins 6 s. La transmission de ce signal emprunte des moyens extérieurs (par exemple selon la Recommandation H.245).

#### C.2 Demande de rafraîchissement rapide

Cette demande fait que le décodeur code son image suivante en mode INTRA avec des paramètres de codage propres à éviter tout débordement de mémoire tampon. La transmission de ce signal emprunte des moyens extérieurs (par exemple selon la Recommandation H.245).

#### C.3 Fin de gel d'image

Signal émis par un codeur en réponse à une demande de rafraîchissement rapide; le décodeur passe du mode gel au mode affichage normal des images codées. Ce signal est transmis par le mot PTYPE (voir 5.1.3) contenu dans l'en-tête de la première image codée en réponse à la demande de rafraîchissement rapide.

#### C.4 Communication multipoint à présence continue (CPM) (*continuous presence multipoint*) (mode non utilisé pour la Recommandation H.324)

La présente Recommandation prévoit donc la possibilité de négocier un mode de communication multipoint à présence continue dans lequel jusqu'à quatre flux binaires QCIF indépendants peuvent être multiplexés sous la forme de quatre «sous-flux binaires» dans un même flux binaire vidéo nouveau, au moyen des champs indicateurs PSBI et GSBI. L'échange de possibilités des terminaux est effectué par des moyens extérieurs (par exemple selon la Recommandation H.242).

En mode de présence CPM, le champ CPM doit être mis à «1» dans chacun des flux binaires H.263 indépendants. Les indicateurs de sous-flux binaire (SBI) (*sub-bit stream indicator*) contenus dans les en-têtes d'image et de groupe GOB de chaque flux binaire H.263 indiquent les numéros de flux binaire auxquels appartiennent ces en-têtes et toutes les informations suivantes jusqu'au prochain en-tête d'image ou de groupe GOB contenus dans le flux binaire vidéo composite.

Chaque sous-flux binaire est considéré comme un flux binaire H.263 normal. Il doit donc être conforme aux possibilités qui sont échangées par moyens externes. Les informations relatives aux différents flux binaires H.263 ne sont pas transmises dans un ordre spécialement prédéfini. Un indicateur SBI peut avoir une valeur quelconque, indépendamment des indicateurs SBI précédents. Les fréquences d'image peuvent également être différentes selon les divers flux binaires H.263. Les informations contenues dans chaque flux binaire individuel sont également indépendantes de toutes les autres informations contenues dans les autres flux binaires. Par exemple, les mots de code GFID d'un sous-flux binaire donné ne sont pas influencés par les mots de code GFID ou PTYPE d'autres sous-flux binaires.

### Annexe D

#### Mode de codage par vecteurs cinétiques non restreints (Cette annexe fait partie intégrante de la présente Recommandation)

La présente annexe décrit le mode de codage H.263 par vecteurs cinétiques non restreints. La possibilité de ce mode H.263 est signalée par des moyens externes (par exemple selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le mot PTYPE.

#### D.1 Vecteurs cinétiques pointant hors des limites d'image

Dans le mode de prédiction par défaut de la présente Recommandation, les vecteurs cinétiques sont restreints de telle manière que tous les pixels sur lesquels ils pointent soient à l'intérieur de la zone d'image codée (voir 4.2.3). Dans le mode de codage par vecteurs cinétiques non restreints, cette restriction est toutefois supprimée et les vecteurs cinétiques *sont* donc autorisés à pointer hors de l'image. Lorsqu'un pixel pointé par un vecteur cinétique est à l'extérieur de la zone d'image codée, on fait alors appel à un pixel périphérique. On détermine ce pixel périphérique en limitant le vecteur cinétique à la dernière position de pixel entier située à l'intérieur de la zone d'image codée. La limitation du vecteur cinétique est effectuée pixel par pixel et séparément pour chaque composante du vecteur cinétique.

Par exemple, si le mode vectoriel cinétique sans restriction est utilisé pour une image au format QCIF, la valeur de pixel pour la composante de luminance de l'image référencée est donnée par la formule suivante:

$$\text{Rumv}(x, y) = \text{R}(x', y')$$

où:

$x, y, x', y'$	sont les coordonnées spatiales dans le domaine des pixels;
$\text{Rumv}(x, y)$	est la valeur de pixel de l'image référencée au point $(x, y)$ en mode de vecteurs cinétiques non restreints;
$\text{R}(x', y')$	est la valeur de pixel de l'image référencée au point $(x', y')$ en mode de vecteurs cinétiques non restreints;
$x' = 0$	si $x < 0$ ;
$x' = 175$	si $x > 175$ ;
$x' = x$	dans les autres cas;

$$\begin{aligned}
y' &= 0 && \text{si } y < 0; \\
y' &= 143 && \text{si } y > 143; \\
y' &= y && \text{dans les autres cas;}
\end{aligned}$$

et la surface d'image codée par la valeur  $R(x', y')$  est définie par les coordonnées suivantes:  $0 \leq x' \leq 175$ ,  $0 \leq y' \leq 143$ . Ces limites correspondent à des positions de pixels entiers; dans ces limites, le point  $(x', y')$  peut également être une position en demi-pixels.

## D.2 Extension de la gamme des vecteurs cinétiques

Dans le mode de prédiction par défaut, les valeurs des composantes horizontales et verticales des vecteurs cinétiques sont limitées à l'intervalle  $[-16, 15,5]$  (ce qui vaut également pour les composantes vectorielles cinétiques anticipées et différées pour les images de type B). Dans le mode de vecteurs cinétiques non restreints, l'intervalle maximal des composantes vectorielles est cependant  $[-31,5, 31,5]$ , avec la restriction que seules les valeurs s'inscrivant dans une étendue de  $[-16, 15,5]$  autour du prédicteur pour chaque composante vectorielle cinétique peuvent être atteintes si ce prédicteur est dans l'intervalle  $[-15,5, 16]$ . Si le prédicteur est à l'extérieur de l'intervalle  $[-15,5, 16]$ , on pourra atteindre toutes les valeurs comprises dans l'intervalle  $[-31,5, 31,5]$  ayant le même signe que le prédicteur, plus la valeur zéro. Donc, si la composante vectorielle cinétique est  $MV_c$  et si le prédicteur de cette composante est  $P_c$ , on a:

$$\begin{aligned}
-31,5 \leq MV_c \leq 0 &&& \text{si } -31,5 \leq P_c \leq -16 \\
-16 + P_c \leq MV_c \leq 15,5 + P_c &&& \text{si } -15,5 \leq P_c \leq 16 \\
0 \leq MV_c \leq 31,5 &&& \text{si } 16,5 \leq P_c \leq 31,5
\end{aligned}$$

Dans le mode des vecteurs cinétiques non restreints, l'interprétation du Tableau 10 est la suivante pour les mots MVD, MVD<sub>2-4</sub> et MVDB:

- si le prédicteur de la composante vectorielle cinétique est dans l'intervalle  $[-15,5, 16]$ , seule la première colonne des différences vectorielles est applicable;
- si le prédicteur de la composante vectorielle cinétique est hors de l'intervalle  $[-15,5, 16]$ , on doit extraire du Tableau 11 la différence vectorielle qui produit une composante vectorielle comprise dans l'intervalle  $[-31,5, 31,5]$  avec le même signe que son prédicteur ( $y$  compris zéro).

Le prédicteur pour les données  $MV_D$  et  $MVD_{2-4}$  est défini comme étant la médiane de l'étendue des composantes vectorielles  $MV_1$ ,  $MV_2$  et  $MV_3$ , telles que définies dans 6.1.1 et dans F.2. Pour les données MVDB, le prédicteur  $P_c = (TR_B \times MV) / TR_D$  (où  $MV$  représente une composante vectorielle pour un bloc de  $8 * 8$  éléments de luminance dans une image de type P (voir également G.4).

## Annexe E

### Mode de codage arithmétique syntaxique (SAC)

(Cette annexe fait partie intégrante de la présente Recommandation)

#### E.1 Introduction

En codage/décodage à longueur variable (VLC/VLD) (*variable length coding/decoding*) conformément à l'article 5, chaque symbole est codé en longueur variable à l'aide d'une table spécifique utilisant la syntaxe du codeur. Cette table contient normalement les longueurs et les valeurs des mots à codage VLC. Le symbole est mis en correspondance avec une entrée de la table au cours d'une opération de recherche dans la table. Puis le mot de code binaire spécifié par l'entrée est transféré normalement vers une mémoire tampon pour envoi au récepteur. En décodage à longueur variable (VLD), le flux binaire reçu est comparé à chaque entrée d'une table spécifique utilisant la syntaxe du codeur. Cette table doit être la même que celle qui a été utilisée dans le codeur pour coder le symbole en cours. L'entrée adaptée qui a été trouvée dans la table est ensuite remise en correspondance avec le symbole correspondant, qui représente le résultat final du décodeur VLD et qui est ensuite utilisé pour récupérer les images vidéo. Ce processus de codage/décodage à longueur variable (VLC/VLD) implique que chaque symbole soit codé sous forme d'un nombre entier et fixe d'éléments binaires. La suppression (que l'on peut réaliser par codage arithmétique) de cette restriction relative au nombre entier et fixe d'éléments binaires pour les symboles peut procurer des réductions du débit binaire final.

La présente annexe décrit le mode facultatif de codage arithmétique syntaxique (SAC) (*syntax-based arithmetic*) de la présente Recommandation. Dans ce mode, toutes les opérations correspondantes de codage/décodage à longueur variable de la présente Recommandation sont remplacées par des opérations de codage/décodage arithmétique. La capacité d'activation de ce mode est signalée par des moyens externes (par exemple selon la Recommandation H.245). L'emploi de ce mode est indiqué dans le mot PTYPE.

## E.2 Spécification du codeur en mode SAC

En mode de codage SAC, un symbole est codé au moyen d'une table spécifique d'entiers (ou d'un modèle) utilisant la syntaxe du codeur et au moyen de la procédure suivante, qui est spécifiée en langage C.

```
#define    q1    16384
#define    q2    32768
#define    q3    49152
#define    top   65535

static long    low, high, opposite_bits, length;

void    encode_a_symbol(int index, int cumul_freq[ ])
{
    length = high - low + 1;
    high = low - 1 + (length * cumul_freq[index]) / cumul_freq[0];
    low += (length * cumul_freq[index+1]) / cumul_freq[0];
    for ( ; ; ) {
        if (high < q2) {
            send out a bit "0" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "1" to PSC_FIFO;
                opposite_bits--;
            }
        }
        else if (low >= q2) {
            send out a bit "1" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "0" to PSC_FIFO;
                opposite_bits--;
            }
            low -= q2;
            high -= q2;
        }
        else if (low >= q1 && high < q3) {
            opposite_bits += 1;
            low -= q1;
            high -= q1;
        }
        else break;

        low *= 2;
        high = 2 * high + 1;
    }
}
```

Les valeurs des paramètres low, high et opposite\_bits sont respectivement initialisées à 0, top et 0. L'élément PSC\_FIFO est un registre du type premier entré-premier sorti (*first in first out*) pour la mise en mémoire tampon des bits issus du codeur arithmétique. Le modèle est spécifié au moyen du paramètre cumul\_freq[ ] et le symbole est spécifié au moyen de son index dans le modèle.

## E.3 Spécification du décodeur en mode SAC

Dans un décodeur en mode SAC, un symbole est décodé au moyen d'un modèle syntaxique spécifique et au moyen de la procédure suivante, qui est spécifiée en langage C.

```
static long    low, high, code_value, bit, length, index, cum;

int    decode_a_symbol(int cumul_freq[ ])
{
    length = high - low + 1;
    cum = (-1 + (code_value - low + 1) * cumul_freq[0]) / length;
    for (index = 1; cumul_freq[index] > cum; index++);
}
```

```

high = low - 1 + (length * cumul_freq[index-1]) / cumul_freq[0];
low += (length * cumul_freq[index]) / cumul_freq[0];
for ( ; ; ) {
    if (high < q2);
    else if (low >= q2) {
        code_value -= q2;
        low -= q2;
        high -= q2;
    }
    else if (low >= q1 && high < q3) {
        code_value -= q1;
        low -= q1;
        high -= q1;
    }
    else break;

    low *= 2;
    high = 2 * high + 1;
    get bit from PSC_FIFO;
    code_value = 2 * code_value + bit;
}
return (index-1);
}
}

```

De nouveau, le modèle est spécifié au moyen du paramètre `cumul_freq[ ]`. Le symbole décodé est renvoyé, au moyen de son index, dans le modèle. L'élément `PSC_FIFO` est un registre du type premier entré-premier sorti pour la mise en mémoire tampon du flux binaire entrant. Le décodeur est initialisé de façon à commencer le décodage d'un flux binaire arithmétique codé, au moyen de la procédure suivante.

```

void decoder_reset()
{
    code_value = 0;
    low = 0;
    high = top;
    for (int i = 1; i <= 16; i++) {
        get bit from PSC_FIFO;
        code_value = 2 * code_value + bit;
    }
}

```

#### E.4 Syntaxe

Comme dans le mode par table de codes à longueur variable (VLC), la syntaxe des symboles est subdivisée en quatre couches: les images, les groupes de blocs, les macroblocs et les blocs. La syntaxe des trois couches supérieures reste exactement la même. Celle de la couche des blocs reste également très similaire mais est illustrée sur la Figure E.1.

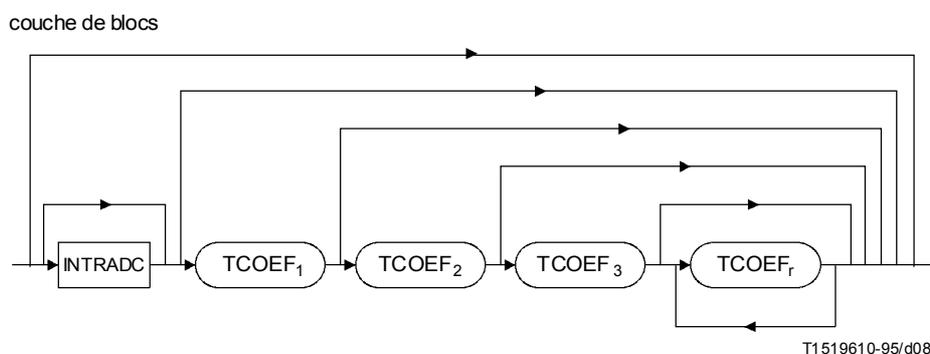


FIGURE E.1/H.263

Structure de la couche des blocs en codage SAC

Dans la Figure E.1, les séquences TCOEF<sub>1</sub>, TCOEF<sub>2</sub>, TCOEF<sub>3</sub> et TCOEF<sub>r</sub> sont les symboles LAST-RUN-LEVEL définis au 5.4.2 et peuvent être, respectivement, le premier, le deuxième, le troisième, etc. des symboles. Les séquences TCOEF<sub>1</sub>, TCOEF<sub>2</sub>, TCOEF<sub>3</sub> et TCOEF<sub>r</sub> ne sont présentes que lorsqu'un, deux, trois ou plus de trois coefficients sont respectivement présents dans la couche des blocs.

## E.5 Registre PSC\_FIFO

Dans un codeur ou un décodeur, le registre PSC\_FIFO est un tampon de type premier entré-premier sorti de capacité supérieure à 17 bits. Dans le registre PSC\_FIFO d'un codeur, on localise et on évite les émulations interdites des mots PSC et GBSC en insérant un «1» de bourrage après chaque apparition successive de 14 zéros (ne faisant pas partie du mot PSC ou GBSC). Dans le registre PSC\_FIFO du décodeur, le premier «1» qui suit chaque chaîne de 14 zéros est supprimé; si par contre une séquence de 14 zéros est suivie d'un «0», cela indique qu'un mot PSC ou GBSC autorisé a été détecté. L'emplacement exact du mot PSC ou GBSC est déterminé par le prochain «1» suivant la séquence de zéros.

## E.6 Symboles de longueur fixe

Les symboles de longueur fixe forment trois chaînes possibles: PSC--TR--PTYPE--PQUANT--CPM--(PSBI)--(TRB-DBQUANT)--PEI--(PSPARE--PEI--...), (GSTUF)--GBSC--GN--(GSBI)--GFID--GQUANT, et (ESTUF)--EOS--(PSTUF). Ces chaînes sont envoyées directement au registre PSC\_FIFO comme dans le mode normal par table de codage VLC selon la présente Recommandation du côté codeur, et sont directement transférées du registre PSC\_FIFO dans le décodeur après la détection d'un mot PSC ou GBSC ou EOS autorisé.

Si une chaîne de symboles de longueur fixe n'est pas la première dans une session vidéo, le codeur arithmétique a besoin d'être réinitialisé avant que l'appel de la procédure suivante ne fasse envoyer cette chaîne de symboles de longueur fixe. Cette procédure doit aussi être appelée à la fin d'une session vidéo ou avant une fin de séquence (EOS).

```
void    encoder_flush()
{
    opposite_bits++;
    if (low < q1) {
        send out a bit "0" to PSC_FIFO;
        while (opposite_bits > 0) {
            send out a bit "1" to PSC_FIFO;
            opposite_bits--;
        }
    }
    else {
        send out a bit "1" to PSC_FIFO;
        while (opposite_bits > 0) {
            send out a bit "0" to PSC_FIFO;
            opposite_bits--;
        }
    }
    low = 0;
    high = top;
}
```

Dans le décodeur, la procédure **decoder\_reset** est appelée après chaque chaîne de symboles de longueur fixe.

## E.7 Symboles de longueur variable

Les modèles pour les symboles de longueur variable sont décrits dans E.8. Les index indiqués dans les tables de codage de longueur variable de l'article 5 sont utilisés pour indexer les entiers contenus dans les modèles.

Les modèles des mots COD et MCBPC contenus dans des images P sont nommés par les symboles *cumf\_COD* et *cumf\_MCBPC*. L'index pour le mot COD=0 est 0 et l'index pour COD=1 est 1. Les index pour le mot MCBPC sont définis dans le Tableau 4 pour les images I et dans le Tableau 5 pour les images P. Le modèle pour le mot MCBPC contenu dans les images I est nommé par le symbole *cumf\_MCBPC\_intra*.

Le modèle pour le mot MODB est *cumf\_MODB*. Les index pour MODB sont définis dans le Tableau 8. Le modèle pour CBPB<sub>n</sub>, n=1,2,...,4, est *cumf\_YCBPB* et le modèle pour CBPB<sub>n</sub>, n=5,6, est *cumf\_UVCBPB*, avec l'index 0 pour CBPB<sub>n</sub>=0 et l'index 1 pour CBPB<sub>n</sub>=1.

Le modèle pour le mot CBPY est *cumf\_CBPY* dans les macroblocs INTER et *cumf\_CBPY\_intra* dans les macroblocs INTRA. Le modèle pour le mot DQUANT est *cumf\_DQUANT*. L'indexage pour CBPY et DQUANT est défini, respectivement, dans le Tableau 10 et dans le Tableau 9.

Le modèle pour les mots MVD, MVD<sub>2-4</sub> et MVDB est *cumf\_MVD* et le modèle pour le mot INTRADC est *cumf\_INTRADC*. L'indexage est défini, respectivement, dans le Tableau 11 et dans le Tableau 12.

Un mot TCOEF non étendu par échappement se compose d'un symbole de type TCOEF1/2/3/r, suivi d'un symbole SIGN indiquant le signe du mot TCOEF. Les modèles des symboles TCOEF1, TCOEF2, TCOEF3 et TCOEFr dans les blocs INTER sont *cumf\_TCOEF1*, *cumf\_TCOEF2*, *cumf\_TCOEF3*, *cumf\_TCOEFr*. Les modèles pour les blocs INTRA sont *cumf\_TCOEF1\_intra*, *cumf\_TCOEF2\_intra*, *cumf\_TCOEF3\_intra*, *cumf\_TCOEFr\_intra*. Pour tous les mots de type TCOEF, l'indexage est défini dans le Tableau 13. Le modèle du symbole SIGN est *cumf\_SIGN*. L'indexage du symbole SIGN est: 0 pour signe positif et 1 pour signe négatif.

Les modèles pour les mots LAST, RUN, LEVEL après échappement (ESCAPE) sont *cumf\_LAST* (*cumf\_LAST\_intra*), *cumf\_RUN* (*cumf\_RUN\_intra*), *cumf\_LEVEL* (*cumf\_LEVEL\_intra*) pour les blocs INTER (ou INTRA). L'indexage du mot LAST est: 0 pour LAST=0 et 1 pour LAST=1, alors que l'indexage pour les mots RUN et LEVEL est défini dans le Tableau 14.

## E.8 Modèles de codage SAC

```
int cumf_COD[3]={16383, 6849, 0};
```

```
int cumf_MCBPC[22]={16383, 4105, 3088, 2367, 1988, 1621, 1612, 1609, 1608, 496, 353, 195, 77, 22, 17, 12, 5, 4, 3, 2, 1, 0};
```

```
int cumf_MCBPC_intra[10]={16383, 7410, 6549, 5188, 442, 182, 181, 141, 1, 0};
```

```
int cumf_MODB[4]={16383, 6062, 2130, 0};
```

```
int cumf_YCBPB[3]={16383, 6062, 0};
```

```
int cumf_UVCBPB[3]={16383, 491, 0};
```

```
int cumf_CBPY[17]={16383, 14481, 13869, 13196, 12568, 11931, 11185, 10814, 9796, 9150, 8781, 7933, 6860, 6116, 4873, 3538, 0};
```

```
int cumf_CBPY_intra[17]={16383, 13619, 13211, 12933, 12562, 12395, 11913, 11783, 11004, 10782, 10689, 9928, 9353, 8945, 8407, 7795, 0};
```

```
int cumf_DQUANT[5]={16383, 12287, 8192, 4095, 0};
```

```
int cumf_MVD[65]={16383, 16380, 16369, 16365, 16361, 16357, 16350, 16343, 16339, 16333, 16326, 16318, 16311, 16306, 16298, 16291, 16283, 16272, 16261, 16249, 16235, 16222, 16207, 16175, 16141, 16094, 16044, 15936, 15764, 15463, 14956, 13924, 11491, 4621, 2264, 1315, 854, 583, 420, 326, 273, 229, 196, 166, 148, 137, 123, 114, 101, 91, 82, 76, 66, 59, 53, 46, 36, 30, 26, 24, 18, 14, 10, 5, 0};
```

```
int cumf_INTRADC[255]={16383, 16380, 16379, 16378, 16377, 16376, 16370, 16361, 16360, 16359, 16358, 16357, 16356, 16355, 16343, 16238, 16237, 16236, 16230, 16221, 16220, 16205, 16190, 16169, 16151, 16130, 16109, 16094, 16070, 16037, 16007, 15962, 15938, 15899, 15854, 15815, 15788, 15743, 15689, 15656, 15617, 15560, 15473, 15404, 15296, 15178, 15106, 14992, 14868, 14738, 14593, 14438, 14283, 14169, 14064, 14004, 13914, 13824, 13752, 13671, 13590, 13515, 13458, 13380, 13305, 13230, 13143, 13025, 12935, 12878, 12794, 12743, 12656, 12596, 12521, 12443, 12359, 12278, 12200, 12131, 12047, 12002, 11948, 11891, 11828, 11744, 11663, 11588, 11495, 11402, 11288, 11204, 11126, 11039, 10961, 10883, 10787, 10679, 10583, 10481, 10360, 10227, 10113, 9961, 9828, 9717, 9584, 9485, 9324, 9112, 9019, 8908, 8766, 8584, 8426, 8211, 7920, 7663, 7406, 7152, 6904, 6677, 6453, 6265, 6101, 5904, 5716, 5489, 5307, 5056, 4850, 4569, 4284, 3966, 3712, 3518, 3342, 3206, 3048, 2909, 2773, 2668, 2596, 2512, 2370, 2295, 2232, 2166, 2103, 2022, 1956, 1887, 1830, 1803, 1770, 1728, 1674, 1635, 1599, 1557, 1500, 1482, 1434, 1389, 1356, 1317, 1284, 1245, 1200, 1179, 1140, 1110, 1092, 1062, 1044, 1035, 1014, 1008, 993, 981, 954, 936, 912, 894, 876, 864, 849, 828, 816, 801, 792, 777, 756, 732, 690, 660, 642, 615, 597, 576, 555, 522, 489, 459, 435, 411, 405, 396, 387, 375, 360, 354, 345, 344, 329, 314, 293, 278, 251, 236, 230, 224, 215, 214, 208, 199, 193, 184, 178, 169, 154, 127, 100, 94, 73, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 20, 19, 18, 17, 16, 15, 9, 0};
```

```
int cumf_TCOEF1[104]={16383, 13455, 12458, 12079, 11885, 11800, 11738, 11700, 11681, 11661, 11651, 11645, 11641, 10572, 10403, 10361, 10346, 10339, 10335, 9554, 9445, 9427, 9419, 9006, 8968, 8964, 8643, 8627, 8624, 8369, 8354, 8352, 8200, 8192, 8191, 8039, 8036, 7920, 7917, 7800, 7793, 7730, 7727, 7674, 7613, 7564, 7513, 7484, 7466, 7439, 7411, 7389, 7373, 7369, 7359, 7348, 7321, 7302, 7294, 5013, 4819, 4789, 4096, 4073, 3373, 3064, 2674, 2357, 2177, 1975, 1798, 1618, 1517, 1421, 1303, 1194, 1087, 1027, 960, 890, 819, 758, 707, 680, 656, 613, 566, 534, 505, 475, 465, 449, 430, 395, 358, 335, 324, 303, 295, 286, 272, 233, 215, 0};
```

int cumf\_TCOEF2[104]={16383, 13582, 12709, 12402, 12262, 12188, 12150, 12131, 12125, 12117, 12113, 12108, 12104, 10567, 10180, 10070, 10019, 9998, 9987, 9158, 9037, 9010, 9005, 8404, 8323, 8312, 7813, 7743, 7726, 7394, 7366, 7364, 7076, 7062, 7060, 6810, 6797, 6614, 6602, 6459, 6454, 6304, 6303, 6200, 6121, 6059, 6012, 5973, 5928, 5893, 5871, 5847, 5823, 5809, 5796, 5781, 5771, 5763, 5752, 4754, 4654, 4631, 3934, 3873, 3477, 3095, 2758, 2502, 2257, 2054, 1869, 1715, 1599, 1431, 1305, 1174, 1059, 983, 901, 839, 777, 733, 683, 658, 606, 565, 526, 488, 456, 434, 408, 380, 361, 327, 310, 296, 267, 259, 249, 239, 230, 221, 214, 0};

int cumf\_TCOEF3[104]={16383, 13532, 12677, 12342, 12195, 12112, 12059, 12034, 12020, 12008, 12003, 12002, 12001, 10586, 10297, 10224, 10202, 10195, 10191, 9223, 9046, 8999, 8987, 8275, 8148, 8113, 7552, 7483, 7468, 7066, 7003, 6989, 6671, 6642, 6631, 6359, 6327, 6114, 6103, 5929, 5918, 5792, 5785, 5672, 5580, 5507, 5461, 5414, 5382, 5354, 5330, 5312, 5288, 5273, 5261, 5247, 5235, 5227, 5219, 4357, 4277, 4272, 3847, 3819, 3455, 3119, 2829, 2550, 2313, 2104, 1881, 1711, 1565, 1366, 1219, 1068, 932, 866, 799, 750, 701, 662, 605, 559, 513, 471, 432, 403, 365, 336, 312, 290, 276, 266, 254, 240, 228, 223, 216, 206, 199, 192, 189, 0};

int cumf\_TCOEFr[104]={16383, 13216, 12233, 11931, 11822, 11776, 11758, 11748, 11743, 11742, 11741, 11740, 11739, 10203, 9822, 9725, 9691, 9677, 9674, 8759, 8609, 8576, 8566, 7901, 7787, 7770, 7257, 7185, 7168, 6716, 6653, 6639, 6276, 6229, 6220, 5888, 5845, 5600, 5567, 5348, 5327, 5160, 5142, 5004, 4900, 4798, 4743, 4708, 4685, 4658, 4641, 4622, 4610, 4598, 4589, 4582, 4578, 4570, 4566, 3824, 3757, 3748, 3360, 3338, 3068, 2835, 2592, 2359, 2179, 1984, 1804, 1614, 1445, 1234, 1068, 870, 739, 668, 616, 566, 532, 489, 453, 426, 385, 357, 335, 316, 297, 283, 274, 266, 259, 251, 241, 233, 226, 222, 217, 214, 211, 209, 208, 0};

int cumf\_TCOEF1\_intra[104]={16383, 13383, 11498, 10201, 9207, 8528, 8099, 7768, 7546, 7368, 7167, 6994, 6869, 6005, 5474, 5220, 5084, 4964, 4862, 4672, 4591, 4570, 4543, 4397, 4337, 4326, 4272, 4240, 4239, 4212, 4196, 4185, 4158, 4157, 4156, 4140, 4139, 4138, 4137, 4136, 4125, 4124, 4123, 4112, 4111, 4110, 4109, 4108, 4107, 4106, 4105, 4104, 4103, 4102, 4101, 4100, 4099, 4098, 4097, 3043, 2897, 2843, 1974, 1790, 1677, 1552, 1416, 1379, 1331, 1288, 1251, 1250, 1249, 1248, 1247, 1236, 1225, 1224, 1223, 1212, 1201, 1200, 1199, 1198, 1197, 1196, 1195, 1194, 1193, 1192, 1191, 1190, 1189, 1188, 1187, 1186, 1185, 1184, 1183, 1182, 1181, 1180, 1179, 0};

int cumf\_TCOEF2\_intra[104]={16383, 13242, 11417, 10134, 9254, 8507, 8012, 7556, 7273, 7062, 6924, 6839, 6741, 6108, 5851, 5785, 5719, 5687, 5655, 5028, 4917, 4864, 4845, 4416, 4159, 4074, 3903, 3871, 3870, 3765, 3752, 3751, 3659, 3606, 3580, 3541, 3540, 3514, 3495, 3494, 3493, 3474, 3473, 3441, 3440, 3439, 3438, 3425, 3424, 3423, 3422, 3421, 3420, 3401, 3400, 3399, 3398, 3397, 3396, 2530, 2419, 2360, 2241, 2228, 2017, 1687, 1576, 1478, 1320, 1281, 1242, 1229, 1197, 1178, 1152, 1133, 1114, 1101, 1088, 1087, 1086, 1085, 1072, 1071, 1070, 1069, 1068, 1067, 1066, 1065, 1064, 1063, 1062, 1061, 1060, 1059, 1058, 1057, 1056, 1055, 1054, 1053, 1052, 0};

int cumf\_TCOEF3\_intra[104]={16383, 12741, 10950, 10071, 9493, 9008, 8685, 8516, 8385, 8239, 8209, 8179, 8141, 6628, 5980, 5634, 5503, 5396, 5327, 4857, 4642, 4550, 4481, 4235, 4166, 4151, 3967, 3922, 3907, 3676, 3500, 3324, 3247, 3246, 3245, 3183, 3168, 3084, 3069, 3031, 3030, 3029, 3014, 3013, 2990, 2975, 2974, 2973, 2958, 2943, 2928, 2927, 2926, 2925, 2924, 2923, 2922, 2921, 2920, 2397, 2298, 2283, 1891, 1799, 1591, 1445, 1338, 1145, 1068, 1006, 791, 768, 661, 631, 630, 615, 592, 577, 576, 561, 546, 523, 508, 493, 492, 491, 476, 475, 474, 473, 472, 471, 470, 469, 468, 453, 452, 451, 450, 449, 448, 447, 446, 0};

int cumf\_TCOEFr\_intra[104]={16383, 12514, 10776, 9969, 9579, 9306, 9168, 9082, 9032, 9000, 8981, 8962, 8952, 7630, 7212, 7053, 6992, 6961, 6940, 6195, 5988, 5948, 5923, 5370, 5244, 5210, 4854, 4762, 4740, 4384, 4300, 4288, 4020, 3968, 3964, 3752, 3668, 3511, 3483, 3354, 3322, 3205, 3183, 3108, 3046, 2999, 2981, 2974, 2968, 2961, 2955, 2949, 2943, 2942, 2939, 2935, 2934, 2933, 2929, 2270, 2178, 2162, 1959, 1946, 1780, 1651, 1524, 1400, 1289, 1133, 1037, 942, 849, 763, 711, 591, 521, 503, 496, 474, 461, 449, 442, 436, 426, 417, 407, 394, 387, 377, 373, 370, 367, 366, 365, 364, 363, 362, 358, 355, 352, 351, 350, 0};

int cumf\_SIGN[3]={16383, 8416, 0};

int cumf\_LAST[3]={16383, 9469, 0};

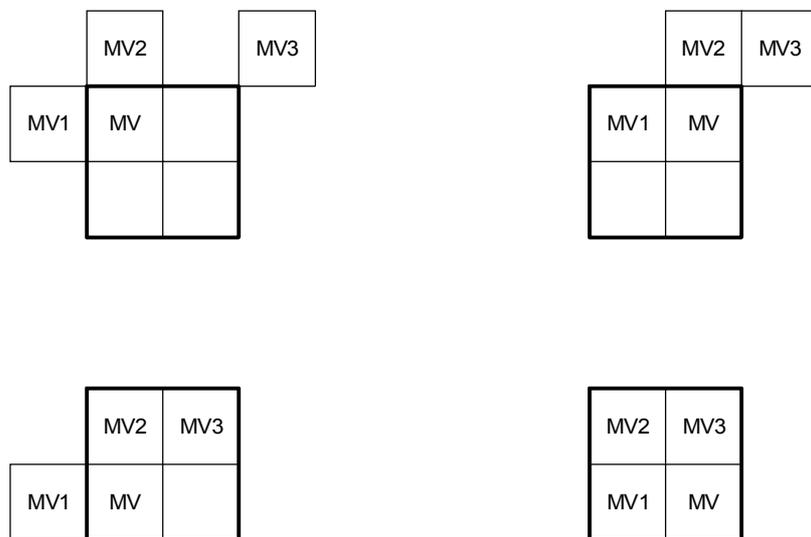
int cumf\_LAST\_intra[3]={16383, 2820, 0};

int cumf\_RUN[65]={16383, 15310, 14702, 13022, 11883, 11234, 10612, 10192, 9516, 9016, 8623, 8366, 7595, 7068, 6730, 6487, 6379, 6285, 6177, 6150, 6083, 5989, 5949, 5922, 5895, 5828, 5774, 5773, 5394, 5164, 5016, 4569, 4366, 4136, 4015, 3867, 3773, 3692, 3611, 3476, 3341, 3301, 2787, 2503, 2219, 1989, 1515, 1095, 934, 799, 691, 583, 435, 300, 246, 206, 125, 124, 97, 57, 30, 3, 2, 1, 0};

int cumf\_RUN\_intra[65]={16383, 10884, 8242, 7124, 5173, 4745, 4246, 3984, 3034, 2749, 2607, 2298, 966, 681, 396, 349, 302, 255, 254, 253, 206, 159, 158, 157, 156, 155, 154, 153, 106, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};



On obtient les vecteurs en ajoutant les prédicteurs aux différences vectorielles indiquées par les mots MVD et  $MVD_{2-4}$  comme lorsqu'un seul vecteur cinétique est présent pour un macrobloc, conformément aux règles du 6.1.1. De nouveau, on calcule les prédicteurs séparément pour les composantes horizontale et verticale. Les prédicteurs candidats MV1, MV2 et MV3 sont cependant redéfinis comme indiqué dans la Figure F.1. Si un seul vecteur par macrobloc est présent, les prédicteurs MV1, MV2 et MV3 sont définis comme pour le bloc 8 \* 8 numéroté 1 dans la Figure 5 (cette définition est donnée dans le coin supérieur gauche des quatre sous-figures de la Figure F.1).



T1519620-95/d09

FIGURE F.1/H.263

**Redéfinition des prédicteurs candidats (MV1, MV2 et MV3) pour chacun des blocs de luminance d'un macrobloc**

Si quatre vecteurs sont utilisés, chacun de ces vecteurs cinétiques est utilisé pour tous les pixels contenus dans un des quatre blocs de luminance du macrobloc. Le numérotage des vecteurs cinétiques est équivalent à celui des quatre blocs de luminance indiqués sur la Figure 5. Le vecteur cinétique  $MVD_{CHR}$ , applicable aux deux blocs de chrominance, se calcule par la somme des quatre vecteurs de luminance et par division de cette somme par 8; les valeurs composant les vecteurs résultants, de résolution jusqu'au 16<sup>e</sup> pixel, sont modifiées pour les rapprocher de la plus proche position d'un demi-pixel, comme indiqué dans le Tableau F.1.

TABLEAU F.1/H.263

**Modification des composantes vectorielles de chrominance en résolution jusqu'au seizième pixel**

position du 16 <sup>e</sup> pixel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	/16
position résultante	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	/2

Les valeurs des demi-pixels sont déterminées par interpolation bilinéaire, comme indiqué au 6.1.2. En mode de prédiction avancé, la prédiction de la luminance est obtenue par compensation de mouvement superposée, comme décrit dans F.3. La prédiction de la chrominance est obtenue par application du vecteur cinétique  $MVD_{CHR}$  à tous les pixels contenus dans les deux blocs de chrominance (comme cela est fait dans le mode de prédiction par défaut).

### F.3 Compensation de mouvement par superposition pour la luminance

Chaque pixel d'un bloc de prédiction de luminance à  $8 * 8$  éléments est une somme pondérée de trois valeurs de prédiction, divisée par 8 (avec arrondissement). Afin d'obtenir ces trois valeurs de prédiction, on fait appel à trois vecteurs cinétiques: le vecteur cinétique du bloc de luminance en cours, et deux sur quatre vecteurs «distants»:

- le vecteur cinétique du bloc situé à gauche ou à droite du bloc de luminance en cours;
- le vecteur cinétique du bloc situé au-dessus ou au-dessous du bloc de luminance en cours.

Les vecteurs cinétiques distants, issus d'autres groupes de blocs, sont utilisés de la même façon que les vecteurs cinétiques distants qui sont issus du groupe de blocs en cours.

Pour chaque pixel, on utilise les vecteurs cinétiques distants des blocs situés aux deux limites de bloc les plus proches. C'est-à-dire que, pour la moitié supérieure du bloc, on utilisera le vecteur cinétique correspondant au bloc situé au-dessus du bloc en cours (voir la Figure F.3). De même, pour la moitié gauche du bloc, on utilisera le vecteur cinétique correspondant au bloc situé à gauche du bloc en cours, alors que pour la moitié droite du bloc, on utilisera le vecteur cinétique correspondant au bloc situé à droite du bloc en cours (voir la Figure F.4).

La création de chaque pixel,  $\bar{p}(i, j)$  dans un bloc de prédiction à  $8 * 8$  éléments de luminance, est régie par l'équation suivante:

$$\bar{p}(i, j) = (q(i, j) \times H_0(i, j) + r(i, j) \times H_1(i, j) + s(i, j) \times H_2(i, j) + 4) / 8$$

où  $q(i, j)$ ,  $r(i, j)$  et  $s(i, j)$  sont les pixels extraits de l'image référencée comme suit:

$$q(i, j) = p(i + MV^{0,x}, j + MV^{0,y})$$

$$r(i, j) = p(i + MV^{1,x}, j + MV^{1,y})$$

$$s(i, j) = p(i + MV^{2,x}, j + MV^{2,y})$$

Dans ces expressions,  $(MV^{0,x}, MV^{0,y})$  désigne le vecteur cinétique pour le bloc en cours,  $(MV^{1,x}, MV^{1,y})$  désigne le vecteur cinétique du bloc situé soit au-dessus, soit au-dessous, et  $(MV^{2,x}, MV^{2,y})$  désigne le vecteur cinétique situé soit à gauche, soit à droite du bloc en cours qui a été défini ci-dessus.

Les matrices  $H_0(i, j)$ ,  $H_1(i, j)$  et  $H_2(i, j)$  sont définies dans les Figures F.2, F.3 et F.4, où les  $(i, j)$  désignent respectivement la colonne et la rangée de la matrice.

Si l'un des macroblocs environnants n'a pas été codé, le vecteur cinétique distant qui correspond à ce bloc est mis à zéro. Si l'un des blocs environnants a été codé INTRA, le vecteur cinétique distant qui correspond à ce bloc est remplacé par le vecteur cinétique du bloc en cours, sauf en mode de trames PB. Dans ce cas (bloc à codage INTRA en mode de trames PB), on utilise le vecteur cinétique du bloc INTRA (voir également l'Annexe G). Si le bloc en cours est situé à la périphérie de l'image et qu'il n'y ait pas de bloc environnant, on remplace le vecteur cinétique distant de ce bloc par le vecteur cinétique en cours. Dans tous les cas, si le bloc en cours est situé en bas du macrobloc (pour les blocs numéros 3 et 4, voir la Figure 5), on remplace le vecteur cinétique distant, qui correspond à un bloc de luminance  $8 * 8$  du macrobloc situé au-dessous du macrobloc en cours, par le vecteur cinétique correspondant au bloc en cours.

Les valeurs de pondération pour la prédiction sont indiquées dans les Figures F.2, F.3 et F.4 ci-après.

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

FIGURE F.2/H.263

Valeurs de pondération,  $H_0$ , pour la prédiction avec un vecteur cinétique du bloc de luminance en cours

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

FIGURE F.3/H.263

Valeurs de pondération,  $H_1$ , pour la prédiction avec des vecteurs cinétiques des blocs de luminance situés au-dessus ou au-dessous du bloc de luminance en cours

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

FIGURE F.4/H.263

Valeurs de pondération,  $H_2$ , pour la prédiction avec des vecteurs cinétiques des blocs de luminance situés à gauche ou à droite du bloc de luminance en cours

## Annexe G

### Mode de codage par trames PB

(Cette annexe fait partie intégrante de la présente Recommandation)

#### G.1 Introduction

La présente annexe décrit le mode facultatif H.263 de codage par trames PB. La possibilité de ce mode est signalée par des moyens extérieurs (par exemple selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le mot PTYPE.

Une trame PB se compose de deux images codées sous forme d'un seul élément. Les lettres PB correspondent aux initiales des types d'image décrits dans la Recommandation H.262: images P et images B. Une trame PB se compose donc d'une image P (prédite à partir de la précédente image P décodée) et d'une image B (prédite à partir, d'une part, de la précédente image P décodée et, d'autre part, de l'image P en cours de décodage). Le terme *image B* a été retenu parce que certaines de ses parties peuvent être prédites bilatéralement à partir des images passées et futures. Le processus de prédiction est illustré sur la Figure G.1.

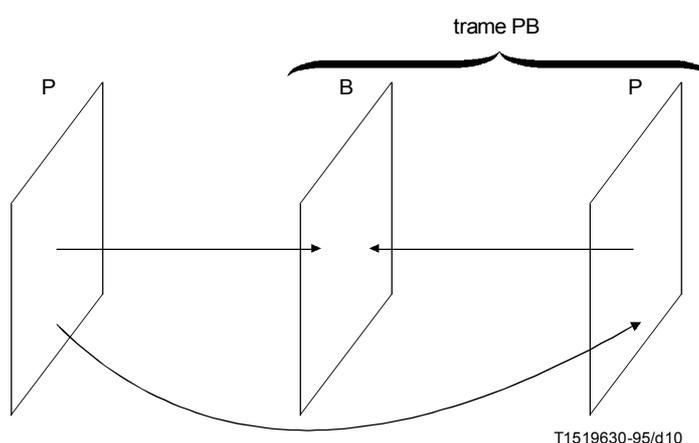


FIGURE G.1/H.263

#### Prédiction en mode de codage par trames PB

#### G.2 Trames PB et blocs INTRA

Lorsque l'on utilise des trames PB, le mode de codage INTRA a la signification suivante (voir également 5.3.2):

- les blocs de type P sont à codage INTRA;
- les blocs de type B sont à codage INTER, avec prédiction comme pour un bloc INTER.

Si l'on utilise des trames PB, des données vectorielles cinétiques (MVD) (*motion vector data*) sont également incluses pour les macroblocs INTRA contenus dans les images pour lesquelles le mot PTYPE indique «INTER». Dans ce cas, le vecteur n'est utilisé que pour le bloc de type B. Les mots de code MVD<sub>2,4</sub> ne sont jamais utilisés pour le mode INTRA (voir également le Tableau 7). En mode de prédiction avancé comme en mode de trames PB, si un des blocs environnants a été codé en mode INTRA, le vecteur cinétique distant qui correspond à ce bloc n'est pas remplacé par le vecteur cinétique du bloc en cours et c'est le vecteur cinétique 'INTRA' distant qui est utilisé.

### G.3 Couche des blocs

Dans une trame PB, un macrobloc se compose de douze blocs. On transmet d'abord les données pour les six blocs de type P comme dans le mode H.263 par défaut, puis les données pour les six blocs de type B (voir également 5.4). La structure de la couche des blocs est représentée sur la Figure 10. Le mot INTRADC est présent pour chaque bloc P du macrobloc si le mot MCBPC indique le type MB 3 ou 4 (voir le Tableau 4 et le Tableau 5). Le mot INTRADC n'est pas présent pour les blocs B. Le mot TCOEF est présent pour les blocs P s'il est indiqué par MCBPC ou par CBPY; TCOEF est présent pour les blocs B s'il est indiqué par CBPB.

### G.4 Calcul des vecteurs pour l'image B dans une trame PB

Les vecteurs pour l'image B sont calculés comme suit (voir également 6.1.1). Soit une composante vectorielle MV en unités de demi-pixel qui doit être utilisée dans l'image P (MV représentant une composante vectorielle pour un bloc de luminance  $8 * 8$ ; si un seul vecteur par macrobloc est transmis, la composante MV a la même valeur pour chacun des quatre blocs de luminance  $8 * 8$ ). Dans ce cas, l'on a besoin des deux composantes vectorielles – anticipée  $MV_F$  et différée  $MV_B$  – pour la prédiction de l'image B. Ces composantes vectorielles anticipées et différées sont déduites de la composante MV et finalement affinées par une composante vectorielle delta qui est fournie par les données MVDB.

- $TR_D$     incrément de la référence temporelle TR à partir du dernier en-tête d'image (voir 5.1.2). Si  $TR_D$  est négatif, on prend  $TR_D = TR_D + 256$ ;
- $TR_B$     voir 5.1.7.

Supposons que  $MV_D$  soit la composante vectorielle delta qui est indiquée par la donnée MVDB et qui correspond à la composante vectorielle MV. Si la donnée MVDB est présente, la même composante  $MV_D$  indiquée par la donnée MVDB est utilisée pour chacun des quatre blocs de luminance B contenus dans le macrobloc.

Dans ce cas, les composantes  $MV_F$  et  $MV_B$  sont données en demi-pixels par les formules suivantes:

$$MV_F = (TR_B \times MV) / TR_D + MV_D$$

$$MV_B = ((TR_B - TR_D) \times MV) / TR_D \quad \text{si } MV_D \text{ est égal à } 0$$

$$MV_B = MV_F - MV \quad \text{si } MV_D \text{ n'est pas égal à } 0$$

où la barre oblique «/» indique une division par troncature. On suppose que les coordonnées reflètent la position réelle dans le temps des images P et B. On tire parti du fait que l'étendue des valeurs de la composante  $MV_F$  est bornée. Chaque mot à codage VLC pour MVDB représente une paire de valeurs différentielles. Une seule de ces paires donnera une valeur de  $MV_F$  s'inscrivant dans l'intervalle permis (par défaut  $[-16, 15,5]$ ; en mode de vecteurs cinétiques non restreints  $[-31,5, 31,5]$ ). Les formules pour les composantes  $MV_F$  et  $MV_B$  sont également utilisées dans le cas de blocs INTRA, où les données vectorielles ne sont utilisées que pour prédire des blocs B.

Pour les blocs de chrominance, la composante  $MV_F$  est déterminée par calcul de la somme des quatre vecteurs de luminance  $MV_F$  correspondants puis par division de cette somme par 8; les composantes vectorielles de résolution jusqu'au 16<sup>e</sup> pixel sont modifiées en fonction de la plus proche position de demi-pixel, comme indiqué dans le Tableau F.2. La composante  $MV_B$  pour la chrominance est déterminée par calcul de la somme des quatre vecteurs de luminance  $MV_B$  correspondants et par division de cette somme par 8; les composantes vectorielles résultantes en résolution jusqu'au 16<sup>e</sup> pixel sont modifiées en fonction de la plus proche position de demi-pixel, comme indiqué dans le Tableau F.2.

Une valeur positive de la composante horizontale ou verticale du vecteur cinétique signifie que la prédiction est formée à partir des pixels contenus dans l'image référencée, situés spatialement à droite ou au-dessous des pixels en cours de prédiction.

### G.5 Prédiction d'un bloc de type B dans une trame PB

Dans ce paragraphe, un bloc est défini comme ayant  $8 \times 8$  éléments. La procédure suivante s'applique aux blocs de luminance comme aux blocs de chrominance. On calcule d'abord les vecteurs anticipés et différés. On suppose que le macrobloc P (luminance et chrominance) est d'abord décodé, reconstruit et écrêté (voir 6.3.2). Ce macrobloc est appelé  $P_{REC}$  et sert de base, avec sa valeur prédite, pour le calcul de la prédiction du bloc B.

La prédiction du bloc de type B a deux modes, qui sont utilisés pour différentes parties du bloc:

- pour les pixels dont le vecteur différé –  $MV_B$  – pointe à l'intérieur du macrobloc  $P_{REC}$ , on utilise la prédiction bidirectionnelle. A cette fin, on calcule la moyenne de la prédiction anticipée sur la base de la composante  $MV_F$  relative à la précédente image P décodée et de la prédiction différée sur la base de la composante  $MV_B$  relative au macrobloc  $P_{REC}$ . Cette moyenne est la demi-somme des deux prédictions (division par troncature);
- pour tous les autres pixels, on fait appel à la prédiction anticipée sur la base de la composante  $MV_F$  relative à la précédente image P décodée.

La Figure G.2 indique quelle partie d'un bloc est prédite en mode bidirectionnel (partie ombrée du bloc B) et quelle partie n'est prédite qu'en mode anticipé (reste du bloc B).

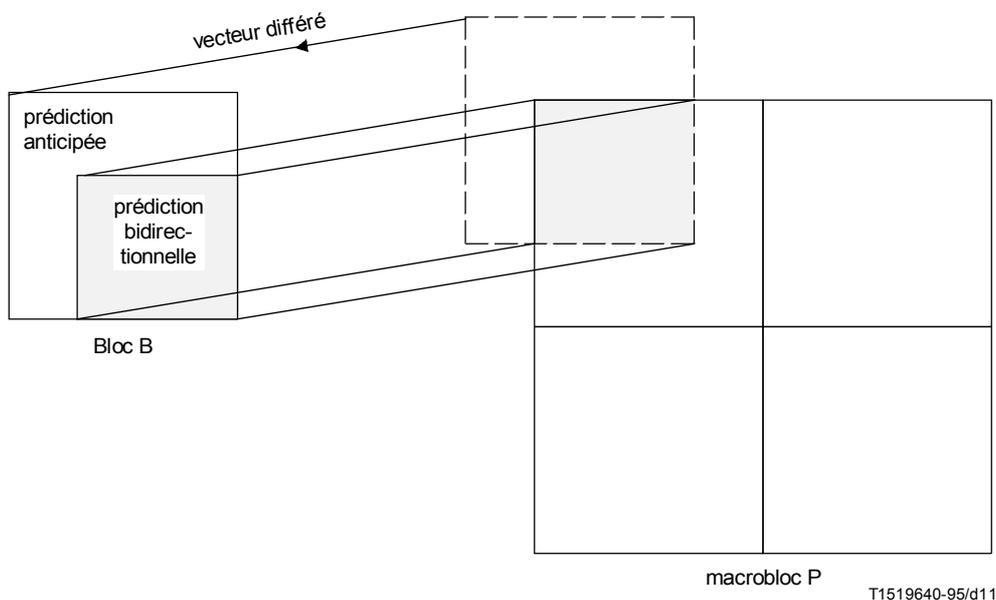


FIGURE G.2/H.263

**Prédiction anticipée et prédiction différée pour un bloc B**

La prédiction bidirectionnelle est utilisée pour les pixels dont le vecteur différé –  $MV_B$  – pointe à l'intérieur du macrobloc  $P_{REC}$ . Ces pixels sont définis par les procédures suivantes, qui sont spécifiées en langage C.

Définitions:

- nh position horizontale du bloc à l'intérieur d'un macrobloc (0 ou 1)
- nv position verticale du bloc à l'intérieur d'un macrobloc (0 ou 1)
- mh (nh, nv) composante vectorielle horizontale du bloc (nh, nv) en demi-pixels
- mv (nh, nv) composante vectorielle verticale du bloc (nh, nv) en demi-pixels
- mhc composante vectorielle horizontale pour la chrominance
- mhv composante vectorielle verticale pour la chrominance

## Procédure pour la luminance

```
for (nh = 0; nh <= 1; nh++) {
    for (nv = 0; nv <= 1; nv++) {
        for (i = nh * 8 + max(0,(-mh(nh,nv)+1)/2 - nh * 8);
            i <= nh * 8 + min(7,15-(mh(nh,nv)+1)/2 - nh * 8); i++) {
            for (j = nv * 8 + max(0,(-mv(nh,nv)+1)/2 - nv * 8);
                j <= nv * 8 + min(7,15-(mv(nh,nv)+1)/2 - nv * 8); j++) {
                predict pixel (i,j) bidirectionally
            }
        }
    }
}
```

## Procédure pour la chrominance

```
for (i = max(0,(-mhc+1)/2); i <= min(7,7-(mhc+1)/2); i++) {
    for (j = max(0,(-mvc+1)/2); j <= min(7,7-(mvc+1)/2); j++) {
        predict pixel (i,j) bidirectionally;
    }
}
```

Les pixels non prédits bidirectionnellement ne sont prédits qu'en mode anticipé.

# Annexe H

## Correction d'erreur directe pour signal vidéo codé

(Cette annexe fait partie intégrante de la présente Recommandation)

### H.1 Introduction

La présente annexe décrit une méthode facultative de correction d'erreur directe (à codage et verrouillage de trames) pour la transmission de données vidéo à codage H.263. Cette correction d'erreur directe (CED) peut être utilisée lorsque aucune CED n'est assurée par des moyens externes, par exemple au niveau du multiplex ou du système. Elle n'est pas utilisée pour le codage H.324. Aussi bien le verrouillage de trames que le code de correction d'erreur directe sont les mêmes que dans la Recommandation H.261.

### H.2 Verrouillage de trame de correction d'erreur

Pour que les informations de parité relatives aux données vidéo et à la correction d'erreur puissent être identifiées par un décodeur, un schéma de verrouillage des trames de correction d'erreur est inclus: il consiste en une multitrame de huit trames, comportant chacune 1 bit de verrouillage de trame, 1 bit indicateur de bourrage (Fi), 492 bits de données codées (ou un remplissage de «1») et 18 bits de parité (voir la Figure H.1). Pour chaque multitrame, le schéma formé par les bits de verrouillage des huit trames élémentaires est le suivant:

$$(S_1S_2S_3S_4S_5S_6S_7S_8) = (00011011)$$

Le fanion indicateur de remplissage (Fi) peut être mis à zéro par un codeur. Dans ce cas, 492 bits de remplissage consécutifs (uniquement des «1») sont émis à la place de 492 bits de données codées. Cette méthode peut être utilisée pour les données de bourrage (voir 3.6).

### H.3 Code de correction d'erreur

Le code de correction d'erreur est de type Bose-Chaudhuri-Hocquenghem (BCH) sans voie de retour (511,493). Son utilisation par le décodeur est facultative. La parité est calculée sur une séquence codée de 493 bits, comprenant 1 bit indicateur de remplissage (Fi) et 492 bits de données vidéo codées.

Le polynôme générateur est le suivant:

$$g(x) = (x^9 + x^4 + 1)(x^9 + x^6 + x^4 + x^3 + 1)$$

Exemple: Pour les données d'entrée de «01111 . . . 11» (493 bits), les bits de la parité de correction résultante sont: «011011010100011011» (18 bits).

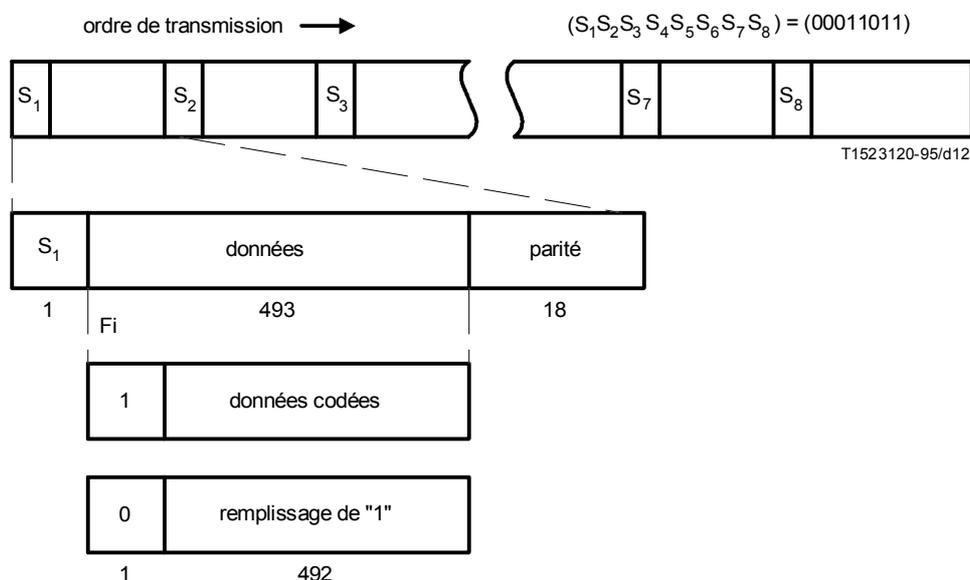


FIGURE H.1/H.263

Trame de correction d'erreur

#### H.4 Temps de rétablissement pour le verrouillage de trame de correction d'erreur

Il y a lieu que trois séquences consécutives (24 bits) de verrouillage des trames de correction d'erreur soient reçues avant que l'alignement des trames soit considéré comme réalisé. Il convient que le décodeur soit conçu de telle manière que le verrouillage des trames soit rétabli moins de 34 000 bits après un déphasage de trame du correcteur d'erreur.

NOTE – A cette fin, on admet que les données vidéo ne contiennent pas trois émulations, non déphasées, de la séquence de verrouillage de trame de correction d'erreur, au cours de la période de reverrouillage.



## SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Réseau téléphonique et RNIS
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission
<b>Série H</b>	<b>Transmission des signaux autres que téléphoniques</b>
Série I	Réseau Numérique avec Intégration des Services
Série J	Transmission des signaux radiophoniques et télévisuels
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques, et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophoniques et télévisuels
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie alphabétique
Série T	Equipements terminaux et protocoles des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de communications de données et communication entre systèmes ouverts
Série Z	Langages de programmation

