



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.263

(02/98)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructure des services audiovisuels – Codage des
images vidéo animées

**Codage vidéo pour communications à faible
débit**

Recommandation UIT-T H.263

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE H
SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Caractéristiques des canaux de transmission pour des usages autres que téléphoniques	H.10–H.19
Emploi de circuits de type téléphonique pour la télégraphie à fréquence vocale	H.20–H.29
Circuits et câbles téléphoniques utilisés pour les divers types de transmission télégraphique et de transmissions simultanées	H.30–H.39
Circuits de type téléphonique utilisés en bélinographie	H.40–H.49
Caractéristiques des signaux de données	H.50–H.99
CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.399

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

RECOMMANDATION UIT-T H.263

CODAGE VIDEO POUR COMMUNICATIONS A FAIBLE DEBIT

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 restant afin de réduire la redondance spatiale. Le codeur de la source peut traiter cinq formats de source vidéo normalisés: sub-QCIF, QCIF, CIF, 4CIF et 16CIF, ainsi qu'une large gamme de formats vidéo personnalisés.

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, la présente Recommandation décrit seize options de codage négociables, destinées à améliorer les performances en matière de compression et à prendre en charge des capacités additionnelles. Des informations complémentaires additionnelles peuvent aussi être incluses dans le flux binaire, afin de permettre une capacité d'affichage améliorée et aussi une utilisation externe.

Source

La Recommandation UIT-T H.263, révisée par la Commission d'études 16 de l'UIT-T (1997-2000), a été approuvée le 6 février 1998 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'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.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

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.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT avait été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

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

		Page
1	Domaine d'application.....	1
2	Références normatives	1
3	Brève spécification.....	2
3.1	Entrée et sortie vidéo.....	2
3.2	Sortie et entrée numérique.....	2
3.3	Fréquence d'échantillonnage	2
3.4	Algorithme de codage de la source	3
3.4.1	Mode de communication multipoint à présence continue et à multiplex vidéo.....	3
3.4.2	Mode de codage par vecteurs cinétiques non restreints	3
3.4.3	Mode de codage arithmétique syntaxique.....	3
3.4.4	Mode de prédiction avancé	3
3.4.5	Mode de codage par trames PB.....	3
3.4.6	Correction d'erreur sans voie de retour.....	4
3.4.7	Mode de codage INTRA avancé	4
3.4.8	Mode avec filtre de dégroupage.....	4
3.4.9	Mode structuré par tranches	4
3.4.10	Informations complémentaires d'amélioration	4
3.4.11	Mode de codage par trames PB amélioré.....	5
3.4.12	Mode de sélection d'image de référence.....	5
3.4.13	Mode d'échelonnabilité temporelle, SNR et spatiale.....	5
3.4.14	Mode de rééchantillonnage d'image de référence.....	5
3.4.15	Mode de rafraîchissement à résolution réduite	6
3.4.16	Mode de décodage par segments indépendants.....	6
3.4.17	Autre mode INTER VLC	6
3.4.18	Mode de quantification modifié	6
3.5	Débit.....	6
3.6	Stockage temporaire des données.....	6
3.7	Symétrie de transmission.....	7
3.8	Traitement des erreurs	7
3.9	Fonctionnement multipoint	7
4	Codeur de source	7
4.1	Format de la source	7
4.2	Algorithme de codage de la source vidéo.....	10
4.2.1	Groupes de blocs, tranches, macroblocs et blocs.....	11

	Page
4.2.2	Prédiction 13
4.2.3	Compensation de mouvement 14
4.2.4	Quantification..... 14
4.3	Contrôle du codage..... 15
4.4	Rafraîchissement forcé..... 15
4.5	Alignement des codes de déclenchement par les octets 15
5	Syntaxe et sémantique 15
5.1	Couche des images 24
5.1.1	Code de déclenchement d'image (PSC, <i>picture start code</i>) (22 bits)..... 25
5.1.2	Référence temporelle (TR, <i>temporal reference</i>) (8 bits)..... 25
5.1.3	Informations de type (PTYPE, <i>type information</i>) (longueur variable)..... 25
5.1.4	Champ PTYPE Plus (<i>PLUSPTYPE</i>) (longueur variable) 26
5.1.5	Format d'image personnalisé (CPFMT, <i>custom picture format</i>) (23 bits) 30
5.1.6	Rapport d'aspect de pixel étendu (EPAR, <i>extended pixel aspect ratio</i>) (16 bits) 31
5.1.7	Code de fréquence de base d'image personnalisée (CPCFC, <i>custom picture clock frequency code</i>) (8 bits) 31
5.1.8	Référence temporelle étendue (ETR, <i>extended temporal reference</i>) (2 bits) 32
5.1.9	Indicateur de vecteurs cinétiques non restreints illimités (UUI, <i>unlimited unrestricted motion vectors indicator</i>) (longueur variable) 32
5.1.10	Bits de sous-mode du mode structuré par tranches (SSS, <i>slice structured submode</i>) (2 bits)..... 32
5.1.11	Numéro de la tranche d'amélioration (ELNUM, <i>enhancement layer number</i>) (4 bits) 32
5.1.12	Numéro de la couche de référence (RLNUM, <i>reference layer number</i>) (4 bits) 32
5.1.13	Fanions de mode de sélection d'image de référence (RPSMF, <i>reference picture selection mode flags</i>) (3 bits) 33
5.1.14	Indication de référence temporelle aux fins de prédiction (TRPI, <i>temporal reference for prediction indication</i>) (1 bit) 33
5.1.15	Référence temporelle aux fins de prédiction (TRP, <i>temporal reference for prediction</i>) (10 bits) 33
5.1.16	Indication de message par canal de retour (BCI, <i>back-channel message indication</i>) (longueur variable) 33
5.1.17	Message par canal de retour (BCM, <i>back-channel message</i>) (longueur variable)..... 34
5.1.18	Paramètres de rééchantillonnage d'image de référence (RPRP, <i>reference picture resampling parameters</i>) (longueur variable) 34
5.1.19	Information sur le quantificateur (PQUANT, <i>quantizer information</i>) (5 bits) 34
5.1.20	Communication multipoint à présence continue et à multiplex vidéo (CPM, <i>continuous presence multipoint and video multiplex</i>) (1 bit)..... 34

	Page	
5.1.21	Indicateur de sous-flux binaire (PSBI, <i>picture sub-bitstream indicator</i>) (2 bits) 34	34
5.1.22	Référence temporelle pour les images B dans les trames PB (TR _B , <i>temporal reference for B-pictures in PB-frames</i>) (3/5 bits) 34	34
5.1.23	Information de quantification pour images B dans les trames PB (DBQUANT, <i>quantization information for B-pictures in PB frames</i>) (2 bits) 35	35
5.1.24	Informations supplémentaires d'insertion (PEI, <i>extra insertion information</i>) (1 bit)..... 35	35
5.1.25	Informations complémentaires d'amélioration (PSUPP, <i>supplemental enhancement information</i>) (0/8/16... bits)..... 35	35
5.1.26	Bourrage (ESTUF, <i>stuffing</i>) (longueur variable) 35	35
5.1.27	Fin de séquence (EOS, <i>end of sequence</i>) (22 bits)..... 35	35
5.1.28	Bourrage (PSTUF, <i>stuffing</i>) (longueur variable)..... 36	36
5.2	Couche des groupes de blocs..... 36	36
5.2.1	Bourrage (GSTUF, <i>stuffing</i>) (longueur variable)..... 36	36
5.2.2	Code de déclenchement d'un groupe de blocs (GBSC, <i>group of block start code</i>) (17 bits) 36	36
5.2.3	Numéro de groupe (GN, <i>group number</i>) (5 bits) 36	36
5.2.4	Indicateur de sous-flux binaire d'un groupe de blocs (GSBI, <i>GOB sub-bitstream indicator</i>) (2 bits) 37	37
5.2.5	Identificateur de trame GOB (GFID, <i>GOB frame ID</i>) (2 bits) 37	37
5.2.6	Information sur le quantificateur (GQUANT, <i>quantizer information</i>) (5 bits) 37	37
5.3	Couche des macroblocs 37	37
5.3.1	Indication de macrobloc codé (COD, <i>coded macroblock indication</i>) (1 bit) 38	38
5.3.2	Type de macrobloc et structure de bloc codée pour la chrominance (MCBPC, <i>macroblock type & coded block pattern for chrominance</i>) (longueur variable) 38	38
5.3.3	Mode de macrobloc pour blocs d'images B (MODB, <i>macroblock mode for B-blocks</i>) (longueur variable)..... 41	41
5.3.4	Structure de bloc codée pour blocs d'images B (CBPB, <i>coded block pattern for B-blocks</i>) (6 bits) 41	41
5.3.5	Structure de bloc codée pour la luminance (CBPY, <i>coded block pattern for luminance</i>) (longueur variable) 41	41
5.3.6	Informations sur le quantificateur (DQUANT, <i>quantizer information</i>) (2 bits/longueur variable) 41	41
5.3.7	Données de vecteur cinétique (MVD, <i>motion vector data</i>) (longueur variable)..... 43	43
5.3.8	Données de vecteur cinétique (MVD ₂₋₄ , <i>motion vector data</i>) (longueur variable)..... 44	44

5.3.9	Données de vecteur cinétique pour macrobloc de type B (MVDB, <i>motion vector data for B-macroblock</i>) (longueur variable)	44
5.4	Couche des blocs	44
5.4.1	Coefficient DC pour les blocs INTRA (INTRADC, <i>DC coefficient for INTRA blocks</i>) (8 bits).....	45
5.4.2	Coefficient de transformée (TCOEF, <i>transform coefficient</i>) (longueur variable).....	45
6	Processus de décodage	48
6.1	Compensation de mouvement	48
6.1.1	Vecteurs cinétiques différentiels	49
6.1.2	Interpolation pour la prédiction en fraction de pixel.....	50
6.2	Décodage des coefficients	51
6.2.1	Quantification inverse	51
6.2.2	Ecrêtage des niveaux de reconstruction	51
6.2.3	Positionnement en zigzag.....	51
6.2.4	Transformation inverse	52
6.3	Reconstruction des blocs	52
6.3.1	Sommation	52
6.3.2	Ecrêtage.....	52
	Annexe A – Spécification de la précision de la transformée inverse	53
	Annexe B – Décodeur fictif de référence	54
	Annexe C – Considérations relatives aux communications multipoints	56
C.1	Demande de gel d'image.....	56
C.2	Demande de rafraîchissement rapide.....	56
C.3	Fin de gel d'image.....	56
C.4	Communication multipoint à présence continue et à multiplex vidéo (CPM, <i>continuous presence multipoint and video multiplexing</i>).....	57
C.4.1	Code de fin de sous-flux binaire (EOSBS, <i>end of sub-bitstream</i>) (23 bits)..	58
C.4.2	Indicateur de fin de sous-flux binaire (ESBI, <i>ending sub-bitstream indicator</i>) (2 bits)	58
	Annexe D – Mode de codage par vecteurs cinétiques non restreints	59
D.1	Vecteurs cinétiques pointant hors des limites d'image.....	59
D.1.1	Restrictions appliquées aux valeurs des vecteurs cinétiques	60

	Page
D.2 Extension de la gamme des vecteurs cinétiques.....	60
Annexe E – Mode de codage arithmétique syntaxique (SAC).....	62
E.1 Introduction.....	62
E.2 Spécification du codeur en mode SAC.....	63
E.3 Spécification du décodeur en mode SAC.....	64
E.4 Syntaxe.....	65
E.5 Registre PSC_FIFO.....	65
E.6 Symboles de la couche des en-têtes.....	65
E.7 Symboles des couches des macroblocs et des blocs.....	66
E.8 Modèles de codage SAC.....	67
Annexe F – Mode de prédiction avancé.....	70
F.1 Introduction.....	70
F.2 Quatre vecteurs cinétiques par macrobloc.....	70
F.3 Compensation de mouvement par superposition pour la luminance.....	71
Annexe G – Mode de codage par trames PB.....	74
G.1 Introduction.....	74
G.2 Trames PB et blocs INTRA.....	75
G.3 Couche des blocs.....	75
G.4 Calcul des vecteurs pour l'image B dans une trame PB.....	75
G.5 Prédiction d'un bloc de type B dans une trame PB.....	76
Annexe H – Correction d'erreur sans voie de retour pour signal vidéo codé.....	78
H.1 Introduction.....	78
H.2 Verrouillage de trames de correction d'erreur.....	78
H.3 Code de correction d'erreur.....	78
H.4 Temps de rétablissement pour le verrouillage de trames de correction d'erreur.....	79
Annexe I – Mode de codage INTRA avancé.....	79
I.1 Introduction.....	79
I.2 Syntaxe.....	80
I.3 Processus de décodage.....	80
Annexe J – Mode avec filtre de dégroupage.....	88
J.1 Introduction.....	88
J.2 Relation avec les modes UMV et AP (Annexes D et F).....	88
J.3 Définition du filtre de limite de dégroupage.....	89

	Page
Annexe K – Mode structuré par tranches	93
K.1 Introduction	93
K.2 Structure de la couche des tranches.....	94
K.2.1 Bourrage (SSTUF, <i>stuffing</i>) (longueur variable).....	94
K.2.2 Code de déclenchement de tranche (SSC, <i>slice start code</i>) (17 bits).....	94
K.2.3 Bit de prévention d'émulation de tranche 1 (SEPBI, <i>slice emulation prevention bit 1</i>) (1 bit)	94
K.2.4 Indicateur de sous-flux binaire de tranche (SSBI, <i>slice sub-bitstream indicator</i>) (4 bits)	95
K.2.5 Adresse de macrobloc (MBA, <i>macroblock address</i>) (5/6/7/9/11/12/13/14 bits)	95
K.2.6 Bit de prévention d'émulation de tranche 2 (SEPBI2, <i>slice emulation prevention bit 2</i>) (1 bit)	95
K.2.7 Information sur le quantificateur (SQUANT, <i>quantizer information</i>) (5 bits)	96
K.2.8 Indicateur de largeur de tranche dans les macroblocs (SWI, <i>slice width indication in macroblocks</i>) (3/4/5/6/7 bits).....	96
K.2.9 Bit de prévention d'émulation de tranche 3 (SEPBI3, <i>slice emulation prevention bit 3</i>) (1 bit)	96
Annexe L – Spécification d'information d'amélioration complémentaire.....	97
L.1 Introduction	97
L.2 Format de PSUPP.....	97
L.3 Ne rien faire.....	98
L.4 Demande de gel d'image complète.....	98
L.5 Demande de gel d'image partielle.....	98
L.6 Demande de gel d'image partielle avec redimensionnement.....	99
L.7 Demande de fin de gel d'image partielle	99
L.8 Etiquette d'instantané d'image complète.....	99
L.9 Etiquette d'instantané d'image partielle	100
L.10 Etiquette de déclenchement de segment de temps vidéo.....	100
L.11 Etiquette de fin de segment de temps vidéo	100
L.12 Etiquette de déclenchement de segment de raffinement progressif	100
L.13 Etiquette de fin de segment de raffinement progressif.....	101
L.14 Information d'incrustation	101
L.15 Type de fonction étendue	104

	Page
Annexe M – Mode de codage par trames PB amélioré	104
M.1 Introduction	104
M.2 Modes de prédiction par macroblocs B_{PB}	104
M.2.1 Prédiction bidirectionnelle	104
M.2.2 Prédiction anticipée.....	105
M.2.3 Prédiction différée.....	105
M.3 Calcul des vecteurs pour la prédiction bidirectionnelle du macrobloc B.....	105
M.4 Tableau MODB	105
Annexe N – Mode de sélection de l'image de référence.....	106
N.1 Introduction	106
N.2 Algorithme de codage de source vidéo	106
N.3 Canal pour les messages transmis par canal de retour	107
N.3.1 Mode de canal logique distinct	108
N.3.2 Mode vidéomux	108
N.4 Syntaxe	108
N.4.1 Canal vers l'avant	108
N.4.2 Syntaxe de message transmis par canal de retour (BCM).....	110
N.5 Processus de décodage	112
Annexe O – Mode d'échelonnabilité temporelle, SNR et spatiale	113
O.1 Aperçu général.....	113
O.1.1 Echelonnabilité temporelle.....	113
O.1.2 Echelonnabilité SNR.....	114
O.1.3 Echelonnabilité spatiale	115
O.1.4 Echelonnabilité multicouche.....	116
O.2 Ordre de transmission des images	117
O.3 Syntaxe de la couche des images.....	119
O.4 Syntaxe de la couche des macroblocs	120
O.4.1 Indication de macrobloc codé (COD, <i>coded macroblock indication</i>) (1 bit)	121
O.4.2 Champ MBTYPE/MCBPC (VLC)	121
O.4.3 Structure de bloc codée pour la chrominance (CBPC, <i>coded block pattern for chrominance</i>) (longueur variable)	123
O.4.4 Structure de bloc codée pour la luminance (CBPY) (longueur variable).....	123
O.4.5 Information sur le quantificateur (DQUANT) (2 bits/longueur variable)....	124
O.4.6 Données de vecteur cinétique (MVDFW, MVDBW) (longueur variable)...	124

	Page
O.5	Décodage des vecteurs cinétiques 124
O.5.1	Vecteur cinétiques différentiels..... 124
O.5.2	Vecteurs cinétiques en mode direct..... 124
O.6	Filtres d'interpolation..... 124
Annexe P – Rééchantillonnage d'image de référence..... 127	
P.1	Introduction 127
P.2	Syntaxe 130
P.2.1	Précision de déplacement de distorsion (WDA, <i>warping displacement accuracy</i>) (2 bits) 130
P.2.2	Paramètres de déformation (longueur variable)..... 131
P.2.3	Mode de remplissage (FILL_MODE) (2 bits)..... 132
P.2.4	Spécification des couleurs de remplissage (Y_FILL, C _B _EPB, C _B _FILL, C _R _EPB, C _R _FILL) (26 bits) 132
P.3	Algorithme de rééchantillonnage 132
P.4	Exemple de mise en œuvre..... 135
P.4.1	Déplacements de points virtuels..... 135
P.4.2	Algorithme de rééchantillonnage 136
P.5	Rééchantillonnage de facteur quatre 138
P.5.1	Suréchantillonnage de facteur quatre 138
P.5.2	Sous-échantillonnage de facteur quatre..... 140
Annexe Q – Mode de rafraîchissement à résolution réduite..... 141	
Q.1	Introduction 141
Q.2	Procédure de décodage..... 142
Q.2.1	Préparation de la référence..... 143
Q.2.2	Décodage de la couche des macroblocs 143
Q.2.3	Enregistrement de l'image 144
Q.2.4	Affichage..... 144
Q.3	Extension de l'image de référence 145
Q.4	Reconstruction des vecteurs cinétiques 146
Q.5	Matrices agrandies de compensation de mouvement par superposition de blocs pour la luminance 148
Q.6	Suréchantillonnage du bloc d'erreur de prédiction reconstruite à résolution réduite. 150
Q.6.1	Procédure de suréchantillonnage pour les pixels à l'intérieur d'un bloc 16 × 16 d'erreur de prédiction reconstruite..... 152
Q.6.2	Procédure de suréchantillonnage pour les pixels à la limite du bloc 16 × 16 d'erreur de prédiction reconstruite..... 153

	Page
Q.7 Filtre de limite de bloc.....	153
Q.7.1 Définition du filtre par défaut de limite de bloc.....	154
Q.7.2 Définition du filtre de limite de bloc lorsque le mode avec filtre de dégrouper est utilisé.....	154
Annexe R – Mode de décodage par segments indépendants.....	155
R.1 Introduction.....	155
R.2 Fonctionnement du mode.....	155
R.3 Contraintes imposées à l'utilisation du mode.....	156
R.3.1 Contrainte sur la forme des segments.....	156
R.3.2 Contrainte sur les variations de forme des segments.....	156
Annexe S – Autre mode INTER VLC.....	157
S.1 Introduction.....	157
S.2 Autre mode INTER VLC pour les coefficients.....	157
S.2.1 Mode d'action du codeur.....	157
S.2.2 Mode d'action du décodeur.....	157
S.3 Autre mode INTER VLC pour CBPY.....	158
Annexe T – Mode de quantification modifié.....	158
T.1 Introduction.....	158
T.2 Mise à jour modifiée de DQUANT.....	158
T.2.1 Modification de QUANT par échelon de faible valeur.....	159
T.2.2 Sélection arbitraire de QUANT.....	159
T.3 Valeur modifiée de l'échelon de quantification pour les coefficients de chrominance.....	159
T.4 Gamme modifiée des valeurs de coefficient.....	160
T.5 Restrictions d'utilisation.....	161
Appendice I – Dépistage d'erreurs.....	161
I.1 Introduction.....	161
I.2 Dépistage d'erreurs.....	162
Appendice II – Amélioration facultative recommandée.....	163
II.1 Introduction.....	163
II.2 Niveaux de prise en charge de modes préférés.....	163
II.2.1 Modes préférés du niveau 1.....	164
II.2.2 Modes préférés du niveau 2.....	164
II.2.3 Modes préférés du niveau 3.....	165
II.3 Formats d'image et fréquences de base d'image.....	166

Recommandation H.263

CODAGE VIDEO POUR COMMUNICATIONS À FAIBLE DEBIT

(révisée en 1998)

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 de source vidéo est fondée sur la Recommandation H.261. Seize options de codage négociables sont présentées afin d'améliorer les performances et d'augmenter la fonctionnalité. La présente Recommandation contient la Version 2 de la Recommandation H.263, entièrement compatible avec la Recommandation originale, les modifications consistant simplement en l'ajout de caractéristiques facultatives au contenu de la Version 1 originale.

2 Références normatives

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.

- [1] Recommandation UIT-R BT.601-5 (1995), *Paramètres de codage en studio de la télévision numérique pour des formats standards d'image 4:3 (normalisé) et 16:9 (écran panoramique)*.

La Référence [1] est mentionnée ici pour définir l'espace chromatique (Y , C_B , C_R) et sa représentation en nombres entiers de 8 bits pour les images utilisées par les codecs vidéo conçus selon la présente Recommandation. (La Référence [1] ne sert à définir aucun autre aspect de la présente Recommandation.)

Les Recommandations additionnelles suivantes de l'UIT-T sont mentionnées à titre d'illustration dans le présent texte et ne constituent pas 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.

- [2] Recommandation UIT-T H.223 (1996), *Protocole de multiplexage pour communications multimédias à faible débit*.
- [3] Recommandation UIT-T H.242 (1997), *Procédures pour l'établissement de communications entre terminaux audiovisuels sur des canaux numériques d'un débit allant jusqu'à 2 Mbit/s*.
- [4] Recommandation UIT-T H.245 (1998), *Protocole de commande pour communications multimédias*.
- [5] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s*.

- [6] Recommandation UIT-T H.262 (1995) | ISO/CEI 13818-2:1996, *Technologies de l'information – Codage générique des images animées et du son associé: données vidéo.*
- [7] Recommandation UIT-T H.324 (1998), *Terminal pour communication multimédia à faible débit.*

3 Brève spécification

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

3.1 Entrée et sortie vidéo

Afin qu'une seule Recommandation puisse s'appliquer aux régions utilisant les normes de télévision à 625 et à 525 lignes, les formats de source normalisés sur lesquels le codeur de source travaille se fondent sur un format intermédiaire commun (CIF, *common intermediate format*). Il est également possible d'utiliser la négociation externe (par exemple, la Recommandation H.245), afin de permettre l'utilisation d'une large gamme de formats sources personnalisés facultatifs. 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.

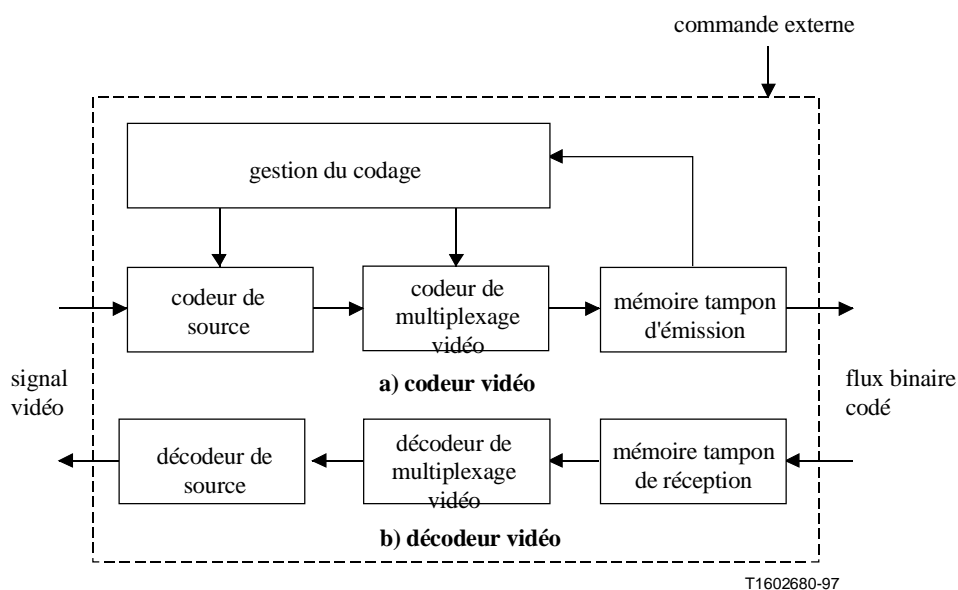


Figure 1/H.263 – Diagramme synoptique du codec vidéo

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 près 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 Recommandation H.263, seize options de codage négociables peuvent être utilisées, soit ensemble ou séparément (sous réserve de certaines restrictions). Des informations complémentaires additionnelles peuvent aussi être incluses dans le flux binaire afin d'améliorer la capacité d'affichage et de permettre l'utilisation externe. Une méthode de correction d'erreur sans voie de retour est indiquée pour application au flux binaire résultant, le cas échéant. Les options de codage négociables, la correction d'erreur sans voie de retour et l'utilisation des informations supplémentaires sont décrites dans les sous-paragraphes qui suivent.

3.4.1 Mode de communication multipoint à présence continue et à multiplex vidéo

Dans ce mode facultatif, jusqu'à quatre "sous-flux binaires" peuvent être envoyés dans le même canal vidéo. Cette caractéristique est conçue pour utilisation dans les applications de communication multipoint à présence continue ou d'autres situations dans lesquelles des canaux logiques distincts ne sont pas disponibles, mais dans lesquelles on désire utiliser des flux binaires vidéo multiples (voir également l'Annexe C).

3.4.2 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 grands, ce qui est particulièrement utile dans le cas d'un mouvement de caméra et de grands formats d'image.

3.4.3 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 de bits (voir aussi l'Annexe E).

3.4.4 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 8×8 éléments d'image au lieu d'un seul vecteur à 16×16 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 un gain subjectif car la compensation OBMC provoque moins d'artefacts de groupage.

3.4.5 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 à la fois de la précédente image P décodée et 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 substantiellement le débit (voir également l'Annexe G). Toutefois, un mode de codage par trames PB amélioré est aussi indiqué (voir également l'Annexe M). Le mode de codage par trames PB original est retenu ici uniquement pour des raisons de compatibilité avec les systèmes établis avant l'adoption du mode de codage par trames PB amélioré.

3.4.6 Correction d'erreur sans voie de retour

Une méthode de correction d'erreur sans voie de retour est spécifiée pour utilisation au besoin pour la protection du flux binaire vidéo. Cette méthode de correction d'erreur sans voie de retour est la même que la méthode de code BCH spécifiée dans la Recommandation H.261 (voir également l'Annexe H).

3.4.7 Mode de codage INTRA avancé

Dans ce mode facultatif, les blocs INTRA sont d'abord prédits à partir des blocs INTRA voisins, avant le codage (voir également l'Annexe I). Des tables distinctes de codage à longueur variable (VLC, *variable length code*) sont définies pour les blocs INTRA. La technique est appliquée aux macroblocs INTRA, tant à l'intérieur des images INTRA qu'à l'intérieur des images INTER. Ce mode améliore nettement les performances en matière de compression par rapport au codage INTRA de la syntaxe de base de la Recommandation H.263.

3.4.8 Mode avec filtre de dégroupage

Dans ce mode facultatif, un filtre est appliqué aux limites des blocs 8×8 des images I et P décodées, afin de réduire les artefacts de groupage (voir également l'Annexe J). Le filtre a pour but d'atténuer l'occurrence des artefacts des limites de bloc de l'image décodée. Le filtre influe sur l'image utilisée pour la prédiction des images subséquentes et se trouve donc dans la boucle de prédiction de mouvement.

3.4.9 Mode structuré par tranches

Dans ce mode facultatif, une couche de "tranches" remplace la couche de groupes de blocs de la syntaxe de flux binaire (voir également l'Annexe K). Ce mode a pour but de fournir une capacité améliorée de résistance aux erreurs, afin que le flux binaire puisse plus facilement être utilisé avec un système de transmission à transport de paquets sous-jacent, et aussi de minimiser le retard vidéo. Une tranche est semblable à un groupe de blocs, dans le sens où il s'agit d'une couche de la syntaxe qui se trouve entre la couche des images et la couche des macroblocs. Toutefois, l'utilisation d'une couche de tranches permet une subdivision souple de l'image, par opposition à la division fixe et à l'ordre fixe de transmission qu'exige la structure à groupe de blocs.

3.4.10 Informations complémentaires d'amélioration

Des informations complémentaires additionnelles peuvent être incluses dans le flux binaire pour signaler une capacité améliorée d'affichage ou pour donner de l'information aux fins d'utilisation externe (voir également l'Annexe L). Ces informations complémentaires peuvent servir à signaler un gel d'image complète ou d'image partielle, ou une demande de fin de gel, avec ou sans redimensionnement; elles peuvent aussi servir à étiqueter des images ou séquences d'images particulières dans le flux vidéo, pour utilisation externe, et peuvent servir à transmettre de l'information d'incrustation aux fins de composition d'images vidéo. Les informations complémentaires peuvent être présentes dans le flux binaire même si le décodeur n'est pas capable de donner la capacité améliorée de l'utiliser ou même de l'interpréter correctement – il est admis que

les décodeurs rejettent simplement les informations complémentaires, à moins que l'exigence de fournir la capacité demandée ait été négociée par des moyens externes.

3.4.11 Mode de codage par trames PB amélioré

Ce mode facultatif représente une amélioration par rapport à l'option de mode de codage par trames PB (voir également les Annexes G et M). La principale différence entre les deux modes est que dans le mode de codage par trames PB amélioré, chaque bloc B peut être prédit de manière anticipée en utilisant un vecteur cinétique distinct ou être prédit de manière différée à l'aide d'un vecteur zéro. Ce mode améliore nettement l'efficacité du codage dans les cas où les vecteurs P réduits ne sont pas de bons candidats pour la prédiction B. La prédiction différée est particulièrement utile en cas de transition entre la trame P précédente et la trame PB.

3.4.12 Mode de sélection d'image de référence

Ce mode facultatif améliore les performances des communications vidéo en temps réel sur un canal susceptible aux erreurs, en permettant la prédiction temporelle à partir d'images autres que l'image de référence envoyée le plus récemment (voir également l'Annexe N). Ce mode peut servir avec des messages d'état de canal de retour qui sont renvoyés au codeur pour l'informer de la bonne réception ou non de son flux binaire. Dans les environnements de canaux susceptibles aux erreurs, ce mode permet au codeur d'optimiser son codage vidéo en fonction des conditions du canal.

3.4.13 Mode d'échelonnabilité temporelle, SNR et spatiale

Ce mode facultatif prend en charge l'échelonnabilité temporelle, SNR et spatiale (voir également l'Annexe O). L'échelonnabilité suppose que le flux binaire soit composé d'une couche de base et d'une ou plusieurs couches d'amélioration associées. La couche de base est un flux binaire séparément décodable. Les couches d'amélioration peuvent être décodées en conjonction avec la couche de base, afin d'augmenter la qualité perçue, soit en augmentant la fréquence image, soit en augmentant la qualité d'image, ou encore en augmentant la taille de l'image. L'échelonnabilité SNR se rapporte à l'information d'amélioration visant à augmenter la qualité de l'image sans augmenter la résolution de l'image. L'échelonnabilité spatiale se rapporte à l'information d'amélioration visant à augmenter la qualité de l'image en augmentant la résolution de l'image soit horizontalement, soit verticalement, ou dans les deux sens. L'échelonnabilité temporelle est également prise en charge par l'utilisation d'images B. Une image B constitue une amélioration d'échelonnabilité contenant des images qui peuvent être prédites de manière bilatérale à partir de deux images d'une couche de référence, l'une précédant dans le temps l'image en cours et l'autre la suivant dans le temps. Les images B permettent à l'information de couche d'amélioration de servir à augmenter la qualité perçue en augmentant la fréquence image de la séquence vidéo améliorée qui est affichée. Ce mode peut être utile pour les réseaux hétérogènes comportant des capacités diverses de largeur de bande et aussi en conjonction avec les schémas de correction d'erreur.

3.4.14 Mode de rééchantillonnage d'image de référence

La présente Recommandation présente une syntaxe permettant de prendre en charge un mode facultatif dans lequel l'image de référence utilisée pour la prédiction de l'image vidéo est traitée par une opération de rééchantillonnage avant d'être utilisée pour former un prédicteur pour l'image d'entrée en cours (voir également l'Annexe P). Ce mode permet la sélection dynamique efficace d'une résolution d'image appropriée pour le codage vidéo et permet aussi la prise en charge de la distorsion d'image pour utilisation en tant que compensateur de mouvement global ou comme générateur d'effets spéciaux.

3.4.15 Mode de rafraîchissement à résolution réduite

Ce mode facultatif permet le rafraîchissement avec une résolution réduite par rapport à une image de référence ayant une résolution supérieure (voir également l'Annexe Q). On s'attend à ce que ce mode soit utilisé pour le codage d'une scène très active, car il permet au codeur d'augmenter la fréquence d'image à laquelle les parties en mouvement d'une scène peuvent être représentées, tout en maintenant une représentation de résolution plus élevée dans les parties plus statiques de la scène.

3.4.16 Mode de décodage par segments indépendants

Ce mode facultatif permet de construire une image sans aucune dépendance de données par rapport aux segments d'image vidéo à groupes de blocs ou à groupes de blocs multiples, ou aux limites des tranches (voir également l'Annexe R). Ce mode donne une robustesse aux erreurs en empêchant la propagation de données erronées à travers les limites des zones de segment d'image vidéo.

3.4.17 Autre mode INTER VLC

Ce mode facultatif augmente l'efficacité du codage d'image INTER, lorsque des changements sont évidents dans l'image (voir également l'Annexe S). L'amélioration d'efficacité est obtenue en permettant d'utiliser un code VLC conçu à l'origine pour les images INTRA également pour certains coefficients d'image INTER.

3.4.18 Mode de quantification modifié

Ce mode facultatif augmente la capacité de contrôle du débit pour le codage, réduit l'erreur de quantification de chrominance, augmente la gamme des coefficients DCT pouvant être représentés et impose certaines restrictions aux valeurs des coefficients (voir également l'Annexe T). Ce mode modifie la sémantique du paramètre de valeur de l'échelon de quantification différentielle du flux binaire en élargissant la gamme des modifications de valeur de l'échelon qui peuvent être spécifiées. Il réduit aussi la valeur de l'échelon de quantification utilisé pour les données de chrominance. La gamme des niveaux de coefficients DCT est élargie pour s'assurer que toute valeur possible de coefficient puisse être codée avec la précision que permet la valeur de l'échelon. Certaines restrictions sont également imposées aux coefficients dans ce mode, afin d'augmenter les performances de détection d'erreur et de minimiser la complexité du décodeur.

3.5 Débit

Le rythme de transmission est fourni de manière externe. 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 macrobloc et structure de bloc codée pour la chrominance (MCBPC, *macroblock type and coded block pattern for chrominance*) (voir les Tableaux 7 et 8); ou, si l'on utilise la correction d'erreur sans voie de retour (FEC, *forward error correction*), au moyen de trames de bourrage de type FEC (voir l'Annexe H).

Le nombre de bits 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 de la plus grande taille d'image (picture size) qui a été négociée pour utilisation dans le flux binaire (voir le Tableau 1). La taille d'image est exprimée comme étant la largeur d'image par la hauteur d'image pour la composante de luminance

(Y) et est mesurée en pixels. 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 valeur supérieure soit d'abord négociée par des moyens externes, par exemple la Recommandation H.245.

Quand le mode d'échelonnabilité temporelle, SNR et spatiale (Annexe O) est employé, le nombre de bits envoyés pour chaque image de chaque couche d'amélioration ne doit pas dépasser la valeur maximale spécifiée par BPPmaxKb.

Tableau 1/H.263 – Valeur minimale du paramètre BPPmaxKb pour différents formats d'image source

Taille de l'image Y en pixels	Valeur minimale du paramètre BPPmaxKb
jusqu'à 25 344 (ou QCIF)	64
de 25 360 à 101 376 (ou CIF)	256
de 101 392 à 405 504 (ou 4CIF)	512
405 520 et plus	1024

3.7 Symétrie de transmission

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

3.8 Traitement des erreurs

Il y a lieu d'assurer le traitement des erreurs par des moyens externes (par exemple la Recommandation H.223). S'ils ne sont pas fournis par des moyens externes (par exemple 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*) (ou de tranches si l'on utilise l'Annexe K) de sa prochaine image en mode INTRA, avec des paramètres de codage permettant d'empêcher 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, si le mode structuré par tranches (voir l'Annexe K) n'est pas utilisé. La méthode de transmission de ces signaux relève de moyens externes (par exemple 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 ayant un format de source qui est défini en fonction:

- 1) du format d'image, déterminé par le nombre de pixels par ligne, par le nombre de lignes par image et par le rapport d'aspect de pixel (PAR, *pixel aspect ratio*);

- 2) du rythme des images, déterminé par la fréquence de base d'image (PCF, *picture clock frequency*). Par exemple, le format intermédiaire commun (CIF, *common intermediate format*) compte 352 pixels par ligne, 288 lignes, un rapport d'aspect de pixel de 12:11 et une fréquence de base d'image de 30 000/1001 images par seconde.

Le codeur de source agit sur des images non entrelacées apparaissant à une fréquence de base d'image (PCF) de 30 000/1001 (soit environ 29,97) fois par seconde, que l'on appelle PCF CIF. Il est aussi possible de négocier l'utilisation d'une fréquence PCF facultative par des moyens externes. Une fréquence PCF personnalisée est donnée par l'expression $1\ 800\ 000 / (\text{diviseur d'horloge} * \text{facteur de conversion d'horloge})$ où le diviseur d'horloge peut prendre une valeur de 1 à 127, et où le facteur de conversion d'horloge peut être soit 1000, soit 1001. La tolérance de la fréquence de base 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 BT.601-5.

- 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. Il est aussi possible de négocier un format d'image personnalisé. Dans tous ces formats d'image, la structure d'échantillonnage de la luminance est de dx pixels par ligne et de dy lignes par image en disposition orthogonale. L'échantillonnage de chacune des deux composantes de différence de couleur est effectué tous les $dx/2$ pixels par ligne et toutes les $dy/2$ lignes par image, en disposition orthogonale. Les valeurs des coefficients dx , dy , $dx/2$ et $dy/2$ sont indiquées dans le Tableau 2 pour chacun des formats d'image normalisés.

Tableau 2/H.263 – Nombre de pixels par ligne et nombre de lignes pour chacun des formats d'image normalisés 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 tous les 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 d'aspect de pixel est le même pour chacun de ces formats d'image normalisés et correspond à celui qui est défini pour les formats QCIF et CIF dans la Recommandation H.261: $(288/3):(352/4)$, ce qui se simplifie en 12:11 en nombres premiers entre eux. La zone d'image couverte par tous les formats d'image normalisés (sauf le format sub-QCIF) a un rapport d'aspect de 4:3.

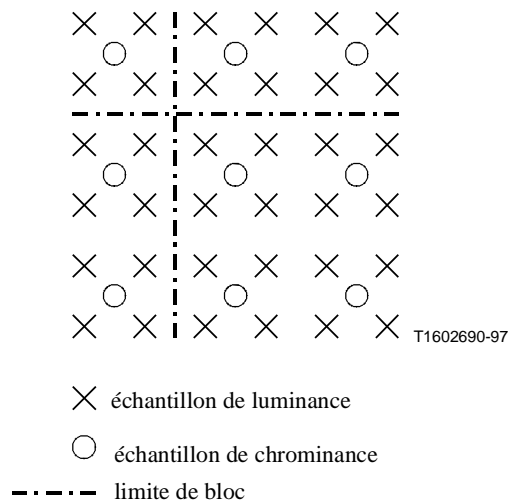


Figure 2/H.263 – Positionnement des échantillons de luminance et de chrominance

Les formats d'image personnalisés peuvent avoir un rapport d'aspect de pixel personnalisé tel qu'indiqué dans le Tableau 3, si l'utilisation d'un rapport d'aspect de pixel personnalisé est d'abord négociée par des moyens externes. Les formats d'image personnalisés peuvent comprendre n'importe quel nombre de lignes et n'importe quel nombre de pixels par ligne, à condition que le nombre de lignes soit divisible par quatre et se trouve dans la gamme [4, ... 1152], et à condition que le nombre de pixels par ligne soit aussi divisible par quatre et se trouve dans la gamme [4, ... 2048]. Pour les formats d'image ayant une largeur ou une hauteur qui n'est pas divisible par 16, l'image est décodée de la même manière que si la largeur ou la hauteur avait la taille immédiatement supérieure qui serait divisible par 16, puis l'image est tronquée à droite et en bas pour atteindre la largeur et la hauteur appropriées, aux fins d'affichage uniquement.

Tableau 3/H.263 – Rapports d'aspect de pixel personnalisés

Rapport d'aspect de pixel (PAR)	Largeur de pixel:hauteur de pixel
carré	1:1
CIF	12:11
type à 525 lignes pour image 4:3	10:11
CIF pour image 16:9	16:11
type à 525 lignes pour image 16:9	40:33
PAR étendu	m:n, m et n nombres premiers entre eux

Tous les décodeurs et codeurs doivent être capables de fonctionner en utilisant la fréquence de base d'image CIF. Certains décodeurs et codeurs peuvent aussi prendre en charge des fréquences de base d'image personnalisées. Tous les décodeurs doivent être capables de fonctionner dans le format d'image sub-QCIF. Tous les décodeurs doivent également être en mesure de fonctionner dans le format d'image QCIF. Certains décodeurs peuvent aussi fonctionner en format CIF, 4CIF ou 16CIF, ou en format d'image personnalisé. Les codeurs doivent être capables de fonctionner avec un des formats d'image 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 format CIF, 4CIF, 16CIF ou en format d'image personnalisé. Des moyens externes signalent les formats facultatifs et les fréquences de base d'image qui peuvent être traités par le décodeur, par exemple selon la Recommandation H.245. Pour avoir un

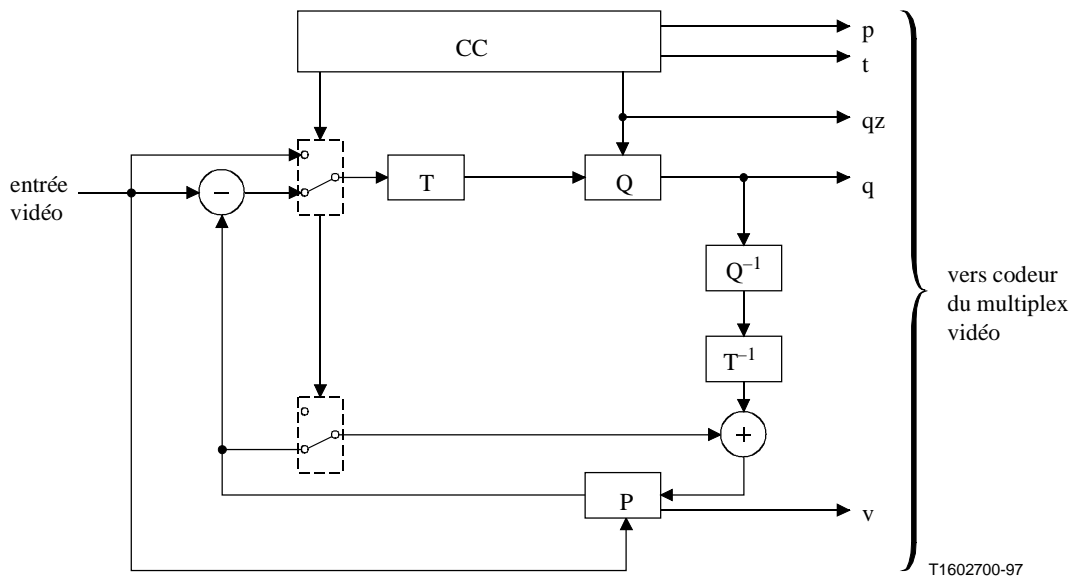
aperçu général complet sur les formats d'image et algorithmes de codage vidéo possibles, se reporter à la description du terminal, par exemple selon la Recommandation H.324.

NOTE – Pour le format CIF, le nombre de pixels par ligne est compatible, à toutes fins pratiques, 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 BT.601-5.

Il doit être possible de limiter la fréquence d'image maximale des codeurs, en ayant un nombre minimal 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, l'image P et l'images B d'une unité de trames PB doivent être considérées comme formant deux images distinctes.

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.



- T transformation
- Q quantificateur
- P mémoire d'image avec retard variable de compensation de mouvement
- CC gestion de 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

4.2.1 Groupes de blocs, tranches, macroblocs et blocs

Chaque image est subdivisée soit en groupes de blocs (GOB), soit en tranches.

Un groupe de blocs comprend jusqu'à $k * 16$ lignes, k dépendant du nombre de lignes dans le format d'image et aussi de l'utilisation ou non du mode facultatif de rafraîchissement à résolution réduite (voir l'Annexe Q). Les relations de dépendance sont illustrées au Tableau 4. Si le nombre de lignes est inférieur ou égal à 400 et que le mode optionnel de rafraîchissement à résolution réduite n'est pas utilisé, alors $k = 1$. Si le nombre de lignes est inférieur ou égal à 800 et que le mode optionnel de rafraîchissement à résolution réduite est utilisé, ou si le nombre de lignes est supérieur à 400, alors $k = 2$. Si le nombre de lignes est supérieur à 800, alors $k = 4$. Quand on utilise des tailles d'image personnalisées, le nombre de lignes dans le groupe de blocs inférieur (dernier groupe) peut être inférieur à $k * 16$ si le nombre de lignes de l'image n'est pas divisible par $k * 16$. Toutefois, chaque groupe de blocs dans chacun des formats d'image normalisés compte $k * 16$ lignes, puisque le nombre de lignes de chaque format d'image normalisé est un multiple entier de $k * 16$. Ainsi, par exemple, si le mode facultatif à résolution réduite n'est pas utilisé, le nombre de groupes de blocs par image est de 6 pour le format sub-QCIF, 9 pour QCIF et 18 pour CIF, 4CIF et 16CIF. La numérotation des groupes GOB est faite par balayage vertical 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 de chaque groupe de blocs comprennent un en-tête GOB (qui peut être vide) suivi des données relatives aux macroblocs. Les données relatives aux groupes GOB sont transmises groupe par groupe, en ordre de numérotation croissant.

Tableau 4/H.263 – Valeur du paramètre k selon la définition de la taille des groupes de blocs

Nombre de lignes dy	Valeur de k quand le mode RRU n'est pas utilisé	Valeur de k quand le mode RRU est utilisé
4,...,400	1	2
404,...,800	2	2
804,...,1152	4	4

Le mode structuré en tranches est décrit à l'Annexe K. Les tranches sont semblables aux groupes de blocs dans le sens où elles constituent une couche à macroblocs multiples de la syntaxe, mais les tranches ont une forme et une utilisation plus souples que les groupes de blocs, et les tranches peuvent apparaître dans le flux binaire dans n'importe quel ordre, sous réserve de certaines conditions.

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

Chaque groupe GOB est divisé en macroblocs. La structure des macroblocs dépend de l'utilisation ou non du mode facultatif de rafraîchissement à résolution réduite (RRU, *reduced-resolution update*) (voir l'Annexe Q). Sauf lorsque le mode RRU est utilisé, chaque 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 donc à 8 pixels par 8 lignes de composante Y, C_B ou C_R. Sauf lorsque le mode RRU est utilisé, 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 quatre rangées de macroblocs pour le format 16CIF.

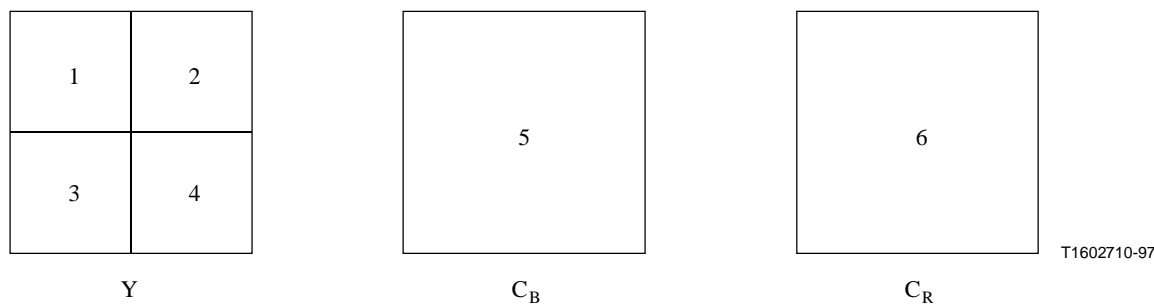


Figure 5/H.263 – Disposition des blocs dans un macrobloc

Lorsque le mode RRU est utilisé, un macrobloc correspond à 32 pixels par 32 lignes de composante Y et, spatialement, aux 16 pixels par 16 lignes des composantes C_B et C_R . Chaque bloc de luminance ou de chrominance correspond à 16 pixels par 16 lignes de composante Y, C_B ou C_R . De plus, un groupe de blocs comprend une seule rangée de macroblocs pour les formats CIF et 4CIF et deux rangées de macroblocs pour le format 16CIF.

La numérotation des macroblocs est faite par 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 relatives aux 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 de contrôle du codage. Les blocs transmis sont transformés et les coefficients résultants sont quantifiés et subissent un codage entropique.

4.2.2 Prédiction

La forme primaire de 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 temporelle est appliquée est appelé codage INTER; si aucune prédiction temporelle n'est appliquée, le mode de codage est appelé codage INTRA. Ce dernier mode peut être signalé au niveau des images (INTRA pour les images I ou INTER pour les images P) ou au niveau du macrobloc dans les images P. Dans le mode facultatif de codage par trames PB, les images de type B sont toujours codées en mode INTER. Les images B sont en partie prédites bilatéralement (voir l'Annexe G).

Au total, la Recommandation H.263 compte sept types d'image de base (dont les deux premiers seulement sont obligatoires), qui sont principalement définis en fonction de leur structure de prédiction:

- 1) INTRA: image n'ayant aucune image de référence aux fins de prédiction (aussi appelée image I);
- 2) INTER: image utilisant une image de référence précédente dans le temps (aussi appelée image P);
- 3) PB: trame représentant deux images et utilisant une image de référence précédente dans le temps (voir l'Annexe G);
- 4) trame PB améliorée: trame fonctionnellement semblable à une trame PB, mais normalement meilleure (voir l'Annexe M);
- 5) B: image ayant deux images de référence, l'une précédant l'image B dans le temps et l'autre lui succédant dans le temps et ayant la même taille d'image (voir l'Annexe O);
- 6) EI: image ayant une image de référence simultanée dans le temps dont la taille d'image est égale ou inférieure (voir l'Annexe O);
- 7) EP: image ayant deux images de référence, l'une précédant l'image EP dans le temps et l'autre étant simultanée dans le temps et ayant une taille d'image égale ou inférieure (voir l'Annexe O).

L'expression image de "référence" ou "d'ancrage" utilisée dans la présente Recommandation signifie une image qui contient des données pouvant être utilisées en tant que référence, comme une base pour le décodage d'une autre image. Cette utilisation par référence est aussi connue sous le nom de "prédiction", bien que ce terme puisse parfois indiquer l'utilisation dans le sens temporel inverse.

4.2.3 Compensation de mouvement

Le décodeur acceptera un seul vecteur par macrobloc ou, si l'on utilise le mode de prédiction avancé ou le mode avec filtre de dégroupage, un ou quatre vecteurs par macrobloc (voir les Annexes F et J). 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. De façon semblable, un macrobloc du mode par trames PB amélioré (voir l'Annexe M) peut inclure un vecteur cinétique anticipé additionnel. Les macroblocs d'image B (voir l'Annexe O) peuvent être transmis avec un vecteur cinétique anticipé et un vecteur différé, et les images EP peuvent être transmises avec un vecteur cinétique anticipé.

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 la gamme $[-16, 15,5]$ (ce qui est également valable pour les composantes des vecteurs cinétiques anticipés et différés pour images B).

Dans le mode de codage par vecteurs cinétiques non restreints, la gamme maximale des composantes vectorielles est cependant augmentée. Si le champ PLUSPTYPE est absent, la gamme est $[-31,5, 31,5]$, avec la restriction que seules pourront être atteintes les valeurs comprises dans la gamme $[-16, 15,5]$ autour du prédicteur de chaque composante vectorielle cinétique, si ce prédicteur est compris dans la gamme $[-15,5, 16]$. Si le champ PLUSPTYPE est absent et que le prédicteur est à l'extérieur de la gamme $[-15,5, 16]$, toutes les valeurs comprises dans la gamme $[-31,5, 31,5]$, ayant le même signe que le prédicteur, plus la valeur zéro, pourront être atteintes. Si le champ PLUSPTYPE est présent, les valeurs de vecteur cinétiques sont moins restreintes (voir également l'Annexe D).

Dans le mode de rafraîchissement à résolution réduite, la gamme des vecteurs cinétiques a environ doublé de taille, et chaque composante vectorielle est restreinte à la valeur zéro ou à une valeur d'un demi-entier. Par conséquent, la gamme de chaque composante vectorielle cinétique est $[-31,5, 30,5]$ dans le mode par défaut de rafraîchissement à résolution réduite (voir l'Annexe Q) et est plus large si le mode de codage par vecteurs cinétiques non restreints est aussi utilisé (voir également l'Annexe D).

Une valeur positive de la composante horizontale ou verticale du vecteur cinétique 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 lorsqu'on utilise le mode de codage par vecteurs cinétiques non restreints, le mode de prédiction avancé ou le mode avec filtre de dégroupage (voir les Annexes D, F et J), ou dans les images B et EP du mode d'échelonnabilité temporelle, SNR et spatiale (voir l'Annexe O).

4.2.4 Quantification

Sauf dans le cas où le mode facultatif de codage INTRA avancé ou le mode de quantification modifié est utilisé, le nombre de quantificateurs est de 1 pour le premier coefficient des blocs de type INTRA et de 31 pour tous les autres coefficients. À 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 uniformément avec une valeur d'échelon de 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. Pour la quantification utilisant le mode de codage INTRA avancé, voir l'Annexe I. Pour la quantification utilisant le mode de quantification modifié, se reporter à l'Annexe T.

NOTE – Pour les échelons de quantification de faible valeur, on ne peut pas représenter toute la dynamique des coefficients de la transformée, à moins que le mode facultatif de quantification modifié ne soit utilisé.

4.3 Contrôle du codage

On peut faire varier divers paramètres afin de contrôler 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 du contrôle ne fait l'objet d'aucune recommandation particulière.

Lorsqu'on fait appel au sous-échantillonnage temporel, celui-ci se fait par mise à l'écart 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 Rafrâichissement forcé

On obtient cette fonction en forçant l'utilisation de l'algorithme de codage en mode INTRA. La séquence de rafrâichissement 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. Une exigence semblable s'applique lorsqu'on utilise les images EP facultatives (voir l'Annexe O), en vertu de laquelle chaque macrobloc doit être codé en mode INTRA ou dans un mode plus évolué au moins une fois toutes les 132 fois où des coefficients sont transmis pour ce macrobloc.

4.5 Alignement des codes de déclenchement par les octets

L'alignement des codes de déclenchement par les octets consiste à insérer un mot de bourrage codé se composant de moins de 8 bits zéro 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, de tranche et EOSBS doivent être alignés par les octets. Les codes de déclenchement de groupe GOB et de fin de séquence EOS peuvent être alignés par les octets.

NOTE 1 – Le nombre de bits consacrés à une image donnée est variable mais toujours un multiple de 8 bits.

NOTE 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

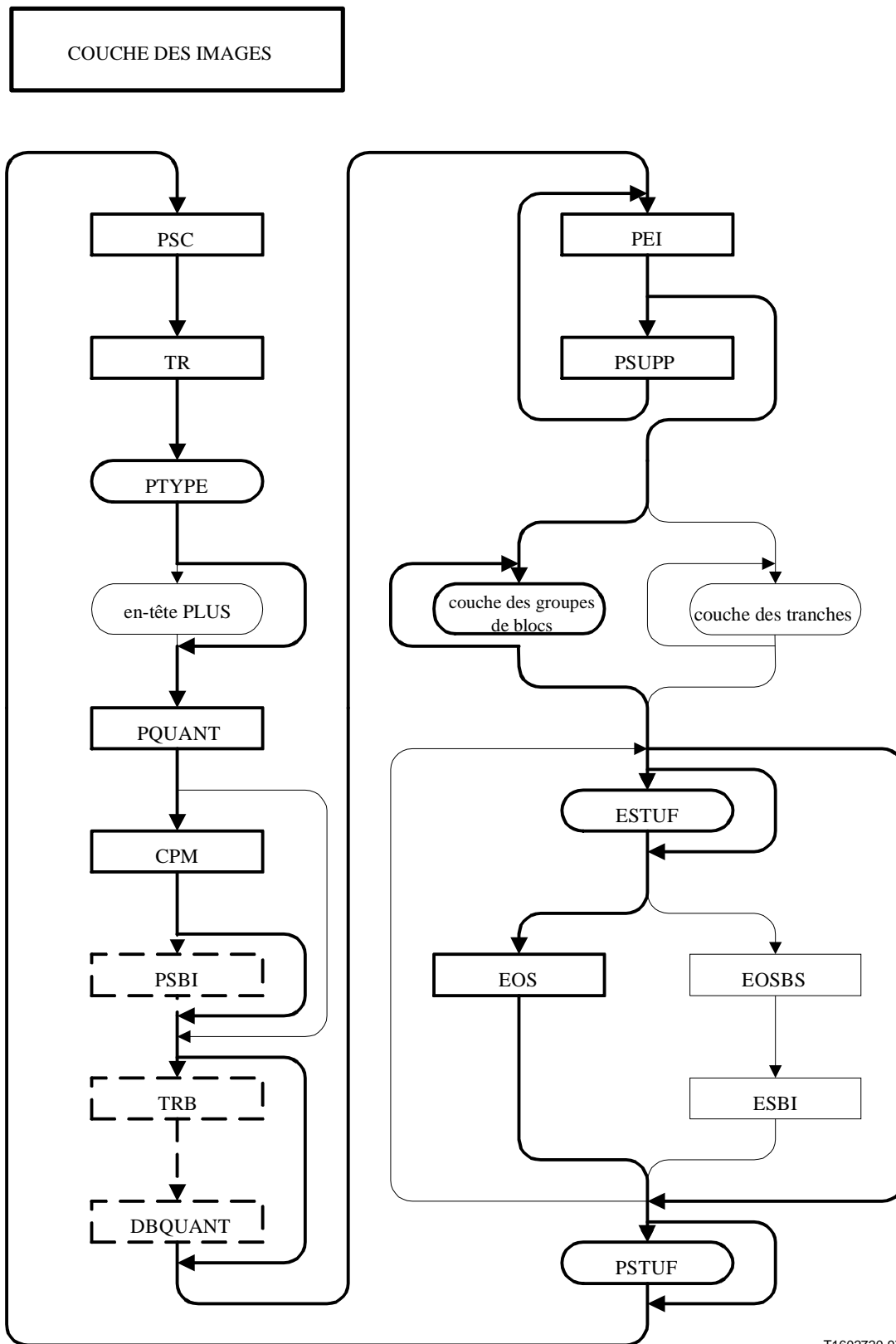
La syntaxe vidéo est disposée selon une structure hiérarchique à quatre couches primaires. Celles-ci sont, de haut en bas, les suivantes:

- couche des images;
- couche des groupes de blocs, des tranches ou des segments d'image vidéo;
- couche des macroblocs;
- couche des blocs.

Le diagramme syntaxique est représenté à la Figure 6. Pour l'interprétation du diagramme, on peut se guider sur les indications suivantes:

- 1) les trajets fléchés montrent les flux possibles d'éléments syntaxiques. Tout élément syntaxique de longueur zéro est considéré comme étant absent aux fins du diagramme de trajets fléchés (ainsi, par exemple, il existe un trajet fléché contournant le champ PSTUF, malgré la nature obligatoire du champ PSTUF, car la longueur du champ PSTUF pourrait être égale à zéro);
- 2) les abréviations et la sémantique de chaque élément syntaxique sont définis dans les paragraphes ultérieurs;
- 3) l'ensemble d'éléments syntaxiques et de trajets fléchés illustrés par des traits continus épais indiquent le flux de syntaxe du mode de fonctionnement "de base", sans qu'aucune amélioration facultative ne soit utilisée. (Cette syntaxe existait dans la Version 1 de la présente Recommandation et reste inchangée.);
- 4) l'ensemble d'éléments syntaxiques et les trajets fléchés illustrés par des traits pointillés épais indiquent les éléments additionnels du flux de syntaxe correspondant aux améliorations facultatives qui sont présentes tant dans la Version 1 que dans la Version 2 de la présente Recommandation. (Cette syntaxe reste inchangée.);
- 5) l'ensemble d'éléments syntaxiques et les trajets fléchés illustrés par des traits continus fins indiquent les éléments additionnels nouveaux du flux de syntaxe correspondant aux améliorations facultatives qui sont spécifiques aux caractéristiques facultatives additionnelles ajoutées dans la Version 2. (Cette syntaxe n'existait pas dans la Version 1.);
- 6) les champs d'élément syntaxique représentés avec des limites rectangulaires indiquent des champs de longueur fixe, tandis que ceux qui ont des limites arrondies indiquent des champs de longueur variable. Un élément syntaxique (DQUANT) est indiqué avec les deux types de limites, car il peut avoir une longueur fixe ou une longueur variable;
- 7) un champ de longueur fixe est défini comme un champ dont la longueur ne dépend pas des données du contenu du champ lui-même. La longueur de ce champ est soit toujours la même, soit elle est déterminée par les données précédentes du flux de syntaxe;
- 8) le terme "couche" est utilisé pour se reporter à toute partie de la syntaxe qui peut être comprise et qui peut figurer sur le diagramme en tant qu'entité distincte.

Sauf indication contraire, le bit de poids fort est transmis en premier. Il s'agit du bit 1 et c'est le bit 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 mis à "1". Les bits tenus en réserve ne doivent pas être utilisés avant que leur fonction ait été spécifiée par l'UIT-T.

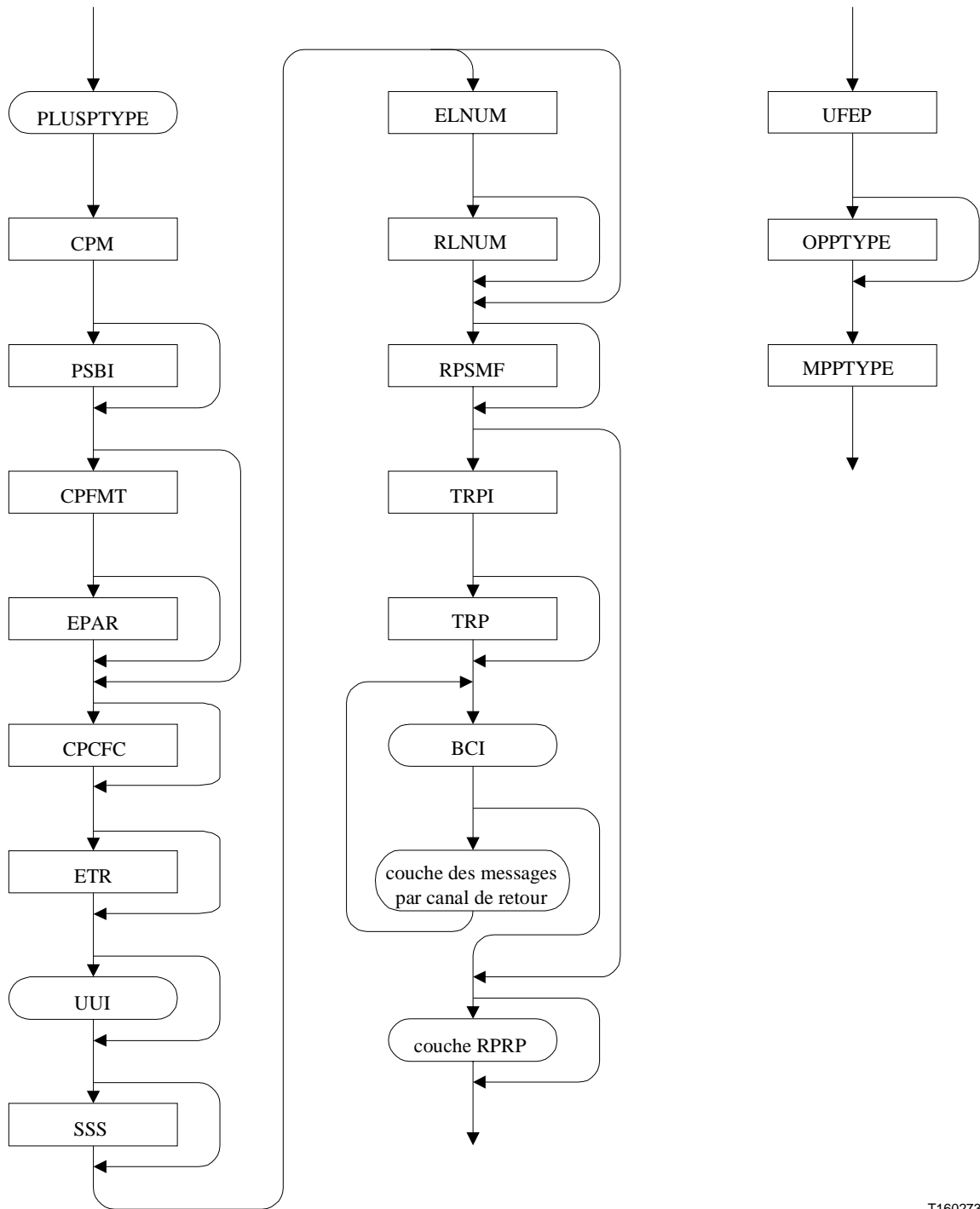


T1602720-97

Figure 6/H.263 (partie 1 de 7) – Diagramme syntaxique du flux vidéo

en-tête PLUS

PLUSPTYPE



T1602730-97

Figure 6/H.263 (partie 2 de 7) – Diagramme syntaxique du flux vidéo

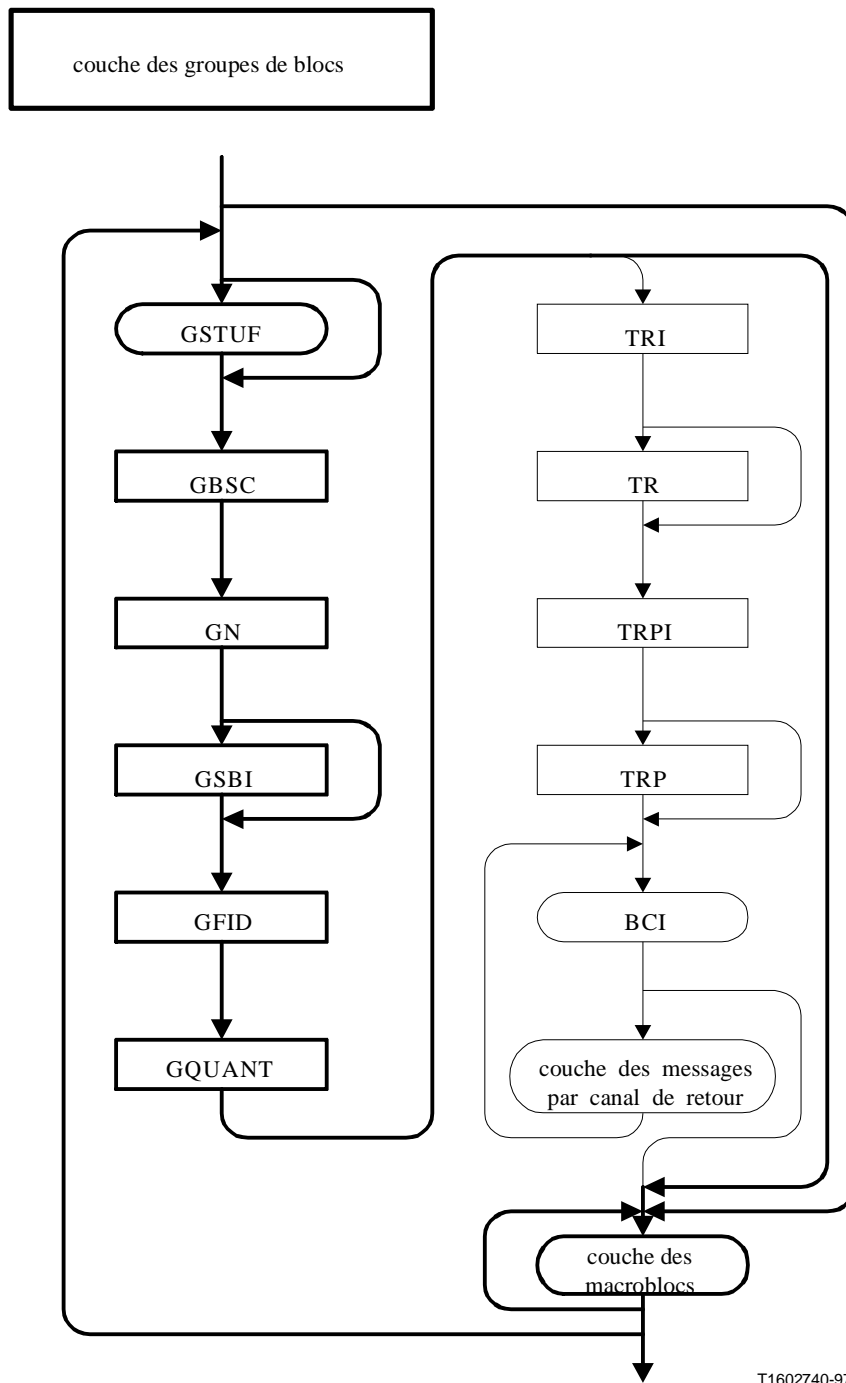
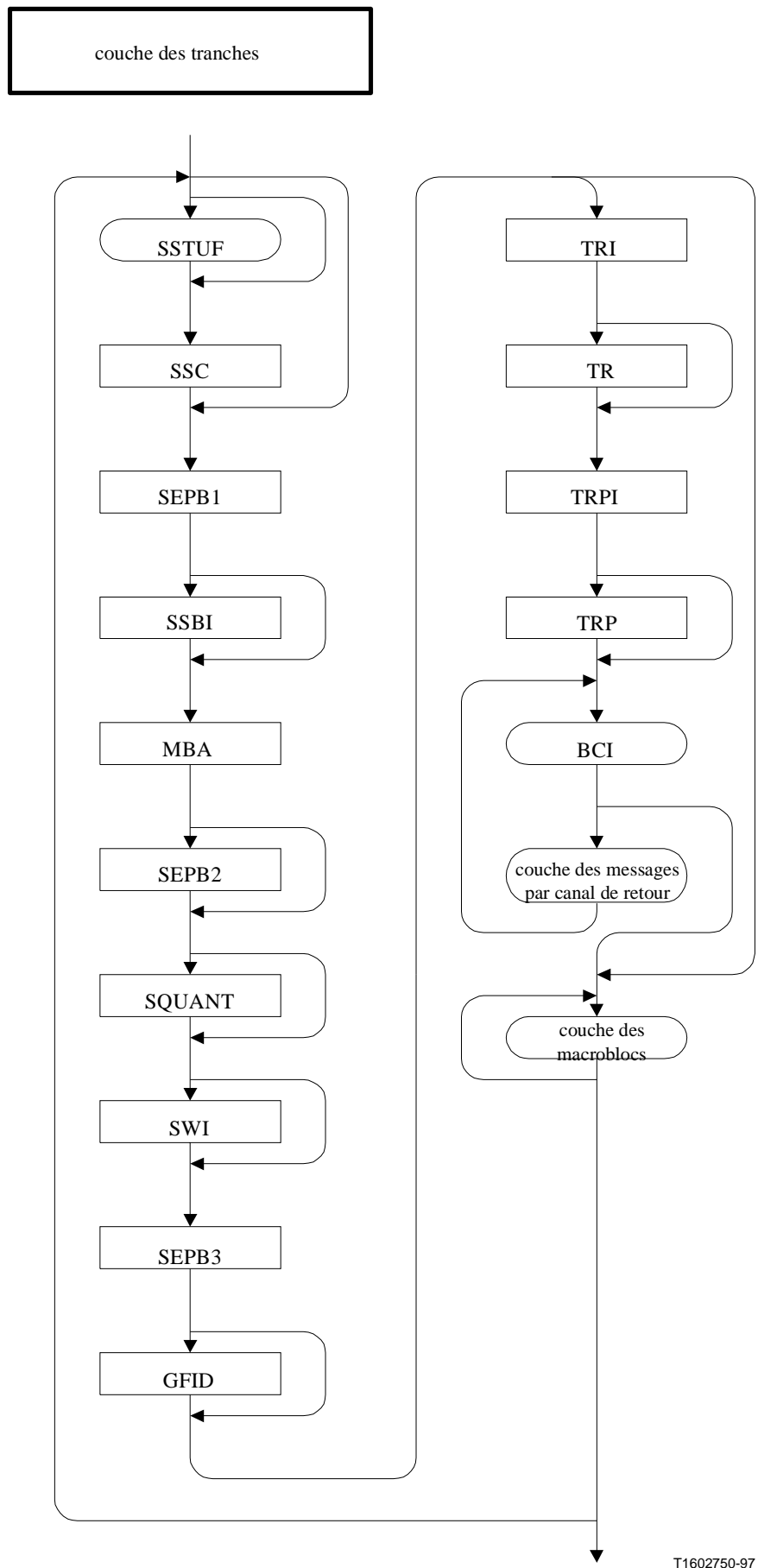


Figure 6/H.263 (partie 3 de 7) – Diagramme syntaxique du flux vidéo



T1602750-97

Figure 6/H.263 (partie 4 de 7) – Diagramme syntaxique du flux vidéo

couche des canaux logiques distincts BCM

couche des messages par canal de retour

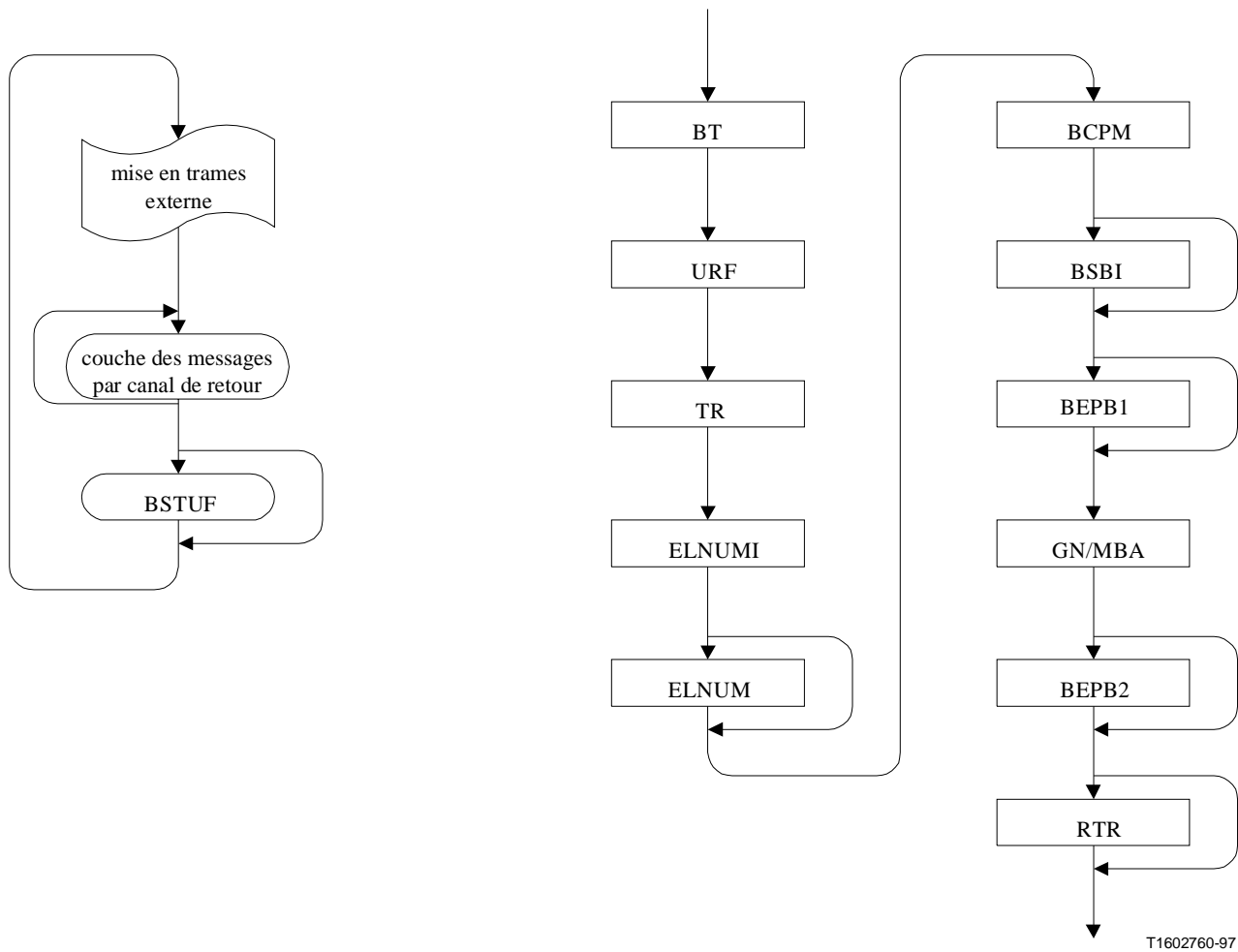
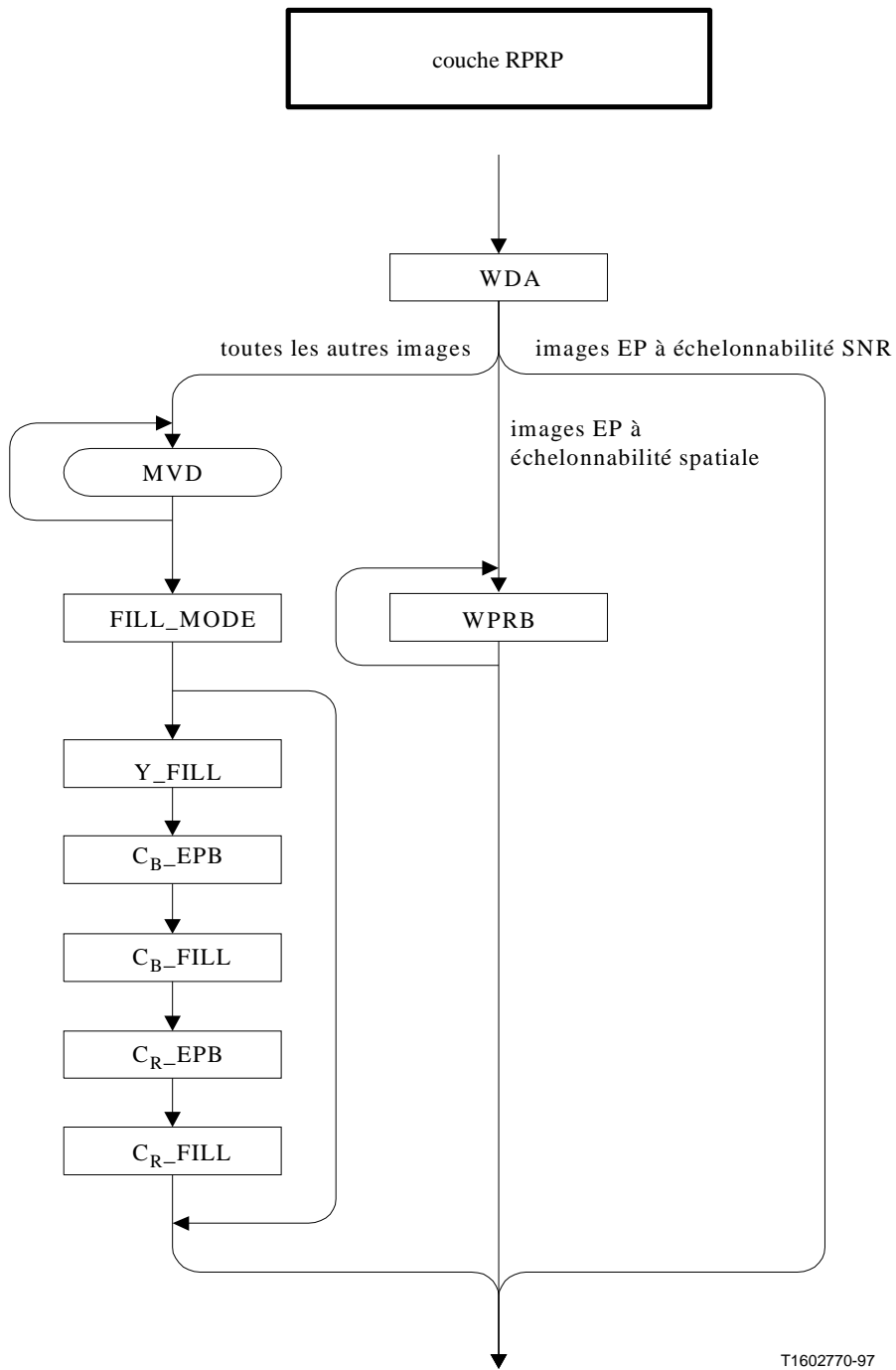
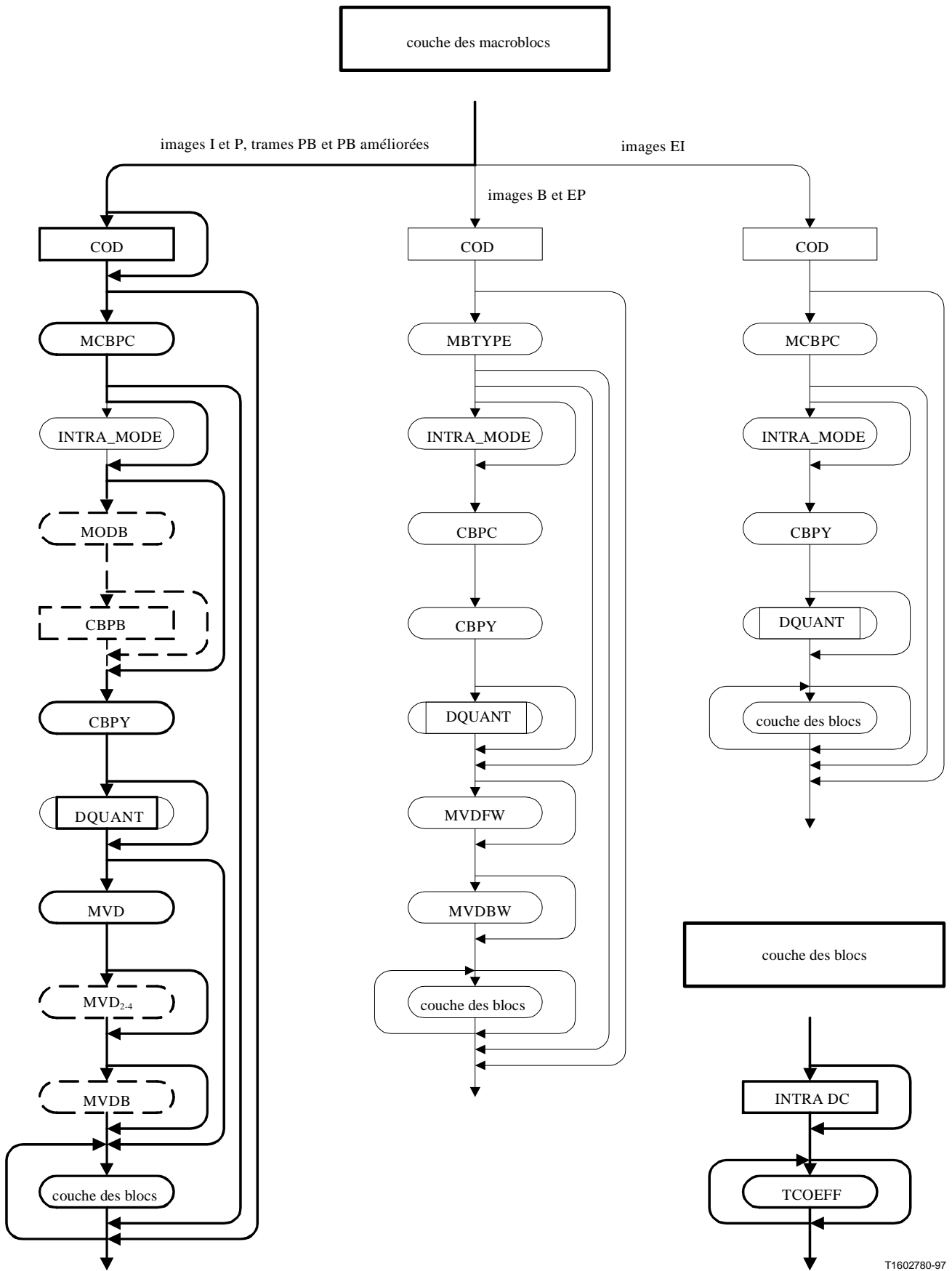


Figure 6/H.263 (partie 5 de 7) – Diagramme syntaxique du flux vidéo



T1602770-97

Figure 6/H.263 (partie 6 de 7) – Diagramme syntaxique du flux vidéo



T1602780-97

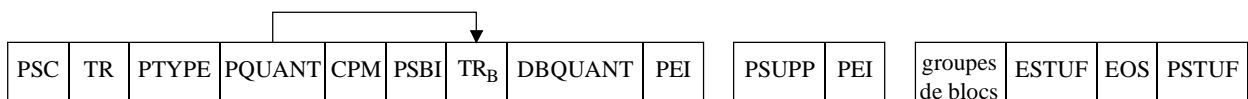
Figure 6/H.263 (partie 7 de 7) – Diagramme syntaxique du flux vidéo

5.1 Couche des images

Pour chaque image, les données comportent un en-tête d'image suivi des données de groupe de blocs ou de tranches, suivies, le cas échéant, d'un code de fin de séquence (EOS, *end-of-sequence*) facultatif et de bits de bourrage. Cette structure est représentée sur la Figure 7 pour les images qui ne comprennent pas le champ facultatif PLUSPTYPE. L'indicateur de sous-flux binaire (PSBI, *picture sub-bitstream indicator*) n'est présent que s'il est indiqué par le champ communication multipoint à présence continue et à multiplex vidéo (CMP, *continuous presence multipoint and video multiplex*). Les codes de référence temporelle pour les trames B (TR_B, *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 champ PTYPE (information de type) indique l'utilisation du mode de codage par "trames PB" (sauf si le champ PLUSPTYPE est présent et indique l'utilisation du champ DBQUANT).

Le champ facultatif de données PLUSPTYPE est présent quand les bits 6-8 du champ PTYPE l'indiquent. Quand il est présent, un jeu additionnel de données est inclus dans le flux binaire, lequel jeu suit immédiatement le champ PTYPE et précède le champ PQUANT. De plus, les champs CPM et PSBI sont avancés dans l'en-tête d'image quand le champ PLUSPTYPE est présent, de sorte qu'ils apparaissent immédiatement après le champ PLUSPTYPE plutôt que de se trouver après le champ PQUANT. Le format des données additionnelles qui suivent le champ PLUSPTYPE est indiqué à la Figure 8. Tous les champs de ces données d'en-tête d'image additionnelles suivant le champ PLUSPTYPE sont facultatifs, selon que leur présence est indiquée ou non dans le champ PLUSPTYPE. Quand on utilise le mode structuré par tranches (voir l'Annexe K), les tranches remplacent les groupes de blocs (GOB) à l'emplacement indiqué à la Figure 7.

Des combinaisons de champs PSUPP et PEI peuvent ne pas être présentes, mais peuvent être répétées si elles sont présentes. Les champs EOS et EOSBS+ESBI peuvent être absents, tandis que le champ ESTUF ne peut être présent que si le champ EOS ou EOSBS l'est également. Le champ EOS ne doit pas être répété, sauf si au moins un code de déclenchement d'image apparaît entre chaque paire de codes EOS. Les en-têtes des images rejetées ne sont pas transmis.



T1602790-97

**Figure 7/H.263 – Structure de la couche des images
(sans les champs facultatifs liés au champ PLUSPTYPE)**

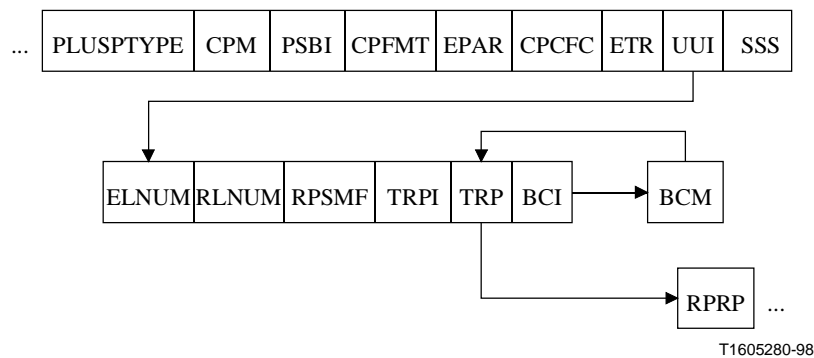


Figure 8/H.263 – Structure des champs facultatifs liés au champ PLUSPTYPE (se trouvant immédiatement après le champ PTYPE, s'ils sont présents)

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

Le code PSC est un mot de 22 bits. Sa valeur est 0000 0000 0000 0000 1 00000. Tous les codes de déclenchement d'image doivent être alignés par les octets, ce qui doit être effectué par insertion de bits PSTUF selon les besoins, 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)

La valeur de TR est formée en ajoutant, à sa valeur contenue dans l'en-tête d'image de référence transmis précédemment dans le temps, 1 plus le nombre d'images sautées ou d'images autres que de référence à la fréquence de base d'image depuis la dernière image transmise. L'interprétation du champ TR dépend de la fréquence de base d'image active. Selon la fréquence de base d'image CIF normalisée, TR est un nombre de 8 bits qui peut prendre 256 valeurs. Les opérations arithmétiques sont effectuées sur les huit bits de poids faible seulement. Si une fréquence de base d'image personnalisée est signalée en utilisation, le champ TR étendu du 5.1.8 et le champ TR forment un nombre de 10 bits dans lequel TR emmagasine les huit bits de plus faible poids (LSB, *least significant bit*) et ETR emmagasine les deux bits de plus fort poids (MSB, *most significant bit*). Les opérations arithmétiques sont alors effectuées sur les dix bits. En mode facultatif de codage par trames PB ou par trames PB amélioré, la référence temporelle TR ne s'applique qu'aux images P. Pour la référence temporelle de la partie image B des trames PB ou PB améliorées, voir 5.1.22.

5.1.3 Informations de type (PTYPE, *type information*) (longueur variable)

Informations sur l'image complète:

- bit 1: toujours "1", afin d'empêcher l'émulation du code de déclenchement;
- bit 2: toujours "0", par opposition à la Recommandation H.261;
- bit 3: indicateur de partage d'écran: "0" non activé, "1" activé;
- bit 4: indicateur de caméra document: "0" non activé, "1" activé;
- bit 5: fin de gel d'image complète: "0" non activé, "1" activé;
- bits 6-8: format de source: "000" interdit, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" réservé, "111" PTYPE étendu;

Si les bits 6-8 n'ont pas une valeur de "111", qui indique un champ PTYPE étendu (PLUSPTYPE), les cinq bits qui suivent sont également présents dans le champ PTYPE:

- bit 9: type de codage d'image: "0" INTRA (image de type I), "1" INTER (image de type P);

- bit 10: mode facultatif par vecteurs cinétiques non restreints (voir l'Annexe D): "0" non activé, "1" activé;
- bit 11: mode facultatif de codage arithmétique syntaxique (voir l'Annexe E): "0" non activé, "1" activé;
- bit 12: mode facultatif de prédiction avancé (voir l'Annexe F): "0" non activé, "1" activé;
- bit 13: mode facultatif de codage par trames PB (voir l'Annexe G): "0" image normale de type I ou P, "1" trame PB.

L'indicateur de partage d'écran est un signal qui indique que les moitiés supérieure et inférieure de l'image décodée pourraient 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 complète est un signal issu d'un codeur qui répond à une demande de retransmission en mode paquets (s'il n'y a pas eu d'accusé de réception) ou à une demande de rafraîchissement rapide (voir également l'Annexe C) ou encore à une demande de gel d'image (voir également l'Annexe L). 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 en cours doit être de type I, sauf si un champ PTYPE étendu est indiqué dans les bits 6-8 et si la capacité d'utiliser le mode facultatif de rééchantillonnage d'image de référence a été négocié par des moyens externes (par exemple selon la Recommandation H.245).

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".

Les bits 6-8 ne doivent pas avoir une valeur de "111", qui indique la présence d'un champ PTYPE étendu (PLUSPTYPE), sauf si la capacité a été négociée par des moyens externes (par exemple selon la Recommandation H.245) pour permettre l'utilisation d'un format de source personnalisé ou d'un ou de plusieurs des autres modes facultatifs disponibles seulement par l'utilisation d'un champ PTYPE étendu (voir les Annexes I à K et M à T). Quand les bits 6-8 n'ont pas la valeur de "111", tous les modes additionnels disponibles seulement par l'utilisation d'un champ PTYPE étendu doivent être considérés comme ayant été mis à un état "inactive" et doivent être considérés comme restant à cet état, à moins qu'ils ne soient explicitement commutés plus tard dans le flux binaire.

5.1.4 Champ PTYPE Plus (*PLUSPTYPE*) (longueur variable)

Il s'agit d'un mot de code de 12 ou 30 bits qui est présent seulement si la présence du champ PTYPE étendu est indiqué dans les bits 6-8 de PTYPE.

Le champ PLUSPTYPE comprend jusqu'à trois sous-champs; soit UFEP, OPPTYPE et MPPTYPE. Le sous-champ OPPTYPE est présent seulement si le champ UFEP a une valeur particulière.

5.1.4.1 Sous-champ de mise à jour du champ PTYPE étendu (UFEP, *update full extended PTYPE*) (3 bits)

Il s'agit d'un mot de code de longueur fixe de 3 bits qui est présent seulement si le champ "PTYPE étendu" est indiqué dans les bits 6-8 du champ PTYPE. Mis à "000", il indique que seuls les champs PTYPE étendus qui ont besoin d'être signalés dans chaque en-tête d'image (MPPTYPE) sont inclus dans l'en-tête d'image en cours. Mis à "001", il indique que tous les champs PTYPE étendus sont inclus dans l'en-tête d'image en cours. Si l'image est de type INTRA ou EI, ce champ doit être mis à "001".

En outre, si le champ PLUSPTYPE est présent dans chacune des images d'une séquence continue d'images, ce champ doit être mis à "001", au moins aussi souvent que le spécifie une période de fin de temporisation de cinq secondes ou de cinq images, selon la période la plus longue. Plus spécifiquement, la période de fin de temporisation exige que UFEP = "001" apparaisse dans le champ PLUSPTYPE (si ce champ PLUSPTYPE est présent dans chacune des images en cause) du premier en-tête d'image ayant une référence temporelle indiquant une gamme de temps supérieure ou égale à cinq secondes depuis la dernière occurrence de UFEP = "001", ou de la cinquième image après la dernière occurrence de UFEP = "001" (selon l'exigence qui donne la plus longue période de temps mesurée par référence temporelle).

Les codeurs devraient mettre le sous-champ UFEP à "001" plus souvent dans les environnements susceptibles aux erreurs. Les valeurs du sous-champ UFEP autres que "000" et "001" sont réservées.

5.1.4.2 Partie facultative du champ PLUSPTYPE (OPPTYPE, *optional part of PLUSPTYPE*) (18 bits)

Si le sous-champ UFEP est à "001", alors les bits qui suivent sont présents dans le champ PLUSPTYPE:

- bits 1-3 format de source: "000" réservé, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" format de source personnalisé, "111" réservé;
- bit 4 fréquence PCF personnalisée facultative: "0" PCF CIF, "1" PCF personnalisé;
- bit 5 mode facultatif de codage par vecteurs cinétiques non restreints (UMV, *unrestricted motion vector*) (voir l'Annexe D): "0" non activé, "1" activé;
- bit 6 mode facultatif de codage arithmétique syntaxique (SAC, *syntax-based arithmetic coding*) (voir l'Annexe E): "0" non activé, "1" activé;
- bit 7 mode facultatif de prédiction avancé (AP, *advanced prediction*) (voir l'Annexe F): "0" non activé, "1" activé;
- bit 8 mode facultatif de codage INTRA avancé (AIC, *advanced INTRA coding*) (voir l'Annexe D): "0" non activé, "1" activé;
- bit 9 mode facultatif avec filtre de dégroupage (DF, *deblocking filter*) (voir l'Annexe J): "0" non activé, "1" activé;
- bit 10 mode facultatif structuré par tranches (SS, *slice structured*) (voir l'Annexe K): "0" non activé, "1" activé;
- bit 11 mode facultatif de sélection d'image de référence (RPS, *reference picture selection*) (voir l'Annexe N): "0" non activé, "1" activé;
- bit 12 mode facultatif de décodage par segments indépendants (ISD, *independent segment decoding*) (voir l'Annexe R): "0" non activé, "1" activé;
- bit 13 autre mode facultatif INTER VLC (AIV, *alternative INTER VLC*) (voir l'Annexe S): "0" non activé, "1" activé;
- bit 14 mode facultatif de quantification modifié (MQ, *modified quantization*) (voir l'Annexe T): "0" non activé, "1" activé;
- bit 15 égal à "1" pour empêcher l'émulation du code de déclenchement;
- bit 16 réservé, doit être égal à "0";
- bit 17 réservé, doit être égal à "0";
- bit 18 réservé, doit être égal à "0".

5.1.4.3 Partie obligatoire du champ PLUSPTYPE quand le champ PLUSPTYPE est présent (MPPTYPE, *mandatory part of PLUSPTYPE when PLUSPTYPE present* (MPPTYPE)) (9 bits)

Quelle que soit la valeur du sous-champ UFEP, les 9 bits suivants sont toujours également présents dans le champ PLUSPTYPE:

- bits 1-3 Code de type d'image:
 - "000" image I (INTRA);
 - "001" image P (INTER);
 - "010" trame PB améliorée (voir l'Annexe M);
 - "011" image B (voir l'Annexe O);
 - "100" image EI (voir l'Annexe O);
 - "101" image EP (voir l'Annexe O);
 - "110" réservé;
 - "111" réservé;
- bit 4 mode facultatif de rééchantillonnage d'image de référence (RPR, *reference picture resampling*) (voir l'Annexe P): "0" non activé, "1" activé;
- bit 5 mode facultatif de rafraîchissement à résolution réduite (RRU) (voir l'Annexe Q): "0" non activé, "1" activé;
- bit 6 type d'arrondissement (RTYPE, *rounding type*) (voir 6.1.2);
- bit 7 réservé, doit être égal à "0";
- bit 8 réservé, doit être égal à "0";
- bit 9 égal à "1" pour empêcher l'émulation du code de déclenchement.

Le codeur devrait contrôler le type d'arrondissement, de sorte que les images P, les trames PB améliorées et les images EP aient des valeurs différentes pour le bit 6 (type d'arrondissement pour les images P) par rapport à leurs images de référence pour la compensation du mouvement. Le bit 6 peut avoir une valeur arbitraire si l'image de référence est une image I ou une image EI. Le bit 6 peut être mis à "1" seulement si les bits 1-3 indiquent une image P, une trame PB améliorée ou une image EP. Pour les autres types d'image, ce bit doit toujours être mis à "0".

5.1.4.4 Sémantique du champ PLUSPTYPE

La partie obligatoire du champ PLUSPTYPE comprend des caractéristiques susceptibles de changer d'image en image. Ce sont essentiellement les bits qui indiquent le type d'image parmi les images I, P, PB améliorée, B, EI et EP. (Il est à noter que le mode de trames PB de l'Annexe G ne peut pas être utilisé si le champ PLUSPTYPE est présent – le mode de trames PB amélioré de l'Annexe M devrait être utilisé à la place.) Toutefois, ces caractéristiques comprennent aussi des indications pour l'utilisation des modes RPR et RRU, car ceux-ci peuvent changer également d'image en image.

Les caractéristiques qui resteront vraisemblablement inchangées d'une image à l'autre (sauf dans les cas évidents discutés au 5.1.4.5) ont été mises dans la partie facultative du champ PLUSPTYPE. Quand le sous-champ UFEP est à 000, l'information de mode manquante est déduite du type d'image et de l'information de mode envoyée dans un champ PLUSPTYPE précédent pour lequel le sous-champ UFEP était égal à 001.

Si le champ PLUSPTYPE est présent, mais que UFEP est égal à 000, alors:

- 1) pour une image P ou une trame PB améliorée (voir l'Annexe M), le rapport d'aspect de pixel, la largeur d'image et la hauteur d'image sont inchangés par rapport à ceux de l'image de référence;
- 2) pour une image B à échelonnabilité temporelle (voir l'Annexe O) dans une couche d'amélioration, le numéro de couche de référence (RLNUM, *reference layer number*) est le même que le numéro de couche d'amélioration (ELNUM, *enhancement layer number*) si la dernière image envoyée dans la couche d'amélioration était une image EI ou EP. Si la dernière image envoyée dans la couche d'amélioration était une image B, le numéro de couche de référence est le même que le numéro de couche de référence de la dernière image B. Le rapport d'aspect de pixel, la largeur d'image et la hauteur d'image sont inchangés par rapport à ceux de l'image de la couche de référence subséquente dans le temps.

Il convient de noter que si des images EI ou EP de l'entourage temporel existent dans la même couche d'amélioration que l'image B, le champ RLNUM (explicite ou implicite) doit toujours être égal au champ ELNUM. Il faut noter également que le rapport d'aspect de pixel, la largeur d'image et la hauteur d'image d'une image B (explicite ou implicite) doivent toujours être égaux à ceux de son image de couche de référence subséquente dans le temps;

- 3) Pour une image EP à échelonnabilité SNR/spatiale (voir l'Annexe O), le rapport d'aspect de pixel, la largeur d'image et la hauteur d'image restent inchangés par rapport à ceux de l'image de référence précédente dans le temps dans la même couche d'amélioration.

5.1.4.5 Restrictions de mode pour certains types d'image et règles d'inférence de mode

Certains modes ne s'appliquent pas à certains types d'image. En particulier:

- 1) les modes suivants ne s'appliquent pas à l'intérieur des images I (INTRA): mode de codage par vecteurs cinétiques non restreints (voir l'Annexe D), mode de prédiction avancé (voir l'Annexe F), autre mode INTER VLC (voir l'Annexe S), mode de rééchantillonnage d'image de référence (voir l'Annexe P) et mode de rafraîchissement à résolution réduite (voir l'Annexe Q);
- 2) les modes suivants ne s'appliquent pas à l'intérieur des images B (voir l'Annexe O): mode de codage arithmétique syntaxique (voir l'Annexe E), mode avec filtre de dégroupage (voir l'Annexe J) et mode de prédiction avancé (voir l'Annexe F);
- 3) les modes suivants ne s'appliquent pas à l'intérieur des images EI (voir l'Annexe O): mode par vecteurs cinétiques non restreints (voir l'Annexe D), mode de codage arithmétique syntaxique (voir l'Annexe E), mode de prédiction avancé (voir l'Annexe F), mode de rééchantillonnage d'image de référence (voir l'Annexe P), mode de rafraîchissement à résolution réduite (voir l'Annexe Q) et autre mode INTER VLC (voir l'Annexe S);
- 4) les modes suivants ne s'appliquent pas à l'intérieur des images EP (voir l'Annexe O): mode de codage arithmétique syntaxique (voir l'Annexe E) et mode de prédiction avancé (voir l'Annexe F).

Un ou plusieurs des modes énumérés dans la liste précédente peuvent avoir un fanion de mode ayant une valeur "1" dans la partie facultative du champ PLUSPTYPE à l'intérieur d'une image d'un type qui est interdit pour ce mode (types I, B, EI ou EP). Cette condition est permise et doit être interprétée sous réserve des règles d'inférence de mode qui suivent.

Les états des modes sont soumis aux règles d'inférence de mode suivantes:

- 1) une fois qu'un fanion de mode a été mis à "1" dans la partie facultative du champ PLUSPTYPE, l'image en cours et chacune des images subséquentes du flux binaire doit se voir assigner un état "activé" pour ce mode;

- 2) un état inféré "non activé" doit être assigné à tout mode qui ne s'applique pas à l'intérieur d'une image ayant le code de type d'image en cours. Toutefois, chaque image subséquente du flux binaire doit avoir un état déduit "activé" pour ce mode (sauf si cela résulte aussi en un conflit évident – qui doit être résolu de la même manière). Dans le cas des flux binaires échelonnables par couches (voir l'Annexe O), l'état de mode doit être déduit seulement à partir de l'intérieur de la même couche du flux binaire;
- 3) l'inférence d'état doit continuer jusqu'à ce qu'une image de la même couche soit envoyée, qui, soit contienne la partie facultative du champ PLUSTYPE soit ne contienne pas ce champ. Si une nouvelle image contenant la partie facultative du champ PLUSPTYPE est envoyée, l'état envoyé dans le nouveau message doit l'emporter sur l'ancien état. Si une image est envoyée sans contenir le champ PLUSPTYPE (image dans laquelle les bits 6-8 du champ PTYPE ne sont pas à "111"), un état "non activé" doit être assigné à tous les modes qui ne sont pas explicitement mis à "activé" dans le champ PTYPE, et tous les modes doivent continuer à avoir un état inféré "non activé" jusqu'à ce qu'une nouvelle image contenant la partie facultative du champ PLUSPTYPE soit envoyée;
- 4) deux modes n'exigent pas d'inférence d'état de mode, puisque les fanions de mode de ces modes apparaissent dans la partie obligatoire du champ PLUSPTYPE. Il s'agit du mode de rééchantillonnage d'image de référence (Annexe P) et du mode de rafraîchissement à résolution réduite (Annexe Q). Le fanion de mode de l'un ou l'autre de ces modes ne doit pas être mis à 1, à moins que l'image en cours permette l'utilisation de ce mode. Par exemple, le bit de mode de rafraîchissement à résolution réduite ne doit pas être mis à 1 dans une image INTRA.

5.1.4.6 Restrictions relatives à l'interaction des modes

Certains modes ne peuvent pas être utilisés en combinaison avec certains autres modes.

- 1) Le mode de codage arithmétique syntaxique (voir l'Annexe E) ne doit pas être utilisé avec l'autre mode INTER VLC (voir l'Annexe S) ni avec le mode de quantification modifié (voir l'Annexe T).
- 2) Si le champ PLUSPTYPE est présent, le mode par vecteurs cinétiques non restreints (voir l'Annexe D) ne doit pas être utilisé avec le mode de codage arithmétique syntaxique (voir l'Annexe E).
- 3) Le mode de décodage par segments indépendants (voir l'Annexe R) ne doit pas être utilisé avec le mode de rééchantillonnage d'image de référence (voir l'Annexe P).
- 4) Le mode de décodage par segments indépendants (voir l'Annexe R) ne doit pas être utilisé avec le mode structuré par tranches sans l'utilisation simultanée du sous-mode de tranches rectangulaires du mode structuré par tranches (voir l'Annexe K).

5.1.4.7 Emplacement dans l'en-tête d'image des champs CPM (1 bit) et PSBI (2 bits)

L'emplacement des champs CPM et PSBI dans l'en-tête d'image dépend de la présence ou non du champ PLUSPTYPE (voir 5.1.20 et 5.1.21). Si le champ PLUSPTYPE est présent, alors le champ CPM le suit immédiatement dans l'en-tête d'image. Si le champ PLUSPTYPE n'est pas présent, le champ CPM suit immédiatement le champ PQUANT dans l'en-tête d'image. Le champ PSBI suit toujours immédiatement le champ CPM (si CPM = "1").

5.1.5 Format d'image personnalisé (CPFMT, *custom picture format*) (23 bits)

Mot de code de longueur fixe de 23 bits qui est présent seulement si l'utilisation d'un format d'image personnalisé est signalée dans le champ PLUSPTYPE et si le sous-champ UFEP est à "001". Quand il est présent, le champ CPFMT comprend:

- bits 1-4 code de rapport d'aspect de pixel: index à 4 bits de la valeur de rapport d'aspect de pixel (PAR) dans le Tableau 5. Pour le PAR étendu, le rapport d'aspect de pixel exact doit être précisé dans le champ EPAR (voir 5.1.6);
- bits 5-13 indication de largeur d'image (PWI, *picture width indication*): gamme de [0, ..., 511]; nombre de pixels par ligne = (PWI + 1) * 4;
- bit 14 égal à "1" pour empêcher l'émulation du code de déclenchement;
- bits 15-23 indication de hauteur d'image (PHI, *picture height indication*): gamme [1, ..., 288]; nombre de lignes = PHI * 4.

Tableau 5/H.263 – Définition du code de rapport d'aspect de pixel (PAR)

Code PAR	Rapport d'aspect de pixel
0000	interdit
0001	1:1 (carré)
0010	12:11 (CIF pour image 4:3)
0011	10:11 (type 525 lignes pour image 4:3)
0100	16:11 (CIF allongé pour image 16:9)
0101	40:33 (type 525 lignes allongé pour image 16:9)
0110-1110	réservé
1111	PAR étendu

5.1.6 Rapport d'aspect de pixel étendu (EPAR, *extended pixel aspect ratio*) (16 bits)

Mot de code de longueur fixe de 16 bits qui est présent seulement si le champ CPFMT est présent et si le rapport d'aspect de pixel (PAR) étendu y est indiqué. Quand il est présent, le champ EPAR comprend:

- bits 1-8 largeur PAR: "0" est interdit. Représentation binaire naturelle de la largeur PAR;
- bits 9-16 hauteur PAR: "0" est interdit. Représentation binaire naturelle de la hauteur PAR.

La largeur PAR et la hauteur PAR doivent être exprimées en nombres premiers entre eux.

5.1.7 Code de fréquence de base d'image personnalisée (CPCFC, *custom picture clock frequency code*) (8 bits)

Mot de code de longueur fixe de 8 bits présent seulement si le champ PLUSPTYPE est présent, si le sous-champ UFEP est égal à 001 et si une fréquence de base d'image personnalisée est signalée dans le champ PLUSPTYPE. Quand il est présent, le champ CPCFP comprend:

- bit 1 code de conversion d'horloge: "0" indique un facteur de conversion d'horloge de 1000 et "1" indique 1001;
- bits 2-8 diviseur d'horloge: "0" est interdit. Représentation binaire naturelle de la valeur du diviseur d'horloge.

La fréquence de base d'image personnalisée est donnée par l'expression $1\ 800\ 000 / (\text{diviseur d'horloge} * \text{facteur de conversion d'horloge})$ Hz.

Le compteur de référence temporelle doit compter en unités de l'inverse de la fréquence de base d'image (PCF), en secondes. Quand la fréquence PCF change par rapport à celle qui était spécifiée pour l'image précédente, la référence temporelle de l'image en cours est mesurée en fonction de la

fréquence PCF précédente, de sorte que la nouvelle fréquence PCF ne s'applique que pour l'interprétation de la référence temporelle des images futures.

5.1.8 Référence temporelle étendue (ETR, *extended temporal reference*) (2 bits)

Mot de code de longueur fixe de 2 bits présent seulement si une fréquence de base d'image personnalisée est utilisée (quelle que soit la valeur du sous-champ UFEP). Il s'agit des deux bits de plus fort poids du nombre de 10 bits défini au 5.1.2.

5.1.9 Indicateur de vecteurs cinétiques non restreints illimités (UII, *unlimited unrestricted motion vectors indicator*) (longueur variable)

Mot de code de longueur variable de 1 ou 2 bits, présent seulement si le mode facultatif de codage par vecteurs cinétiques non restreints est indiqué dans le champ PLUSPTYPE et si le sous-champ UFEP est égal à 001. Quand le champ UII est présent, il indique la limite effective de la gamme des vecteurs cinétiques utilisée.

- UII = "1" la gamme des vecteurs cinétiques est limitée conformément aux Tableaux D.1 et D.2.
- UII = "01" la gamme des vecteurs cinétiques n'est pas limitée sauf par la taille d'image.

5.1.10 Bits de sous-mode du mode structuré par tranches (SSS, *slice structured submode*) (2 bits)

Mot de code de longueur fixe de 2 bits présent seulement si le mode facultatif structuré par tranches (voir l'Annexe K) est indiqué dans le champ PLUSPTYPE et si le sous-champ UFEP est égal à 001. Si le mode structuré par tranches est utilisé mais que UFEP n'est pas égal à 001, les dernières valeurs envoyées pour le champ SSS restent valables.

- bit 1 tranches rectangulaires, "0" indique des tranches de forme libre, tandis que "1" indique des tranches rectangulaires;
- bit 2 ordre arbitraire des tranches, "0" indique un ordre séquentiel, tandis que "1" indique un ordre arbitraire.

5.1.11 Numéro de la tranche d'amélioration (ELNUM, *enhancement layer number*) (4 bits)

Mot de code de longueur fixe de 4 bits présent seulement si le mode facultatif d'échelonnabilité temporelle, SNR et spatiale est utilisé (quelle que soit la valeur du sous-champ UFEP). La couche d'amélioration particulière est identifiée par un numéro de couche d'amélioration, ELNUM. La correspondance d'image entre les couches est obtenue par référence temporelle. La taille d'image est soit indiquée à l'intérieur de chaque couche d'amélioration en utilisant les champs existants de format de source, soit inférée par relation à la couche de référence. La première couche d'amélioration au-dessus de la couche de base est appelée couche d'amélioration numéro 2, alors que la couche de base porte le numéro 1.

5.1.12 Numéro de la couche de référence (RLNUM, *reference layer number*) (4 bits)

Mot de code de longueur fixe de 4 bits présent seulement si le mode facultatif d'échelonnabilité temporelle, SNR et spatiale est utilisé (voir l'Annexe O) et si le sous-champ UFEP est égal à 001. Le numéro de couche pour les images utilisées comme référence ou comme ancrage est identifié par un numéro de couche de référence (RLNUM). La correspondance temporelle entre les couches est obtenue au moyen de la référence temporelle.

Il convient de noter que pour les images B d'une couche d'amélioration pour lesquelles des images EI ou EP de l'entourage temporel sont présentes dans la même couche d'amélioration, le champ RLNUM doit être égal au champ ELNUM (voir l'Annexe O).

5.1.13 Fanions de mode de sélection d'image de référence (RPSMF, *reference picture selection mode flags*) (3 bits)

Mot de code de longueur fixe de 3 bits qui est présent seulement si le mode de sélection d'image de référence est utilisé et si le sous-champ UFEP est égal à 001. Quand il est présent, le champ RPSMF indique quels types de message par canal de retour sont nécessaires pour le codeur. Si le mode de sélection d'image de référence est utilisé mais que le champ RPSMF n'est pas présent, la dernière valeur de RPSMF qui a été envoyée doit rester en vigueur.

- 100: les signaux ACK et NACK ne sont pas nécessaires;
- 101: les signaux ACK doivent être retournés;
- 110: les signaux NACK doivent être retournés;
- 111: les signaux ACK et NACK doivent être retournés;
- 000-011: réservé.

5.1.14 Indication de référence temporelle aux fins de prédiction (TRPI, *temporal reference for prediction indication*) (1 bit)

Mot de code de longueur fixe d'un bit qui est présent seulement si le mode facultatif de sélection d'image de référence est utilisé (quelle que soit la valeur du sous-champ UFEP). Quand il est présent, le champ TRPI indique la présence du champ TRP qui suit:

- 0: champ TRP non présent;
- 1: champ TRP présent.

Le champ TRPI doit être égal à 0 chaque fois que l'en-tête d'image indique une image I ou EI.

5.1.15 Référence temporelle aux fins de prédiction (TRP, *temporal reference for prediction*) (10 bits)

Quand il est présent (ce qui est indiqué par le champ TRPI), le champ TRP indique la référence temporelle qui est utilisée pour la prédiction du codage, sauf dans le cas des images B. Pour les images B, l'image ayant la référence temporelle TRP est utilisée pour la prédiction dans le sens avant. (La prédiction dans le sens temporel inverse utilise toujours l'image immédiatement subséquente dans le temps.) Le champ TRP est un nombre de dix bits. Si une fréquence de base d'image personnalisée n'était pas utilisée pour l'image de référence, les deux bits de plus fort poids de la référence temporelle sont zéro et les bits de plus faible poids contiennent les huit bits TR trouvés dans l'en-tête d'image de l'image de référence. Si une fréquence de base d'image personnalisée était utilisée pour l'image de référence, la référence temporelle est un nombre de dix bits équivalents à la concaténation des champs ETR et TR provenant de l'en-tête d'image de référence.

Lorsque le champ TRP n'est pas présent, c'est la plus récente image de référence principale (temporellement antérieure) qui doit être utilisée pour la prédiction, comme en dehors du mode de sélection d'image de référence. Le champ TRP est valide jusqu'au prochain champ PSC, GSC ou SSC.

5.1.16 Indication de message par canal de retour (BCI, *back-channel message indication*) (longueur variable)

Champ de longueur variable d'un ou deux bits présent seulement si le mode facultatif de sélection d'image de référence est utilisé. Quand il est mis à "1", ce champ signale la présence du champ facultatif de message par canal de retour vidéo (BCM) qui suit. "01" indique l'absence ou la fin du champ de message par canal de retour vidéo. Des combinaisons de BCM et de BCI peuvent être présentes, et peuvent être répétées quand elles sont présentes. Le champ BCI doit être mis à "01" si le sous-mode multiplex vidéo du mode facultatif de sélection d'image de référence n'est pas utilisé.

5.1.17 Message par canal de retour (BCM, *back-channel message*) (longueur variable)

Message transmis par canal de retour avec syntaxe spécifiée au N.4.2, présent seulement si le champ précédent BCI est présent et mis à "1".

5.1.18 Paramètres de rééchantillonnage d'image de référence (RPRP, *reference picture resampling parameters*) (longueur variable)

Champ de longueur variable présent seulement si le bit RPR du mode facultatif de rééchantillonnage d'image de référence est mis à "1" dans le champ PLUSPTYPE. Ce champ comporte les paramètres du mode de rééchantillonnage d'image de référence (voir l'Annexe P). Il est à noter que le mode de rééchantillonnage d'image de référence peut aussi être appelé implicitement par l'occurrence d'un en-tête d'image correspondant à une image codée INTER ayant une taille d'image qui diffère de celle de l'image codée précédente, auquel cas le champ RPRP n'est pas présent et le bit du mode de rééchantillonnage d'image de référence n'est pas mis à "1".

5.1.19 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 champ GQUANT ou DQUANT ultérieur. Ce mot de code est la représentation binaire naturelle des valeurs de quantification (QUANT) qui, leur pas étant la moitié de l'échelon de quantification, vont de 1 à 31.

5.1.20 Communication multipoint à présence continue et à multiplex vidéo (CPM, *continuous presence multipoint and video multiplex*) (1 bit)

Mot de code de 1 bit qui signale l'utilisation du mode facultatif de communication multipoint à présence continue et à multiplex vidéo (CPM); "0" indique que le mode est non activé, "1" qu'il est activé. Pour l'utilisation du bit CPM, voir l'Annexe C. Le champ CPM suit immédiatement PLUSPTYPE si ce champ est présent, sinon il suit le champ PQUANT dans l'en-tête d'image.

5.1.21 Indicateur de sous-flux binaire (PSBI, *picture sub-bitstream 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 et à multiplex vidéo est indiqué par le champ 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). Le champ PSBI suit immédiatement le champ CPM, si celui-ci a la valeur "1" (l'emplacement des champs CPM et PSBI dans l'en-tête d'image dépend de la présence du champ PLUSPTYPE).

5.1.22 Référence temporelle pour les images B dans les trames PB (TR_B , *temporal reference for B-pictures in PB-frames*) (3/5 bits)

Le champ TR_B est présent si le champ PTYPE ou PLUSPTYPE indique "trame PB" ou "trame PB améliorée" (voir également les Annexes G et M), ainsi que le nombre d'images non transmises ou d'images autres que de référence (à 29,97 Hz ou à la fréquence de base d'image personnalisée indiquée dans le champ CPCFC) depuis la dernière image P ou I, ou depuis la partie P d'une trame PB ou d'une trame PB améliorée, et avant la partie image B de la trame PB ou PB améliorée. Ce mot de code est la représentation binaire naturelle du nombre d'images non transmises plus un. Il a une longueur de 3 bits pour la fréquence de base d'image CIF normalisée et est étendu à 5 bits lors de l'utilisation d'une fréquence de base d'image personnalisée. Le nombre maximal d'images non transmises est de 6 pour la fréquence de base d'image CIF normalisée et de 30 lors de l'utilisation d'une fréquence de base d'image personnalisée.

5.1.23 Information de quantification pour images B dans les trames PB (DBQUANT, *quantization information for B-pictures in PB frames*) (2 bits)

Le champ DBQUANT est présent si le champ PTYPE ou PLUSPTYPE indique "trame PB" ou "trame PB améliorée" (voir également les Annexes G et M). Dans le processus de décodage, on obtient un paramètre de quantification QUANT pour chaque macrobloc. Avec les 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 champ DBQUANT indique la relation entre QUANT et BQUANT selon la définition du Tableau 6. 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, résultant de l'utilisation du Tableau 6, est supérieure à 31, elle est limitée à 31.

Tableau 6/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	$(8 \times \text{QUANT})/4$

5.1.24 Informations supplémentaires d'insertion (PEI, *extra insertion information*) (1 bit)

Bit qui, lorsqu'il est mis à "1", signale la présence du champ de données facultatif suivant.

5.1.25 Informations complémentaires d'amélioration (PSUPP, *supplemental enhancement information*) (0/8/16... bits)

Si le champ PEI est mis à "1", il est suivi de 9 bits comprenant 8 bits de données (PSUPP) puis d'un autre champ PEI pour indiquer si un nouveau champ de 9 bits suit, et ainsi de suite. Les codeurs doivent utiliser le champ PSUPP selon les indications de l'Annexe L. Les décodeurs qui ne prennent pas en charge les capacités étendues décrites à l'Annexe L doivent être conçus de manière à ignorer le champ PSUPP si le champ PEI est mis à 1. Ainsi, la compatibilité vers l'amont sera garantie pour les capacités étendues de l'Annexe L, de sorte qu'un flux binaire qui utilise les capacités étendues puisse aussi être utilisé, sans modification, par les décodeurs qui ne prennent pas en charge ces capacités.

5.1.26 Bourrage (ESTUF, *stuffing*) (longueur variable)

Mot de code de longueur variable, se composant de moins de 8 bits zéro. Les codeurs peuvent insérer ce mot de code directement avant un mot de code EOS. Les codeurs doivent insérer ce mot de code selon les besoins afin d'obtenir l'alignement obligatoire par les octets, directement, avant un mot de code EOSBS. Si le champ ESTUF est présent, son dernier bit doit être le dernier bit (de plus faible poids) d'un octet, de manière que le début du mot de code EOS ou EOSBS soit aligné sur une limite d'octet. Les décodeurs doivent être conçus de façon à ignorer le mot ESTUF. Se reporter à l'Annexe C pour la description du champ EOSBS et de son utilisation.

5.1.27 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 non ce mot de code, qui peut être aligné sur une limite d'octets. Pour cela, on insère le mot ESTUF avant le code EOS, de manière que le premier bit du code EOS soit le premier bit (de

plus fort poids) d'un octet. Le champ EOS ne doit pas être répété, sauf si au moins un code de déclenchement d'image apparaît entre chaque paire de codes EOS.

5.1.28 Bourrage (PSTUF, *stuffing*) (longueur variable)

Mot de code de longueur variable composé de moins de 8 bits zéro. Les codeurs doivent insérer ce mot de code pour l'alignement par les octets du code PSC suivant. Le dernier bit du mot PSTUF doit être le dernier bit (de plus faible poids) 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 ignorer 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 bits (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 9. 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-têtes GOB non vides. Le mot GSTUF peut être présent lorsque le mot GBSC est présent. Les champs GN, GFID et GQUANT sont présents lorsque le mot GBSC est présent. Le mot selon GSBI est présent lorsque le champ CPM a la valeur "1" dans l'en-tête d'image.



Figure 9/H.263 – Structure de la couche des groupes de blocs

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

Mot de code de longueur variable, se composant de moins de 8 bits zéro. 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 plus faible poids) d'un octet, de manière que le début du mot GBSC soit aligné sur une limite d'octet. Les décodeurs doivent être conçus de façon à ignorer 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 sur des limites d'octet. 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 plus fort poids) d'un octet.

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

Mot de code de longueur fixe de 5 bits qui sont la représentation binaire du numéro de groupe de blocs. Pour le groupe ayant le numéro 0, l'en-tête GOB (y compris les codes GSTUF, GBSC, GN, GSBI, GFID et GQUANT) est vide, car le numéro de groupe 0 est utilisé dans le code de déclenchement d'image (PSC). Les numéros de groupe 1 à 17 sont utilisés dans les en-têtes GOB des formats d'image normalisés. Les numéros de groupe 1 à 24 sont utilisés dans les en-têtes GOB des

formats d'image personnalisés. Les numéros de groupe 16 à 28 sont émulés dans l'en-tête de tranche (voir l'Annexe K), quand CPM = "0", et les numéros de groupe 25 à 27 et 29 sont émulés dans l'en-tête de tranche (voir l'Annexe K), quand CPM = "1". Le numéro de groupe 31 est utilisé dans le code EOS, tandis que le numéro de groupe 30 est utilisé dans le code EOSBS.

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

Mot de code de longueur fixe de 2 bits, qui n'est présent que si CPM = "1" dans l'en-tête d'image. 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 de 2 bits, qui doit avoir la même valeur dans chaque en-tête de groupe GOB (ou de tranche) 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 transmise, l'identificateur GFID doit avoir la même valeur que dans cette image précédente, à condition que le champ PLUSPTYPE ne soit pas présent. Si cependant le mot PTYPE diffère, dans un certain en-tête d'image, du mot PTYPE contenu dans l'en-tête de l'image précédemment transmise, la valeur du mot GFID dans cette image doit différer de la valeur contenue dans l'image précédente.

Si le champ PLUSPTYPE est présent, la valeur du mot GFID doit être la même que pour l'image précédente (dans la même couche) si les champs PTYPE et PLUSPTYPE, ainsi que tous les champs présents parmi CPFMT, EPAR, CPCFC, SSS, ELNUM, RLNUM, UUI, RPSMF et RPRP restent les mêmes que dans l'image précédente; sinon, le mot GFID 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 de 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 la 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 10. L'indication de macrobloc codé (COD, *coded macroblock indication*) n'est présente que dans les images qui ne sont pas de type "INTRA", 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 l'image est de type "INTRA". Le code de mode de macrobloc pour blocs d'images B (MODB) est présent pour les macroblocs de type 0-4 si le champ PTYPE indique "trame PB". Les codes CBPY, DQUANT, MVD et MVD₂₋₄ 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 MVD₂₋₄ ne sont présents qu'en mode de prédiction avancé (voir l'Annexe F) ou en mode avec filtre de dégroupage (voir l'Annexe J). 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. Pour le codage de la couche des macroblocs dans les images B, EI et EP, voir l'Annexe O.

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD ₂	MVD ₃	MVD ₄	MVDB	données de bloc
-----	-------	------	------	------	--------	-----	------------------	------------------	------------------	------	-----------------

Figure 10/H.263 – Structure de la couche des macroblocs

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

Bit 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 bit COD n'est présent que dans les images qui ne sont pas du type "INTRA", pour chaque macrobloc de ces images.

NOTE – En mode de prédiction avancé (voir l'Annexe F), la compensation de mouvement par superposition de blocs est également effectuée si COD est mis à "1"; de même, dans le mode avec filtre de dégroupage (voir l'Annexe J), le filtre de dégroupage peut également avoir une incidence sur la valeur de certains pixels des macroblocs dont le champ 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)

Le mot de code MCBPC, de longueur variable, renseigne sur le type de macrobloc et sur la structure de bloc codée pour la chrominance. Les codes correspondant au champ MCBPC sont donnés aux Tableaux 7 et 8. Le champ MCBPC est toujours inclus dans les macroblocs codés.

Tableau 7/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

Tableau 8/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

Tableau 8/H.263 – Table des codes de longueur variable (VLC) pour le mot MCBPC (images de type P) (fin)

Index	Type de macrobloc	CBPC (56)	Nombre de bits	Code
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
21	5	00	11	0000 0000 010
22	5	01	13	0000 0000 0110 0
23	5	10	13	0000 0000 0111 0
24	5	11	13	0000 0000 0111 1

Un mot de code additionnel est donné dans les tableaux aux fins de bourrage. Ce mot de code devrait être ignoré par les décodeurs. Si une trame PB améliorée est indiquée par les bits 1-3 du champ MPPTYPE et si un format de source personnalisé est indiqué dans les bits 1-3 du champ OPPTYPE, le champ MBA ne doit pas indiquer de bourrage avant le premier macrobloc de l'image (afin d'empêcher l'émulation de code de déclenchement).

Le type de macrobloc renseigne sur le macrobloc et sur les éléments de données qui sont présents. Les types de macrobloc et les éléments inclus sont indiqués dans les Tableaux 9 et 10. Le macrobloc de type 5 (index 21-24 du Tableau 8) ne doit pas être présent, sauf si un champ PTYPE étendu (PLUSPTYPE) est présent dans l'en-tête d'image et si soit le mode de prédiction avancé (voir l'Annexe F), soit le mode avec filtre de dégroupage (voir l'Annexe J) est utilisé, et il ne doit pas être présent pour le premier macrobloc d'une image. De plus, les codeurs ne doivent pas permettre à un code MCBPC correspondant à un macrobloc du type 5 de suivre immédiatement sept zéros consécutifs dans le flux binaire (ce qui pourrait se produire à cause de certains codes INTRADC donnés, suivis par COD = 0), afin d'empêcher l'émulation du code de déclenchement. Au besoin, les codes des macroblochs du type 5 peuvent être précédés de bits de bourrage afin de répondre à cette exigence (pour les macroblochs autres que le premier d'une image).

Tableau 9/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 ₂₋₄
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	5	INTER4V+Q	X	X	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 croix "X" indique que l'élément est présent dans le macrobloc.

Tableau 10/H.263 – Types de macroblocs et éléments de données inclus pour trames PB

Type d'image	Type de macrobloc	Nom	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD ₂₋₄
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	INTER4V	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	5	INTER4V+Q	X	X	X	X	(X)	X	X	(X)	X
INTER	bourrage	-	X	X							

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

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 de transformée non INTRADC (INTRADC étant le coefficient dc des blocs INTRA, voir 5.4.1) est transmis, sauf si le mode facultatif de codage INTRA avancé est utilisé. Si un coefficient non INTRADC quelconque est présent pour le bloc N avec $N = 5$ et 6 , on a $CBPC_N = 1$; sinon, $CBPC_N = 0$, dans la structure de bloc codée. Si le mode de codage INTRA avancé est utilisé, l'utilisation est la même, mais le coefficient INTRADC est indiqué de la même manière que les autres coefficients (voir l'Annexe I). 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 7 et 8.

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

Le champ MODB est présent pour les macroblocs de type 0-4 si le champ 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 11 définit les mots de code pour le champ MODB. Ce champ MODB est codé différemment pour les trames PB améliorées, comme l'indique l'Annexe M.

Tableau 11/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 croix "X" indique que l'élément est présent dans le macrobloc.

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 $CBPB_N$ dans la structure de bloc codée. La numérotation des blocs est indiquée dans la Figure 5, le bit le plus à gauche du mot CBPB correspondant au bloc numéro 1.

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 (Y) contenus dans le macrobloc pour lesquels au moins un coefficient de transformée non INTRADC (INTRADC étant le coefficient dc de transformée de blocs INTRA, voir 5.4.1) est transmis, sauf si le mode de codage INTRA avancé est utilisé. Si le mode de codage INTRA avancé est utilisé, le coefficient INTRADC est indiqué de la même manière que les autres coefficients (voir l'Annexe I).

$CBPY_N = 1$ si un coefficient non INTRADC quelconque est présent pour le bloc N. Sinon $CBPY_N = 0$, pour chaque bit $CBPY_N$ 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 correspondant au bloc numéro 1. Pour une certaine structure $CBPY_N$, différents mots de code sont utilisés pour les macroblocs INTER et INTRA, comme indiqué dans le Tableau 13.

5.3.6 Informations sur le quantificateur (DQUANT, *quantizer information*) (2 bits/longueur variable)

Si le mode de quantification modifié n'est pas utilisé, le mot de code DQUANT à 2 bits définit un changement de l'information QUANT. Le Tableau 12 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 limitée à 1 ou à 31 selon le cas. Si le mode de quantification modifié est utilisé, le mot de code DQUANT est de longueur variable, tel que défini dans l'Annexe T.

Tableau 12/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 13/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 champ MVD est inclus pour tous les macroblocs INTER (ainsi que pour les macroblocs INTRA en mode "trames PB"). Il se compose d'un mot de code de longueur variable pour la composante horizontale, suivi d'un mot de code de longueur variable pour la composante verticale. Les codes de longueur variable sont indiqués dans le Tableau 14. Si le mode de codage par vecteurs cinétiques non restreints est utilisé et si le champ PLUSPTYPE est présent, les vecteurs cinétiques sont codés en utilisant le Tableau D.3 au lieu du Tableau 14 (voir l'Annexe D).

Tableau 14/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

Tableau 14/H.263 – Table des codes de longueur variable pour MVD (*fin*)

Index	Différences vectorielles		Nombre de bits	Codes
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

5.3.8 Données de vecteur cinétique (MVD₂₋₄, *motion vector data*) (longueur variable)

Les trois mots de code MVD₂₋₄ sont inclus si cela est indiqué par les champs PTYPE et MCBPC. Ils consistent chacun en un mot de code de longueur variable pour la composante horizontale, suivi d'un mot de code de longueur variable pour la composante verticale de chaque vecteur. Les codes de longueur variable sont indiqués dans le Tableau 14. Les mots MVD₂₋₄ ne sont présents que dans le mode de prédiction avancé (voir l'Annexe F) ou dans le mode avec filtre de dégroupage (voir l'Annexe J).

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 ou de trames PB amélioré si cela est indiqué par le champ MODB. Il consiste en un mot de code de longueur variable pour la composante horizontale, suivi d'un mot de code de longueur variable pour la composante verticale de chaque vecteur. Les codes de longueur variable sont indiquées dans le Tableau 14. Pour l'utilisation des mots MVDB, voir les Annexes G et M.

5.4 Couche des blocs

En dehors du mode de trames PB, un macrobloc se compose de quatre blocs de luminance et d'un bloc de chacun des deux types de blocs de différence de couleur (voir la Figure 5). La structure de la

couche des blocs est représentée à la Figure 11. Le coefficient INTRADC est présent pour chaque bloc du macrobloc si le champ MCBPC indique que le macrobloc est du type 3 ou 4 (voir les Tableaux 7 et 8). Le coefficient 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 coefficient INTRADC est présent pour chaque bloc P du macrobloc si le champ MCBPC indique que le macrobloc est du type 3 ou 4 (voir les Tableaux 7 et 8). Le coefficient INTRADC n'est pas présent pour les blocs B. Le mot TCOEF est présent pour les blocs P s'il a été indiqué par MCBPC ou CBPY. 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.



Figure 11/H.263 – Structure de la couche des blocs

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

Mot de code de 8 bits. Le code 0000 0000 n'est pas utilisé. Le code 1000 0000 n'est pas utilisé, le niveau de reconstruction de 1024 étant codé comme étant 1111 1111 (voir le Tableau 15).

Tableau 15/H.263 – Niveaux de reconstruction pour coefficient DC en mode INTRA

Index	Code de longueur fixe	Niveau de reconstruction avant transformation 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

5.4.2 Coefficient de transformée (TCOEF, transform coefficient) (longueur variable)

Les événements (EVENT) les plus fréquents sont codés avec les codes de longueur variable indiquées dans le Tableau 16. Le dernier bit "s" indique le signe du niveau: "0" pour positif et "1" pour négatif.

Un événement (EVENT) est une combinaison d'une dernière indication de coefficient non égal à zéro (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 16 n'est pas interdite. Pour l'octet LEVEL, le code 0000 0000 est interdit, et le code 1000 0000 est interdit à moins que le mode de quantification modifié soit utilisé (voir l'Annexe T). Le Tableau 17 indique les codes pour RUN et LEVEL.

Tableau 16/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

Tableau 16/H.263 – Table des codes de longueur variable pour le mot TCOEF (suite)

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
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
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

Tableau 16/H.263 – Table des codes de longueur variable pour le mot TCOEF (*fin*)

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
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	ECHAPPEMENT			7	0000 011

Tableau 17/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	voir le texte
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

6 Processus de décodage

6.1 Compensation de mouvement

Le présent sous-paragraphe 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 de codage par vecteurs cinétiques non restreints. On trouvera

à l'Annexe F une description de la compensation de mouvement en mode de prédiction avancé. Pour une description de la compensation de mouvement dans le mode de rafraîchissement à résolution réduite, se reporter à l'Annexe Q.

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 les Tableaux 14 et D.3). 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 12. Les prédicteurs sont calculés séparément pour les composantes horizontale et verticale.

Dans les cas particuliers des bords du groupe de blocs, de la tranche 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 en mode de trames PB avec prédiction bidirectionnelle) ou n'a pas été codé (COD = 1);
- 2) le prédicteur candidat MV1 est mis à zéro si le macrobloc correspondant est à l'extérieur de l'image ou de la tranche (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 (en 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; ou en dehors de la tranche, en mode structuré par tranches;
- 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.

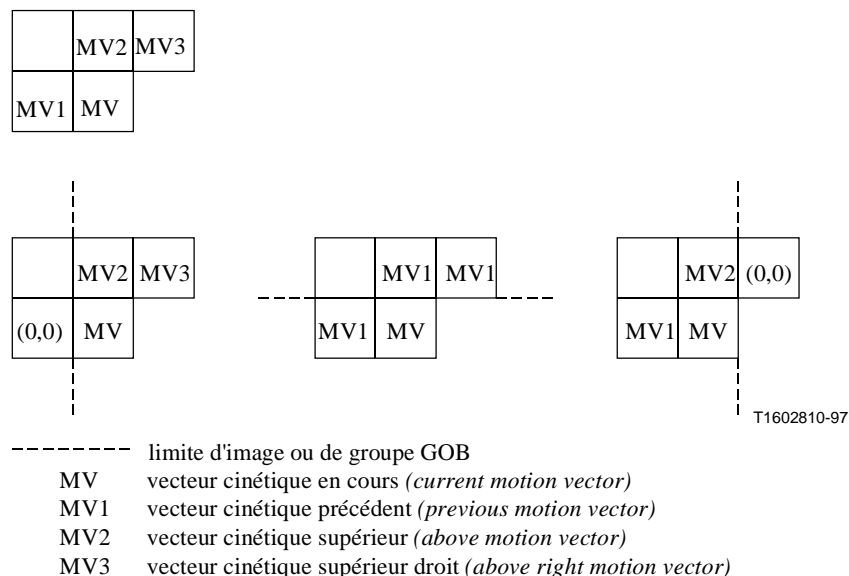


Figure 12/H.263 – Prédiction des vecteurs cinétiques

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 la gamme permise $[-16, 15,5]$. 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. Si le mode de codage par vecteurs cinétiques non restreints est utilisé (voir l'Annexe D), le décodage des vecteurs cinétiques doit se faire selon les indications de D.2.

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 du demi-pixel, comme indiqué dans le Tableau 18.

Tableau 18/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

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 13. Une barre oblique "/" indique une division par troncature.

La valeur du champ RCONTROL est égale à la valeur du bit du type d'arrondissement (RTYPE) (bit 6) du champ MPPTYPE (voir 5.1.4.3), quand le champ de format de source (bits 6-8) du champ PTYPE indique "PTYPE étendu". Sinon, le champ RCONTROL a une valeur implicite de 0. Quelle que soit la valeur du bit RTYPE, la valeur du champ RCONTROL est égale à 0 pour la partie B des trames PB améliorées (voir l'Annexe M).

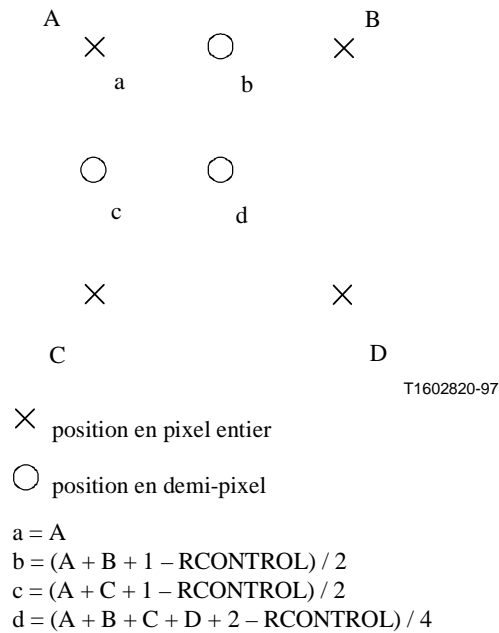


Figure 13/H.263 – Prédiction au demi-pixel près par interpolation bilinéaire

6.2 Décodage des coefficients

6.2.1 Quantification inverse

Le processus de quantification inverse est décrit dans le présent sous-paragraphe, sauf pour le cas où le mode facultatif de codage INTRA avancé est utilisé (voir l'Annexe I). Si LEVEL = "0", le niveau de reconstruction REC = "0". Le niveau de reconstruction du coefficient INTRADC est donné par le Tableau 15. Les niveaux de reconstruction de tous les coefficients non nuls autres que le coefficient INTRADC sont donnés par les formules suivantes:

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

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

On notera que ce processus élimine les nombres de valeur paire. On a constaté que cela empêchait 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: $REC = \text{sign}(LEVEL) \cdot |REC|$

Le terme $\text{sign}(LEVEL)$ est donné par le dernier bit du code de coefficient TCOEF (voir le Tableau 16) ou par le Tableau 17.

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 à la gamme de -2048 à 2047.

6.2.3 Positionnement en zigzag

Les coefficients quantifiés de la transformée sont rangés en blocs de 8×8 éléments, conformément à la séquence indiquée dans la Figure 14, sauf si le mode facultatif de codage INTRA avancé est utilisé (voir l'Annexe I). 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 14/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 8×8 éléments résultants une transformation inverse en cosinus discret à deux dimensions séparables, à 8×8 coefficients. Le résultat 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:

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

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

où:

x, y = coordonnées spatiales dans le domaine des pixels;

u, v = 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. Pour la sommation en mode de rafraîchissement à résolution réduite, se reporter à l'Annexe Q.

6.3.2 Ecrêtage

Pour empêcher 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 les 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

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 8×8 . 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 8×8 , 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:

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

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

où:

x, y = coordonnées spatiales dans le domaine des pixels;

u, v = 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 la gamme $[-2048$ à $+2047]$. On obtient ainsi des données d'entrée à 12 bits pour la transformée inverse.

A.4 Pour chaque bloc de 8×8 données à 12 bits produit en A.3, effectuer une transformée discrète inverse (IDCT) 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 la gamme $[-256$ à $+255]$. Ces blocs de 8×8 éléments constituent les données de sortie de référence de la transformée en cosinus discrète inverse.

A.5 Pour chaque bloc de 8×8 données produit en A.3, utiliser le dispositif à l'essai de transformation en cosinus discrète inverse, en tronquant la sortie à la gamme $[-256$ à $+255]$. Ces blocs de 8×8 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

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

B.1 Fonctionnant en synchronisme, le décodeur HRD et le codeur doivent avoir la même fréquence d'horloge ainsi que la même fréquence de base d'image.

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 de bits par image qui a été négocié pour utilisation dans le flux binaire (voir 3.6). La valeur de B est définie comme suit:

$$B = 4 \cdot R_{\max} / PCF$$

où PCF est la fréquence de base d'image effective et où R_{\max} est le débit vidéo maximal pendant la connexion, en bits par seconde. La fréquence de base d'image effective est la fréquence de base d'image CIF normalisée, à moins qu'une fréquence de base d'image personnalisée ne soit spécifiée dans le champ CPCFC de l'en-tête d'image. Cette valeur de B constitue un minimum. 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 RTGC ou RNIS, liaison simple ou multiple). Elle peut être égale au débit maximal qui est supporté par la liaison physique. La négociation de R_{\max} est effectuée par des moyens externes (comme par exemple, la Recommandation H.245).

B.3 Le décodeur HRD est initialement vide.

B.4 La mémoire tampon du décodeur HRD est examinée aux intervalles de l'horloge d'image $(1000 / PCF \text{ ms})$. Si au moins une image codée complète est dans le tampon, toutes les données de l'image la plus ancienne dans l'ordre du flux binaire 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, et sauf si le mode facultatif d'échelonnabilité temporelle, SNR et spatiale est utilisé, une image complète codée est une image I ou P normale ou une trame PB ou PB améliorée.

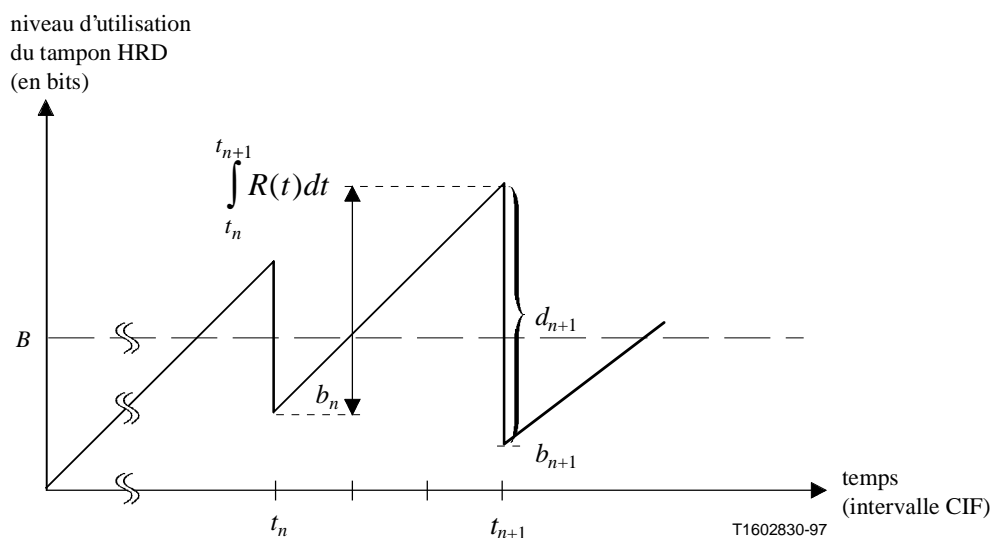
Quand le mode facultatif d'échelonnabilité temporelle, SNR et spatiale est utilisé (voir l'Annexe O), on donne à chaque couche d'amélioration un décodeur HRD supplémentaire, pour lequel une image codée complète est une image EI, EP ou B. Le tampon de la couche de base doit retenir les bits de l'en-tête d'image, à mesure qu'ils arrivent, jusqu'à ce qu'une partie suffisante de l'en-tête d'image soit arrivée pour déterminer si l'image est une image de couche de base ou de couche d'amélioration, et pour déterminer le numéro de la couche d'amélioration. Une fois que l'on peut déterminer que l'image qui arrive appartient à une couche d'amélioration, tous les bits de cette image doivent être instantanément transférés au décodeur de la couche d'amélioration appropriée, et tout bit qui arrive plus tard continue à être mis dans le décodeur de la couche d'amélioration jusqu'à ce qu'une partie suffisante d'un nouvel en-tête d'image soit arrivée pour déterminer si le flux binaire devrait à nouveau être réacheminé dans un autre tampon de décodeur HRD. Le processus d'identification de la couche d'amélioration est instantané et asynchrone, et est donc indépendant des instants de vérification de l'intervalle de l'horloge d'image.

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

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 secondes. La transmission de ce signal emprunte des moyens externes (par exemple, selon la Recommandation H.245). Il convient de noter qu'une commande similaire peut aussi être envoyée au moyen d'information complémentaire d'amélioration dans l'en-tête d'image du flux binaire (voir L.4).

C.2 Demande de rafraîchissement rapide

Cette demande fait que le décodeur code sa prochaine image en mode INTRA avec des paramètres de codage permettant d'empêcher tout débordement de mémoire tampon. La transmission de ce signal emprunte des moyens externes (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 et à multiplex vidéo (CPM, *continuous presence multipoint and video multiplexing*)

NOTE – Mode non utilisé pour la Recommandation H.324.

La présente Recommandation prévoit la possibilité d'un mode négociable de communication multipoint à présence continue et à multiplex vidéo dans lequel jusqu'à quatre flux binaires H.263 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, GSBI, SSBI et ESBI. L'échange de possibilités des terminaux est effectué par des moyens externes (par exemple selon la Recommandation H.242). Les données par canal de retour de sélection d'image de référence pour répondre aux sous-flux binaires indépendants sont prises en charge par l'utilisation des champs BCPM et BSBI.

En mode CPM, le champ CPM doit être mis à "1" dans chacun des sous-flux binaires H.263 indépendants. Les sous-flux binaires sont identifiées par un numéro d'identification de flux en utilisant les indicateurs de sous-flux binaire (SBI, *sub-bitstream indicator*) contenus dans les en-têtes d'image et de groupe GOB ou dans les en-têtes de tranche et EOSBS de chaque flux binaire H.263. Le champ SBI indique le numéro de flux binaire H.263 auquel appartient cet en-tête et toutes les informations suivantes jusqu'au prochain en-tête d'image ou de groupe GOB, ou de tranche, contenu 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 complètement 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. De manière semblable, les règles d'inférence d'état de mode appliquées lorsqu'on utilise un champ de type d'image étendu (PLUSPTYPE) dans l'en-tête d'image et tous les autres aspects de l'opération de flux binaire vidéo doivent fonctionner de manière indépendante et distincte pour chaque sous-flux binaire.

Par convention, le sous-flux binaire ayant le plus petit numéro d'identification de sous-flux binaire (envoyé dans le champ SBI) est considéré comme ayant la priorité la plus élevée dans les cas où un conflit en matière de ressources exigées peut nécessiter un choix par priorité (sauf si une convention différente relativement à la priorité est établie par des moyens externes).

Afin de marquer la fin de chaque sous-flux binaire du mode CPM, une syntaxe est donnée, comme l'illustre la Figure C.1, à condition que la capacité d'envoyer cette syntaxe additionnelle soit d'abord négociée par des moyens externes (alors que le fonctionnement en mode CPM était défini dans la Version 1 de la présente Recommandation, la syntaxe de fin du sous-flux binaire a été ajoutée dans la Version 2 et n'est donc pas considérée comme faisant partie du fonctionnement en mode CPM de la Version 1). La syntaxe de fin de sous-flux binaire (ESTUF + EOSBS + ESBI) marque la fin de chaque sous-flux binaire, plutôt que la fin du flux entier comme le fait un code EOS.

NOTE – Aucune capacité de négociation du fonctionnement des sous-flux binaires du mode CPM tels que définis ici pour la Recommandation H.263 n'a été adoptée dans une Recommandation UIT-T quelconque relative aux terminaux (comme la Recommandation H.324) avant la création de la Version 2 de la présente Recommandation. Par conséquent, toute négociation par des moyens externes du fonctionnement en mode CPM qui sera adoptée dans une future Recommandation de la série H applicable aux terminaux devra comprendre la prise en charge de la syntaxe de fin de sous-flux binaire, sauf indication contraire dans la Recommandation de la série H portant sur les terminaux.

La syntaxe de fin de sous-flux binaire comprend trois parties. Après l'alignement obligatoire sur limite d'octet à l'aide du champ ESTUF, un mot de code EOSBS de 23 bits est envoyé (lequel correspond à un en-tête de groupe de blocs (GOB) pour lequel le champ GN = 30, qui sinon n'est pas utilisé dans la syntaxe, suivi d'un seul bit de valeur zéro, réservé pour usage futur). Le mot de code EOSBS est suivi d'un mot de code ESBI de deux bits indiquant le sous-flux binaire touché. Cette paire de mots de code signifie que l'envoi de données pour le sous-flux associé est terminé et que toute donnée subséquente envoyée relativement au même sous-flux binaire doit être complètement indépendante de celles qui précédaient le champ EOSBS. En particulier, la prochaine image correspondant au sous-flux binaire après le code EOSBS ne doit pas être une image INTER ou tout autre type d'image qui peut utiliser la prédiction temporelle anticipée (une image I ou EI est permise, mais une image P, une trame PB, une trame PB améliorée, une image B ou une image EP ne l'est pas).

La syntaxe des champs EOSBS et ESBI est décrite dans les sous-paragraphes qui suivent. Le champ ESTUF est décrit au 5.1.26.



Figure C.1/H.263 – Diagramme syntaxique pour les indicateurs de fin de sous-flux binaire

C.4.1 Code de fin de sous-flux binaire (EOSBS, *end of sub-bitstream*) (23 bits)

Le code EOSBS est un mot de code de 23 bits. Sa valeur est de 0000 0000 0000 0000 1 11110 0. Il revient au codeur de décider d'insérer ou non ce mot de code. Le code EOSBS ne devrait pas être envoyé, sauf si au moins un en-tête d'image a déjà été envoyé pour le même sous-flux binaire indiqué dans le champ ESBI suivant, et il ne doit pas être envoyé, à moins que la capacité d'envoyer le code EOSBS ait été négociée par des moyens externes. Les codes EOSBS doivent être alignés sur des limites d'octet. Cet alignement est réalisé en insérant le champ ESTUF avant le code de déclenchement EOSBS, de sorte que le premier bit du code de déclenchement EOSBS soit le premier bit (bit de plus fort poids) d'un octet (voir 5.1.26).

Le code EOSBS indique que l'envoi de données pour le sous-flux binaire indiqué est terminé et que le sous-flux binaire a été déclaré terminé, jusqu'à ce qu'il soit redéclenché par l'envoi d'un autre code de déclenchement d'image correspondant à ce sous-flux binaire. Les images subséquentes ayant le même numéro d'identification de sous-flux binaire (ESBI) doivent être complètement indépendantes des images envoyées avant le code EOSBS et ne doivent dépendre d'elles en aucune façon.

Les informations de commande et autres informations associées au flux binaire vidéo en général, sans spécification des sous-flux binaires auxquels ces codes sont supposés s'appliquer (comme une demande de gel d'image ou une demande de rafraîchissement rapide envoyée en H.242) devraient être présumées s'appliquer à tous les sous-flux binaires *actifs* seulement. Un sous-flux binaire est considéré comme actif si au moins un code de déclenchement d'image a été reçu pour le sous-flux binaire et si les dernières données reçues qui étaient applicables à ce sous-flux binaire n'étaient pas celles d'un champ EOS et de champs EOSBS + ESBI.

C.4.2 Indicateur de fin de sous-flux binaire (ESBI, *ending sub-bitstream indicator*) (2 bits)

Le champ ESBI est un mot de code de longueur fixe de deux bits qui suit immédiatement le code EOSBS. Il indique le numéro du sous-flux binaire qui se termine. Sa valeur est la représentation binaire naturelle sur deux bits du numéro de sous-flux binaire.

Mode de codage par vecteurs cinétiques non restreints

La présente annexe décrit le mode de codage de la présente Recommandation 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 champ PTYPE ou PLUSPTYPE.

La gamme des vecteurs cinétiques et la table de codage VLC utilisée pour le codage des différences de vecteurs cinétiques pour le mode de codage par vecteurs cinétiques non restreints dépend de la présence ou non du champ PLUSPTYPE dans l'en-tête d'image. Quand le champ PLUSPTYPE est présent, la gamme des vecteurs cinétiques dépend aussi de la taille d'image et de la valeur du champ UII dans l'en-tête d'image.

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 de codage par vecteurs cinétiques non restreints 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:

$$\begin{aligned} x, y, x', y' &= \text{coordonnées spatiales dans le domaine des pixels;} \\ \text{Rumv}(x, y) &= \text{valeur de pixel de l'image référencée au point } (x, y) \text{ en mode de vecteurs} \\ &\quad \text{cinétiques non restreints;} \\ \text{R}(x', y') &= \text{valeur de pixel de l'image référencée au point } (x', y') \text{ en mode de vecteurs} \\ &\quad \text{cinétiques non restreints;} \\ x' &\begin{cases} = 0 & \text{si } x < 0; \\ = 175 & \text{si } x > 175; \\ = x & \text{dans les autres cas;} \end{cases} \\ y' &\begin{cases} = 0 & \text{si } y < 0; \\ = 143 & \text{si } y > 143; \\ = x & \text{dans les autres cas;} \end{cases} \end{aligned}$$

et la surface d'image codée par la valeur $\text{R}(x', y')$ 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.1.1 Restrictions appliquées aux valeurs des vecteurs cinétiques

Si le champ PLUSPTYPE est présent dans l'en-tête d'image, les valeurs des vecteurs cinétiques sont limitées, de sorte qu'aucun élément de la région 16×16 ou (8×8) choisie n'a une distance horizontale ou verticale de plus de 15 pixels en dehors de la zone d'image codée. Il convient de noter qu'il s'agit là d'une gamme d'extrapolation plus petite que lorsque le champ PLUSPTYPE n'est pas présent.

NOTE 1 – Quand le champ PLUSPTYPE est absent, la gamme d'extrapolation est d'un maximum de 31,5 pixels en dehors de la zone d'image codée quand le mode de codage par vecteurs cinétiques non restreints est utilisé, et de 16 pixels en dehors de la zone d'image codée quand le mode de prédiction avancé (voir l'Annexe F) est utilisé, sans le mode de codage par vecteurs cinétiques non restreints.

NOTE 2 – Lorsque le mode de prédiction avancé est utilisé (voir l'Annexe F), le vecteur cinétique pour chaque région 16×16 ou (8×8) affecte une zone plus étendue en raison de la compensation de mouvement par superposition de blocs. Cela peut provoquer une augmentation de l'étendue d'extrapolation effective pour les vecteurs cinétiques "distants" du mode de prédiction avancé car le nombre de pixels superposés (4 pixels ou 8 pixels si le mode de rafraîchissement à résolution réduite est également utilisé) s'ajoute au nombre de pixels requis pour l'extrapolation (bien que l'étendue des valeurs autorisées pour chaque vecteur cinétique reste la même que lorsque le mode de prédiction avancé n'est pas utilisé).

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 la gamme $[-16, 15,5]$ (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, la gamme maximale des composantes vectorielles est cependant étendue.

Si le champ PLUSPTYPE n'est pas présent dans l'en-tête d'image, la gamme des vecteurs cinétiques est étendue à $[-31,5, 31,5]$, avec la restriction que seules les valeurs s'inscrivant dans une gamme de $[-16, 15,5]$ autour du prédicteur pour chaque composante vectorielle cinétique peuvent être atteintes si ce prédicteur est dans la gamme $[-15,5, 16]$. Si le prédicteur est à l'extérieur de la gamme $[-15,5, 16]$, on pourra atteindre toutes les valeurs comprises dans la gamme $[-31,5, 31,5]$ 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:

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

Dans le mode de vecteurs cinétiques non restreints, l'interprétation du Tableau 14 est la suivante pour les mots MVD, MVD_{2-4} et MVDB:

- si le prédicteur de la composante vectorielle cinétique est dans la gamme $[-15,5, 16]$, seule la première colonne des différences vectorielles est applicable;
- si le prédicteur de la composante vectorielle cinétique est hors de la gamme $[-15,5, 16]$, on doit extraire du Tableau 14 la différence vectorielle qui produit une composante vectorielle comprise dans la gamme $[-31,5, 31,5]$ avec le même signe que son prédicteur (y compris zéro).

Le prédicteur pour les données MVD et MVD_{2-4} est défini comme étant la médiane de la gamme 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).

Si le champ PLUSPTYPE est présent, la gamme de vecteurs cinétiques ne dépend pas de la valeur de prédiction de vecteur cinétique. Si le champ UUI est mis à "1", la gamme des vecteurs cinétiques dépend du format d'image. Pour les formats d'image normalisés jusqu'au format CIF, la gamme est [-32, 31,5], tandis que pour les formats jusqu'à 4CIF, elle est de [-64, 63,5] et pour les formats jusqu'à 16CIF, elle est de [-128, 127,5]; pour les formats d'image personnalisés plus grands, la gamme est de [-256, 255,5]. Les gammes de vecteurs cinétiques horizontaux et verticaux peuvent être différentes pour les formats d'image personnalisés. Les gammes de vecteurs horizontaux et verticaux sont spécifiées aux Tableaux D.1 et D.2.

Tableau D.1/H.263 – Gamme des vecteurs cinétiques horizontaux quand le champ PLUSPTYPE est présent et que UUI = 1

Largeur d'image	Gamme des vecteurs cinétiques horizontaux
4, ..., 352	[-32, 31,5]
356, ..., 704	[-64, 63,5]
708, ..., 1408	[-128, 127,5]
1412, ..., 2048	[-256, 255,5]

Tableau D.2/H.263 – Gamme des vecteurs cinétiques verticaux quand le champ PLUSPTYPE est présent et que UUI = 1

Hauteur d'image	Gamme des vecteurs cinétiques verticaux
4, ..., 288	[-32, 31,5]
292, ..., 576	[-64, 63,5]
580, ..., 1152	[-128, 127,5]

Dans le mode de rafraîchissement à résolution réduite, la gamme spécifiée s'applique aux pseudo-vecteurs cinétiques. Cela suppose que la gamme résultante des vecteurs cinétiques réels double à peu près de taille (voir également l'Annexe Q).

Si le champ UUI est mis à "01", les vecteurs cinétiques ne sont pas limités, sauf par leur distance à la limite de la zone codée, comme l'explique D.1.1. La même limitation s'applique aux vecteurs cinétiques réels (et non pas seulement aux pseudo-vecteurs cinétiques) en mode de rafraîchissement à résolution réduite.

Pour coder les vecteurs cinétiques quand le champ PLUSPTYPE est présent, le Tableau D.3 est utilisé pour coder la différence entre le vecteur cinétique et la prédiction de vecteur cinétique. Chaque entrée du Tableau D.3 comporte une seule valeur (par opposition au Tableau 14). La gamme des vecteurs cinétiques et l'utilisation du Tableau D.3 pour coder les données de vecteur cinétique s'appliquent à tous les types d'image quand le champ PLUSPTYPE est présent.

Les différences de vecteurs cinétiques sont toujours codées comme une paire de composantes horizontale et verticale. Si les deux composantes d'une paire sont égales (0,5, 0,5), six zéros consécutifs sont produits. Pour empêcher l'émulation du code de déclenchement, cette occurrence doit être suivie d'un bit mis à "1". Cela correspond à l'envoi d'une composante vectorielle cinétique zéro supplémentaire.

Tableau D.3/H.263 – Tableau des vecteurs cinétiques utilisé quand le champ PLUSPTYPE est présent

Valeur absolue de la différence vectorielle en unités d'un demi-pixel	Nombre de bits	Codes
0	1	1
1	3	0s0
"x ₀ " + 2 (2:3)	5	0x ₀ 1s0
"x ₁ x ₀ " + 4 (4:7)	7	0x ₁ 1x ₀ 1s0
"x ₂ x ₁ x ₀ " + 8 (8:15)	9	0x ₂ 1x ₁ 1x ₀ 1s0
"x ₃ x ₂ x ₁ x ₀ " + 16 (16:31)	11	0x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₄ x ₃ x ₂ x ₁ x ₀ " + 32 (32:63)	13	0x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 64 (64:127)	15	0x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 128 (128:255)	17	0x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 256 (256:511)	19	0x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 512 (512:1023)	21	0x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₉ x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 1024 (1024:2047)	23	0x ₉ 1x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0
"x ₁₀ x ₉ x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 2048 (2048:4095)	25	0x ₁₀ 1x ₉ 1x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 1s0

Le Tableau D.3 constitue une table réversible de construction régulière. Chaque rangée représente un intervalle de différence de vecteurs cinétiques en unités d'un demi-pixel. Les bits "...x₁x₀" indiquent tous les bits suivant le "1" en tête dans la représentation binaire de la valeur absolue de la différence de vecteurs cinétiques. Le bit "s" indique le signe de la différence de vecteurs cinétiques, "0" indiquant un signe positif et "1" un signe négatif. La représentation binaire de la différence de vecteurs cinétiques est entrelacée de bits qui indiquent si le code continue ou se termine. Par exemple, la différence de vecteurs cinétiques -13 a le signe s = 1 et la représentation binaire 1x₂x₁x₀ = 1101. Elle est codée comme 0 x₂1 x₁1 x₀1 s0 = 0 11 01 11 10. Le 0 en deuxième position du dernier groupe de deux bits indique la fin du code.

ANNEXE E

Mode de codage arithmétique syntaxique (SAC)

E.1 Introduction

En codage/décodage à longueur variable (VLC/VLD, *variable length coding/decoding*) conformément au paragraphe 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 mappé 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 final.

La présente annexe décrit le mode facultatif de codage arithmétique syntaxique (SAC, *syntax-based arithmetic coding*) 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 H.263 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) de la présente Recommandation, 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.

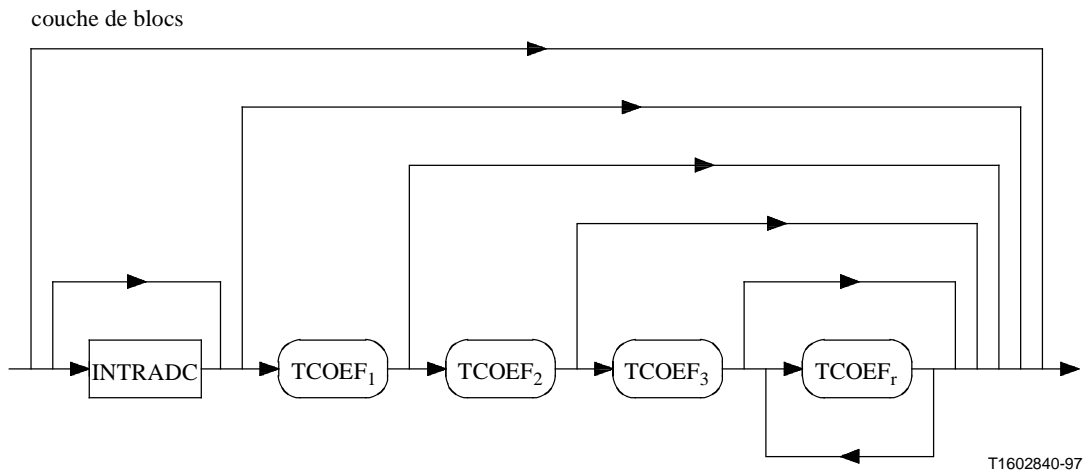


Figure E.1/H.263 – Structure de la couche des blocs en codage SAC

Dans la Figure E.1, les séquences TCOEF1, TCOEF2, TCOEF3 et TCOEFr 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 TCOEF1, TCOEF2, TCOEF3 et TCOEFr 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 la couche des en-têtes

Les couches des en-têtes de la syntaxe sont considérées comme étant les éléments syntaxiques se trouvant au-dessus des couches des blocs et des macroblocs (voir la Figure 6 et la spécification de syntaxe dans le texte). Les niveaux d'en-tête de la syntaxe de base de la Version 1 peuvent former trois chaînes possibles: (PSTUF)--PSC--TR--PTYPE--PQUANT--CPM--(PSBI)--(TRB-DBQUANT)--PEI--(PSUPP--PEI--...), (GSTUF)--GBSC--GN--(GSBI)--GFID--GQUANT et (ESTUF)--(EOS)--(PSTUF). Dans la syntaxe révisée de la Version 2, les niveaux d'en-tête de la syntaxe peuvent avoir d'autres structures (voir la Figure 6 et la spécification de syntaxe dans le texte). Les chaînes syntaxiques du niveau d'en-tête 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, GBSC, SSC, EOS ou EOSBS autorisé.

Si un en-tête n'est pas le premier dans une session vidéo, le codeur arithmétique a besoin d'être réinitialisé avant d'envoyer l'en-tête, en appelant la procédure suivante. Cette procédure doit aussi être appelée à la fin d'une session vidéo si le champ (ESTUF)--EOS [ou (ESTUF)--EOSBS pour le sous-flux binaire pour lequel le dernier en-tête a été envoyé] n'est pas envoyé.

```
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 des couches des macroblocs et des blocs

Les modèles pour les symboles des couches des macroblocs et des blocs sont décrits dans E.8. Les index indiqués dans les tables de codage de longueur variable (VLC) du paragraphe 5 sont utilisés pour indexer les entiers contenus dans les modèles.

Le modèle du mot COD contenu dans des images P est nommé `cumf_COD`. L'index pour le mot COD = 0 est 0 et l'index pour COD = 1 est 1. Le modèle du mot MCBPC contenu dans des images P est nommé `cumf_MCBPC_no4MVQ`, sauf si le champ PLUSPTYPE est présent dans l'en-tête d'image et si soit le mode de prédiction avancé (Annexe F), soit le mode avec filtre de dégroupage (Annexe J) est utilisé, auquel cas le modèle est appelé `cumf_MCBPC_4MVQ`. Les index pour le mot MCBPC sont définis dans le Tableau 7 pour les images I et dans le Tableau 8 pour les images P. Le modèle pour le mot MCBPC contenu dans les images I est nommé `cumf_MCBPC_intra`.

Le modèle pour le mot MODB est `cumf_MODB_G` si l'Annexe G est utilisée ou `cumf_MODB_M` si l'Annexe M est utilisée. Les index pour MODB sont définis respectivement dans le Tableau 11 ou dans le Tableau M.1. Le modèle pour CBP_n, n = 1, 2, ..., 4, est `cumf_YCBPB` et le modèle pour CBP_n, n = 5, 6, est `cumf_UVCBPB`, avec l'index 0 pour CBP_n = 0 et l'index 1 pour CBP_n = 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 les Tableaux 13 et 12.

Le modèle pour les mots MVD, MVD₂₋₄ et MVDB est `cumf_MVD` et le modèle pour le mot INTRADC est `cumf_INTRADC`. L'indexage est défini, respectivement, dans les Tableaux 14 et 15.

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 16. Le modèle du symbole SIGN est cumf_SIGN. L'indexage du symbole SIGN est: 0 pour le signe positif et 1 pour le 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 17.

Le modèle pour le MODE INTRA est cumf_INTRA_AC_DC. L'indexage est défini au Tableau I.1.

E.8 Modèles de codage SAC

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

```
int cumf_MCBPC no4MVQ[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 4MVQ[26]={16383, 6880, 6092, 5178, 4916, 3965, 3880, 3795, 3768, 1491, 1190, 889, 655, 442, 416, 390, 360, 337, 334, 331, 327, 326, 88, 57, 26, 0};
```

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

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

```
int cumf_MODB M[7]={16383, 6717, 4568, 2784, 1370, 655, 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};
```

```
int cumf_LEVEL[255]={16383, 16382, 16381, 16380, 16379, 16378, 16377, 16376, 16375, 16374,
16373, 16372, 16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361,
16360, 16359, 16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348,
16347, 16346, 16345, 16344, 16343, 16342, 16341, 16340, 16339, 16338, 16337, 16336, 16335,
16334, 16333, 16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322,
16321, 16320, 16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16310, 16309,
16308, 16307, 16306, 16305, 16304, 16303, 16302, 16301, 16300, 16299, 16298, 16297, 16296,
16295, 16294, 16293, 16292, 16291, 16290, 16289, 16288, 16287, 16286, 16285, 16284, 16283,
16282, 16281, 16280, 16279, 16278, 16277, 16250, 16223, 16222, 16195, 16154, 16153, 16071,
15989, 15880, 15879, 15878, 15824, 15756, 15674, 15606, 15538, 15184, 14572, 13960, 10718,
7994, 5379, 2123, 1537, 992, 693, 611, 516, 448, 421, 380, 353, 352, 284, 257, 230, 203, 162, 161,
160, 133, 132, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85,
84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58,
57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 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};
```

```
int cumf_LEVEL_intra[255]={16383, 16379, 16378, 16377, 16376, 16375, 16374, 16373, 16372,
16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361, 16360, 16359,
16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348, 16347, 16346,
16345, 16344, 16343, 16342, 16341, 16340, 16339, 16338, 16337, 16336, 16335, 16334, 16333,
16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322, 16321, 16320,
```

16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16268, 16267, 16224, 16223, 16180, 16179, 16136, 16135, 16134, 16133, 16132, 16131, 16130, 16129, 16128, 16127, 16126, 16061, 16018, 16017, 16016, 16015, 16014, 15971, 15970, 15969, 15968, 15925, 15837, 15794, 15751, 15750, 15749, 15661, 15618, 15508, 15376, 15288, 15045, 14913, 14781, 14384, 13965, 13502, 13083, 12509, 12289, 12135, 11892, 11738, 11429, 11010, 10812, 10371, 9664, 9113, 8117, 8116, 8028, 6855, 5883, 4710, 4401, 4203, 3740, 3453, 3343, 3189, 2946, 2881, 2661, 2352, 2132, 1867, 1558, 1382, 1250, 1162, 1097, 1032, 967, 835, 681, 549, 439, 351, 350, 307, 306, 305, 304, 303, 302, 301, 300, 299, 298, 255, 212, 211, 210, 167, 166, 165, 164, 163, 162, 161, 160, 159, 158, 115, 114, 113, 112, 111, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 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};

int cumf_INTRA_AC_DC[4]={16383, 9229, 5461, 0};

ANNEXE F

Mode de prédiction avancé

F.1 Introduction

La présente annexe décrit le mode facultatif de prédiction avancé de la présente Recommandation, y compris la compensation de mouvement par superposition de blocs et la possibilité de quatre vecteurs cinétiques par macrobloc. La possibilité de ce mode est signalée par des moyens externes (par exemple selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le mot PTYPE. Dans le mode de prédiction avancé, les vecteurs cinétiques sont autorisés à traverser les limites d'image, comme c'est le cas dans le mode de vecteurs cinétiques non restreints (pour la description de cette technique, voir D.1). La caractéristique d'extension de la gamme des vecteurs cinétiques dans le mode de vecteurs cinétiques non restreints n'est pas automatiquement incluse dans le mode de prédiction avancé et n'est active que si le mode de vecteurs cinétiques non restreints est choisi. Si le mode de prédiction avancé est utilisé en combinaison avec le mode de trames PB, la compensation de mouvement par superposition n'est utilisée que pour la prédiction des images P et non pour celle des images B.

F.2 Quatre vecteurs cinétiques par macrobloc

La présente Recommandation fait appel à un seul vecteur cinétique par macrobloc, sauf en mode de prédiction avancé ou en mode avec filtre de dégroupage. Dans ce mode, la décision relative au choix entre un ou quatre vecteurs est indiquée par le mot de code MCBPC pour chaque macrobloc. Si un seul vecteur cinétique est transmis pour un certain macrobloc, ce vecteur est défini sous la forme de quatre vecteurs ayant la même valeur. Si le mot MCBPC indique que quatre vecteurs cinétiques sont transmis pour le macrobloc en cours, l'information relative au premier vecteur cinétique est transmise dans le mot de code MVD et l'information relative aux trois autres vecteurs cinétiques est transmise dans les mots de code MVD₂₋₄ (voir aussi 5.3.7 et 5.3.8).

On obtient les vecteurs en ajoutant les prédicteurs aux différences vectorielles indiquées par les mots MVD et MVD₂₋₄ 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).

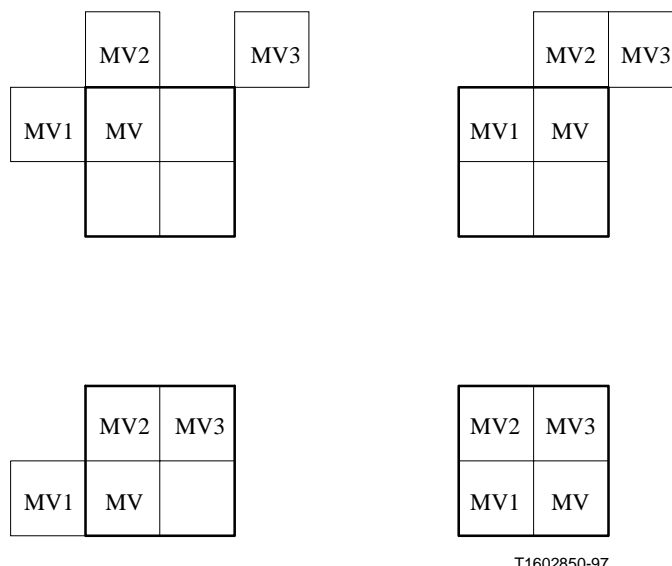


Figure F.1/H.263 – Redéfinition des prédicteurs candidats (MV1, MV2 et MV3) pour chacun des blocs de luminance d'un macrobloc

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. La numérotation des vecteurs cinétiques est équivalente à celle 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^{ème} 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 ^{ème} 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 en 6.1.2. En mode de prédiction avancé, la prédiction de la luminance est obtenue par compensation de mouvement par superposition, 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 vecteurs "distants" sur quatre:

- 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.

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).

Soit (x,y) un point dans une image, mesuré en unités entières de pixels.

Soit (m,n) un index de bloc entier dans une image, donné par:

$$m = x / 8 \quad \text{et} \quad n = y / 8,$$

où "/" indique une division avec troncature.

Soit (i,j) un emplacement de pixel entier dans un bloc de 8×8 , donné par:

$$i = x - m \cdot 8 \quad \text{et} \quad j = y - n \cdot 8,$$

ce qui donne:

$$(x,y) = (m \cdot 8 + i, n \cdot 8 + j).$$

Soit un vecteur cinétique (MV^k_x, MV^k_y) qui peut contenir des décalages de pixel entier ou de demi-pixel, avec $k = 0, 1$ ou 2 . Par exemple, (MV^k_x, MV^k_y) peut être égal à $(-7,0, 13,5)$. Ici, (MV^0_x, MV^0_y) désigne le vecteur cinétique pour le bloc en cours (m,n) , (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 (m,n) qui a été défini ci-dessus.

La création de chaque pixel, $P(x,y)$ dans un bloc de prédiction de luminance à 8×8 éléments, avec l'index de bloc (m,n) est régie par l'équation suivante:

$$P(x,y) = (q(x,y) \cdot H_0(i, j) + r(x,y) \cdot H_1(i, j) + s(x,y) \cdot H_2(i, j) + 4)/8,$$

où $q(x,y)$, $r(x,y)$ et $s(x,y)$ sont les valeurs de prédiction prises de l'image de référence et définis comme suit:

$$q(x, y) = p(x + MV^0_x, y + MV^0_y),$$

$$r(x, y) = p(x + MV^1_x, y + MV^1_y),$$

$$s(x, y) = p(x + MV^2_x, y + MV^2_y),$$

où $p(x + MV^k_x, y + MV^k_y)$ est la valeur de prédiction au point $(x + MV^k_x, y + MV^k_y)$ dans l'image de référence. Il convient de noter que $(x + MV^k_x, y + MV^k_y)$ peut se trouver en dehors de l'image et peut se situer à une position de pixel entier ou de demi-pixel. Dans les cas où l'on utilise des vecteurs cinétiques à demi-pixel, $p(x + MV^k_x, y + MV^k_y)$ se rapporte à la valeur obtenue après l'application du processus d'interpolation décrit au 6.1.2.

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 (i, j) désignent respectivement la colonne et la rangée de la matrice.

Quand ni le mode structuré par tranches (voir l'Annexe K), ni le mode de décodage par segments indépendants (voir l'Annexe R) ne sont utilisés, les vecteurs cinétiques distants provenant d'autres segments d'image vidéo sont utilisés de la même manière que les vecteurs cinétiques distants à l'intérieur du groupe de blocs en cours. Si le mode structuré par tranches ou le mode de décodage par

segments indépendants est utilisé, les vecteurs cinétiques distants correspondant aux blocs d'autres segments de l'image vidéo sont mis à la valeur du vecteur cinétique du bloc en cours, quelles que soient les autres conditions décrites dans le paragraphe qui suit. (Voir l'Annexe R pour la définition d'un segment d'image vidéo.)

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 donc 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

G.1 Introduction

La présente annexe décrit le mode facultatif de codage par trames PB de la présente Recommandation. La possibilité de ce mode est signalée par des moyens externes (par exemple, selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le champ 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 bidirectionnellement à partir des images passées et futures. Le processus de prédiction est illustré par la Figure G.1.

Une version améliorée du mode de codage par trames PB, nommée "mode de codage par trames PB amélioré" est décrit à l'Annexe M. Le mode de codage par trames PB décrit dans la présente annexe est retenu dans la présente Recommandation seulement afin de maintenir la compatibilité avec les systèmes conçus avant l'adoption du mode de codage par trames PB amélioré. Pour cette raison, le mode de codage par trames PB décrit dans la présente annexe ne peut être utilisé avec les caractéristiques additionnelles de la syntaxe qui exigent le recours au champ PLUSPTYPE.

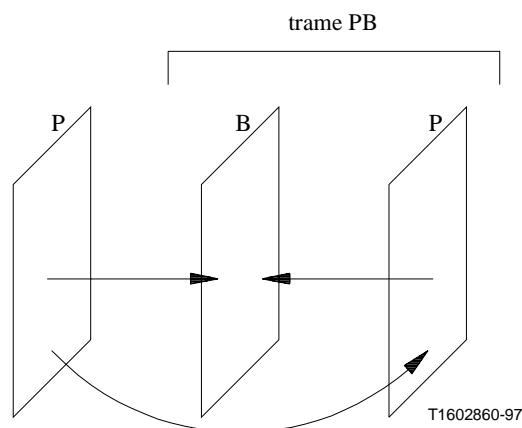


Figure G.1/H.263 – Prédiction en mode de codage par trames PB

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 PTYPE a la valeur "INTER". Dans ce cas, le vecteur n'est utilisé que pour les blocs de type B. Les mots de code MVD_{2-4} ne sont jamais utilisés pour le mode INTRA (voir également le Tableau 10). 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 11. INTRADC est présent pour chaque bloc P du macrobloc si le champ MCBPC indique le type de macrobloc 3 ou 4 (voir les Tableaux 7 et 8). INTRADC n'est pas présent pour les blocs B. 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). 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 améliorées par un vecteur delta qui est fourni par les données $MVDB$.

- TR_D : incrément de la référence temporelle TR (ou la combinaison de la référence temporelle étendue ETR et de la référence temporelle TR dans une trame PB améliorée lorsqu'une

fréquence de base d'image personnalisée est utilisée) à partir du dernier en-tête d'image (voir 5.1.2). Si TR_D est négatif, $TR_D = TR_D + d$, où $d = 256$ pour une fréquence d'image CIF et 1024 pour toute fréquence de base d'image personnalisée.

- TR_B : voir 5.1.2.

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 n'est pas présente, MV_D est mise à zéro. Si MVDB est présente, la même composante MV_D indiquée par 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:

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

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

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

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 la gamme des valeurs de la composante MV_F est bornée. Chaque mot à codage VLC pour MVDB représente une paire de valeurs différentielles. Une seule des valeurs de cette paire donnera une valeur de MV_F s'inscrivant dans la gamme permise (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 $16^{\text{ème}}$ pixel résultantes sont modifiées vers la plus proche position en demi-pixel, tel qu'indiqué dans le Tableau F.1. 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 $16^{\text{ème}}$ pixel sont modifiées vers la plus proche position en demi-pixel, tel qu'indiqué dans le Tableau F.1.

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 de référence, 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 le présent sous-paragraphe, un bloc est défini comme ayant 8×8 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 écrété (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 par rapport à l'image décodée précédente et de la prédiction différée sur la base de la composante MV_B par rapport au macrobloc P_{REC} . Cette moyenne est la moitié de la 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 par rapport à l'image décodée précédente.

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).

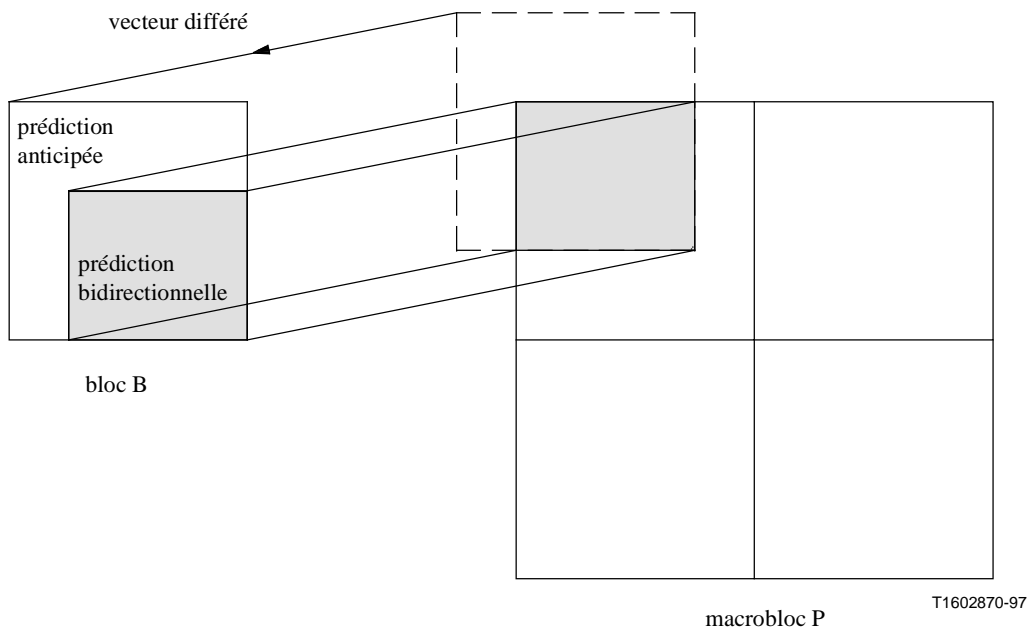


Figure G.2/H.263 – Prédiction anticipée et bidirectionnelle pour un bloc B

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;
- mvc 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 sans voie de retour pour signal vidéo codé

H.1 Introduction

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

H.2 Verrouillage de trames 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 multitrames de huit trames, chaque trame comportant 1 bit de verrouillage de trame, 1 bit indicateur de remplissage (F_i), 492 bits de données codées (ou un bourrage 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:

$$(S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8) = (00011011)$$

L'indicateur de remplissage (F_i) 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 BCH (511, 493) sans voie de retour. Son utilisation par le décodeur est facultative. La parité est calculée sur une séquence codée de 493 bits, comprenant 1 bit indicateur de remplissage (F_i) 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).

H.4 Temps de rétablissement pour le verrouillage de trames 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 ne 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.

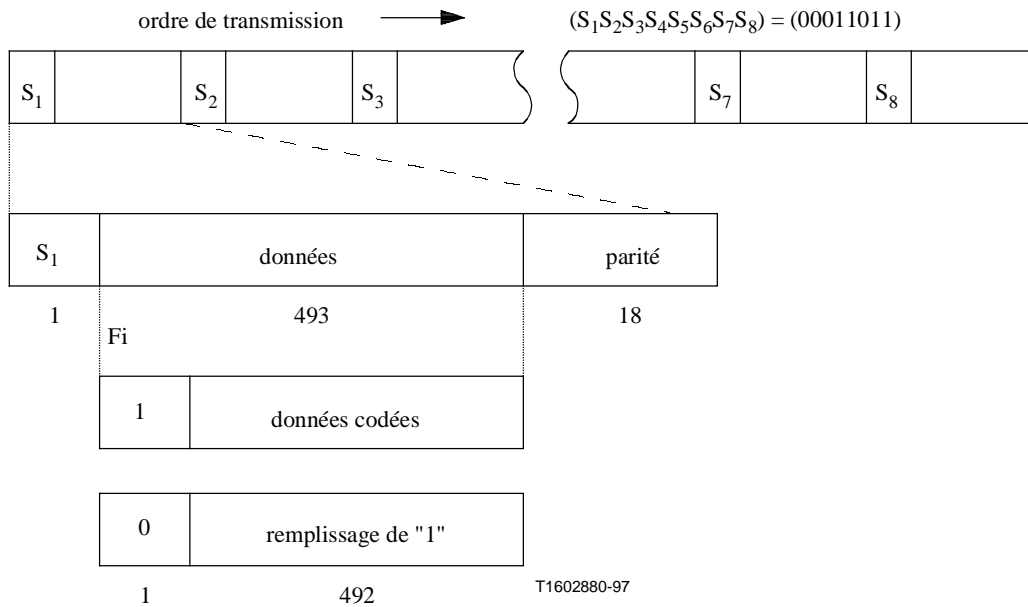


Figure H.1/H.263 – Trame de correction d'erreur

ANNEXE I

Mode de codage INTRA avancé

La présente annexe décrit le mode de codage facultatif INTRA avancé de la présente Recommandation. 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 champ PLUSPTYPE de l'en-tête d'image.

I.1 Introduction

Ce mode facultatif modifie le décodage des macroblocs de type "INTRA" (les autres types de macroblocs ne sont pas touchés). L'efficacité du codage des macroblocs INTRA est améliorée par recours à:

- 1) la prédiction des blocs INTRA en utilisant les blocs INTRA voisins pour la même composante (Y , C_B ou C_R);
- 2) la quantification inverse modifiée pour les coefficients INTRA;
- 3) un code VLC distinct pour les coefficients INTRA.

Un bloc à codage INTRA particulier peut être prédit à partir du bloc au-dessus du bloc en cours de décodage, à partir du bloc à gauche du bloc en cours de décodage ou encore à partir de ces deux blocs. Dans des cas spéciaux, les blocs voisins ne sont pas à codage INTRA ou ne se trouvent pas dans le même segment d'image vidéo. La prédiction par blocs utilise toujours des données correspondant à la même composante de luminance ou de différence de couleur (Y, C_B ou C_R) que le bloc en cours de décodage. Dans la prédiction, les coefficients DC sont toujours prédits d'une manière quelconque. La première rangée de coefficients AC peut être prédite à partir des coefficients du bloc supérieur, la première colonne de coefficients AC peut être prédite à partir de ceux du bloc gauche, ou encore seul le coefficient DC peut être prédit comme moyenne du bloc au-dessus et du bloc gauche, le choix étant signalé macrobloc par macrobloc. Les coefficients AC restants ne sont jamais prédits. La quantification inverse du coefficient INTRADC est modifiée afin de permettre une taille d'échelon de quantification variable, alors que, dans le corps de la présente Recommandation, une taille d'échelon fixe de 8 est utilisée pour les coefficients INTRADC. La quantification inverse de tous les coefficients INTRA est effectuée sans "zone morte" dans l'espace de reconstruction du quantificateur.

I.2 Syntaxe

Lorsqu'on utilise le mode de codage INTRA avancé, la syntaxe de la couche des macroblocs est modifiée de la manière précisée dans la Figure I.1. La syntaxe illustrée par la Figure I.1 est identique à celle qui est définie au 5.3, sauf pour ce qui est de l'insertion d'un champ INTRA_MODE additionnel pour les macroblocs INTRA. INTRA_MODE est présent seulement lorsque le champ MCBPC indique un macrobloc de type INTRA (macrobloc de type 3 ou 4). Le mode de prédiction est codé au moyen du code de longueur variable du Tableau I.1. Un mode de prédiction est transmis pour chaque macrobloc INTRA.

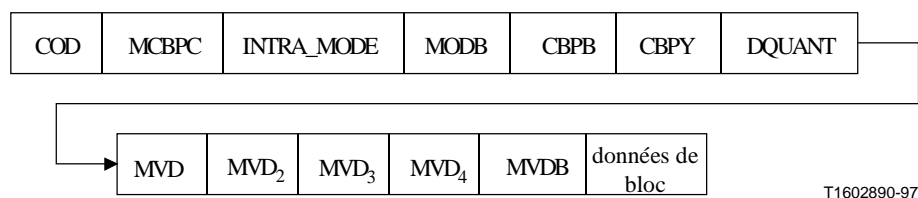


Figure I.1/H.263 – Structure de la couche des macroblocs

Tableau I.1/H.263 – Code de longueur variable (VLC) pour INTRA_MODE

Index	Mode de prédiction	VLC
0	0 (DC seulement)	0
1	1 (DC et AC verticaux)	10
2	2 (DC et AC horizontaux)	11

I.3 Processus de décodage

Deux balayages sont effectués en plus du balayage en zigzag. Les deux balayages additionnels sont illustrés dans les parties a) et b) de la Figure I.2, et le balayage en zigzag est illustré dans la Figure 14.

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

a) balayage horizontal alterné

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

b) balayage vertical alterné (comme dans la Recommandation H.262)

Figure I.2/H.263 – Séquences de balayage DCT pour codage INTRA avancé

Pour les blocs à codage INTRA, si Mode de prédiction = 0, le balayage en zigzag de la Figure 14 est choisi pour tous les blocs du macrobloc; sinon, la direction de la prédiction est utilisée pour sélectionner un balayage pour le macrobloc.

Lorsque Mode de prédiction = 1, le bloc adjacent sur le plan vertical est utilisé pour effectuer la prédiction. Ce mode de prédiction est conçu pour les blocs INTRA où domine un contenu de fréquence horizontale plus fort, de telle sorte que le bloc adjacent dans le sens vertical est utilisé pour prédire le contenu de fréquence horizontale du bloc en cours, la valeur de prédiction étant nulle pour tous les coefficients représentant un contenu AC vertical. La séquence de balayage est choisie de manière à balayer les fréquences horizontales plus fortes avant les verticales, par balayage horizontal alterné.

Lorsque Mode de prédiction = 2, le bloc adjacent sur le plan horizontal est utilisé pour effectuer la prédiction. Ce mode de prédiction est conçu pour les blocs INTRA où domine un contenu de fréquence verticale plus fort, de telle sorte que le bloc adjacent dans le sens horizontal est utilisé pour prédire le contenu de fréquence verticale du bloc en cours, la valeur de prédiction étant nulle pour tous les coefficients représentant un contenu AC horizontal. La séquence de balayage est choisie de manière à balayer les fréquences verticales plus fortes avant les horizontales, par balayage vertical alterné.

Pour les blocs qui ne sont pas à codage INTRA, les blocs de 8×8 de coefficients de transformée sont balayés selon une séquence en zigzag comme dans la Figure 14.

Une table VLC distincte est utilisée pour tous les coefficients AC INTRADC et INTRA. Elle figure au Tableau I.2. Il est à noter que les entrées de mots de code utilisées dans le Tableau I.2 sont identiques à celles qui sont utilisées dans le tableau TCOEF normal (Tableau 16) utilisé lorsque le codage INTRA avancé n'est pas utilisé, avec toutefois une interprétation différente des expressions LEVEL et RUN (sans modification du sens de LAST).

Tableau I.2/H.263 – Code de longueur variable pour INTRA TCOEF

INDEX	LAST	RUN	 LEVEL 	BITS	CODE VLC
0	0	0	1	3	10s
1	0	1	1	5	1111s
2	0	3	1	7	0101 01s
3	0	5	1	8	0010 111s
4	0	7	1	9	0001 1111s
5	0	8	1	10	0001 00101s
6	0	9	1	10	0001 00100s
7	0	10	1	11	0000 1000 01s
8	0	11	1	11	0000 1000 00s
9	0	4	3	12	0000 0000 111s
10	0	9	2	12	0000 0000 110s
11	0	13	1	12	0000 0100 000s
12	0	0	2	4	110s
13	0	1	2	7	0101 00s
14	0	1	4	9	0001 1110s
15	0	1	5	11	0000 0011 11s
16	0	1	6	12	0000 0100 001s
17	0	1	7	13	0000 0101 0000s
18	0	0	3	5	1110s
19	0	3	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	3	4	13	0000 0101 0001s
22	0	0	5	6	0110 1s
23	0	4	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	0	4	6	0110 0s
26	0	5	2	10	0001 0001 0s
27	0	5	3	13	0000 0101 0010s
28	0	2	1	6	0101 1s
29	0	6	2	11	0000 0011 00s
30	0	0	25	13	0000 0101 0011s
31	0	4	1	7	0100 11s
32	0	7	2	11	0000 0010 11s
33	0	0	24	13	0000 0101 0100s
34	0	0	8	7	0100 10s
35	0	8	2	11	0000 0010 10s
36	0	0	7	7	0100 01s
37	0	2	4	11	0000 0010 01s
38	0	0	6	7	0100 00s
39	0	12	1	11	0000 0010 00s
40	0	0	9	8	0010 110s
41	0	0	23	13	0000 0101 0101s
42	0	2	2	8	0010 101s
43	0	1	3	8	0010 100s
44	0	6	1	9	0001 1100s
45	0	0	10	9	0001 1011s
46	0	0	12	10	0001 0000 1s

Tableau I.2/H.263 – Code de longueur variable pour INTRA TCOEF (suite)

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
47	0	0	11	10	0001 0000 0s
48	0	0	18	10	0000 1111 1s
49	0	0	17	10	0000 1111 0s
50	0	0	16	10	0000 1110 1s
51	0	0	15	10	0000 1110 0s
52	0	0	14	10	0000 1101 1s
53	0	0	13	10	0000 1101 0s
54	0	0	20	12	0000 0100 010s
55	0	0	19	12	0000 0100 011s
56	0	0	22	13	0000 0101 0110s
57	0	0	21	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	14	1	10	0000 1100 1s
60	1	20	1	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	19	1	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	0	2	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	4	1	8	0010 001s
69	1	0	3	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	8	1	9	0001 0101s
76	1	7	1	9	0001 0100s
77	1	0	4	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	16	1	10	0000 1011 0s
81	1	15	1	10	0000 1010 1s
82	1	2	2	10	0000 1010 0s
83	1	1	2	10	0000 1001 1s
84	1	0	6	10	0000 1001 0s
85	1	0	5	10	0000 1000 1s
86	1	4	2	11	0000 0001 11s
87	1	3	2	11	0000 0001 10s
88	1	1	3	11	0000 0001 01s
89	1	0	7	11	0000 0001 00s
90	1	2	3	12	0000 0100 100s
91	1	1	4	12	0000 0100 101s
92	1	0	9	12	0000 0100 110s
93	1	0	8	12	0000 0100 111s
94	1	21	1	13	0000 0101 1000s

Tableau I.2/H.263 – Code de longueur variable pour INTRA TCOEF (fin)

INDEX	LAST	RUN	LEVEL	BITS	CODE VLC
95	1	22	1	13	0000 0101 1001s
96	1	23	1	13	0000 0101 1010s
97	1	7	2	13	0000 0101 1011s
98	1	6	2	13	0000 0101 1100s
99	1	5	2	13	0000 0101 1101s
100	1	3	3	13	0000 0101 1110s
101	1	0	10	13	0000 0101 1111s
102	ESCAPE			7	0000 011

Selon la valeur de INTRA_MODE, soit un, soit huit coefficients sont des résidus de prédiction qui doivent être ajoutés à un prédicteur tel que décrit ci-dessous. La Figure I.3 montre trois blocs 8×8 de niveaux finals DCT reconstruits pour une même composante (Y , C_B ou C_R), et qui sont étiquetés $RecA'(u,v)$, $RecB'(u,v)$ et $RecC'(u,v)$, où u et v sont respectivement des indices de colonne (horizontal) et de rangée (vertical). Le processus de reconstruction diffère du traitement décrit au 6.2.1. Les résidus INTRADC sont reconstruits différemment: on utilise une taille d'échelon variable au lieu d'utiliser le Tableau 15, un prédicteur est ensuite ajouté aux valeurs résiduelles afin d'obtenir la valeur de reconstruction finale du coefficient. Les coefficients INTRA autres que INTRADC sont aussi reconstruits d'un manière qui diffère de celle au 6.2.1, au moyen d'un espacement de reconstruction sans "zone morte" et, dans certains cas, en ajoutant un prédicteur afin d'obtenir la valeur de reconstruction finale du coefficient. Le bloc peut contenir des résidus de prédiction DC et AC.

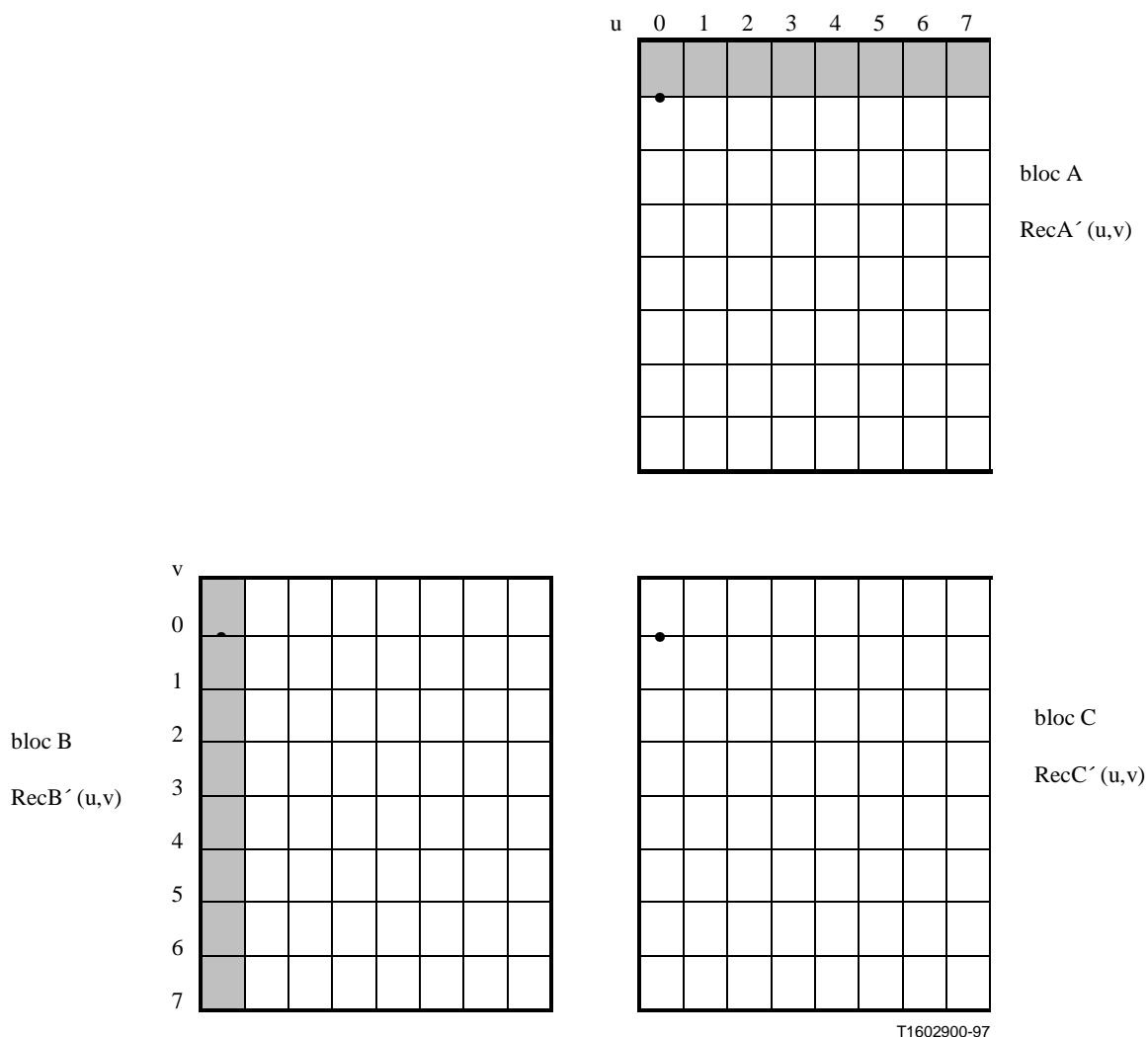


Figure I.3/H.263 – Trois blocs voisins du domaine DCT

Les définitions de MCBPC et CBPY changent lorsque le codage INTRA avancé est utilisé. Lorsque le codage INTRA avancé est utilisé, les coefficients de transformée INTRADC ne sont plus traités comme un cas distinct, ils sont plutôt traités de la même manière que des coefficients AC en ce qui concerne MCBPC et CBPY. Cela signifie qu'un coefficient INTRADC de valeur nulle ne sera pas codé comme un niveau (LEVEL), mais il allongera plutôt l'exécution pour les coefficients AC suivants.

Le processus de quantification inverse de la partie B d'une trame PB améliorée (voir l'Annexe M) n'est pas modifié par l'utilisation du mode de codage INTRA avancé.

Définir $RecC(u,v)$ comme les *résidus* de coefficient reconstruit du bloc en cours. Pour tous les coefficients INTRA, la valeur résiduelle reconstituée est obtenue au moyen de:

$$RecC(u,v) = 2 * QUANT * LEVEL(u,v) \quad u = 0, \dots, 7, v = 0, \dots, 7.$$

NOTE – $LEVEL(u,v)$ représente une quantité ayant une grandeur et un signe dans l'équation ci-dessus.

Définir $RecC'(u,v)$ comme les valeurs *finales* de coefficient reconstruit du bloc en cours (après des ajustements de prédiction, oddification, tel que décrit ci-dessous, et écrêtage). Les valeurs finales de coefficient reconstruit $RecC'(u,v)$ sont récupérées en ajoutant $RecC(u,v)$ à la prédiction appropriée indiquée dans le champ INTRA_MODE, en changeant, au besoin, la valeur du bit de plus faible poids du coefficient DC, et en effectuant un écrêtage.

$RecA'(u,v)$ représente les valeurs *finales* de coefficient reconstruit du bloc immédiatement au-dessus du bloc en cours. $RecB'(u,v)$ représente les valeurs *finales* de coefficient reconstruit du bloc immédiatement à la gauche du bloc en cours.

Les valeurs de coefficient reconstruit pour les blocs A et B peuvent être utilisées pour la prédiction des valeurs de coefficient pour le bloc C si les blocs A et B sont dans le même segment d'image vidéo que le bloc C. On dit qu'un bloc est "dans le même segment d'image vidéo" qu'un autre bloc seulement si toutes les conditions suivantes sont respectées:

- 1) le bloc en question est dans les limites de l'image;
- 2) si l'on est pas en mode structuré par tranches (voir l'Annexe K), le bloc pertinent est dans le même groupe de blocs ou aucun en-tête de groupe de blocs n'est présent pour le groupe de blocs en cours;
- 3) si l'on est en mode structuré par tranches, le bloc pertinent est dans la même tranche.

Le bloc C qui doit être décodé est prédit seulement à partir de blocs INTRA dans le même segment d'image vidéo que le bloc C, comme on le voit plus bas.

Si Mode de prédiction = 0 (prédiction DC seulement) et que les blocs A et B sont tous deux des blocs INTRA dans le même segment d'image vidéo que le bloc C, le coefficient DC du bloc C est prédit à partir de la moyenne (avec troncature) des coefficients DC des blocs A et B. Si un seul des deux blocs A et B est un bloc INTRA dans le même segment d'image vidéo que le bloc C, le coefficient DC de ce seul bloc est utilisé comme prédicteur lorsque Mode de prédiction = 0. Si aucun des deux blocs A et B n'est un bloc INTRA dans le même segment d'image vidéo que le bloc C, la prédiction utilise la valeur 1024 comme prédicteur du coefficient DC.

Si Mode de prédiction = 1 ou 2 (prédiction DC et AC verticale ou DC et AC horizontale) et que le bloc de référence (bloc A ou bloc B) n'est pas un bloc INTRA dans le même segment d'image vidéo que le bloc C, la prédiction utilise la valeur 1024 comme prédicteur du coefficient DC et la valeur 0 comme prédicteur des coefficients AC du bloc C.

Un processus de changement de signe "oddification" est appliqué au coefficient DC afin de minimiser les erreurs d'adaptation de la transformation inverse IDCT. Certaines valeurs de coefficients peuvent entraîner une erreur d'adaptation en raison d'une erreur d'arrondissement entre diverses mises en œuvre de l'IDCT, en particulier certaines valeurs de coefficients (0,0), (0,4), (4,0) et (4,4). Ainsi, un coefficient DC de $8k + 4$ pour certains entiers k a pour effet qu'un bloc ayant subi la transformation inverse a une valeur constante $k + 0,5$, pour laquelle des erreurs mineures peuvent entraîner l'obtention d'arrondissements divergents pour diverses mises en œuvre.

Définir la fonction clipAC() pour indiquer un écrêtage à la gamme -2048 à 2047. Définir la fonction clipDC() pour indiquer un écrêtage à la gamme 0 à 2047. Définir la fonction oddifyclipDC(x) comme suit:

```
If (x is even) {
    result = clipDC(x+1)
} else {
    result = clipDC(x)
}
```

La reconstruction pour chaque mode de prédiction INTRA est alors spécifiée comme suit, où l'opérateur "/" est une division par troncature:

mode 0: prédiction DC seulement.

$$\text{RecC}'(u,v) = \text{clipAC}(\text{RecC}(u,v)) \quad (u,v) \neq (0,0), u = 0, \dots, 7, v = 0, \dots, 7.$$

If (block A and block B are both INTRA coded and are both in the same video picture segment as block C) {

$$\text{tempDC} = \text{RecC}(0,0) + (\text{RecA}'(0,0) + \text{RecB}'(0,0)) / 2$$

} else {

If (block A is INTRA coded and is in the same video picture segment as block C) {

$$\text{tempDC} = \text{RecC}(0,0) + \text{RecA}'(0,0)$$

} else {

If (block B is INTRA coded and is in the same video picture segment as block C) {

$$\text{tempDC} = \text{RecC}(0,0) + \text{RecB}'(0,0)$$

} else {

$$\text{tempDC} = \text{RecC}(0,0) + 1024$$

}

}

}

$$\text{RecC}'(0,0) = \text{oddifyclipDC}(\text{tempDC})$$

Mode 1: prédiction DC et AC pour le bloc ci-dessus.

If (block A is INTRA coded and is in the same video picture segment as block C) {

$$\text{tempDC} = \text{RecC}(0,0) + \text{RecA}'(0,0)$$

$$\text{RecC}'(u,0) = \text{clipAC}(\text{RecC}(u,0) + \text{RecA}'(u,0)) \quad u = 1, \dots, 7,$$

$$\text{RecC}'(u,v) = \text{clipAC}(\text{RecC}(u,v)) \quad u = 0, \dots, 7, v = 1, \dots, 7.$$

} else {

$$\text{tempDC} = \text{RecC}(0,0) + 1024$$

$$\text{RecC}'(u,v) = \text{clipAC}(\text{RecC}(u,v)) \quad (u,v) \neq (0,0), u = 0, \dots, 7, v = 0, \dots, 7$$

}

$$\text{RecC}'(0,0) = \text{oddifyclipDC}(\text{tempDC})$$

Mode 2: DC and AC prediction from the block to the left.

If (block B is INTRA coded and is in the same video picture segment as block C) {

$$\text{tempDC} = \text{RecC}(0,0) + \text{RecB}'(0,0)$$

$$\text{RecC}'(0,v) = \text{clipAC}(\text{RecC}(0,v) + \text{RecB}'(0,v)) \quad v = 1, \dots, 7,$$

$$\text{RecC}'(u,v) = \text{clipAC}(\text{RecC}(u,v)) \quad u = 1, \dots, 7, v = 0, \dots, 7.$$

} else {

$$\text{tempDC} = \text{RecC}(0,0) + 1024$$

$$\text{RecC}'(u,v) = \text{clipAC}(\text{RecC}(u,v)) \quad (u,v) \neq (0,0), u = 0, \dots, 7, v = 0, \dots, 7$$

}

$$\text{RecC}'(0,0) = \text{oddifyclipDC}(\text{tempDC})$$

Mode avec filtre de dégroupage

J.1 Introduction

La présente annexe décrit l'utilisation d'un filtre facultatif de limite de bloc dans la boucle de codage. Le principal objet de ce filtre est la réduction des artefacts liés au groupage. Le filtrage est appliqué aux limites de blocs 8×8 . Les vecteurs cinétiques peuvent avoir une résolution de 8×8 ou 16×16 (voir J.2). Le traitement décrit dans la présente annexe s'applique seulement aux images P, I, EP ou EI ou à la partie image P d'une trame PB améliorée. (La possibilité de filtrage d'images B ou de la partie image B d'une trame PB améliorée n'est pas une question de normalisation, bien qu'un type quelconque de filtrage soit recommandé afin d'améliorer la qualité d'image.) La possibilité de ce mode est signalée par des moyens externes (par exemple, selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le champ PLUSPTYPE de l'en-tête d'image.

J.2 Relation avec les modes UMV et AP (Annexes D et F)

L'utilisation du mode avec filtre de dégroupage a des effets sur la qualité d'image semblables à la compensation de mouvement par superposition de blocs (OBMC) telle qu'elle est définie à l'Annexe F, lorsqu'elle est utilisée seule. L'utilisation conjointe des deux techniques permet l'obtention d'une amélioration additionnelle de la qualité d'image. Le mode de prédiction avancé (voir aussi l'Annexe F) se compose de trois éléments:

- 1) quatre vecteurs cinétiques par macrobloc, tel que défini dans F.2;
- 2) compensation de mouvement par superposition pour la luminance, tel que défini dans F.3;
- 3) vecteurs cinétiques pointant hors des limites d'image, tel que défini dans D.1.

Pour que le mode avec filtre de dégroupage puisse offrir des performances maximales lorsque des considérations relatives à la complexité peuvent empêcher l'utilisation de la partie OBMC du mode de prédiction avancé, le mode avec filtre de dégroupage offre la possibilité d'utiliser quatre vecteurs cinétiques par macrobloc et des vecteurs cinétiques pointant hors des limites d'image.

En bref, les trois options définies aux Annexes D, F et J contiennent les cinq éléments de codage suivants:

- 1) vecteurs cinétiques pointant hors des limites d'image (D.1);
- 2) extension de la gamme des vecteurs cinétiques (D.2);
- 3) quatre vecteurs cinétiques par macrobloc (F.2);
- 4) compensation de mouvement par superposition pour la luminance (F.3);
- 5) filtre de limite de dégroupage (J.3).

Le Tableau J.1 indique lequel des cinq éléments est mis en fonction par l'activation des options des Annexes D, F et J.

Tableau J.1/H.263 – Eléments caractéristiques des modes UMV, AP et DF

Mode de codage par vecteurs cinétiques non restreints	Mode de prédiction avancé	Mode avec filtre de dégroupage	Vecteurs cinétiques pointant hors des limites d'image	Extension de la gamme des vecteurs cinétiques	Quatre vecteurs cinétiques par macrobloc	Compensation de mouvement par superposition pour la luminance	Filtre de limite de dégroupage
non activé	non activé	non activé	non activé	non activé	non activé	non activé	non activé
non activé	non activé	activé	activé	non activé	activé	non activé	activé
non activé	activé	non activé	activé	non activé	activé	activé	non activé
non activé	activé	activé	activé	non activé	activé	activé	activé
activé	non activé	non activé	activé	activé	non activé	non activé	non activé
activé	non activé	activé	activé	activé	activé	non activé	activé
activé	activé	non activé	activé	activé	activé	activé	non activé
activé	activé	activé	activé	activé	activé	activé	activé

J.3 Définition du filtre de limite de dégroupage

Des traitements de filtrage sont appliqués aux limites de blocs 8×8 au codeur ainsi qu'au décodeur. Les données d'image reconstruite (la somme de la prédiction et de l'erreur de prédiction reconstruite) sont écrêtées de manière à maintenir les valeurs dans la gamme 0 à 255, tel que décrit au 6.3.2. Le filtrage est ensuite appliqué, ce qui modifie l'image qui doit être stockée dans la mémoire d'image pour une prédiction future. Les opérations de filtrage comprennent un écrêtage additionnel garantissant que les valeurs de pixel résultantes soient dans la gamme de 0 à 255. Aucun filtrage n'est effectué sur la limite d'une image et, lorsque le mode de décodage par segments indépendants est utilisé, aucun filtrage n'est effectué sur les limites des tranches lorsque le mode structuré par tranches est utilisé (voir les Annexes K et R) ou sur la limite supérieure des groupes de blocs pour lesquels il y a un en-tête de groupe de blocs lorsque le mode structuré par tranches n'est pas utilisé (voir l'Annexe R). Les données de chrominance ainsi que de luminance sont filtrées.

Lorsque le mode décrit dans la présente annexe est utilisé avec le mode de codage par trames PB amélioré de l'Annexe M, la prédiction différée du macrobloc B est fondée sur le macrobloc P reconstruit (nommé P_{REC} au G.5) après l'écrêtage mais avant les opérations effectuées par le filtre de limite de dégroupage. La prédiction anticipée du macrobloc B est fondée sur la version filtrée de l'image décodée auparavant (soit les mêmes données d'image qui sont utilisées pour la prédiction du macrobloc P).

Le filtre de déblocage utilise un ensemble de quatre valeurs de pixel (écrêtées) sur une ligne horizontale ou verticale de l'image reconstruite, notées A, B, C et D, dont A et B appartiennent à un bloc nommé bloc1, et C et D appartiennent à un bloc voisin nommé bloc2, qui se trouve à la droite ou au-dessous du bloc1. La Figure J.1 montre des exemples de position de ces pixels.

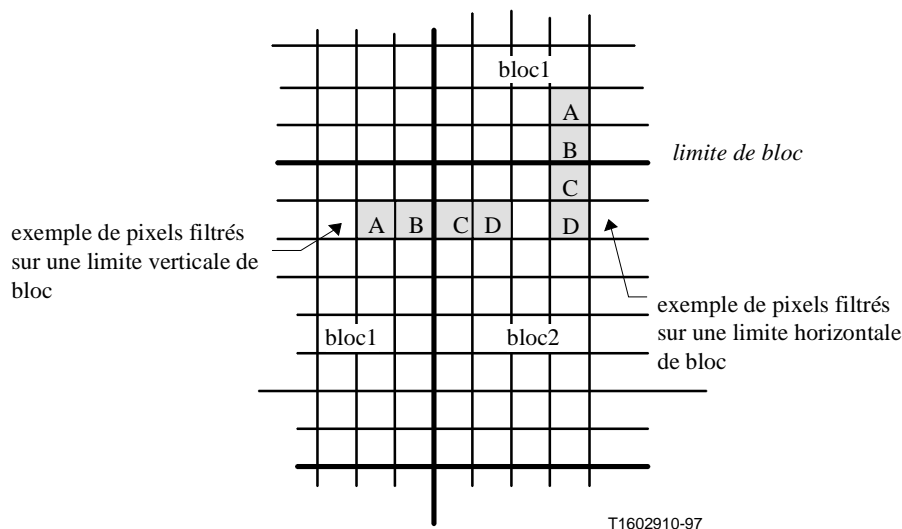


Figure J.1/H.263 – Exemples de positions de pixels filtrés

L'une ou l'autre, ou encore les deux conditions suivantes doivent être respectées afin d'appliquer le filtrage à une limite particulière:

- condition 1: le bloc1 appartient à un macrobloc codé (COD==0 || MB-type == INTRA);
- condition 2: le bloc2 appartient à un macrobloc codé (COD==0 || MB-type == INTRA).

Si le filtrage doit être appliqué sur la limite, A, B, C et D doivent être remplacés par A1, B1, C1, D1 où:

$$B1 = \text{clip}(B + d1)$$

$$C1 = \text{clip}(C - d1)$$

$$A1 = A - d2$$

$$D1 = D + d2$$

$$d = (A - 4B + 4C - D) / 8$$

$$d1 = \text{UpDownRamp}(d, \text{STRENGTH})$$

$$d2 = \text{clipd1}((A - D) / 4, d1/2)$$

$$\text{UpDownRamp}(x, \text{STRENGTH}) = \text{SIGN}(x) * (\text{MAX}(0, \text{abs}(x) - \text{MAX}(0, 2 * (\text{abs}(x) - \text{STRENGTH}))))$$

La valeur de STRENGTH dépend de celle de QUANT et elle commande le degré de filtrage. La relation entre STRENGTH et QUANT est donnée dans le Tableau J.2.

QUANT = paramètre de quantification utilisé pour le bloc2 si ce dernier appartient à un macrobloc codé;

QUANT = paramètre de quantification utilisé pour le bloc1 si le bloc2 n'appartient pas à un macrobloc codé (mais si le bloc1 appartient à un tel macrobloc).

Tableau J.2/H.263 – Relation entre QUANT et STRENGTH pour le filtre

QUANT	STRENGTH	QUANT	STRENGTH
1	1	17	8
2	1	18	8
3	2	19	8
4	2	20	9
5	3	21	9
6	3	22	9
7	4	23	10
8	4	24	10
9	4	25	10
10	5	26	11
11	5	27	11
12	6	28	11
13	6	29	12
14	7	30	12
15	7	31	12
16	7		

La fonction clip(x) est définie conformément au 6.3.2 et la fonction clipd1(x, lim) écrête x de manière à limiter sa valeur à la gamme $\pm \text{abs}(\text{lim})$. Le symbole "/" représente une division par troncature vers zéro.

La Figure J.2 montre comment la valeur d1 varie en fonction de d. Comme résultat, le filtre a un effet seulement si d est plus petit que $2 * \text{STRENGTH}$ (et différent de zéro). Cela a pour but d'empêcher le filtrage de limites réelles fortes dans le contenu d'image. Toutefois, si le mode de rafraîchissement à résolution réduite est utilisé, la valeur de STRENGTH est réglée à l'infini et il en résulte que la valeur de d1 est toujours égale à la valeur de d (voir Q.7.2).

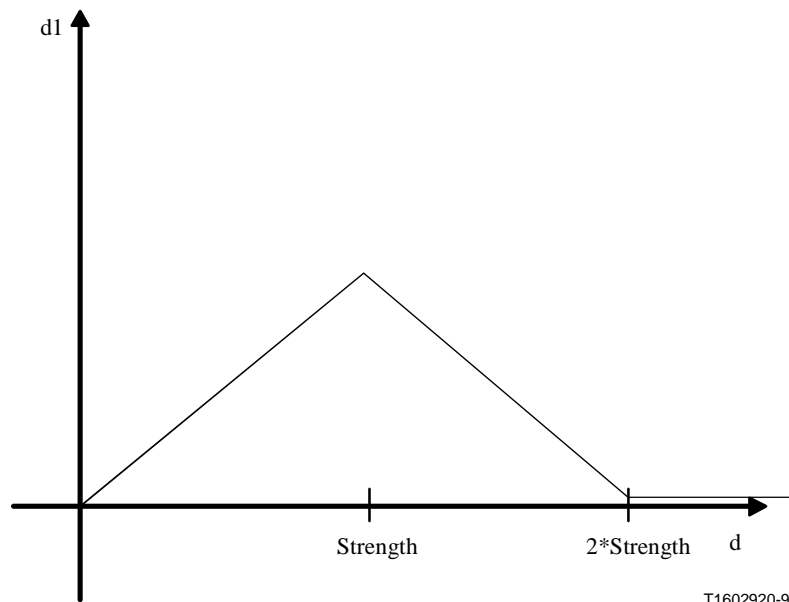


Figure J.2/H.263 – Paramètre d1 comme fonction du paramètre d pour le mode avec filtre de dégroupage

La définition de d1 est conçue afin que les petites erreurs d'adaptation entre le codeur et le décodeur demeurent faibles et ne s'additionnent pas d'image en image dans une séquence vidéo. Cela poserait un problème, par exemple, dans le cas d'une situation où le filtre est simplement mis en marche et arrêté, car une erreur d'adaptation de d de seulement ± 1 pourrait faire en sorte que le filtre soit activé au côté codeur et désactivé du côté décodeur, ou inversement.

En raison des effets de l'arrondissement, l'ordre des limites où le filtrage est effectué doit être précisé.

Filtrage sur les limites horizontales:

ce processus est supposé se dérouler en premier lieu. Plus précisément, les pixels $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}$ qui sont

utilisés dans le filtrage sur une limite horizontale ne doivent pas avoir été touchés par le filtrage précédent sur une limite verticale.

Filtrage sur les limites verticales:

avant l'exécution du filtrage sur une limite verticale au moyen des pixels (A, B, C, D), toutes les modifications des pixels (A, B, C, D) résultant du filtrage appliqué sur une limite horizontale doivent avoir été effectuées.

Il est à noter que si un ou plusieurs des pixels (A, B, C, D) prenant part à un processus de filtrage sont à l'extérieur d'une image, aucun filtrage n'a lieu. De plus, si le mode de décodage par segments indépendants est utilisé (voir l'Annexe R) et qu'un ou plusieurs des pixels (A, B, C, D) prenant part à un processus de filtrage sont dans des segments d'image vidéo différents (voir I.3, au sujet des conditions où un bloc est considéré comme étant dans le même segment d'image vidéo), aucun filtrage n'est effectué.

Mode structuré par tranches

K.1 Introduction

La présente annexe décrit le mode facultatif de codage structuré par tranches de la présente Recommandation. 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 champ PLUSPTYPE de l'en-tête d'image. Afin de faciliter son utilisation optimale dans divers environnements, ce mode comporte deux sous-modes qui peuvent aussi être signalés par des moyens externes (par exemple, la Recommandation H.245). Ces deux sous-modes indiquent si des tranches rectangulaires seront utilisées ou non et/ou si les tranches seront transmises en ordre séquentiel ou selon un ordre arbitraire.

Une tranche est définie comme un en-tête de tranche suivi de macroblocs consécutifs en ordre de balayage. Il y a toutefois une exception pour la tranche qui suit immédiatement le code de déclenchement d'image dans le flux binaire d'une image (qui n'est pas nécessairement la tranche débutant par le macrobloc 0). Dans ce cas, seule une partie de l'en-tête de tranche est transmise tel que décrit au K.2. La couche des tranches définit un segment d'image vidéo et elle est utilisée au lieu de la couche des groupes de blocs dans ce mode facultatif. Un segment de tranche d'image vidéo commence par une limite de macrobloc de l'image et il contient un certain nombre de macroblocs. Des tranches différentes dans une même image ne doivent pas se chevaucher les unes les autres et chaque macrobloc doit appartenir à une et à une seule tranche.

Ce mode contient deux sous-modes, signalés dans le champ SSS de l'en-tête d'image:

- 1) le sous-mode de tranches rectangulaires (RS, *rectangular slice submode*): lorsque le sous-mode RS est utilisé, la tranche occupera une zone rectangulaire dont la largeur est spécifiée par le paramètre SWI de l'en-tête de tranche, en unités de macroblocs, et elle contiendra un certain nombre de macroblocs en ordre de balayage dans la zone rectangulaire. Lorsque le sous-mode de tranches rectangulaires n'est pas utilisé, le champ SWI est absent de l'en-tête de tranche et la tranche contient un certain nombre de macroblocs en ordre de balayage dans l'image dans son ensemble;
- 2) le sous-mode d'ordre arbitraire de tranches (ASO, *arbitrary slice ordering submode*): lorsque ce sous-mode est utilisé, les tranches peuvent figurer dans n'importe quel ordre dans le flux binaire. Lorsque ce sous-mode n'est pas utilisé, les tranches doivent être envoyées dans l'ordre (unique) selon lequel la valeur du champ MBA de l'en-tête de tranche augmente strictement d'une tranche à la tranche subséquente de l'image.

Les limites de tranches sont traitées différemment des limites de macrobloc ordinaire afin que les emplacements des en-têtes de tranche dans le flux binaire tiennent lieu de point de resynchronisation pour le rétablissement en cas d'erreur sur les bits ou de perte de paquet et afin de permettre le décodage de tranches désordonnées dans une image. Ainsi, aucune dépendance de données ne peut traverser les limites des tranches dans l'image en cours, sauf dans le cas du mode avec filtre de dégroupage qui, lorsqu'il est utilisé sans le mode de décodage par segments indépendants, filtre au-delà des limites des blocs de l'image. Toutefois, les vecteurs cinétiques dans une tranche peuvent entraîner des dépendances de données qui traversent les limites des tranches dans l'image de référence utilisée aux fins de prédiction, à moins que le mode facultatif de décodage par segments indépendants soit utilisé.

Les règles suivantes sont adoptées afin de garantir que les emplacements des limites de tranches puissent servir de points de resynchronisation et que les tranches puissent être envoyées de manière non ordonnée sans que cela cause des retards de décodage additionnels:

- 1) les valeurs de prédiction des vecteurs cinétiques sont les mêmes que si un en-tête de groupe de blocs était présent (voir 6.1.1), ce qui empêche l'utilisation de vecteurs cinétiques de blocs à l'extérieur d'une tranche en cours pour la prédiction des valeurs des vecteurs cinétiques dans la tranche;
- 2) le mode de codage INTRA avancé (voir l'Annexe I) traite la limite de tranches comme si elle était une limite d'image du point de vue de la prédiction de valeurs de coefficient DCT de bloc INTRA;
- 3) l'assignation de vecteurs cinétiques distants en vue de la compensation de mouvement par superposition de blocs dans le mode de prédiction avancé empêche aussi l'utilisation de vecteurs cinétiques de blocs en dehors de la tranche en cours comme vecteurs cinétiques distants (voir F.3).

K.2 Structure de la couche des tranches

La structure syntaxique de la couche des tranches est illustrée à la Figure K.1 pour toutes les tranches sauf la tranche qui suit immédiatement le code de déclenchement d'image dans le flux binaire d'une image. Dans le cas de la tranche qui suit le code de déclenchement d'image, seuls les bits de prévention d'émulation (SEPB1, SEPB3 et, conditionnellement, SEPB2, tel que spécifié ci-dessous), le champ MBA et, en sous-mode RS, le champ SWI, sont inclus.

SSTUF	SSC	SEPB1	SSBI	MBA	SEPB2	SQUANT	SWI	SEPB3	GFID	données de macrobloc
-------	-----	-------	------	-----	-------	--------	-----	-------	------	----------------------

Figure K.1/H.263 – Structure de la couche des tranches

Voir 5.2.5 pour GFID et 5.3 pour la description de la couche des macroblocs.

K.2.1 Bourrage (SSTUF, *stuffing*) (longueur variable)

Mot de longueur variable, se composant de moins de 8 bits. Les codeurs doivent insérer ce mot de code directement avant un mot SSC dans tous les cas où il est nécessaire de s'assurer que SSC soit aligné sur une limite d'octet. Si le mot SSTUF est présent, son dernier bit doit être le dernier bit (de plus faible poids) d'un octet, de manière que le début du mot SSC soit aligné en octets. Les décodeurs doivent être conçus de façon à ignorer le mot SSTUF. Il est à noter que des 0 sont utilisés pour le bourrage dans SSTUF.

K.2.2 Code de déclenchement de tranche (SSC, *slice start code*) (17 bits)

Mot de 17 bits de valeur 0000 0000 0000 0000 1. Les codes de déclenchement de tranche doivent être alignés sur des limites d'octet. Cela peut être obtenu par insertion de SSTUF avant le code de déclenchement de manière que le premier bit du code de déclenchement soit le premier (de plus fort poids) d'un octet. Le code de déclenchement de tranche est absent de la tranche qui suit le code de déclenchement d'image.

K.2.3 Bit de prévention d'émulation de tranche 1 (SEPB1, *slice emulation prevention bit 1*) (1 bit)

Un seul bit dont la valeur est toujours "1", et qui est inclus afin de prévenir l'émulation de code de déclenchement.

K.2.4 Indicateur de sous-flux binaire de tranche (SSBI, *slice sub-bitstream indicator*) (4 bits)

Mot de code d'une longueur de quatre bits, présent seulement lorsque CPM = "1" dans l'en-tête d'image. SSBI indique le numéro de sous-flux binaire de la tranche pour le fonctionnement en communication multipoint à présence continue et à multiplex vidéo, tel que décrit à l'Annexe C. Le mappage entre la valeur de SSBI et le numéro de sous-flux est donné dans le Tableau K.1. SSBI est absent pour la tranche qui suit le code de déclenchement d'image.

Tableau K.1/H.263 – Valeur de SSBI et numéros de sous-flux binaire associés

Numéro de sous-flux binaire	Valeur de champ SSBI	Valeur de GN émulée
0	1001	25
1	1010	26
2	1011	27
3	1101	29

K.2.5 Adresse de macrobloc (MBA, *macroblock address*) (5/6/7/9/11/12/13/14 bits)

Mot de code dont la longueur dépend de la taille de l'image en cours et de l'activation ou non du mode de rafraîchissement à résolution réduite (voir l'Annexe Q). Ses bits sont la représentation binaire du numéro de macrobloc du premier macrobloc dans la tranche en cours, compté à partir du début de l'image en ordre de balayage, en commençant au numéro de macrobloc 0 du coin supérieur gauche. MBA identifie de manière unique par quel macrobloc de l'image la tranche en cours commence. Le Tableau K.2 indique les longueurs de mots de code pour ce mot de code. Dans le cas des tailles d'image personnalisées, la longueur de champ est donnée par la première donnée dans le tableau qui correspond à un nombre égal ou supérieur de macroblocs et la valeur maximale est le nombre de macroblocs dans l'image en cours, moins un. En mode de rafraîchissement à résolution réduite, la taille d'image pertinente est la taille d'image de rafraîchissement ayant la plus faible résolution plutôt que la taille d'image indiquée dans l'en-tête d'image (voir l'Annexe Q).

Tableau K.2/H.263 – Spécification du paramètre MBA

Format d'image	Valeur par défaut		Mode RRU	
	Valeur maximale	Longueur de champ	Valeur maximale	Longueur de champ
sous-QCIF	47	6	11	5
QCIF	98	7	29	6
CIF	395	9	98	7
4CIF	1583	11	395	9
16CIF	6335	13	1583	11
2048 × 1152	9215	14	2303	12

K.2.6 Bit de prévention d'émulation de tranche 2 (SEPB2, *slice emulation prevention bit 2*) (1 bit)

Un seul bit dont la valeur est toujours "1", et qui est inclus dans certaines conditions afin d'empêcher l'émulation de code de déclenchement. Pour les tranches autres que celle qui suit le code de déclenchement d'image, SEPB2 est inclus seulement si la longueur du champ MBA est supérieure à 11 bits et que CPM = "0" dans l'en-tête d'image, ou si la longueur du champ MBA est supérieure à 9 bits et que CPM = "1" dans l'en-tête d'image. Pour la tranche qui suit immédiatement le code de déclenchement d'image, SEPB2 est inclus seulement si le sous-mode de tranches rectangulaires est utilisé.

K.2.7 Information sur le quantificateur (SQUANT, *quantizer information*) (5 bits)

Mot de code de longueur fixe de cinq bits qui indique le quantificateur QUANT à utiliser pour cette tranche jusqu'à ce qu'il soit mis à jour par un mot DQUANT ultérieur. Ce mot de code est la représentation binaire naturelle des valeurs de QUANT dont la taille de pas est la demie des échelons de quantification et qui se trouvent dans la gamme de 1 à 31. SQUANT est absent pour la tranche qui suit le code de déclenchement d'image.

K.2.8 Indicateur de largeur de tranche dans les macroblocs (SWI, *slice width indication in macroblocks*) (3/4/5/6/7 bits)

Mot de code présent seulement si le sous-mode de tranches rectangulaires est actif et dont la longueur dépend de la taille de l'image en cours et de l'activation ou non du mode de rafraîchissement à résolution réduite, tel que spécifié dans le Tableau K.3. Dans le cas des tailles d'image personnalisées, la longueur de champ est donnée par la taille de format normalisée suivante égale ou plus longue (QCIF, CIF, ...) et sa valeur maximale est le nombre total de macroblocs couvrant la largeur de l'image, moins un. La dernière rangée du tableau indique la longueur de champ pour les tailles d'image supérieures à 16CIF. SWI désigne la largeur de la tranche rectangulaire en cours dont le premier macrobloc (du coin supérieur gauche) est spécifié par MBA. Le calcul de la largeur réelle de la tranche est donné par la formule suivante:

$$\text{largeur réelle de tranche} = \text{SWI} + 1$$

En mode de rafraîchissement à résolution réduite, la taille d'image pertinente est la taille d'image ayant la plus basse résolution pour l'information de rafraîchissement, plutôt que la taille d'image indiquée dans l'en-tête d'image.

Tableau K.3/H.263 – Spécification du paramètre SWI

Format d'image	Valeur par défaut		Mode RRU	
	Valeur maximale	Longueur de champ	Valeur maximale	Longueur de champ
sous-QCIF	7	4	3	3
QCIF	10	4	5	3
CIF	21	5	10	4
4CIF	43	6	21	5
16CIF	87	7	43	6
largeur de 1412...2048 pixels	127	7	63	6

K.2.9 Bit de prévention d'émulation de tranche 3 (SEPB3, *slice emulation prevention bit 3*) (1 bit)

Un seul bit ayant toujours la valeur "1" afin de prévenir l'émulation de code de déclenchement.

ANNEXE L

Spécification d'information d'amélioration complémentaire

L.1 Introduction

La présente annexe décrit le format de l'information d'amélioration complémentaire envoyée dans le champ PSUPP de la couche des images de la présente Recommandation. La capacité d'un décodeur à assurer une ou toutes les possibilités améliorées décrites dans la présente annexe peut être signalée par des moyens externes (par exemple, selon la Recommandation H.245). Les décodeurs qui n'offrent pas les possibilités améliorées peuvent tout simplement ignorer tous les bits d'information PSUPP apparaissant dans le flux binaire. La présence de cette information d'amélioration complémentaire est indiquée dans PEI et un bit PEI additionnel est inséré entre chacun des octets de données PSUPP, tel que décrit aux 5.1.24 et 5.1.25.

Dans la présente annexe, une distinction est faite entre "image décodée" et "image affichée". Aux fins de la présente annexe, le terme "image affichée" correspond à une image dont le format d'image est le format spécifié pour l'image en cours par la couche des images de la syntaxe du flux binaire vidéo. L'"image affichée" est construite de la manière décrite dans la présente annexe à partir de l'image décodée, de l'image affichée antérieure, de l'information d'amélioration complémentaire décrite dans la présente annexe et, dans certains cas, en partie d'une image de fond contrôlée à l'externe.

L.2 Format de PSUPP

Les données PSUPP se composent d'une indication FTYPE de type de fonction de quatre bits, suivie d'une spécification de quatre bits de taille de données de paramètre DSIZE, suivie d'un nombre DSIZE d'octets de données de paramètre de fonction, suivis, en option, d'une autre indication de type de fonction et ainsi de suite. Une valeur d'indication de type de fonction est définie comme un code d'échappement afin de permettre une extensibilité future permettant la définition de plus de quinze fonctions différentes. Un décodeur qui reçoit une indication de type de fonction qu'il ne reconnaît pas peut ignorer les données de paramètre de fonction correspondant à cette fonction puis vérifier s'il y a une indication de type de fonction ultérieure qu'il peut assurer. Les valeurs définies de FTYPE figurent dans le Tableau L.1.

Tableau L.1/H.263 – Valeurs du type de fonction FTYPE

0	réservé
1	ne rien faire
2	demande de gel d'image complète
3	demande de gel d'image partielle
4	demande de gel d'image partielle avec redimensionnement
5	demande de fin de gel d'image partielle
6	étiquette d'instantané d'image complète
7	étiquette d'instantané d'image partielle

Tableau L.1/H.263 – Valeurs du type de fonction FTYPE (*fin*)

8	étiquette de déclenchement de segment de temps vidéo
9	étiquette de fin de segment de temps vidéo
10	étiquette de déclenchement de segment de raffinement progressif
11	étiquette de fin de segment de raffinement progressif
12	information d'incrustation
13	réservé
14	réservé
15	type de fonction étendue

L.3 Ne rien faire

Aucune action n'est requise par la fonction "ne rien faire". Cette fonction est utilisée afin de prévenir l'émulation de code de déclenchement. Dans tous les cas où les cinq derniers bits ou plus de l'octet final de l'octet PSUPP précédent sont tous des zéros et qu'aucune autre demande de fonction PSUPP ne doit être envoyée, la fonction "ne rien faire" doit être insérée dans PSUPP afin de prévenir la possibilité d'émulation de code de déclenchement. La fonction "ne rien faire" peut aussi être envoyée lorsque ce n'est pas requis par une règle exprimée dans l'énoncé précédent. DSIZE aura la valeur zéro pour la fonction "ne rien faire".

L.4 Demande de gel d'image complète

La fonction demande de gel d'image complète indique que le contenu de l'ensemble de l'image précédente doit demeurer inchangé, sans rafraîchissement de l'image affichée au moyen du contenu de l'image décodée en cours. L'image affichée doit alors demeurer inchangée jusqu'à ce que le bit de fin de gel d'image dans le champ PTYPE en cours ou dans un champ PTYPE ultérieur soit mis à 1 ou jusqu'à ce qu'une fin de temporisation se produise, soit le premier de ces événements qui se produit. La demande vient à échéance en raison de la fin de temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes. La fin de la temporisation peut être empêchée par l'émission d'une autre demande de gel d'image complète avant ou au moment de l'expiration de la période de temporisation (p. ex. en répétant la demande dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). DSIZE doit valoir zéro pour la fonction demande de gel d'image complète.

L.5 Demande de gel d'image partielle

La fonction demande de gel d'image partielle indique que le contenu d'une zone rectangulaire spécifique de l'image vidéo affichée précédemment doit demeurer inchangée, sans rafraîchissement de la zone spécifiée de l'image affichée au moyen du contenu de l'image décodée en cours. La zone spécifiée de l'image affichée doit alors demeurer inchangée jusqu'à ce que le bit de fin de gel d'image dans le champ PTYPE en cours ou dans un champ PTYPE ultérieur soit mis à 1, jusqu'à ce qu'une demande de fin du gel d'image partielle touchant la zone spécifiée soit reçue, jusqu'à ce que le format de source spécifié dans un en-tête d'image diffère de celui d'en-têtes d'image précédents ou jusqu'à ce qu'une fin de temporisation se produise, soit le premier de ces événements qui se produit. Tout changement de format de la source d'image doit agir comme indication de fin du gel pour toutes les demandes actives de gel d'image partielle. La demande vient à échéance en raison de la fin de temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes.

La fin de la temporisation peut être empêchée par l'émission d'une demande identique de gel d'image partielle avant ou au moment de l'expiration de la période de temporisation (p. ex. en répétant la demande dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). DSIZE doit avoir une valeur de 4 pour la demande de gel d'image partielle. Les quatre octets de PSUPP suivants contiennent l'emplacement horizontal et vertical du coin supérieur gauche du rectangle gelé de l'image, et la largeur et la hauteur du rectangle, respectivement, chacun étant exprimé au moyen de huit bits, en unités de huit pixels. Par exemple, une zone large de 24 pixels et haute de 16 pixels dans le coin supérieur gauche de l'affichage vidéo est spécifié par les quatre paramètres (0, 0, 3, 2).

L.6 Demande de gel d'image partielle avec redimensionnement

La fonction demande de gel d'image partielle avec redimensionnement indique que le contenu d'une zone rectangulaire spécifique de l'image vidéo affichée précédemment doit être redimensionné de manière à ce qu'il occupe une plus petite partie de l'image vidéo affichée, qui doit alors demeurer inchangée, sans rafraîchissement de la zone spécifiée de l'image affichée au moyen du contenu de l'image décodée en cours. La zone spécifiée de l'image affichée doit alors demeurer inchangée jusqu'à ce que le bit de fin de gel dans le champ PTYPE en cours ou dans un champ PTYPE ultérieur soit mis à 1, jusqu'à ce qu'une demande de fin du gel d'image partielle touchant la zone spécifiée soit reçue, jusqu'à ce que le format de source spécifié dans un en-tête d'image diffère de celui d'en-têtes d'image précédents ou jusqu'à ce qu'une fin de temporisation se produise, soit le premier de ces événements qui se produit. Tout changement de format de la source d'image doit agir comme indication de fin du gel pour toute demande active de gel d'image partielle avec redimensionnement. La demande vient à échéance en raison de la fin de temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes. La fin de la temporisation peut être empêchée par l'émission d'une demande de gel d'image partielle portant sur la zone visée de l'image affichée avant ou au moment de l'expiration de la période de temporisation (p. ex. émission d'une demande de gel d'image partielle dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). DSIZE doit avoir une valeur de 8 pour la demande de gel d'image partielle avec redimensionnement. Les huit octets de données PSUPP qui suivent contiennent 32 bits qui sont utilisés afin de spécifier la zone rectangulaire de la zone touchée de l'image affichée, puis 32 bits utilisés pour spécifier la zone rectangulaire correspondante de la zone touchée de l'image décodée. La largeur et la hauteur de la zone rectangulaire de l'image décodée doivent toutes deux être égales à 2^i fois la largeur et la hauteur spécifiées pour la zone rectangulaire dans l'image affichée, où i est un entier dans la gamme de 1 à 8. L'emplacement et la taille de chacune de ces deux régions sont spécifiés suivant le même format de spécification que pour la fonction de demande de gel d'image partielle.

L.7 Demande de fin de gel d'image partielle

La fonction de demande de fin de gel d'image partielle indique que le contenu d'une zone rectangulaire spécifique de l'image vidéo affichée doit être rafraîchi par les images décodées en cours et ultérieures. DSIZE doit avoir une valeur de 4 pour la demande de fin de gel d'image partielle. Les quatre octets de données PSUPP qui suivent spécifient une région rectangulaire de l'image affichée suivant le même format de spécification que pour la fonction de demande de gel d'image partielle.

L.8 Etiquette d'instantané d'image complète

La fonction étiquette d'instantané d'image complète indique que l'image en cours est étiquetée pour usage externe en tant qu'instantané d'image fixe du contenu vidéo. DSIZE doit avoir une valeur de 4

pour la fonction étiquette d'instantané d'image complète. Les quatre octets de données PSUPP qui suivent spécifient un numéro d'identification d'instantané pour usage externe.

L.9 Etiquette d'instantané d'image partielle

La fonction étiquette d'instantané d'image partielle indique qu'une zone rectangulaire spécifiée de l'image en cours est étiquetée pour usage externe en tant qu'instantané d'image fixe du contenu vidéo. DSIZE doit avoir une valeur de 8 pour la fonction étiquette d'instantané d'image partielle. Les quatre premiers octets de données PSUPP qui suivent spécifient un numéro d'identification d'instantané pour usage externe et les quatre autres octets de données PSUPP qui suivent spécifient une région rectangulaire de l'image décodée suivant le même format de spécification que pour la fonction demande de gel d'image partielle.

L.10 Etiquette de déclenchement de segment de temps vidéo

La fonction étiquette de déclenchement de segment de temps vidéo indique que le commencement d'une sous-séquence de données vidéo spécifiée est étiquetée comme section utile de contenu vidéo pour usage externe, à partir de l'image en cours. La sous-séquence de données vidéo étiquetée se poursuivra jusqu'à son arrêt par la réception d'une fonction correspondante étiquette de fin de segment de temps vidéo ou jusqu'à la fin de la temporisation, soit le premier de ces événements qui se produit. La sous-séquence étiquetée se terminera en raison de la temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes. La fin de la temporisation peut être empêchée par l'émission d'une étiquette de déclenchement de segment de temps vidéo identique avant ou au moment de l'expiration de la période de temporisation (p. ex. par répétition de la fonction étiquette de déclenchement de segment de temps vidéo dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). DSIZE doit avoir une valeur de 4 pour la fonction étiquette de déclenchement de segment de temps vidéo. Les quatre octets de données PSUPP qui suivent spécifient un numéro d'identification de segment de temps vidéo pour usage externe.

L.11 Etiquette de fin de segment de temps vidéo

La fonction étiquette de fin de segment de temps vidéo indique que la fin d'une sous-séquence de données vidéo spécifiée est étiquetée comme une section utile de contenu vidéo pour usage externe, se terminant à l'image précédente. DSIZE doit avoir une valeur de 4 pour la fonction étiquette de déclenchement de segment de temps vidéo. Les quatre octets de données PSUPP qui suivent spécifient un numéro d'identification de segment de temps vidéo pour usage externe.

L.12 Etiquette de déclenchement de segment de raffinement progressif

La fonction étiquette de déclenchement de segment de raffinement progressif indique le commencement d'une sous-séquence spécifiée de données vidéo qui est étiquetée comme l'image en cours suivie d'une séquence de zéros ou d'autres images de raffinement de la qualité de l'image en cours, plutôt que comme la représentation d'une scène en mouvement continu. La sous-séquence de données vidéo étiquetée se poursuivra jusqu'à son arrêt par la réception d'une fonction étiquette de fin de segment de raffinement progressif correspondant ou jusqu'à la fin de la temporisation, soit le premier de ces événements qui se produit. La sous-séquence étiquetée se terminera en raison de la temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes. La fin de la temporisation peut être empêchée par l'émission d'une étiquette de déclenchement de segment de raffinement progressif identique avant ou au moment de l'expiration de la période de temporisation (p. ex. par répétition de la fonction étiquette de déclenchement de segment de

raffinement progressif dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). DSIZE doit avoir une valeur de 4 pour la fonction étiquette de déclenchement de segment de temps vidéo. Les quatre octets de données PSUPP qui suivent spécifient un numéro d'identification de segment de raffinement progressif pour usage externe.

L.13 Etiquette de fin de segment de raffinement progressif

La fonction étiquette de fin de segment de raffinement progressif indique la fin d'une sous-séquence spécifiée de données vidéo qui est étiquetée comme une image initiale suivie d'une séquence de zéros ou comme plusieurs images de raffinement de la qualité de l'image initiale et se terminant par l'image précédente. DSIZE doit avoir une valeur de 4 pour la fonction de déclenchement de segment de temps vidéo. Les quatre octets de données PSUPP qui suivent spécifient un numéro d'identification de raffinement progressif pour usage externe.

L.14 Information d'incrustation

La fonction information d'incrustation (CKIF, *chroma keying information function*) indique que la technique d'incrustation est utilisée afin de représenter des pixels "transparents" et "semi-transparentes" dans les images vidéo décodées. Lorsqu'ils sont présentés à l'affichage, les pixels "transparentes" ne sont pas affichés. A leur place, une image de fond qui est soit une image de référence antérieure ou une image commandée de l'extérieur est affichée. Les pixels semi-transparentes sont affichés en mélangeant la valeur de pixel dans l'image en cours avec la valeur correspondante dans l'image de fond. Un octet est utilisé pour indiquer la valeur de couleur de l'incrustation pour chaque composante (Y , C_B ou C_R) utilisée pour l'incrustation. Les pixels qui seront "semi-transparentes" seront représentés au moyen de deux valeurs seuils, notées T_1 et T_2 . Mettons que α représente la transparence d'un pixel; $\alpha = 255$ indique que le pixel est opaque et $\alpha = 0$ indique que le pixel est transparent. Pour les autres valeurs de α , la valeur résultante d'un pixel doit être la combinaison pondérée de la valeur du pixel dans l'image en cours et de la valeur du pixel de l'image de fond (spécifiée par des moyens externes). Les valeurs de α peuvent être utilisées pour former une image nommée "carte alpha". La valeur résultante de chaque composante peut être:

$$[\alpha \cdot X + (255 - \alpha) \cdot Z] / 255$$

où X est la valeur de composante du pixel décodé (pour Y , C_B ou C_R) et Z est la valeur de composante du pixel correspondant dans l'image de fond.

La valeur α peut être calculée comme suit. La distance entre la couleur du pixel et la valeur de couleur d'incrustation est d'abord calculée:

$$d = A_Y (X_Y - K_Y)^2 + A_B (X_B - K_B)^2 + A_R (X_R - K_R)^2$$

dans laquelle X_Y , X_B et X_R sont les valeurs Y , C_B et C_R de couleur du pixel décodé, K_Y , K_B et K_R sont les paramètres correspondants de couleur d'incrustation et A_Y , A_B et A_R sont les bits de fanion d'incrustation qui indiquent les composantes de couleur qui serviront de clés. Lorsque la distance d est calculée, la valeur α peut être calculée tel que spécifié dans le pseudo-code suivant:

```
for each pixel
  if (d < T1) then  $\alpha = 0$ ;
  else if (d > T2) then  $\alpha = 255$ ;
  else  $\alpha = [255 \cdot (d - T_1)] / (T_2 - T_1)$ 
```

La méthode précise de réalisation de l'incrustation dans le décodeur n'est toutefois pas précisée dans la présente Recommandation, car la spécification normative de cette méthode n'est pas nécessaire

pour l'interfonctionnement. Le processus est fourni à des fins d'illustration afin d'établir l'interprétation voulue des paramètres de données.

Comme la valeur α dérivée est simplement une fonction de X_Y , X_B et X_R , une table de consultation (LUT, *look-up table*) peut être élaborée afin de réaliser l'opération ci-dessus. Une telle table comportera $2^{8 \times N}$ entrées correspondant à toutes les valeurs des pixels, où N est le nombre de composantes de couleur utilisées comme clés. Chacune des entrées de la table contiendra alors la valeur α correspondante.

DSIZE doit être dans la gamme de 1 à 9 (inclusivement) pour l'information d'incrustation, selon la quantité de données envoyées avec la fonction CKIF. Il ne peut y avoir plus d'une fonction CKIF envoyée avec une image.

Le premier octet qui suit l'octet DSIZE contiendra l'ordre de représentation de l'image en cours – les flux ayant un ordre de représentation inférieur sont supposés constituer l'image de fond des flux dont l'ordre de représentation est supérieur.

Si DSIZE est plus grand que un, l'octet suivant l'octet d'ordre de représentation doit être utilisé pour envoyer six bits-fanions définis comme suit:

- bit 1: A_Y : bit-fanion signalant la présence d'un paramètre d'incrustation K_Y pour les valeurs de luminance Y
- bit 2: A_B : bit-fanion signalant la présence d'un paramètre d'incrustation K_B pour les valeurs de chrominance C_B
- bit 3: A_R : bit-fanion signalant la présence d'un paramètre d'incrustation K_R pour les valeurs de chrominance C_R
- bit 4: A_1 : bit-fanion signalant la présence d'un paramètre de seuil T_1 pour la transparence
- bit 5: A_2 : bit-fanion signalant la présence d'un paramètre de seuil T_2 pour l'opacité
- bit 6: RPB: bit-fanion signalant l'utilisation de l'image de référence comme image de fond
- bit 7: réservé
- bit 8: réservé

DSIZE doit avoir une valeur de 1 ou une valeur de 2 plus le nombre de bits-fanions parmi A_Y , A_B et A_R qui sont mis à 1, plus 2 fois le nombre de bits parmi A_1 et A_2 qui sont mis à 1. Si la valeur de DSIZE est supérieure à 1, un octet additionnel doit être envoyé pour spécifier la valeur de chaque composante de couleur pour chacun des bits-fanions A_Y , A_B et A_R qui sont mis à un et deux octets additionnels doivent être envoyés afin de spécifier chacune des valeurs de seuil indiquées par un fanion parmi T_1 et T_2 . Ces octets doivent suivre dans le même ordre que les bits-fanions.

Si la valeur de DSIZE est égale à 1 ou si les trois bits-fanions de couleur d'incrustation A_Y , A_B et A_R ont une valeur de zéro, les bits-fanions d'incrustation A_Y , A_B et A_R et les couleurs d'incrustation K_Y , K_B et K_R qui ont été utilisés pour l'image incrustée précédente doivent aussi être utilisés pour l'image en cours. Si aucune valeur précédente n'a été envoyée pour la séquence vidéo, les bits-fanions d'incrustation implicites $A_Y = 1$, $A_B = 1$ et $A_R = 1$ et les couleurs d'incrustation implicites $K_Y = 50$, $K_B = 220$ et $K_R = 100$ doivent être utilisés comme valeurs précédentes.

Si la valeur de DSIZE est égale à 1 ou si les deux bits-fanions de seuil d'incrustation A_1 et A_2 ont une valeur de zéro, les valeurs de seuil d'incrustation T_1 et T_2 qui ont été utilisées pour l'image incrustée précédente doivent aussi être utilisés pour l'image en cours. Si aucune valeur précédente n'a été envoyée pour la séquence vidéo, les valeurs-seuils implicites $T_1 = 48$ et $T_2 = 75$ devraient être utilisées comme valeurs précédentes.

Dans la partie de l'image où les pixels sont "semi-transparents" (par exemple, lorsque $T_1 < d < T_2$), les pixels décodés contiennent habituellement la couleur d'incrustation dans les composantes où les

incrustations sont utilisées. Cela peut faire apparaître des artefacts de couleur. Afin de régler ce problème, ces valeurs de pixels peuvent être ajustées avant qu'elles soient mélangées avec la couleur du fond. Un tel processus de correction peut être appliqué aux composantes de couleur utilisées dans l'opération d'incrustation et signalées par des bits-fanions. Ce processus est le suivant:

$$X' = K + (T_2 / d)(X - K)$$

où X est la valeur de composante décodée initiale du pixel et X' est la valeur corrigée.

Comme les valeurs ajustées de pixel X'_y , X'_b et X'_r sont des fonctions de X_y , X_b et X_r , la correction des couleurs peut être réalisée au moyen d'une table de consultation. Cette table de consultation contiendrait 2^{8N} entrées correspondant à toutes les valeurs de pixel, où N est le nombre de composantes de couleur utilisées comme clés. Chaque entrée contiendrait alors les valeurs corrigées correspondantes.

Si le bit-fanion de fond d'image de référence (RPB, *reference picture background*) est mis à "1", cela indique que l'image de référence qui précède dans le temps (avant qu'aucun rééchantillonnage tel que précisé à l'Annexe P n'ait été effectué pour l'image en cours) doit être conservée comme fond (opaque) de l'image en cours et de toutes les images incrustées ultérieures, jusqu'à ce qu'elle soit remplacée par l'arrivée d'une autre image dont le fanion RPB est mis à "1". S'il n'y a pas d'image de référence précédente dans le temps pour l'image en cours (p. ex. si l'image en cours est à codage INTRA ou EI), l'image à laquelle renvoie le bit-fanion RPB est l'image qui aurait normalement été l'image de référence si l'image en cours était de type INTER ou EP, selon le cas. Si le bit-fanion RPB est mis à "0", cela signifie que le fond doit demeurer contrôlé tel que précédemment (soit par contrôle externe, soit au moyen d'une image de référence qui a été stockée précédemment au moment de la réception d'une image précédente pour laquelle RPB était mis à "1").

L'utilisation de l'incrustation qui est invoquée par l'émission de la fonction d'information d'incrustation doit commencer à l'image en cours et se poursuivre jusqu'à une image ultérieure de type INTRA ou EI, ou jusqu'à l'expiration d'une période de temporisation, soit le premier de ces événements qui se produit. L'utilisation de l'incrustation se terminera en raison de la temporisation au bout de cinq secondes ou de cinq images, soit la plus longue de ces deux périodes. La fin de la temporisation peut être empêchée par l'émission d'une fonction d'information d'incrustation identique ou au moment de l'expiration de la période de temporisation (p. ex. par répétition de la fonction d'information d'incrustation dans l'en-tête de la première image avec une référence temporelle indiquant un intervalle de temps supérieur ou égal à cinq secondes depuis l'émission ou dans l'en-tête de la cinquième image après l'émission). Le codeur doit envoyer suffisamment d'information avec la fonction d'information d'incrustation pour permettre une resynchronisation complète à chaque image de type INTRA ou EI et dans chaque intervalle de temporisation (il ne doit pas se fonder sur des valeurs en mémoire ou des valeurs implicites des couleurs ou seuils d'incrustation).

L.15 Type de fonction étendue

L'indication de type de fonction étendue sert à indiquer que l'octet PSUPP suivant contient une fonction étendue. L'utilisation des fonctions étendues est réservée afin que l'UIT ait la possibilité future de définir un grand nombre de fonctions de données PSUPP à compatibilité amont. DSIZE doit avoir une valeur de zéro pour l'indication de type de fonction étendue. Afin de permettre la compatibilité amont de l'utilisation future de l'indication de type de fonction étendue, les décodeurs devront traiter le deuxième ensemble de quatre bits de l'octet qui suit l'indication de type de fonction étendue comme étant une valeur de DSIZE qui indique le nombre d'octets PSUPP ultérieurs qui doivent être sautés pour les données de paramètre de fonction étendue, qui peuvent être suivies d'indications FTYPE additionnelles.

ANNEXE M

Mode de codage par trames PB amélioré

M.1 Introduction

La présente annexe décrit un mode facultatif de codage par trames PB amélioré de la présente Recommandation. Il est donc considéré comme avantageux d'utiliser le mode de codage par trames PB amélioré décrit dans la présente annexe au lieu du mode de codage par trames PB défini à l'Annexe G. La possibilité de ce mode est signalée par des moyens externes (par exemple, selon la Recommandation H.245). L'utilisation de ce mode est signalée dans le champ PLUSPTYPE de l'en-tête d'image.

La plupart des éléments de ce mode sont semblables à ceux du mode de codage par trames PB défini à l'Annexe G.

Afin d'éviter la confusion avec les images B définies à l'Annexe O, les termes image B, macrobloc B et bloc B ne seront pas utilisés dans la présente annexe. Nous utiliserons plutôt la notation B_{PB} pour représenter la "partie B" d'une trame PB améliorée. Lorsque des renvois sont faits à l'Annexe G, les expressions image B et bloc B seront représentées par les termes image B_{PB} et bloc B_{PB} .

La principale différence entre le mode de codage par trames PB et le mode de codage par trames PB amélioré réside dans le fait que dans le mode de codage par trames PB amélioré, le macrobloc B_{PB} dispose d'un mode de prédiction anticipée et différée qui s'ajoute au mode de prédiction bidirectionnelle. Dans la présente annexe, le sigle MVDB (lorsqu'il est présent) désigne un vecteur cinétique anticipé. (Il est à noter que dans l'Annexe G, le sigle MVDB était utilisé pour désigner une amélioration des vecteurs anticipés et différés réduits pour la prédiction bidirectionnelle, plutôt qu'un vecteur cinétique différé distinct.)

La présente annexe relève toutes les différences par rapport à l'Annexe G. L'absence d'indication signifie que la procédure de l'Annexe G est utilisée.

M.2 Modes de prédiction par macroblocs B_{PB}

Il existe trois manières de coder un macrobloc B_{PB} . Les différents modes de codage sont signalés par le paramètre MODB. Les modes de codage des macroblocs B_{PB} sont:

M.2.1 Prédiction bidirectionnelle

Dans le mode de prédiction bidirectionnelle, la prédiction utilise les images de référence précédant et suivant l'image B_{PB} (dans le cas d'une séquence de trames PB améliorées, cela signifie la partie image P de la trame PB améliorée précédant dans le temps et la partie image P de la trame PB améliorée en cours). Cette prédiction équivaut à la prédiction définie à l'Annexe G lorsque $MV_D = 0$. Il est à noter

que, dans ce mode (et seulement dans ce mode), les données de vecteur cinétique (MVD) du macrobloc PB doivent être incluses même si le macrobloc P est à codage INTRA. (Remarquer la différence entre MVD – données de vecteur cinétique – et MV_D – vecteur delta – défini à l'Annexe G.)

M.2.2 Prédiction anticipée

En mode de prédiction anticipée, les données vectorielles contenues dans MVDB sont utilisées pour effectuer la prédiction anticipée à partir de l'image de référence précédente (une image INTRA ou INTER, ou la partie image P d'une trame PB ou PB améliorée). Cela signifie qu'il n'y a toujours qu'un vecteur 16×16 pour le macrobloc B_{PB} dans ce mode de prédiction.

Un prédicteur simple est utilisé pour le codage du vecteur cinétique anticipé. La règle s'appliquant à ce prédicteur est que si le macrobloc en cours n'est pas à la limite gauche de cette image ou tranche et que le macrobloc à la gauche comporte un vecteur cinétique anticipé, le prédicteur du vecteur cinétique anticipé pour le macrobloc en cours est réglé à la valeur du vecteur cinétique anticipé du bloc qui se trouve à gauche; sinon le prédicteur est réglé à zéro. La différence entre le prédicteur et le vecteur cinétique voulu est alors codée en VLC de la même manière que les données vectorielles utilisées pour l'image P (MVD).

En ce qui concerne les vecteurs cinétiques dépassant les limites d'image définis au D.1, la technique décrite s'applique aussi au vecteur B_{PB} anticipé si cette caractéristique est utilisée (ce principe s'applique au mode de prédiction anticipée aussi bien qu'au mode de prédiction bidirectionnelle).

M.2.3 Prédiction différée

Dans le mode de prédiction différée, la prédiction du macrobloc B_{PB} est identique à celle de P_{REC} (définie au G.5). Aucune donnée de vecteur cinétique n'est utilisée pour la prédiction différée.

M.3 Calcul des vecteurs pour la prédiction bidirectionnelle du macrobloc B

Si la prédiction bidirectionnelle est utilisée, les vecteurs anticipé et différé pondérés sont calculés de la manière décrite à l'Annexe G lorsque $MV_D = 0$.

M.4 Tableau MODB

Le Tableau M.1 donne une nouvelle description de MODB (qui remplace le Tableau 11). Ce tableau montre les modes de codage possibles pour un bloc B_{PB} .

Tableau M.1/H.263 – Tableau MODB pour le mode de codage par trames PB amélioré

Index	CBPB	MVDB	Nombre de bits	Code	Mode de codage
0			1	0	prédiction bidirectionnelle
1	x		2	10	prédiction bidirectionnelle
2		x	3	110	prédiction anticipée
3	x	x	4	1110	prédiction anticipée
4			5	11110	prédiction différée
5	x		5	11111	prédiction différée

NOTE – Le symbole "x" indique que l'élément syntaxique associé est présent.

Mode de sélection de l'image de référence

N.1 Introduction

La présente annexe décrit le mode facultatif de sélection d'image de référence de la présente Recommandation qui fait appel à une méthode de prédiction intertrame modifiée nommée "NEWPRED". La possibilité de ce mode H.263 est signalée par des moyens externes (par exemple, selon la Recommandation H.245). La quantité de mémoire-image supplémentaire offerte par le décodeur peut aussi être signalée par des moyens externes afin d'aider à la gestion de la mémoire au codeur. Ce mode peut faire appel à des messages transmis d'un décodeur à un codeur par un canal de retour afin d'informer le codeur des parties de quelles images ont été décodées correctement au décodeur. L'utilisation de ce mode est indiquée dans le champ PLUSTYPE de l'en-tête d'image. Ce mode comporte deux commutateurs de mode avec canal de retour qui indiquent si un canal de retour est utilisé ainsi que le genre de messages qui sont renvoyés par ce canal de retour à partir du décodeur; il comporte aussi un autre sous-mode défini comme le canal pour les messages transmis par canal de retour.

Les deux commutateurs du mode avec canal de retour déterminent les genres de messages transmis par le canal de retour, précisant si des messages ACK (messages d'accusé de réception) ou NACK (messages d'accusé de réception négatif) sont envoyés. Ensemble, les deux commutateurs définissent quatre modes de fonctionnement de base:

- 1) NEITHER: aucune donnée n'est renvoyée par canal de retour du décodeur au codeur;
- 2) ACK: le décodeur renvoie seulement des messages d'accusé de réception;
- 3) NACK: le décodeur renvoie seulement des messages d'accusé de réception négatif;
- 4) ACK+NACK: le décodeur renvoie des messages d'accusé de réception et des messages d'accusé de réception négatif.

Le type de message, parmi les types décrits ci-haut, est indiqué dans l'en-tête d'image.

Il existe aussi deux modes de fonctionnement en ce qui concerne le canal de transmission des messages transmis par canal de retour:

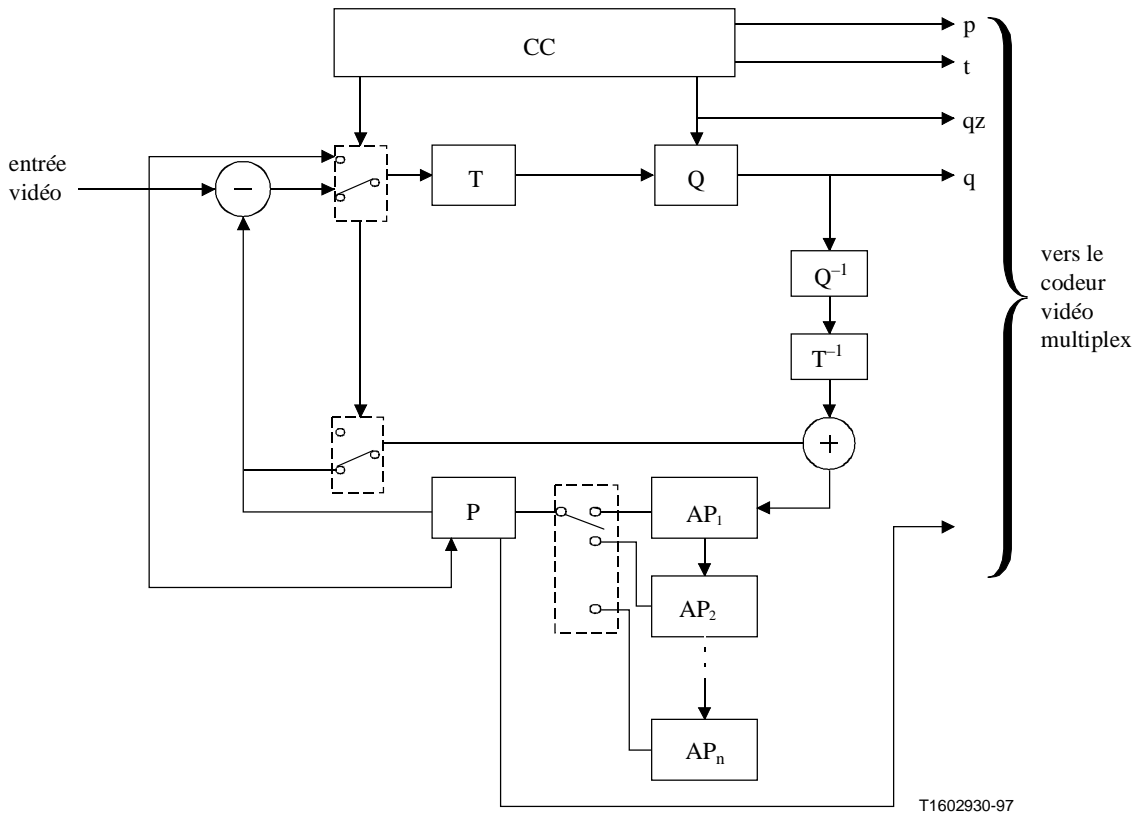
- 1) mode de canal logique distinct: ce mode de fonctionnement transmet les données par canal de retour par l'intermédiaire d'un canal logique distinct dans la couche multiplex du système;
- 2) mode vidéomux: ce mode de fonctionnement transmet par canal de retour les données relatives au signal vidéo reçu dans les données vidéo avant d'un flux vidéo de données codées.

La présente annexe précise une syntaxe s'appliquant aux messages transmis par canal de retour ainsi qu'aux données de canal vers l'avant.

N.2 Algorithme de codage de source vidéo

Le codeur de source de ce mode est montré sous une forme généralisée à la Figure N.1. Cette figure montre une structure qui utilise un certain nombre de mémoires d'image. Le codeur de source peut sélectionner une des mémoires d'image afin d'éliminer la propagation de l'erreur temporelle causée par le codage intertrame. Le mode de décodage par segments indépendants (voir l'Annexe R), qui traite les limites des groupes de blocs dont l'en-tête n'est pas vide ou des tranches comme des limites d'image, peut être utilisé afin d'éviter la propagation des erreurs dues à la compensation de mouvement sur les limites des groupes de blocs ou des tranches lorsque ce mode est appliqué à une unité plus petite que l'image, comme un groupe de blocs ou une tranche. L'information indiquant

quelle est l'image sélectionnée pour la prédiction est comprise dans le flux binaire codé. La stratégie qu'applique le codeur pour sélectionner l'image qui sera utilisée aux fins de la prédiction dépasse l'objet de la présente Recommandation.



- T transformation
- Q quantificateur
- P mémoire d'image avec retard variable à compensation de mouvement
- AP mémoire d'image additionnelle
- CC commande de codage
- p fanion INTRA/INTER
- t fanion d'émission ou non émission
- qz indication de quantificateur
- q index de quantification pour les coefficients de transformation
- v vecteur cinétique

Figure N.1/H.263 – Codeur de source pour NEWPRED

N.3 Canal pour les messages transmis par canal de retour

Ce mode comporte deux modes de fonctionnement sur le plan du genre de canal utilisé pour la transmission des messages transmis par canal de retour. Un de ces modes est le mode de canal logique distinct et l'autre est le mode vidéomux. Le mode de canal logique distinct est un mode préféré et il assure la transmission des messages transmis par canal de retour définis au N.4.2, au moyen du canal logique réservé. Le mode vidéomux est préparé à l'intention des systèmes qui ne peuvent établir un canal distinct supplémentaire pour les messages transmis par canal de retour, en raison de restrictions s'appliquant au nombre de combinaisons de canaux. Le mode vidéomux livre les messages transmis par canal de retour au moyen du même canal logique que les données vidéo avant qui sont transmises dans le sens opposé.

N.3.1 Mode de canal logique distinct

Le mode de canal logique distinct livre les messages transmis par canal de retour par l'intermédiaire d'un canal logique réservé ouvert seulement pour les messages transmis par canal de retour. Le mécanisme d'association avec le canal vers l'avant qui livre les données vidéo est établi par des moyens externes (par exemple, selon la Recommandation H.245). Le fonctionnement avec un canal logique distinct exige un mécanisme de verrouillage de trames externe pour assurer la synchronisation des messages dans le canal de retour, car la syntaxe s'appliquant au canal de retour définie dans la présente Recommandation ne contient pas de mots de fanion de synchronisation.

N.3.2 Mode vidéomux

Le mode vidéomux livre les messages transmis par canal de retour par l'intermédiaire du même canal logique que les données vidéo avant transmises dans le sens opposé. La syntaxe du flux binaire multiplexé est décrite au N.4.1. Les messages transmis par canal de retour peuvent être insérés au moyen d'indications de messages transmis par canal de retour (BCI) dans l'en-tête de groupe de blocs ou de tranche.

N.4 Syntaxe

N.4.1 Canal vers l'avant

La syntaxe s'appliquant aux données transmises par le canal vers l'avant qui transmet le signal vidéo comprimé est modifiée seulement dans la couche des groupes de blocs (GOB) ou des tranches.

La syntaxe de la couche des groupes de blocs est illustrée à la Figure N.2. Les champs TRI, TR, TRPI, TRP, BCI et BCM sont ajoutés à la Figure 9.

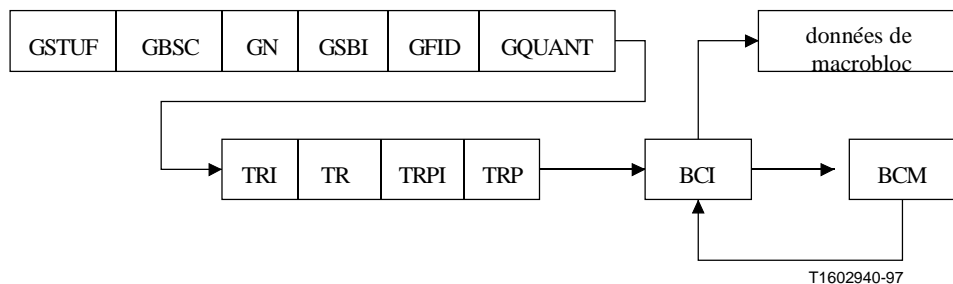


Figure N.2/H.263 – Structure de la couche des groupes de blocs pour NEWPRED

Lorsque le mode structuré par tranches facultatif (voir l'Annexe K) est utilisé, la syntaxe de la couche des tranches est modifiée de la même manière que la couche des groupes de blocs. La syntaxe est illustrée à la Figure N.3.

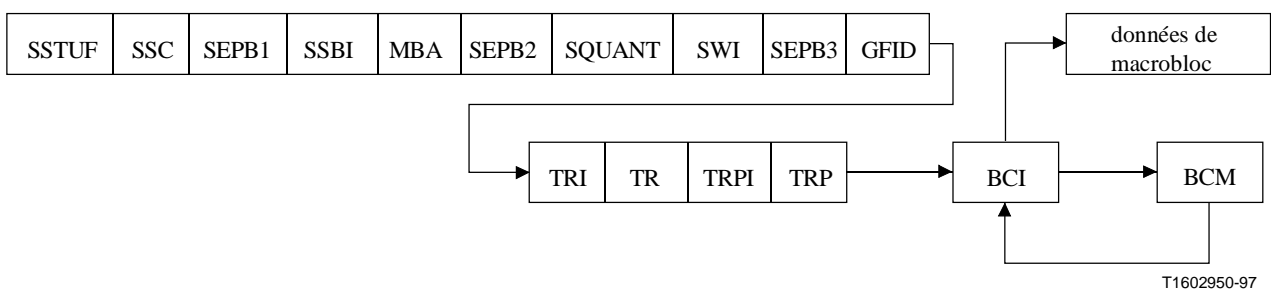


Figure N.3/H.263 – Structure de la couche des tranches pour NEWPRED

N.4.1.1 Indicateur de référence temporelle (TRI, *temporal reference indicator*) (1 bit)

L'indicateur TRI indique si le champ TR suivant est présent ou non.

0: le champ TR n'est pas présent.

1: le champ TR est présent.

N.4.1.2 Référence temporelle (TR, *temporal reference*) (8/10 bits)

Lorsqu'il est présent, le champ TR est un nombre de huit bits à moins qu'une fréquence de base d'image soit utilisée, auquel cas ce champ comporte un nombre de dix bits composé par concaténation des champs ETR et TR de l'en-tête d'image.

N.4.1.3 Indicateur de référence temporelle pour la prédiction (TRPI, *temporal reference for prediction indicator*) (1 bit)

L'indicateur TRPI indique si le champ TRP suivant est présent ou non.

0: le champ TRP n'est pas présent.

1: le champ TRP est présent.

TRPI doit avoir une valeur nulle dans tous les cas où l'image est une image I ou EI.

N.4.1.4 Référence temporelle pour la prédiction (TRP, *temporal reference for prediction*) (10 bits)

Lorsqu'il est présent (tel qu'indiqué par TRPI), le champ TRP indique la référence temporelle utilisée pour la prédiction du codage, sauf dans le cas des images B et de la partie image B d'une trame PB améliorée. Pour les images B ou la partie image B d'une trame PB améliorée, l'image ayant la référence temporelle TRP est utilisée pour la prédiction dans le sens avant. (La prédiction dans le sens inverse dans le temps utilise toujours l'image qui suit immédiatement dans le temps.) TRP est un nombre de dix bits. Si une fréquence de base d'image personnalisée n'est pas utilisée pour l'image de référence, les deux bits de plus fort poids de TR ont une valeur nulle et les bits de plus faible poids contiennent la référence temporelle de huit bits qui se trouve dans l'en-tête d'image de l'image de référence. Si une fréquence de base d'image personnalisée est utilisée pour l'image de référence, TRP est un nombre de dix bits composé par concaténation des champs ETR et TR de l'en-tête de l'image de référence.

Lorsque le champ TRP n'est pas présent, c'est la plus récente image de référence principale (temporellement antérieure) qui doit être utilisée pour la prédiction, comme en dehors du mode de sélection d'image de référence. Le champ TRP est valide jusqu'au prochain champ PSC, GSC ou SSC.

N.4.1.5 Indication de message transmis par canal de retour (BCI, *back-channel message indication*) (longueur variable)

Ce champ contient un ou deux bits; lorsqu'ils sont mis à "1", ils signalent la présence du champ de message vidéo transmis par canal de retour (BCM) suivant. Sinon, la valeur du champ est "01", ce qui indique l'absence ou la fin du champ de message vidéo transmis par canal de retour. Les combinaisons de champs BCM et BCI peuvent être absentes, et elles peuvent être répétées lorsqu'elles sont présentes. BCI doit toujours être mis à "01" si le mode vidéomux n'est pas utilisé.

N.4.1.6 Message transmis par canal de retour (BCM, *back-channel message*) (longueur variable)

Message transmis par canal de retour ayant une syntaxe définie au N.4.2, et qui est présent seulement si le champ BCI précédent est mis à "1".

N.4.2 Syntaxe de message transmis par canal de retour (BCM)

La syntaxe s'appliquant au canal de retour qui transmet les messages d'accusé de réception et les messages d'accusé de réception négatif est illustrée à la Figure N.4. Ce message est renvoyé d'un décodeur à un codeur afin d'indiquer si des données transmises par un canal vers l'avant ont été décodées correctement ou non.

BT	URF	TR	ELNUMI	ELNUM	BCPM	BSBI	BEPB1	GN/MBA	BEPB2	RTR	BSTUF
----	-----	----	--------	-------	------	------	-------	--------	-------	-----	-------

Figure N.4/H.263 – Structure de la syntaxe d'un message transmis par canal de retour (BCM) pour NEWPRED

N.4.2.1 Type de message transmis par canal de retour (BT, *back-channel message type*) (2 bits)

Le champ type de message transmis par canal de retour indique si la partie correspondante du message codé est décodée correctement ou non. Le type de message requis pour le codeur est indiqué dans l'en-tête d'image du canal vers l'avant.

00: réservé pour usage futur.

01: réservé pour usage futur.

10: NACK. Cela signale le décodage erroné de la partie correspondante des données transmises par le canal vers l'avant.

11: ACK. Cela signale le décodage exact de la partie correspondante des données transmises par le canal vers l'avant.

N.4.2.2 Fanion de non-fiabilité (URF, *unreliable flag*) (1 bit)

Le fanion de non-fiabilité est mis à 1 lorsque le décodeur ne dispose pas d'une valeur fiable de TR ou de GN/MBA. (Lorsque la valeur de BT est NACK, le décodeur peut ne pas disposer d'une référence temporelle fiable.)

0: fiable.

1: non fiable.

N.4.2.3 Référence temporelle (TR, *temporal reference*) (10 bits)

Le champ TR contient l'information de référence temporelle correspondant au segment d'image vidéo pour lequel une indication ACK/NACK figure dans le message transmis par canal de retour.

NOTE – la signification du terme "segment d'image vidéo" tel qu'il est employé dans la présente annexe est définie à l'Annexe R. Si une fréquence de base d'image personnalisée n'est pas utilisée pour l'image de référence, les deux bits de plus fort poids du champ TR ont une valeur nulle et les bits de plus faible poids contiennent la référence temporelle de huit bits qui se trouve dans l'en-tête d'image de l'image de référence. Si une fréquence de base d'image personnalisée est utilisée pour l'image de référence, TR est un nombre de dix bits composé par concaténation des champs ETR et TR de l'en-tête de l'image de référence.

N.4.2.4 Indication de numéro de couche d'amélioration (ELNUMI, *enhancement layer number indication*) (1 bit)

Le champ indication de numéro de couche d'amélioration a la valeur "0" à moins que le mode d'échelonnabilité temporelle, SNR et spatiale (Annexe O) soit utilisé dans les données transmises par le canal vers l'avant et que certaines couches d'amélioration du canal vers l'avant soient combinées dans un canal logique et que le message transmis par canal de retour renvoie à une couche

d'amélioration (plutôt qu'à la couche de base), auquel cas l'indication de numéro de couche d'amélioration doit valoir "1".

N.4.2.5 Numéro de couche d'amélioration (ELNUM, *enhancement layer number*) (4 bits)

Le champ numéro de couche d'amélioration est présent si et seulement si la valeur de ELNUMI est "1", auquel cas il contient le numéro de couche de la couche d'amélioration à laquelle il est fait référence dans le message transmis par canal de retour.

N.4.2.6 BCPM (1 bit)

BCPM a la valeur "0" à moins que le mode CPM soit utilisé dans les données transmises par le canal vers l'avant, auquel cas il a la valeur "1". Si BCPM a la valeur "1", cela signifie que BSBI est présent.

N.4.2.7 Indicateur de sous-flux binaire de canal de retour (BSBI, *back-channel sub-bitstream indicator*) (2 bits)

Mot de code de longueur fixe de 2 bits qui est seulement présent si BCPM a la valeur "1". Le BSBI est la représentation binaire naturelle du numéro de sous-flux binaire approprié dans les données transmises par le canal vers l'avant, pour lesquelles une indication ACK/NACK figure dans le message transmis par canal de retour tel que décrit au 5.2.4 et dans l'Annexe C.

N.4.2.8 Bit 1 de prévention d'émulation de canal de retour (BEPB1, *back-channel emulation prevention bit 1*) (1 bit)

Champ qui est présent si et seulement si le mode vidéomux est utilisé. La valeur de ce champ est toujours "1" pour prévenir un code de déclenchement d'émulation.

N.4.2.9 Numéro de groupe de bloc/adresse de macrobloc (GN/MBA, *GOB number/macroblock address*) (5/6/7/9/11/12/13/14 bits)

Un numéro de groupe de blocs ou une adresse de macrobloc est présent dans ce champ. Si le mode facultatif structuré par tranches (voir l'Annexe K) n'est pas utilisé, ce champ contient le numéro de groupe de blocs du commencement du segment d'image vidéo pour lequel une indication NACK/ACK figure dans le message transmis par canal de retour. Si le mode facultatif structuré par tranches est utilisé, ce champ contient l'adresse de macrobloc du commencement de la tranche pour laquelle une indication NACK/ACK figure dans le message transmis par canal de retour. La longueur de ce champ est la longueur spécifiée dans une autre partie de la présente Recommandation pour GN ou MBA.

N.4.2.10 Bit 2 de prévention d'émulation de canal de retour (BEPB2, *back-channel emulation prevention bit 2*) (1 bit)

Champ qui est présent si et seulement si le mode vidéomux est utilisé. La valeur de ce champ est toujours "1" pour prévenir un code de déclenchement d'émulation.

N.4.2.11 Référence temporelle demandée (RTR, *requested temporal reference*) (10 bits)

Le champ référence temporelle demandée est présent seulement si BT a la valeur NACK. RTR indique la référence temporelle demandée pour le groupe de blocs ou la tranche associés au message NACK. Il s'agit habituellement de la référence temporelle du dernier segment d'image vidéo décodé correctement de la position correspondante au décodeur. Si une fréquence de base d'image personnalisée n'est pas utilisée pour l'image de référence demandée, les deux bits de plus fort poids de RTR ont une valeur nulle et les bits de plus faible poids contiennent la référence temporelle de huit bits qui se trouve dans l'en-tête d'image de l'image de référence demandée. Si une fréquence de base d'image personnalisée est utilisée pour l'image de référence demandée, RTR est un nombre de

dix bits composé par concaténation des champs ETR et TR de l'en-tête de l'image de référence demandée.

N.4.2.12 Bourrage (BSTUF, *stuffing*) (longueur variable)

Ce champ est présent si et seulement si le mode de canal logique distinct est utilisé et que le message transmis par canal de retour est le dernier dans une trame extérieure. BSTUF se compose d'un mot de code de longueur variable ne comportant aucun bit ou comportant plusieurs bits de valeur "0". Ce champ est présent seulement à la fin d'une trame extérieure.

N.5 Processus de décodage

Le décodeur de ce mode peut avoir besoin d'un nombre additionnel de mémoires d'image afin de stocker les signaux vidéo décodés correctement ainsi que leur information de référence temporelle (TR). Le décodeur utilise l'image stockée pour laquelle la référence temporelle est une TRP comme image de référence pour décodage intertrame plutôt que la dernière image décodée, si le champ TRP existe dans les données transmises par le canal vers l'avant. Lorsque l'image pour laquelle la référence temporelle est TRP n'est pas disponible au décodeur, le décodeur peut envoyer un signal de rafraîchissement INTRA forcé au codeur par des moyens externes (par exemple, selon la Recommandation H.245). A moins qu'une politique de stockage de trame différente soit négociée par des moyens externes, les segments d'image vidéo décodés correctement doivent être stockés en mémoire pour utilisation comme images de référence ultérieure selon le principe du premier entré, premier sorti, illustré par la Figure N.1 (sauf pour les images B, qui ne sont pas utilisées comme images de référence), et les segments d'image vidéo dont il a été déterminé que le décodage était incorrect ne doivent pas remplacer ceux qui ont été décodés correctement dans cette zone de mémoire.

Le message d'accusé de réception (ACK, *acknowledgment message*) et le message d'accusé de réception négatif (NACK, *non-acknowledgment message*) sont définis comme des messages transmis par canal de retour. Un message ACK peut être renvoyé lorsque le décodeur décode un segment d'image vidéo avec succès. Des messages NACK peuvent être renvoyés lorsque le décodeur ne réussit pas à décoder un segment d'image vidéo, et d'autres messages de ce genre peuvent continuer d'être renvoyés jusqu'à ce que le décodeur obtienne des données de canal vers l'avant attendues, qui comprennent la TRP demandée ou un rafraîchissement INTRA. Le genre de message qui sera renvoyé est indiqué dans le champ RPSMF de l'en-tête d'image des données transmises par le canal vers l'avant.

Dans un scénario d'utilisation connu sous le nom de "codage par redondance vidéo", le mode de sélection de l'image de référence peut être utilisé par certains codeurs d'une manière où plus d'une représentation de la scène représentée est envoyée au même moment dans le temps (habituellement au moyen d'images de référence différentes). Dans le cas où le mode de sélection d'image de référence est utilisé et où des images adjacentes dans le flux binaire ont la même référence temporelle, le décodeur doit considérer cette occurrence comme une indication selon laquelle des exemplaires redondants de contenus de scène à peu près identiques ont été envoyées, et il décodera et utilisera la première de ces images, en ignorant la ou les images redondantes ultérieures.

Mode d'échelonnabilité temporelle, SNR et spatiale

La présente annexe décrit le mode facultatif de la présente Recommandation qui prend en charge de l'échelonnabilité temporelle, SNR et spatiale. Ce mode peut également être utilisé conjointement avec les techniques de contrôle d'erreurs. Les possibilités de ce mode et la mesure dans laquelle ses fonctions sont prises en charge sont signalées par des moyens externes (par exemple, selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le champ PLUSPTYPE.

O.1 Aperçu général

L'échelonnabilité permet le décodage d'une séquence selon différents niveaux de qualité grâce à l'utilisation d'une hiérarchie d'images et d'images améliorées subdivisée en une ou plusieurs couches. Trois types d'images sont utilisés aux fins de l'échelonnabilité, soit les images B, EI et EP, conformément à la description ci-dessous. Chaque image porte un numéro de couche d'amélioration, ELNUM, qui indique à quelle couche elle appartient, et un numéro de couche de référence, RLNUM, qui indique quelle couche est utilisée pour sa prédiction. La couche inférieure est nommée couche de base et porte le numéro 1.

L'échelonnabilité s'effectue au moyen de trois méthodes de base: amélioration temporelle, amélioration SNR et amélioration spatiale.

O.1.1 Echelonnabilité temporelle

L'échelonnabilité temporelle se fonde sur les images B, ou images à codage prédictif bidirectionnel. Les images B permettent la prédiction à partir d'une image reconstruite précédente ou subséquente ou d'images reconstruites précédente et subséquente de la couche de référence. Cette caractéristique donne généralement une efficacité de compression supérieure à celle obtenue à partir d'images P. Ces images B sont différentes de la partie image B d'une trame PB (ou PB améliorée) (voir les Annexes G et M), étant donné qu'il s'agit d'entités distinctes du flux binaire: elles ne sont pas syntaxiquement fusionnées à une image P (ou EP) subséquente.

Les images B (et la partie B de trames PB ou PB améliorées) ne sont pas utilisées comme images de référence aux fins de la prédiction de toute autre image. Cette caractéristique permet d'ignorer au besoin les images B sans répercussions négatives sur les images subséquentes, ce qui assure une échelonnabilité temporelle. La Figure O.1 illustre la structure prédictive des images P et B.

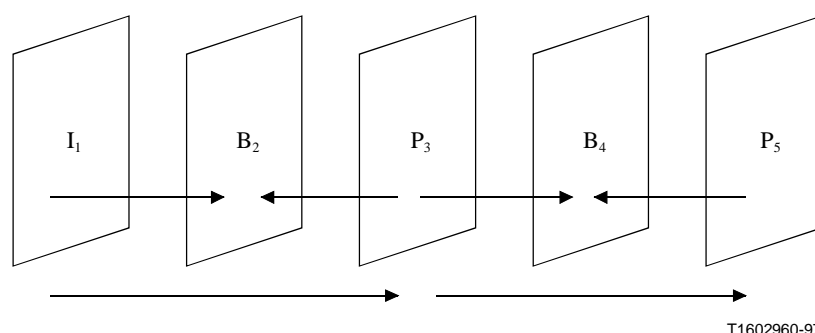


Figure O.1/H.263 – Illustration des dépendances de prédiction des images B

L'emplacement des images B dans le flux binaire est fonction d'un ordre dépendant des données plutôt que d'un ordre strictement temporel. (Cette règle est conforme à l'ordonnement des autres images dans le flux de données mais, pour tous les types d'images autres que les images B, il n'y a pas de conflit entre l'ordre dépendant des données et l'ordre temporel.) Par exemple, si les images d'une séquence vidéo sont numérotées 1, 2, 3, ..., l'ordre des images codées du flux binaire est I₁, P₃, B₂, P₅, B₄, ..., où l'indice renvoie au numéro d'image original (voir la Figure O.1).

Il n'y a pas de limite au nombre d'images B qui peuvent être insérées entre des paires d'images de référence dans la couche de référence (sous réserve de prévenir toute ambiguïté temporelle due au surremplissage du champ de référence temporelle dans l'en-tête d'image). Cependant, le nombre maximal de ces images peut être fixé par des moyens externes (par exemple, la Recommandation H.245).

La hauteur, la largeur et le rapport d'aspect de pixel d'une image B doivent toujours être les mêmes que ceux de son image de couche de référence qui la suit dans le temps.

Les vecteurs cinétiques peuvent dépasser les limites d'image des images B.

O.1.2 Echelonnabilité SNR

L'autre méthode de base utilisée pour obtenir l'échelonnabilité fait appel à l'amélioration spatiale/SNR. L'échelonnabilité spatiale et l'échelonnabilité SNR sont similaires, sauf en ce qui a trait à l'utilisation de l'interpolation, qui est décrite ci-dessous. Étant donné que la compression introduit des artefacts et des distorsions, la différence entre une image reconstruite et l'image originale entrée dans le codeur est (presque toujours) une image de valeur autre que zéro, qui contient ce que l'on peut nommer l'erreur de codage. Normalement, cette erreur de codage est perdue au codage et n'est jamais récupérée. Grâce à l'échelonnabilité SNR, les images correspondant à l'erreur de codage peuvent également être codées et transmises au décodeur pour améliorer l'image décodée. Ces données supplémentaires permettent d'accroître le rapport signal/bruit (SNR, *signal-to-noise ratio*) de l'image vidéo, d'où le terme "échelonnabilité SNR". La Figure O.2 illustre le flux des données pour l'échelonnabilité SNR. Les flèches verticales provenant de la couche inférieure indiquent que l'image de la couche d'amélioration est prédite à partir d'une approximation reconstruite de cette image dans la couche de référence (inférieure).

Si la prédiction est fondée uniquement sur la couche inférieure, l'image de la couche d'amélioration est nommée "image EI". Il est cependant possible de créer une image modifiée à codage prédictif bidirectionnel en utilisant une image précédente de la couche d'amélioration et une image de référence simultanée dans le temps de la couche inférieure. Ce type d'image est nommé "image EP", ou "image P améliorée". Le cheminement de prédiction des images EI et EP est présenté à la Figure O.2. (Bien que cela ne soit pas explicitement illustré à la Figure O.2, une image EI de la couche d'amélioration peut avoir une image P comme image de référence dans la couche inférieure, et une image EP peut avoir une image I comme image de référence dans la couche inférieure.)

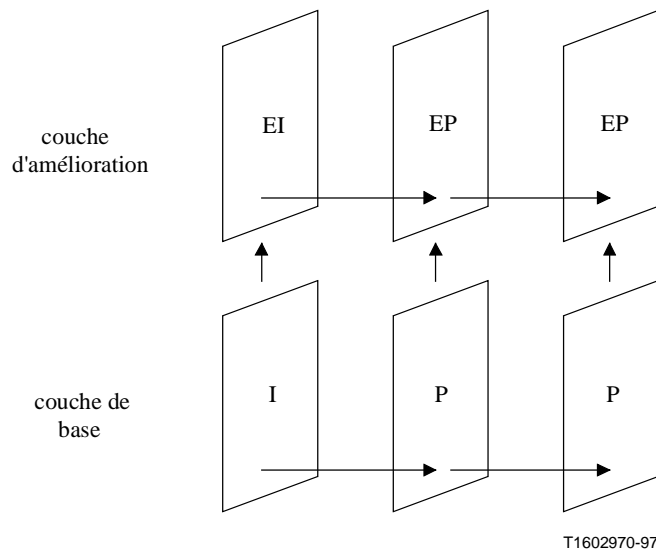


Figure O.2/H.263 – Illustration de l'échelonnabilité SNR

Dans le cas des images EI et EP, la prédiction fondée sur la couche de référence n'utilise pas de vecteurs cinétiques. Cependant, comme dans le cas des images P normales, les images EP font appel à des vecteurs cinétiques lorsque la prédiction se fait à partir d'une image de référence antérieure dans le temps de la même couche.

O.1.3 Echelonnabilité spatiale

La troisième et dernière méthode d'échelonnabilité du mode d'échelonnabilité temporelle, SNR et spatiale est l'échelonnabilité spatiale, qui est étroitement liée à l'échelonnabilité SNR. La seule différence entre ces deux types d'échelonnabilité est que, avant d'être utilisée pour prédire l'image dans la couche d'amélioration spatiale, l'image de la couche de référence est interpolée par un facteur de deux, soit horizontalement ou verticalement (échelonnabilité spatiale unidimensionnelle), soit horizontalement et verticalement (échelonnabilité spatiale bidimensionnelle). Les filtres d'interpolation utilisés aux fins de cette opération sont décrits au O.6. Pour qu'un décodeur puisse recourir à l'échelonnabilité spatiale, il peut devoir posséder une fonction de format d'image personnalisé. Par exemple, si le format de la couche de base était sub-QCIF (128×96), le format de l'image de la couche d'amélioration spatiale bidimensionnelle serait de 256×192 , ce qui n'est pas un format d'image standard. De même, si le format de la couche de base était QCIF (176×144), avec le rapport d'aspect de pixel standard de 12:11, le format de l'image de la couche d'amélioration spatiale horizontale unidimensionnelle serait de 352×144 et le rapport d'aspect de pixel, de 6:11. Dans ces deux cas, il faudrait alors utiliser un format d'image personnalisé pour la couche d'amélioration. Par contre, un format d'image personnalisé n'est pas nécessaire dans le cas de l'utilisation d'une couche de base QCIF avec une couche d'amélioration spatiale bidimensionnelle CIF. L'échelonnabilité spatiale est illustrée à la Figure O.3.

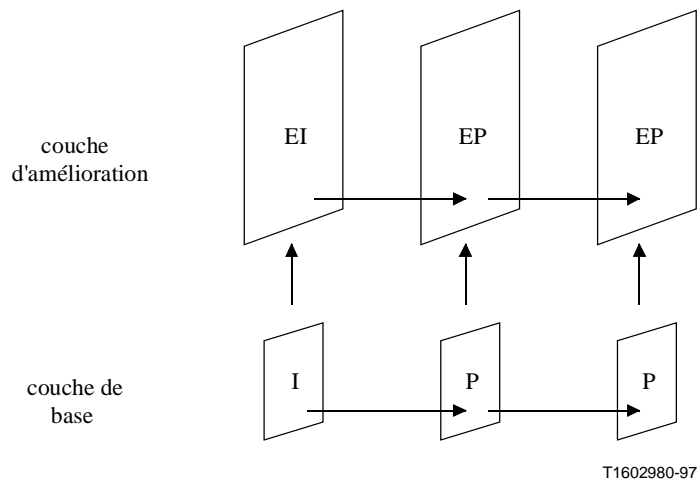


Figure O.3/H.263 – Illustration de l'échelonnabilité spatiale

A l'exception de l'utilisation d'un processus de sur-échantillonnage pour accroître la taille de l'image de la couche de référence avant de l'utiliser comme référence aux fins du processus de codage, le traitement et la syntaxe des images en échelonnabilité spatiale sont fonctionnellement identiques à ceux des images à échelonnabilité SNR.

Etant donné qu'il y a très peu de différence syntaxique entre les images utilisant l'échelonnabilité spatiale et celles utilisant l'échelonnabilité SNR, les images utilisées à l'une ou l'autre de ces fins sont nommées "image EI" et "images EP".

L'image de la couche de base utilisée pour la prédiction ascendante dans une image EI ou EP peut être une image I, une image P, ou la partie P d'une trame PB ou PB améliorée (mais non une image B, ni la partie B d'une trame PB ou PB améliorée).

O.1.4 Echelonnabilité multicouche

Il est possible d'insérer dans le temps des images B non seulement entre des images de types I, P, PB et PB améliorée, mais également entre des images de types EI et EP, peu importe qu'il s'agisse d'images résultant d'une amélioration spatiale ou SNR. Il est également possible d'avoir plus d'une couche d'amélioration SNR ou spatiale en plus d'une couche de base. Ainsi, un flux binaire multicouche échelonnable peut être une combinaison de couches SNR, de couches spatiales et d'images B. Cependant, la taille d'une image ne peut pas diminuer quand le nombre de couches augmente. La taille demeure la même ou augmente d'un facteur de 2 dans une ou deux dimensions. La Figure O.4 illustre un flux binaire multicouche échelonnable.

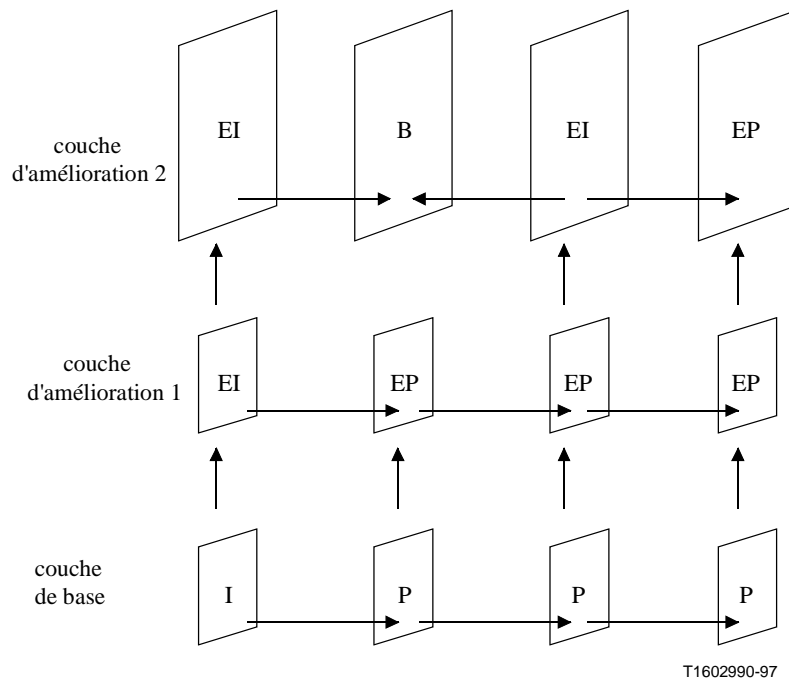


Figure O.4/H.263 – Illustration de l'échelonnabilité multicouche

Dans le cas de l'échelonnabilité multicouche, l'image d'une couche de référence, utilisée pour fins de prédiction ascendante d'une image EI ou EP, peut être une image I, P, EI ou EP, ou encore la partie P d'une trame PB ou PB améliorée de la couche de base (mais non une image B, ni la partie B d'une trame PB ou PB améliorée).

Comme dans le cas à deux couches, les images B peuvent se trouver dans n'importe quelle couche. Cependant, toute image dans une couche d'amélioration dont la position temporelle est la même que celle d'une image B dans sa couche de référence doit être une image B ou la partie B d'une trame PB ou PB améliorée. Cela maintient le caractère "jetable" des images B. Il faut toutefois noter que des images B peuvent se trouver dans des couches qui n'ont pas d'images correspondantes dans les couches inférieures. Cela permet à un codeur de transmettre des signaux vidéo améliorés ayant une fréquence d'image supérieure à celle des couches inférieures.

Le numéro de couche d'amélioration et le numéro de couche de référence pour chaque image d'amélioration (B, EI ou EP) sont indiqués respectivement dans les champs ELNUM et RLNUM de l'en-tête d'image (lorsqu'ils sont présents). Voir les règles d'inférence décrites au 5.1.4.4 pour les cas où ces champs sont absents. Si une image B se trouve dans une couche d'amélioration dans laquelle se trouvent également des images d'échelonnabilité SNR ou spatiale voisines dans le temps, le numéro de la couche de référence (RLNUM) de l'image B doit être le même que celui de la couche d'amélioration (ELNUM).

La hauteur, la largeur et le rapport d'aspect de pixel d'une image B doivent toujours être les mêmes que ceux de son image subséquente dans le temps de la couche de référence.

O.2 Ordre de transmission des images

Les images dépendant d'autres images doivent se trouver dans le flux binaire après les images dont elles sont dépendantes.

L'ordre syntaxique du flux binaire est défini de sorte que, dans le cas des images de référence (c'est-à-dire les images de types I, P, EI ou EP ou la partie P de trames PB ou PB améliorées), les deux règles ci-dessous doivent être respectées:

- 1) toutes les images de référence ayant la même référence temporelle doivent se trouver dans le flux binaire dans un ordre croissant de couche d'amélioration (étant donné que l'image de référence de la couche inférieure sert à décoder l'image de référence de la couche immédiatement supérieure);
- 2) toutes les images de référence simultanées dans le temps dont il est question en 1) ci-dessus doivent se trouver dans le flux binaire avant toute image B pour laquelle l'une quelconque de ces images de référence est la première image de référence subséquente dans le temps dans la couche de référence de l'image B (pour réduire le retard de décodage de toutes les images de référence pouvant servir de références aux images B).

Ensuite, les images B ayant des références temporelles antérieures doivent suivre (ordonnées dans le temps dans chaque couche d'amélioration).

L'emplacement de chaque image B dans le flux binaire doit être conforme aux règles suivantes:

- 1) l'emplacement de l'image B dans le flux binaire doit être après celui de sa première image de référence subséquente dans le temps dans la couche de référence (étant donné que le décodage des images B dépend généralement du décodage préalable de cette image de référence);
- 2) son emplacement dans le flux binaire doit être après celui de toutes les images de référence qui sont simultanées dans le temps avec la première image de référence subséquente dans le temps dans la couche de référence (pour réduire le retard de décodage de toutes les images de référence pouvant servir de références aux images B);
- 3) son emplacement dans le flux binaire doit être avant celui de toute image supplémentaire subséquente dans le temps, autre qu'une image B, dans sa couche de référence (sans quoi il faudrait augmenter la capacité de stockage des images de la couche de référence);
- 4) son emplacement dans le flux binaire doit être après celui de toutes les images EI et EP qui sont simultanées dans le temps avec la première image de référence subséquente dans le temps;
- 5) son emplacement dans le flux binaire doit être avant celui de toutes les images subséquentes dans le temps dans la même couche d'amélioration (sinon, il se produirait des retards inutiles et il faudrait augmenter la capacité de stockage des images de la couche d'amélioration).

La Figure O.5 illustre deux ordres possibles de transmission des images conformément aux règles ci-dessus relativement à la structure de couches présentée (les chiffres dans les cases pointillées indiquent l'ordre dans le flux binaire, les deux ordres possibles étant séparés par une virgule).

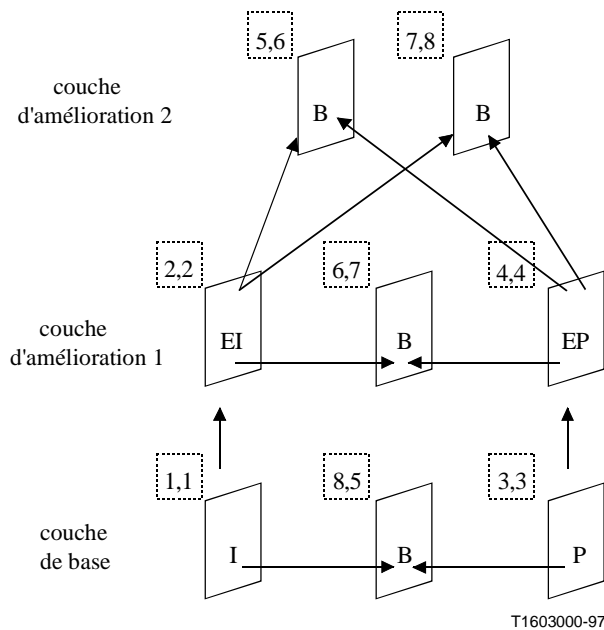


Figure O.5/H.263 – Exemple d'ordre de transmission des images

O.3 Syntaxe de la couche des images

Le numéro de couche d'amélioration (ELNUM) (voir 5.1.11) est toujours présent dans le cas de toute image d'amélioration (B, EI ou EP), et il ne doit pas être présent dans le cas d'images I ou P ou de trames PB ou PB améliorées. Le numéro de couche de référence (RLNUM) (voir 5.1.12) est présent dans le cas de certaines images d'amélioration et il est inféré dans d'autres cas, tel que décrit au 5.1.12.

Il y a exactement une couche de base et ses champs ELNUM et RLNUM égalent 1. Le champ RLNUM donne le numéro de couche d'amélioration des images de référence anticipées et différées des images B, pour l'image de référence ascendante des images EI et EP. Les images de référence de la couche de base peuvent être des images I, PB, PB améliorées et P, dont aucune ne contient un champ ELNUM ou RLNUM dans l'en-tête d'image (leurs valeurs par défaut sont de 1).

Dans le cas des images B, le champ RLNUM doit être inférieur ou égal au champ ELNUM; dans le cas des images EI et EP, RLNUM doit être inférieur à ELNUM.

Le champ ELNUM peut différer du numéro de couche utilisé au niveau du système. Étant donné que les images B n'ont pas d'autres images qui dépendent d'elles, elles peuvent même être placées dans une couche d'amélioration distincte par des composants de système externes à la présente Recommandation (par exemple, selon les Recommandations H.245 et H.223). En outre, il revient au réalisateur de décider si les images d'amélioration sont acheminées dans des canaux vidéo distincts ou si elles sont multiplexées avec les images de la couche de base.

Conformément aux dispositions du 5.1.4.5, le mode avec filtre de dégroupage (voir l'Annexe J) ne s'applique pas aux images B, étant donné que les images B ne servent pas à la prédiction d'autres images. L'application d'un filtre de dégroupage à ces images est une technique entièrement postérieure au traitement, et elle n'est pas visée par la présente Recommandation. Cependant, l'utilisation d'un type quelconque de filtre de dégroupage pour les images B est préconisée, et un filtre similaire au filtre décrit dans l'Annexe J pourrait très bien servir à cette fin.

La référence temporelle (TR) (voir 5.1.2) est définie exactement comme dans le cas des images I et P.

Il ne doit pas y avoir de champs TR_B (voir 5.1.22) ou $DBQUANT$ (voir 5.1.23) dans l'en-tête d'image des images B, EI ou EP.

O.4 Syntaxe de la couche des macroblocs

La syntaxe de la couche des macroblocs est la même pour les images B et EP, étant donné que toutes deux utilisent deux images de référence de façon similaire. Cependant, l'interprétation varie légèrement selon le type d'image. La Figure O.6 présente la syntaxe des images B et EP. Le champ MBTYPE indique le mode de prédiction utilisé: prédiction en mode direct, prédiction anticipée, prédiction différée/ascendante ou prédiction bidirectionnelle. Le champ MBTYPE est défini différemment pour les images B et EP, tel que décrit ci-dessous.

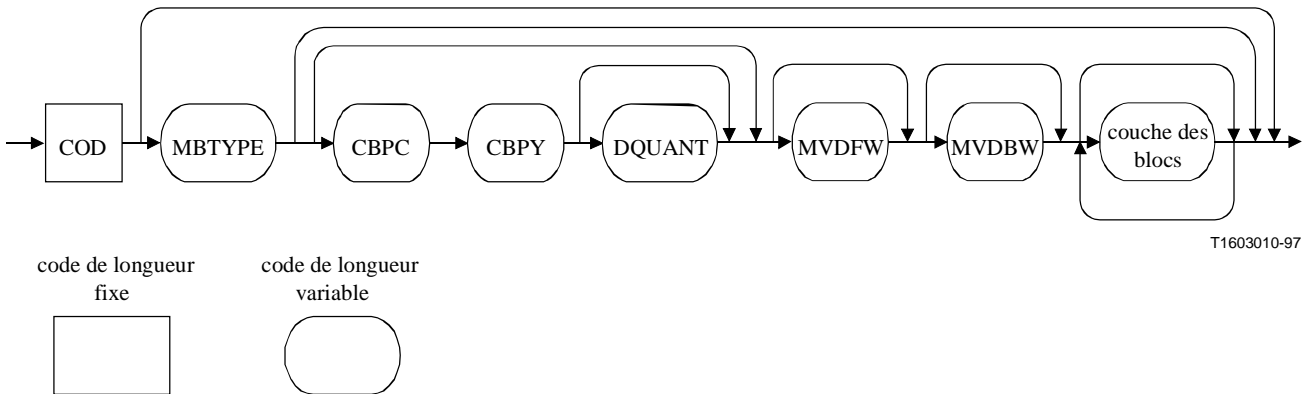


Figure O.6/H.263 – Syntaxe des macroblocs pour les images EP et B

Le mode de prédiction direct n'est disponible que pour les images B. Il s'agit d'un mode de prédiction bidirectionnel similaire au mode bidirectionnel décrit à l'Annexe M (mode de codage par trames PB amélioré). La seule différence est qu'il n'y a pas de restriction concernant les pixels qui peuvent faire l'objet d'une prédiction différée, étant donné que l'image complète de prédiction différée est connue du décodeur. Le mode bidirectionnel utilise des vecteurs cinétiques distincts pour la prédiction anticipée et pour la prédiction différée. Dans les modes direct et bidirectionnel, la valeur des pixels de prédiction est calculée en établissant la moyenne des pixels de prédiction anticipée et différée. La moyenne est calculée en divisant la somme des deux prédictions par deux (division par troncature). En mode direct, quand il y a quatre vecteurs cinétiques dans le macrobloc de référence, les quatre vecteurs cinétiques sont utilisés comme dans le mode de codage par trames PB amélioré (Annexe M).

Dans le cas des images B, la prédiction anticipée se fait à partir d'une image de référence antérieure dans la couche de référence. La prédiction différée se fait à partir d'une image de référence subséquente dans le temps dans la couche de référence.

Dans le cas des images EP, la prédiction anticipée se fait à partir d'une image EI ou EP antérieure dans la même couche. La prédiction ascendante se fait à partir de l'image de référence simultanée dans le temps (peut-être interpolée) dans la couche de référence. Aucun vecteur cinétique n'est utilisé aux fins de la prédiction ascendante (qui se trouve, du point de vue syntaxique, au même emplacement que la prédiction différée des images B), mais un vecteur cinétique peut être utilisé pour la prédiction anticipée.

La syntaxe des macroblocs des images EI est légèrement différente. Comme l'illustre la Figure O.7, les champs MBTYPE et CBPC sont combinés en un champ MCBPC. Aucune prédiction anticipée

n'est utilisée; on utilise uniquement une prédiction ascendante fondée sur l'image de référence simultanée dans le temps dans la couche de référence. Aucun vecteur cinétique n'est utilisé.

Dans le cas des images B et EP, l'utilisation de vecteurs cinétiques pointant hors des limites d'image peut se faire conformément au D.1 (par contre, l'extension de la gamme des vecteurs cinétiques telle que décrite au D.2 n'est active que si le mode de codage par vecteurs cinétiques non restreints est également utilisé).

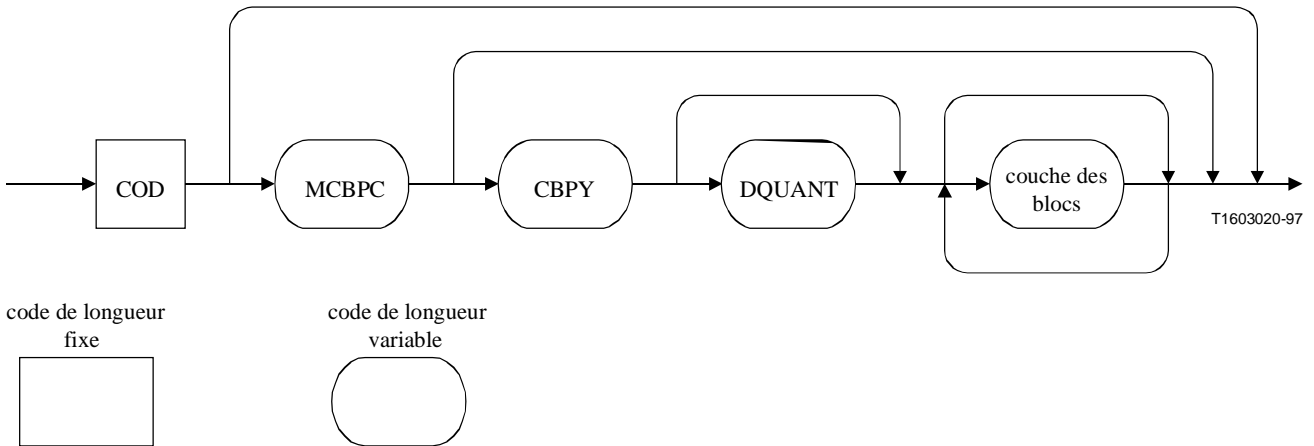


Figure O.7/H.263 – Syntaxe des macroblocs pour les images EI

O.4.1 Indication de macrobloc codé (COD, *coded macroblock indication*) (1 bit)

Si le bit est mis à "0", le macrobloc est codé. Si le bit est mis à "1", aucune autre information n'est transmise relativement au macrobloc et celui-ci est traité comme un macrobloc "sauté", tel que décrit ci-dessous.

O.4.2 Champ MBTYPE/MCBPC (VLC)

Il y a différents tableaux de codes MBTYPE concernant les images B et EP. Dans le cas des images EI, il y a un tableau de codes MCBPC. Le Tableau O.1 est le tableau de codes MBTYPE applicable aux images B. Le Tableau O.2 est le tableau de codes MBTYPE applicable aux images EP. Le Tableau O.3 est le tableau de codes MCBPC applicable aux images EI.

Dans le cas des images B, le type de prédiction "directe (macrobloc sauté)" indique l'absence de champ MBTYPE et de toute donnée dans le macrobloc transmis, et indique que le décodeur déduit les vecteurs cinétiques anticipé et différé et la prédiction bidirectionnelle correspondante. Cela est signalé par le bit COD. Les types de prédiction "anticipée (sans texture)", "différée (sans texture)" et "bidirectionnelle (sans texture)" pour les images B indiquent une prédiction anticipée, différée ou bidirectionnelle sans coefficients, avec un vecteur cinétique transmis pour la prédiction anticipée ou différée, et deux vecteurs cinétiques transmis pour la prédiction bidirectionnelle.

Dans le cas des images EP, le type de prédiction "anticipée (macrobloc sauté)" indique qu'aucune donnée additionnelle n'est transmise pour le macrobloc, de sorte que le décodeur devrait utiliser la prédiction anticipée avec un vecteur cinétique zéro et aucun coefficient. Les types de prédiction "ascendante (sans texture)" et "bidirectionnelle (sans texture)" pour une image EP indiquent une prédiction ascendante ou bidirectionnelle sans coefficients et avec vecteur(s) cinétique(s) zéro.

Dans le cas des images EP, le type de prédiction "ascendante (macrobloc sauté)" indique qu'aucune donnée additionnelle n'est transmise pour le macrobloc, de sorte que le décodeur devrait utiliser la prédiction ascendante avec un vecteur cinétique zéro et aucun coefficient.

Tableau O.1/H.263 – Codes MBTYPE VLC pour les images B

Index	Type de prédiction	MVDFW	MVDB W	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	directe (sauté)					(COD = 1)	0
0	directe			X		11	2
1	directe + Q			X	X	0001	4
2	anticipée (sans texture)	X				100	3
3	anticipée	X		X		101	3
4	anticipée + Q	X		X	X	0011 0	5
5	différée (sans texture)		X			010	3
6	différée		X	X		011	3
7	différée + Q		X	X	X	0011 1	5
8	bidirectionnelle (sans texture)	X	X			0010 0	5
9	bidirectionnelle	X	X	X		0010 1	5
10	bidirectionnelle + Q	X	X	X	X	0000 1	5
11	INTRA			X		0000 01	6
12	INTRA + Q			X	X	0000 001	7
13	bouillage					0000 0000 1	9

Tableau O.2/H.263 – Codes MBTYPE VLC pour les images EP

Index	Type de prédiction	MVDFW	MVDB W	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	anticipée (sauté)					(COD = 1)	0
0	anticipée	X		X		1	1
1	anticipée + Q	X		X	X	001	3
2	ascendante (sans texture)					010	3
3	ascendante			X		011	3
4	ascendante + Q			X	X	0000 1	5
5	bidirectionnelle (sans texture)					0001 0	5
6	bidirectionnelle	X		X		0001 1	5
7	bidirectionnelle + Q	X		X	X	0000 01	6
8	INTRA			X		0000 001	7
9	INTRA + Q			X	X	0000 0001	8
10	bouillage					0000 0000 1	9

Tableau O.3/H.263 – Codes MCBPC VLC pour les images EI

Index	Type de prédiction	Structure de bloc codée (56)	CBPY	DQUANT	MCBPC	Bits
–	ascendante (sauté)				(COD = 1)	0
0	ascendante	00	X		1	1
1	ascendante	01	X		001	3
2	ascendante	10	X		010	3
3	ascendante	11	X		011	3
4	ascendante + Q	00	X	X	0001	4
5	ascendante + Q	01	X	X	0000 001	7
6	ascendante + Q	10	X	X	0000 010	7
7	ascendante + Q	11	X	X	0000 011	7
8	INTRA	00	X		0000 0001	8
9	INTRA	01	X		0000 1001	8
10	INTRA	10	X		0000 1010	8
11	INTRA	11	X		0000 1011	8
12	INTRA+Q	00	X	X	0000 1100	8
13	INTRA+Q	01	X	X	0000 1101	8
14	INTRA+Q	10	X	X	0000 1110	8
15	INTRA+Q	11	X	X	0000 1111	8
16	bourrage				0000 0000 1	9

O.4.3 Structure de bloc codée pour la chrominance (CBPC, *coded block pattern for chrominance*) (longueur variable)

Lorsqu'il est présent, le champ CBPC indique la structure de bloc codée pour la chrominance, telle que décrite au Tableau O.4. Le champ CBPC est présent uniquement pour les images EP et B; sa présence est alors indiquée par le champ MBTYPE (voir les Tableaux O.1 et O.2).

Tableau O.4/H.263 – Codes CBPC VLC

Index	Structure de bloc codée (56)	CBPC	Bits
0	00	0	1
1	01	10	2
2	10	111	3
3	11	110	3

O.4.4 Structure de bloc codée pour la luminance (CBPY) (longueur variable)

Lorsqu'il est présent, le champ CBPY indique quels blocs de la partie luminance du macrobloc sont présents. Sa présence est toujours indiquée par le champ MBTYPE (voir les Tableaux O.1 et O.2 et O.3). Le champ CBPY est codé conformément au 5.3.5 et au Tableau 13. Les macroblocs à codage prédictif ascendant des images EI et EP, les macroblocs à codage prédictif bidirectionnel des images

EP et les macroblocs INTRA des images EI, EP et B utilisent la définition CBPY pour les macroblocs INTRA. Les autres types de macroblocs des images EI, EP et B utilisent la définition CBPY pour les macroblocs INTER.

O.4.5 Information sur le quantificateur (DQUANT) (2 bits/longueur variable)

Le code DQUANT est utilisé comme dans les autres types de macroblocs d'image. Voir 5.3.6 et l'Annexe T.

O.4.6 Données de vecteur cinétique (MVDFW, MVDBW) (longueur variable)

Le champ MVDFW, quand il est présent, indique les données de vecteur cinétique du vecteur anticipé. Le champ MVDBW, quand il est présent, indique les données de vecteur cinétique du vecteur différé (permis uniquement dans le cas des images B). Les mots de code de longueur variable sont donnés au Tableau 14 et au Tableau D.3, pour les cas où le mode de codage par vecteurs cinétiques non restreints est utilisé (voir l'Annexe D).

O.5 Décodage des vecteurs cinétiques

O.5.1 Vecteur cinétiques différentiels

Les vecteurs cinétiques pour les blocs à codage prédictif anticipé, différé ou bidirectionnel sont codés de façon différentielle. Pour récupérer les vecteurs cinétiques des macroblocs, une prédiction est ajoutée aux différences vectorielles cinétiques. Les prédictions sont créées d'une manière similaire à celle décrite au 6.1.1, sauf que les vecteurs cinétiques anticipés sont prédits uniquement à partir des vecteurs cinétiques anticipés des macroblocs voisins et que les vecteurs cinétiques différés sont prédits uniquement à partir des vecteurs cinétiques différés des macroblocs voisins. Les règles de décision décrites au 6.1.1 s'appliquent dans les cas particuliers des bords d'images, de groupes de blocs ou de tranches. Si un macrobloc voisin n'a pas de vecteur cinétique du même type (anticipé ou différé), le prédicteur candidat pour ce macrobloc est zéro pour ce type de vecteur cinétique.

O.5.2 Vecteurs cinétiques en mode direct

Dans le cas des macroblocs codés en mode direct, aucune différence vectorielle n'est transmise. Les vecteurs cinétiques anticipés et différés sont plutôt calculés directement à partir des vecteurs P consécutifs dans le temps, tel que décrit en G.4, le champ MV_D étant toujours égal à zéro. Ces vecteurs dérivés ne servent pas à la prédiction d'autres vecteurs cinétiques.

O.6 Filtres d'interpolation

Les Figures O.8 et O.9 illustrent la méthode d'interpolation d'image pour l'échelonnabilité spatiale bidimensionnelle. La première figure présente l'interpolation des pixels intérieurs et la seconde, l'interpolation près des limites de l'image. Il s'agit de la même technique que celle utilisée à l'Annexe Q et, dans certains cas, à l'Annexe P. Les Figures O.10 et O.11 illustrent la méthode d'interpolation d'image pour l'échelonnabilité spatiale unidimensionnelle. La Figure O.10 présente l'interpolation horizontale des pixels intérieurs. L'interpolation verticale se fait de façon analogue. La Figure O.11 illustrent l'interpolation des pixels près des limites de l'image. Ici également, l'interpolation verticale se fait de façon analogue. Il s'agit de la même technique que celle utilisée dans certains cas à l'Annexe P.

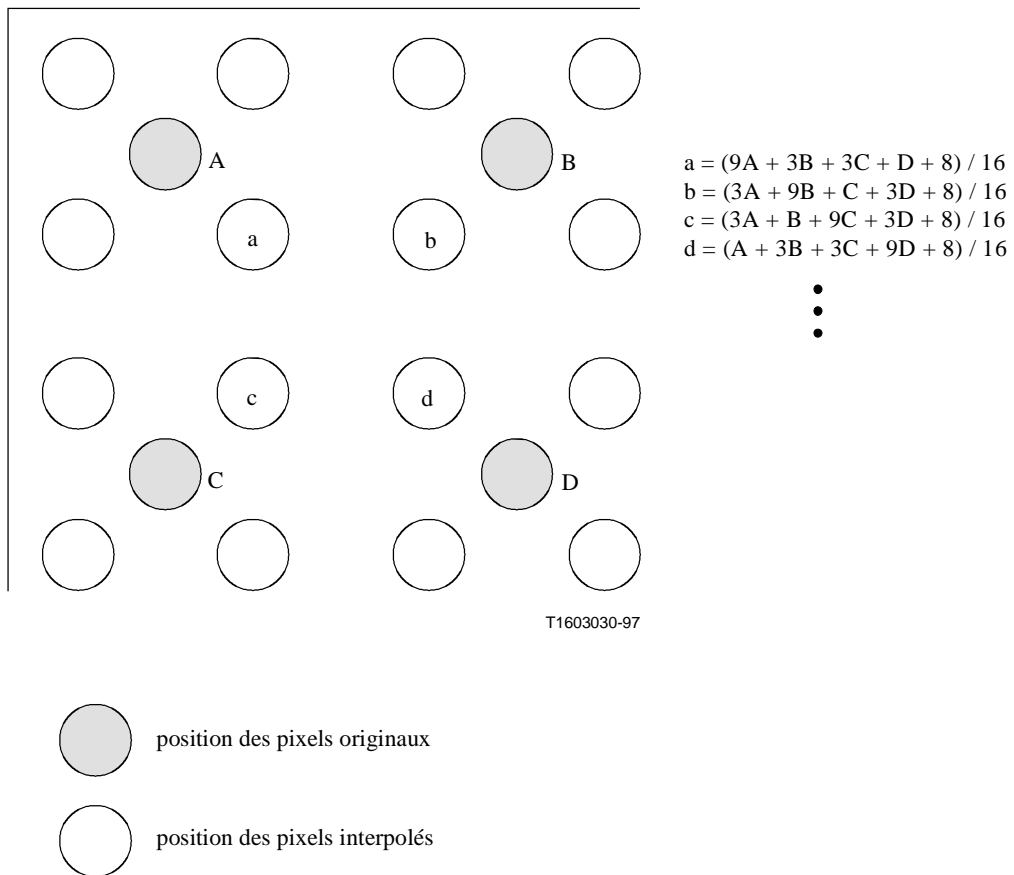


Figure O.8/H.263 – Méthode d'interpolation des pixels pour l'échelonnabilité bidimensionnelle

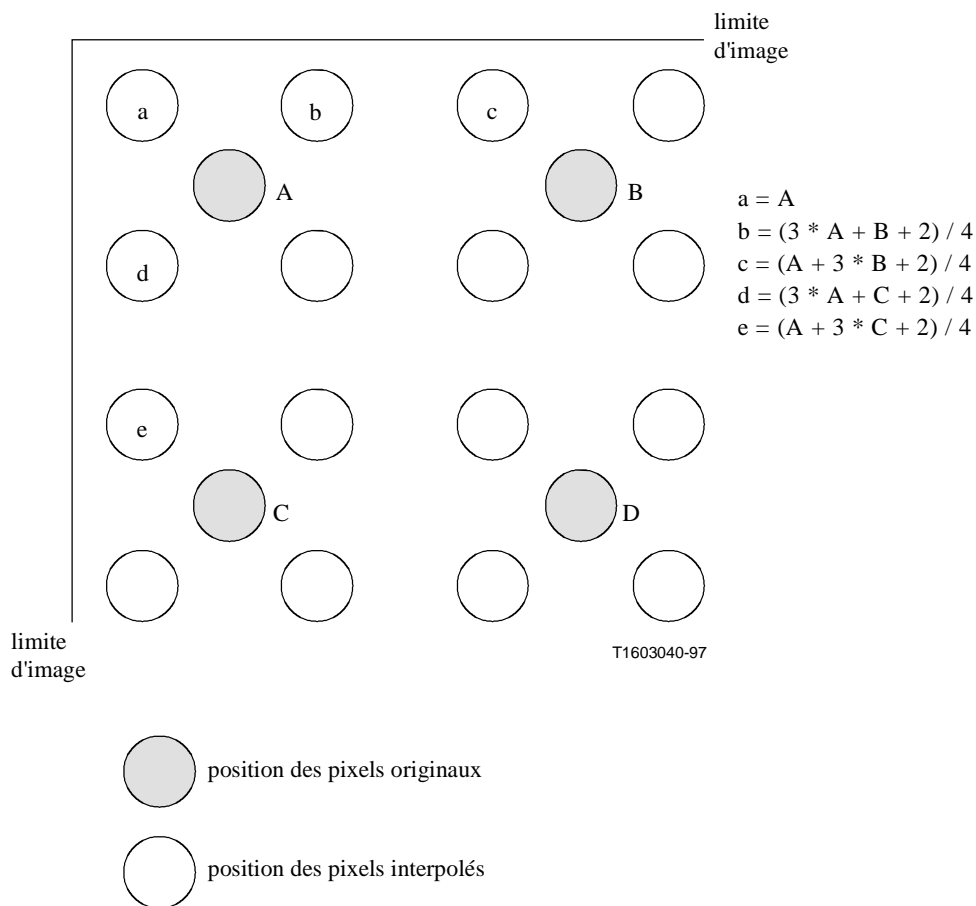


Figure O.9/H.263 – Méthode d'interpolation bidimensionnelle près des limites

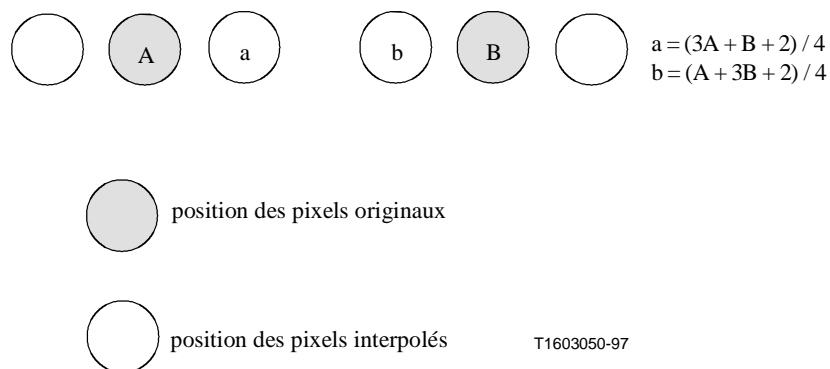


Figure O.10/H.263 – Méthode d'interpolation des pixels pour l'échelonnabilité unidimensionnelle

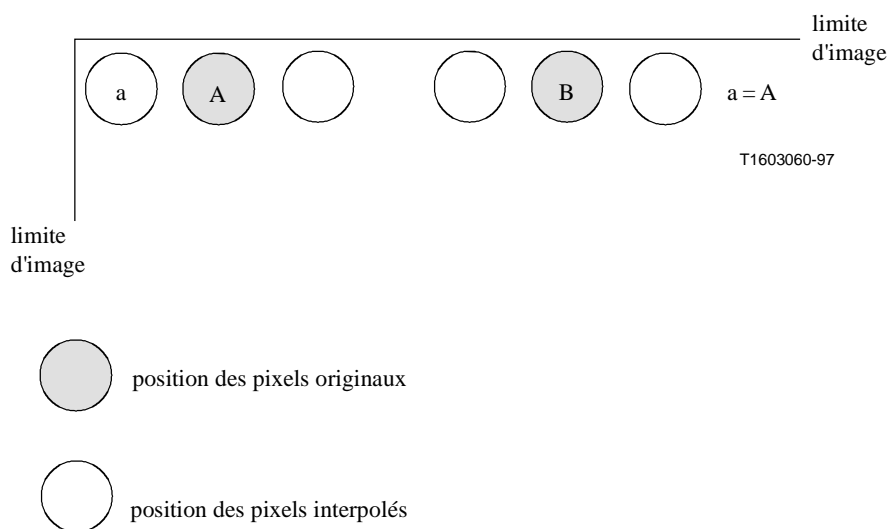


Figure O.11/H.263 – Méthode d'interpolation unidimensionnelle près des limites d'image

ANNEXE P

Rééchantillonnage d'image de référence

P.1 Introduction

La présente annexe décrit le mode d'utilisation et la syntaxe d'un procédé de rééchantillonnage qui peut être appliqué à l'image de référence décodée précédente, pour produire une image "déformée" qui servira à prédire l'image en cours. Cette syntaxe de rééchantillonnage permet de spécifier la relation entre l'image en cours et une image antérieure ayant un format de source différent et permet aussi de préciser une déformation par "mouvement global" de la forme, de la taille et de l'emplacement de l'image antérieure par rapport à l'image en cours. En particulier, le mode de rééchantillonnage d'image de référence peut servir à modifier de manière adaptative la résolution des images durant le codage. Les coefficients d'interpolation bilinéaire sont obtenus à l'aide d'un algorithme rapide. La possibilité d'utilisation de ce mode et la mesure dans laquelle ses fonctions sont prises en charge sont déterminées par des moyens externes (par exemple, selon la Recommandation H.245). Ce mode peut être utilisé dans le cadre de scénarios restreints qui peuvent être définis au cours de la négociation sur la capacité (par exemple, pour prendre en charge uniquement le redimensionnement de facteur quatre des images, ou uniquement la déformation des images de résolution d'un demi-pixel, ou pour prendre en charge le redimensionnement et les déplacements d'image arbitraires).

NOTE – La transformation par défaut d'image entre images de résolutions différentes est définie pour conserver l'alignement spatial des limites de la région d'image et des positions relatives des échantillons de luminance et de chrominance. Ceci peut avoir des implications sur la conception des opérations de rééchantillonnage utilisées pour engendrer des images de résolutions diverses au codeur et pour afficher des images de résolutions diverses après le décodage (en particulier en ce qui concerne les décalages des positions spatiales causés par des déphasages induits au cours du rééchantillonnage). De plus, puisque ce mode peut être utilisé pour des variations adaptatives dynamiques de la résolution des images, le fonctionnement avec ce mode peut bénéficier de la négociation par des moyens externes pour afficher l'image décodée à une résolution supérieure à celle de l'image codée, afin de pouvoir passer d'une taille d'image codée à une autre sans avoir à redimensionner l'image affichée.

Si le bit de rééchantillonnage d'image de référence dans le champ PLUSPTYPE n'est pas mis à 1, PLUSPTYPE étant présent et l'image étant une image INTER, B, ou EP ou une trame PB améliorée, et si l'image a une taille différente de celle de l'image codée antérieure dans le temps, une telle condition invoque le rééchantillonnage d'image de référence avec des paramètres de déformation (voir P.2.2) définis égaux à zéro, un mode de remplissage (voir P.2.3) correspondant à l'écrêtage (*clip*) et une précision de déplacement (voir P.2.1) au $\frac{1}{16}$ de pixel près. De ce fait, le processus de rééchantillonnage peut facilement servir de variation à codage prédictif dans la résolution de l'image. Dans les cas simples de variation de résolution d'un facteur quatre, comme lorsqu'on passe du format CIF au format 4CIF, le processus de rééchantillonnage se réduit au même filtre simple que pour l'échelonnabilité spatiale (Annexe O) ou pour le rafraîchissement à résolution réduite (Annexe Q), sauf pour l'application du contrôle de l'arrondissement.

Si l'image est une image EP et si le bit de rééchantillonnage d'image de référence est mis à 1 dans le champ PLUSPTYPE de l'en-tête d'image sur la couche de référence, ce bit devra également être mis à 1 dans l'en-tête de l'image EP dans la couche d'amélioration.

Si une image B utilise le mode de rééchantillonnage d'image de référence, le processus de rééchantillonnage sera appliqué à l'image d'ancrage antérieure dans le temps et non pas à la suivante dans le temps. L'image d'ancrage antérieure dans le temps à laquelle est appliqué le processus de rééchantillonnage sera l'image décodée (c.-à-d. avant tout rééchantillonnage appliqué pour le mode de rééchantillonnage d'image de référence si ce mode est aussi invoqué pour l'image de référence suivante). L'image d'ancrage suivante dans le temps sera de même taille que l'image B.

Si le mode de rééchantillonnage d'image de référence est invoqué pour une trame PB améliorée, un ensemble de paramètres de déformation est envoyé et l'image de référence rééchantillonnée sert de référence à la fois pour la partie B et pour la partie P de la trame PB améliorée.

Le rééchantillonnage d'image de référence est défini à partir du déplacement des quatre coins de la région d'image en cours. Pour le champ de luminance de l'image en cours ayant pour dimension horizontale H et pour dimension verticale V , on définit quatre vecteurs cinétiques conceptuels \underline{v}^{00} , \underline{v}^{H0} , \underline{v}^{0V} , et \underline{v}^{HV} , correspondant respectivement aux coins supérieur gauche, supérieur droit, inférieur gauche et inférieur droit de l'image. Ces vecteurs indiquent comment déplacer les coins de l'image en cours pour les faire coïncider avec les coins correspondants de l'image décodée précédente, comme à la Figure P.1. Les unités de ces vecteurs sont les mêmes que dans la grille de l'image de référence. Pour produire un vecteur $\underline{v}(x, y)$ en un point de valeur réelle (x, y) à l'intérieur de l'image en cours, on utilise une approximation de l'interpolation bilinéaire, comme dans le cas suivant:

$$\underline{v}(x, y) = \left(1 - \frac{y}{V}\right) \left[\left(1 - \frac{x}{H}\right) \underline{v}^{00} + \left(\frac{x}{H}\right) \underline{v}^{H0} \right] + \left(\frac{y}{V}\right) \left[\left(1 - \frac{x}{H}\right) \underline{v}^{0V} + \left(\frac{x}{H}\right) \underline{v}^{HV} \right]$$

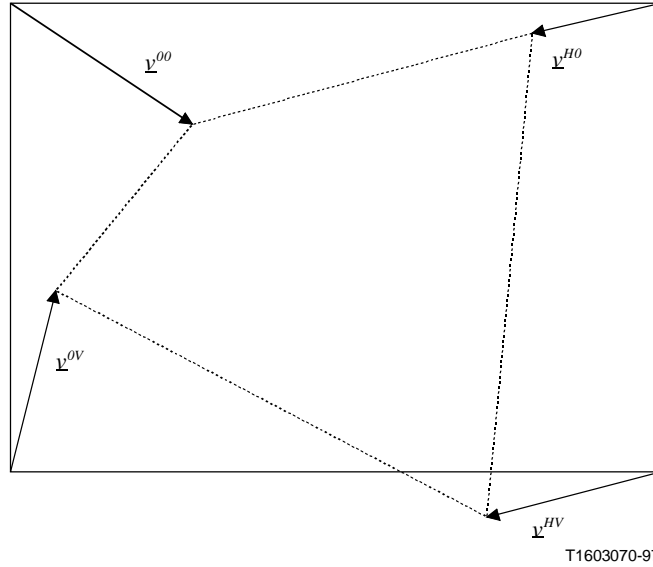


Figure P.1/H.263 – Exemple de vecteurs cinétiques conceptuels utilisés pour la déformation

La dimension horizontale H et la dimension verticale V de l'image en cours, ainsi que la dimension horizontale H_r et la dimension verticale V_r de l'image de référence sont les valeurs indiquées par l'entête d'image, qu'elles soient ou non divisibles par 16. Si la largeur ou la hauteur de l'image n'est pas divisible par 16, on engendrera la zone additionnelle en ajoutant des pixels à l'image rééchantillonnée à l'aide du même mode de remplissage que dans le processus de rééchantillonnage.

Pour simplifier la description, on définit les vecteurs de rééchantillonnage \underline{r}^0 , \underline{r}^x , \underline{r}^y et \underline{r}^{xy} de la manière suivante:

$$\begin{aligned} \underline{r}^0 &= \underline{v}^{00} \\ \underline{r}^x &= \underline{v}^{H0} - \underline{v}^{00} \\ \underline{r}^y &= \underline{v}^{0V} - \underline{v}^{00} \\ \underline{r}^{xy} &= \underline{v}^{00} - \underline{v}^{H0} - \underline{v}^{0V} + \underline{v}^{HV} \end{aligned}$$

A l'aide de ces définitions, on réécrit l'équation d'interpolation bilinéaire sous la forme:

$$\underline{v}(x, y) = \underline{r}^0 + \left(\frac{x}{H}\right) \underline{r}^x + \left(\frac{y}{V}\right) \underline{r}^y + \left(\frac{x}{H}\right) \left(\frac{y}{V}\right) \underline{r}^{xy}$$

Pour la déformation, on suppose que le coin supérieur gauche a pour coordonnées $(x, y) = (0, 0)$ et que chaque pixel a une largeur et une hauteur égales à l'unité, de sorte que les centres des pixels sont situés aux points $(x, y) = \left(i_L + \frac{1}{2}, j_L + \frac{1}{2}\right)$ pour $i_L = 0, \dots, H - 1$ et $j_L = 0, \dots, V - 1$, l'indice L indiquant que i_L et j_L se rapportent au champ de luminance. (Puisque le rapport d'aspect de pixel est en général constant ou que la conversion du rapport d'aspect est effectuée par ce rééchantillonnage, il n'est pas nécessaire de faire intervenir le rapport d'aspect réel des pixels pour les

déformations.) Avec cette convention, les déplacements en x et en y aux points en question dans le champ de luminance de l'image de référence sont:

$$v_x(i_L, j_L) = \frac{1}{HV} \left[HVr_x^0 + \left(i_L + \frac{1}{2}\right) Vr_x^x + \left(j_L + \frac{1}{2}\right) Hr_x^y + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_x^{xy} \right]$$

$$v_y(i_L, j_L) = \frac{1}{HV} \left[HVr_y^0 + \left(i_L + \frac{1}{2}\right) Vr_y^x + \left(j_L + \frac{1}{2}\right) Hr_y^y + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_y^{xy} \right]$$

Comme toutes les positions et les phases doivent être calculées par rapport au centre du pixel du coin supérieur gauche, qui a pour coordonnées $(x, y) = \left(\frac{1}{2}, \frac{1}{2}\right)$, les grandeurs qui nous intéressent sont:

$$x_R(i_L, j_L) - \frac{1}{2} = \left(i_L + \frac{1}{2}\right) + v_x(i_L, j_L) - \frac{1}{2}$$

$$= \frac{1}{HV} \left[HVr_x^0 + \left(i_L + \frac{1}{2}\right) (HV + Vr_x^x) + \left(j_L + \frac{1}{2}\right) Hr_x^y + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_x^{xy} \right] - \frac{1}{2}$$

$$y_R(i_L, j_L) - \frac{1}{2} = \left(i_L + \frac{1}{2}\right) + v_y(i_L, j_L) - \frac{1}{2}$$

$$= \frac{1}{HV} \left[HVr_y^0 + \left(j_L + \frac{1}{2}\right) Vr_y^x + \left(j_L + \frac{1}{2}\right) (HV + Hr_y^y) + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_y^{xy} \right] - \frac{1}{2}$$

Après avoir déterminé une position dans l'image de référence décodée précédente à l'aide d'une approximation de ces équations, on utilise une interpolation bilinéaire telle que décrite plus loin dans la présente annexe pour obtenir une valeur pour le pixel rééchantillonné.

Chaque vecteur de rééchantillonnage peut se décomposer en deux composantes, la première décrivant la déformation géométrique et la seconde représentant toute différence de dimensions entre l'image prédite (dimension horizontale H et dimension verticale V) et l'image de référence (dimension horizontale H_R et dimension verticale V_R). Cette décomposition est la suivante:

$$\underline{v}^{00} = \underline{v}_{warp}^{00} + \underline{v}_{size}^{00} = \underline{v}_{warp}^{00} + (0,0)$$

$$\underline{v}^{H0} = \underline{v}_{warp}^{H0} + \underline{v}_{size}^{H0} = \underline{v}_{warp}^{H0} + (H_R - H, 0)$$

$$\underline{v}^{0V} = \underline{v}_{warp}^{0V} + \underline{v}_{size}^{0V} = \underline{v}_{warp}^{0V} + (0, V_R - V)$$

$$\underline{v}^{HV} = \underline{v}_{warp}^{HV} + \underline{v}_{size}^{HV} = \underline{v}_{warp}^{HV} + (H_R - H, V_R - V)$$

P.2 Syntaxe

Lorsque le bit de rééchantillonnage d'image de référence est mis à 1 dans le champ PLUSTYPE de l'en-tête d'image, le champ RPRP de l'en-tête d'image contient des paramètres qui gouvernent le processus de rééchantillonnage d'image de référence et qui comprennent: un champ de précision de déplacement de distorsion (WDA, *warping displacement accuracy*) de deux bits, parfois huit paramètres de déformation ou raffinements de paramètres de déformation à un bit, et un mode de remplissage, comme il est décrit dans le présent sous-paragraphe.

P.2.1 Précision de déplacement de distorsion (WDA, *warping displacement accuracy*) (2 bits)

Un champ à deux bits de précision de déplacement de distorsion WDA apparaît d'abord dans le champ RPRP du flux binaire pour indiquer la précision des déplacements pour chaque pixel. Une valeur "10" signifie que les déplacements en x et en y pour chaque pixel sont quantifiés avec une précision d'un demi-pixel. Une valeur égale à "11" signifie que les déplacements sont déterminés au $\frac{1}{16}$ de pixel près. L'utilisation d'autres valeurs est réservée.

P.2.2 Paramètres de déformation (longueur variable)

Lorsque les paramètres de rééchantillonnage d'image de référence sont envoyés pour une image INTER ou une image B, ou pour une trame PB améliorée, huit paramètres de déformation sont inclus dans l'en-tête d'image à l'aide du code de longueur variable (VLC) représenté au Tableau D.3. Pour une image EP utilisant l'échelonnabilité SNR, ce sont les paramètres de déformation de la couche inférieure qui sont utilisés et aucun paramètre de déformation n'est transmis. Si le bit de rééchantillonnage d'image de référence est mis à 1 dans le champ PLUSPTYPE de l'en-tête d'image pour une image EP utilisant l'échelonnabilité spatiale, on perfectionne les paramètres de déformation de la couche inférieure jusqu'à atteindre la précision nécessaire pour la couche en cours en multipliant par deux le paramètre de déformation pour chaque dimension suréchantillonnée (paramètres de déformation avec indice x et/ou y) de la couche inférieure et en ajoutant la valeur d'un bit additionnel qui est envoyé à la place du paramètre de déformation correspondant pour définir le bit le moins significatif du paramètre de déformation.

Les huit paramètres de déformation entiers (ou leurs raffinements à un bit) sont envoyés dans l'ordre suivant:

$$w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}, \text{ et } w_y^{xy}$$

Lorsqu'il ne s'agit pas de raffinements à un bit, l'envoi de ces paramètres est similaire à l'envoi de différences de vecteurs cinétiques dans le mode de codage par vecteurs cinétiques non restreints lorsque PLUSPTYPE est présent, en utilisant le Tableau D.3 sans restriction sur la gamme de valeurs des paramètres (c.-à-d., une gamme de -4095 à $+4095$). Comme pour le codage de paires de différence de vecteurs cinétiques, un bit de prévention d'émulation sera ajouté au besoin après l'envoi de chaque paire de paramètres de déformation, de sorte que si le mot de code tout à zéro (la valeur $+1$ en unités d'un demi-pixel) du Tableau D.3 est utilisé pour les deux paramètres de déformation de la paire (w_x^0 et w_y^0 , w_x^x et w_y^y , ou w_x^{xy} et w_y^{xy}), et la paire de mots de code est suivie par un seul bit égal à 1 pour empêcher l'émulation d'un code de déclenchement.

Ces huit paramètres de déformation sont interprétés comme des déplacements de coins d'image par rapport aux déplacements que produirait la suppression de la composante de redimensionnement des vecteurs de rééchantillonnage. Les paramètres de déformation sont mis à l'échelle pour représenter les décalages d'un demi-pixel dans le champ de luminance de l'image en cours et la gamme de valeurs de ces paramètres s'étend de -4095 à $+4095$. Les paramètres de déformation sont définis à partir des vecteurs de rééchantillonnage par les relations:

$$\begin{aligned}
w_x^0 &= 2r_x^0 & w_y^0 &= 2r_y^0 \\
w_x^x &= 2(r_x^x - (H_R - H)) & w_y^x &= 2r_y^x \\
w_x^y &= 2r_x^y & w_y^y &= 2(r_y^y - (V_R - V)) \\
w_x^y &= 2r_x^y & w_y^{xy} &= 2r_y^{xy}
\end{aligned}$$

P.2.3 Mode de remplissage (FILL_MODE) (2 bits)

Pour une image INTER ou B ou pour une trame PB améliorée, dans l'en-tête d'image, juste après les paramètres de déformation codés en VLC (codage à longueur variable) se trouvent deux bits qui définissent l'action du mode de remplissage à effectuer pour les valeurs des pixels dont la position calculée dans l'image de référence se trouve à l'extérieur de la région de l'image de référence. La signification de ces deux bits est illustrée au Tableau P.1 et leur emplacement est indiqué à la Figure P.2. Pour une image EP, l'action du mode de remplissage est la même que pour la couche de référence et les deux bits du mode de remplissage ne sont pas envoyés.

Tableau P.1/H.263 – Bits et action du mode de remplissage

Bits de remplissage	Action de remplissage
0 0	<i>couleur</i>
0 1	<i>noir</i>
1 0	<i>gris</i>
1 1	<i>écrêtage</i>

Si le mode de remplissage est *écrêtage*, les coordonnées des points dans l'image de référence antérieure sont limités indépendamment comme dans le mode de codage par vecteurs cinétiques non restreints, de sorte que les valeurs des pixels à l'extérieur de l'image de référence antérieure sont estimées par extrapolation à partir des valeurs des pixels à la limite de l'image. Si le mode de remplissage est *noir*, les échantillons de luminance à l'extérieur de la région de l'image de référence antérieure ont pour valeur $Y = 16$ et les échantillons de chrominance une valeur de $C_B = C_R = 128$. Si le mode de remplissage est *gris*, les valeurs des échantillons de luminance et de chrominance sont $Y = C_B = C_R = 128$. Si le mode de remplissage est *couleur*, des champs additionnels sont envoyés pour préciser une couleur de remplissage, comme nous allons le voir au sous-paragraphe qui suit.

P.2.4 Spécification des couleurs de remplissage (Y_FILL, C_B_EPB, C_B_FILL, C_R_EPB, C_R_FILL) (26 bits)

Si le mode de remplissage est *couleur* et si l'image n'est pas une image EP, les bits de remplissage sont alors suivis dans le flux binaire par trois entiers à huit bits, Y_fill , C_B_fill , et C_R_fill , qui spécifient une couleur de remplissage précise. Entre ces trois entiers à huit bits se trouvent deux bits de prévention d'émulation (C_B_EPB et C_R_EPB) dont chacun est égal à 1. La Figure P.2 représente le format de cette spécification de couleur, qui est uniquement présente lorsque le mode de remplissage est *couleur*. Chaque champ entier à huit bits est envoyé à l'aide de sa représentation naturelle. Pour une image EP, l'action de remplissage (et la couleur de remplissage) est la même que pour la couche de référence et la spécification de couleur de remplissage n'est pas envoyée.

FILL_MODE	Y_FILL	C _B _EPB	C _B _FILL	C _R _EPB	C _R _FILL
-----------	--------	---------------------	----------------------	---------------------	----------------------

Figure P.2/H.263 – Format des données de mode de remplissage et de spécification de couleur de remplissage

P.3 Algorithme de rééchantillonnage

La méthode décrite dans le présent sous-paragraphe donnera des résultats mathématiques équivalents à ceux de la méthode utilisée pour engendrer les échantillons de l'image de référence rééchantillonnée. En fonction des paramètres de déformation entiers

$w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}$, et w_y^{xy} , les paramètres entiers

$u_x^{00}, u_y^{00}, u_x^{H0}, u_y^{H0}, u_x^{0V}, u_y^{0V}, u_x^{HV}$, et u_y^{HV} qui désignent les déplacements en x et en y aux coins du

champ de luminance avec une précision au $\frac{1}{32}$ de pixel près (on obtient les déplacements réels en divisant ces valeurs par 32) sont définis par:

$$\begin{aligned}
u_x^{00} &= 16w_x^0 & u_y^{00} &= 16w_y^0 \\
u_x^{H0} &= 16(w_x^0 + w_x^x + 2(H_R - H)) & u_y^{H0} &= 16(w_y^0 + w_y^x) \\
u_x^{0V} &= 16(w_x^0 + w_x^y) & u_y^{0V} &= 16(w_y^0 + w_y^y + 2(V_R - V)) \\
u_x^{HV} &= 16(w_x^0 + w_x^x + w_x^y + w_x^{xy} + 2(H_R - H)) & u_y^{HV} &= 16(w_y^0 + w_y^x + w_y^y + w_y^{xy} + 2(V_R - V))
\end{aligned}$$

Ensuite, H' et V' , qui désignent la dimension horizontale et la dimension verticale de la *trame* virtuelle, sont définis comme les plus petits entiers qui vérifient la condition suivante:

$$H' \geq H, V' \geq V, H' = 2^m, V' = 2^n, m \text{ et } n \text{ étant des entiers positifs}$$

Lorsqu'on effectue une extrapolation bilinéaire sur les vecteurs de coins du champ de luminance, les paramètres entiers $u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}$, et u_y^{RB} qui désignent les déplacements en x et en y du champ de luminance aux *points virtuels* $(x, y) = (0, 0), (H', 0), (0, V')$ et (H', V') avec une précision au $\frac{1}{32}$ de pixel près (on obtient les déplacements réels en divisant ces valeurs par 32) sont définis par:

$$\begin{aligned}
u_x^{LT} &= u_x^{00} & u_y^{LT} &= u_y^{00} \\
u_x^{RT} &= ((H - H')u_x^{00} + H'u_x^{H0}) // H & u_y^{RT} &= ((H - H')u_y^{00} + H'u_y^{H0}) // H \\
u_x^{LB} &= ((V - V')u_x^{00} + V'u_x^{0V}) // V & u_y^{LB} &= ((V - V')u_y^{00} + V'u_y^{0V}) // V \\
u_x^{RB} &= ((V - V')((H - H')u_x^{00} + H'u_x^{H0}) + V'((H - H')u_x^{0V} + H'u_x^{HV})) // (HV) \\
u_y^{RB} &= ((V - V')((H - H')u_y^{00} + H'u_y^{H0}) + V'((H - H')u_y^{0V} + H'u_y^{HV})) // (HV)
\end{aligned}$$

où $//$ désigne la division entière qui arrondit le quotient à l'entier le plus proche et qui arrondit les valeurs demi-entières à l'entier suivant en s'éloignant de zéro.

Dans la suite de la présente annexe, on suppose que les centres des pixels sont situés aux points $(x, y) = \left(i + \frac{1}{2}, j + \frac{1}{2}\right)$ pour les champs de luminance et de chrominance. Les paramètres entiers i, j sont définis comme suit:

- $i = 0, \dots, H - 1$ et $j = 0, \dots, V - 1$ pour la luminance;
- $i = 0, \dots, \frac{H}{2} - 1$ et $j = 0, \dots, \frac{V}{2} - 1$ pour la chrominance.

Ceci sous-entend que l'on utilise des systèmes de coordonnées différents pour la luminance et la chrominance, comme l'indique la Figure P.3. Si l'on prend le système de coordonnées utilisé pour la chrominance, les paramètres entiers $u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB},$ et u_y^{RB} définis plus haut peuvent également être considérés comme les déplacements en x et en y du champ de chrominance aux points virtuels $(x, y) = (0, 0), (H'/2, 0), (0, V'/2)$ et $(H'/2, V'/2)$ avec une précision au $\frac{1}{64}$ de pixel près (on obtient les déplacements réels en divisant ces valeurs par 64). A l'aide de ces paramètres et d'un paramètre additionnel S , qui est par définition égal à 2 pour la luminance et à 1 pour la chrominance, les algorithmes de rééchantillonnage pour les champs de luminance et de chrominance sont définis à partir d'équations communes.

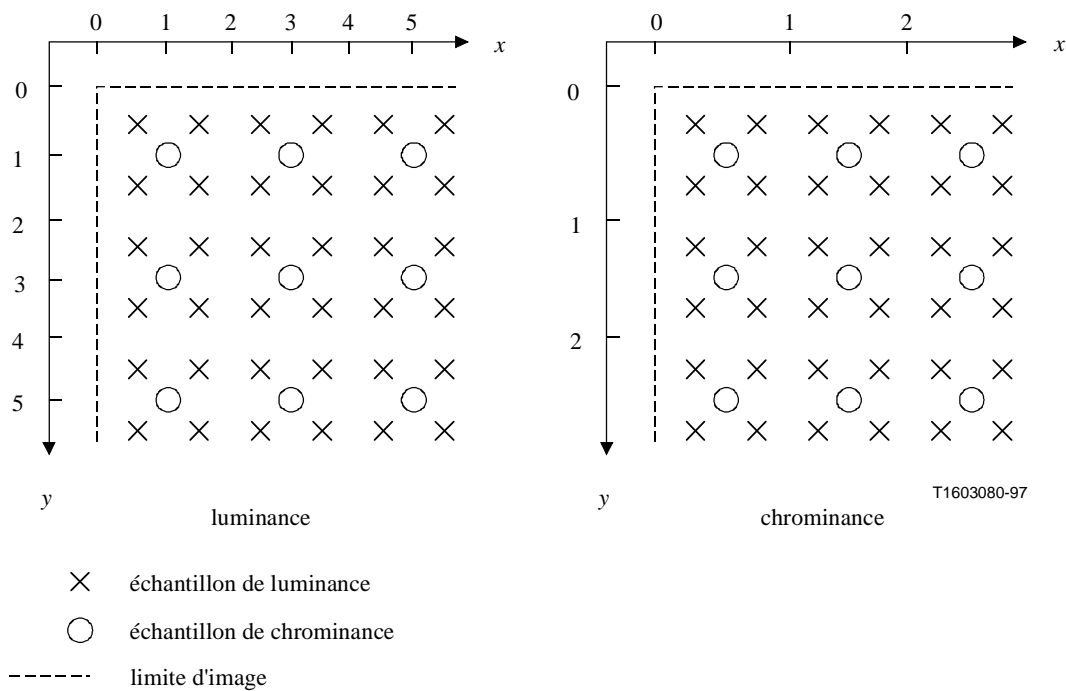


Figure P.3/H.263 – Systèmes de coordonnées pour les champs de luminance et de chrominance

Les paramètres entiers $u_x^L(j), u_y^L(j), u_x^R(j),$ et $u_y^R(j)$ qui désignent les déplacements en x et y du champ d'image au point $(x, y) = \left(0, j + \frac{1}{2}\right)$ et $\left(SH'/2, j + \frac{1}{2}\right)$ avec une précision au $\frac{s}{64}$ de pixel près (on obtient les déplacements réels en divisant ces valeurs par $64/S$) sont définis à l'aide d'une interpolation linéaire à une dimension du type:

$$u_x^L(j) = \left((SV' - 2j - 1)u_x^{LT} + (2j + 1)u_x^{LB} \right) // (SV') \quad u_y^L(j) = \left((SV' - 2j - 1)u_y^{LT} + (2j + 1)u_y^{LB} \right) // (SV')$$

$$u_x^R(j) = \left((SV' - 2j - 1)u_x^{RT} + (2j + 1)u_x^{RB} \right) // (SV') \quad u_y^R(j) = \left((SV' - 2j - 1)u_y^{RT} + (2j + 1)u_y^{RB} \right) // (SV')$$

où $\square // \square$ désigne une division entière qui arrondit le quotient à l'entier le plus proche et qui arrondit les valeurs demi-entières à l'entier supérieur en valeur absolue.

Enfin, les paramètres qui déterminent le point transformé dans l'image de référence deviennent:

$$I_R(i, j) = Pi + \left((SH' - 2i - 1)u_x^L(j) + (2i + 1)u_x^R(j) + 32H'/P \right) // // (64H'/P)$$

$$J_R(i, j) = Pj + \left((SH' - 2i - 1)u_y^L(j) + (2i + 1)u_y^R(j) + 32H'/P \right) // // (64H'/P)$$

$$i_R(i, j) = I_R(i, j) // // P \quad j_R(i, j) = J_R(i, j) // // P$$

$$\emptyset_x = I_R(i, j) - (I_R(i, j) // // P)P \quad \emptyset_y = J_R(i, j) - (J_R(i, j) // // P)P$$

où:

$\square // \square$ division entière avec troncature vers l'infini négatif;

\square / \square division entière (n'entraînant pas de perte de précision dans ce cas);

P précision de déplacement en x et y ($P = 2$ lorsque WDA = $\square 10 \square$ et $P = 16$ lorsque WDA = $\square 11 \square$ ou s'il est absent; la définition de WDA est donnée au P.2.1);

$\left(\frac{I_R(i, j)}{P} + \frac{1}{2}, \frac{J_R(i, j)}{P} + \frac{1}{2} \right)$ (x, y) emplacement du point transformé ($I_R(i, j)$ et $J_R(i, j)$ sont tous deux entiers);

$\left(i_R(i, j) + \frac{1}{2}, j_R(i, j) + \frac{1}{2} \right)$ (x, y) emplacement du point d'échantillonnage près du point transformé ($i_R(i, j)$ et $j_R(i, j)$ sont tous deux entiers);

$(\emptyset_x, \emptyset_y)$ coefficients d'interpolation bilinéaire du point transformé (\emptyset_x et \emptyset_y sont tous deux entiers).

On peut simplifier la résolution de cette équation en remplaçant les divisions par des opérations de décalage binaire, puisque $64H'/P = 2^{m+2}$ lorsque $P = 16$ et $64H'/P = 2^{m+5}$ lorsque $P = 2$.

A l'aide de ces paramètres, la valeur d'échantillon $E_P(i, j)$ du pixel situé en $(x, y) = \left(i + \frac{1}{2}, j + \frac{1}{2} \right)$ dans l'image rééchantillonnée s'obtient par interpolation bilinéaire:

$$E_P(i, j) = \left((P - \emptyset_y) \left((P - \emptyset_x) E_R(i_R, j_R) + \emptyset_x E_R(i_R + 1, j_R) \right) \right. \\ \left. + \emptyset_y \left((P - \emptyset_x) E_R(i_R, j_R + 1) + \emptyset_x E_R(i_R + 1, j_R + 1) \right) + P^2 / 2 - 1 + RCRPR \right) / P^2$$

où \square / \square désigne une division par troncature. i_R et j_R sont les notations simplifiées de $i_R(i, j)$ et $j_R(i, j)$, et $E_R(i_R, j_R)$ désigne la valeur d'échantillon du pixel situé en $(x, y) = \left(i_R + \frac{1}{2}, j_R + \frac{1}{2} \right)$ dans

l'image de référence après extrapolation à l'aide du mode de remplissage qui convient, le cas échéant. La valeur du paramètre RCRPR est définie de la manière suivante:

- pour une image B (ou la partie B d'une trame PB améliorée) dont l'image d'ancrage suivante dans le temps est une image P, RCRPR est égal au bit de type d'arrondissement (RTYPE)

dans MPPTYPE (voir 5.1.4.3) de cette image P suivante dans le temps. Cela sous-entend que pour les trames PB améliorées, RCRPR a la même valeur pour la partie P et la partie B;

- pour les autres types d'image, RCRPR est égal au bit RTYPE de l'image en cours.

P.4 Exemple de mise en œuvre

Le présent sous-paragraphe présente un exemple de mise en œuvre de l'algorithme décrit au sous-paragraphe précédent sous forme de pseudo-code.

P.4.1 Déplacements de points virtuels

Lorsqu'une grande image est codée, l'application directe de l'équation permettant d'obtenir les paramètres u_x^{RB} et u_y^{RB} définis au P.3 peut nécessiter de faire intervenir des variables requérant plus de 32 bits pour leur représentation binaire. Pour les systèmes qui ne peuvent pas utiliser facilement de registres à nombres entiers ou à virgule flottante à 64 bits, nous donnons ci-dessous un exemple d'algorithme qui ne requiert pas de variables ayant plus de 32 bits pour le calcul de u_x^{RB} et u_y^{RB} .

Puisque H , V , H' et V' sont divisibles par 4, la définition de u_x^{RB} peut s'écrire:

$$u_x^{RB} = \left((V_Q - V_{Q'}) \left((H_Q - H_{Q'}) u_x^{00} + H_{Q'} u_x^{H0} \right) + V_{Q'} \left((H_Q - H_{Q'}) u_x^{0V} + H_{Q'} u_x^{HV} \right) \right) // A$$

où $H_Q = H/4$, $V_Q = V/4$, $H_{Q'} = H'/4$, $V_{Q'} = V'/4$, $A = H_Q V_Q$, et $\blacksquare // \blacksquare$ désigne une division entière qui arrondit le quotient au nombre entier le plus proche et qui arrondit les valeurs demi-entières au nombre entier suivant en s'éloignant de zéro. Ensuite, les paramètres T_T et T_B sont définis par

$$T_T = (H_Q - H_{Q'}) u_x^{00} + H_{Q'} u_x^{H0}$$

$$T_B = (H_Q - H_{Q'}) u_x^{0V} + H_{Q'} u_x^{HV}$$

pour la simplicité de la description. A l'aide de l'opérateur $\blacksquare // \blacksquare$ qui désigne la division entière avec troncature vers l'infini négatif et l'opérateur $\blacksquare \% \blacksquare$ défini par $a \% b = a - (a // b) b$, on peut obtenir la valeur de u_x^{RB} par l'intermédiaire du pseudo-code suivant:

```

q = (V_Q - V_Q') * (T_T // A) + V_Q' * (T_B // A) + ((V_Q - V_Q') * (T_T % A) + V_Q' * (T_B % A)) // A;
r = ((V_Q - V_Q') * (T_T % A) + V_Q' * (T_B % A)) % A;
if (q < 0)
    u_x^{RB} = q + (r + (A - 1) / 2) / A;
sinon
    u_x^{RB} = q + (r + A / 2) / A;

```

On peut également calculer la valeur de u_y^{RB} à l'aide de cet algorithme.

P.4.2 Algorithme de rééchantillonnage

Pour simplifier la description de l'algorithme, on définit une fonction `prior_sample` (échantillon antérieur). Cette fonction sert à produire une valeur de pixel pour n'importe quel point entier (i_p, j_p) par rapport à la grille d'échantillonnage de l'image de référence précédente:

```

clip(x_min, x, x_max) {
    if (x < x_min) {
        return x_min;
    } else if (x > x_max) {
        return x_max;
    } else {

```

```

        return x;
    }
}
prior_sample (ip, jp) {
    if (FILL_MODE = clip) {
        ic = clip (0, ip, S*HR / 2-1);
        jc = clip (0, jp, S*VR / 2-1);
        return prior_ref[ic, jc];
    } else {
        if ((ip < 0) OR (ip > S*HR / 2-1) or (jp < 0) OR (jp > S*VR / 2-1) {
            return fill_value;
        } else {
            return prior_ref[ip, jp];
        }
    }
}
}

```

Dans le pseudo-code, `prior_ref [i, j]` représente l'échantillon situé à la colonne i et la rangée j dans l'image de référence précédente dans le temps.

On définit ensuite une fonction filtre qui met en œuvre l'interpolation bilinéaire décrite au P.3. On suppose que tous les arguments de la fonction suivante sont entiers et que les coefficients de l'interpolation bilinéaire \emptyset_x et \emptyset_y sont quantifiés dans la gamme 0, ..., $P-1$ (inclusivement).

```

filter(x0, y0,  $\emptyset_x$ ,  $\emptyset_y$ ) {
    return [(P- $\emptyset_y$ ) * ((P- $\emptyset_x$ ) * prior_sample(x0, y0) +  $\emptyset_x$  * prior_sample(x0+1, y0)) +
             $\emptyset_y$  * ((P- $\emptyset_x$ ) * prior_sample(x0, y0+1) +  $\emptyset_x$  * prior_sample(x0+1, y0+1))] +
            P2 / 2 - 1 + RCRPR / P2;
}

```

Enfin, on peut à partir de ces fonctions définir la méthode servant à déformer l'image de référence pour produire une prédiction de l'image en cours. Les pixels de l'image de prédiction peuvent être produits dans un ordre de balayage récurrent. On suppose que les valeurs $u_x^L(j)$, $u_y^L(j)$, $u_x^R(j)$, et $u_y^R(j)$ sont déjà calculées et entrées dans les variables u_x^L , u_y^L , u_x^R , et u_y^R . Le paramètre D étant défini par $D = 64H' / P$ et sachant que $H' = 2^m$, on obtient les valeurs d'échantillon des pixels de la $j^{\text{ème}}$ rangée du champ rééchantillonné (la rangée la plus haute est par définition la rangée numéro 0) à l'aide du pseudo-code suivant:

```

axi = D * P + 2 * (uxR - uxL);
ayi = 2 * (uyR - uyL);
ax = uxL * S * 2m + (uxR - uxL); + D / 2;
ay = j * D * P + uyL * S * 2m + (uyR - uyL); + D / 2;
pour (i = 0; i < S * H / 2; i++) {
    IR = ax /// D;
    JR = ay /// D;
    IR = IR /// P;
}

```

```

jR = JR /// P ;
Øx = IR - (iR * P) ;
Øy = JR - (jR * P) ;
new_ref[i, j] = filter(iR, jR, Øx, Øy) ;
ax += axi ;
ay += ayi ;
}

```

où toutes les variables utilisées dans ce code sont entières et new_ref (nouvelle_réf) = [i, j] indique l'échantillon engendré pour la colonne *i* et la rangée *j* dans l'image de référence rééchantillonnée. D'après la définition des paramètres, toutes les divisions dans ce code peuvent être remplacées par des opérations de décalage binaire. Par exemple, si $P = 16$;

```

IR = ax /// D ;
JR = ay /// D ;
iR = IR /// P ;
jR = JR /// P ;
Øx = IR - (iR * P) ;
Øy = JR - (jR * P) ;

```

peuvent s'écrire, en supposant que a_x , a_y , I_R , et J_R soient des variables entières codées en binaire en représentation de complément à deux, sous la forme:

```

IR = ax >> (m+2) ;
JR = ay >> (m+2) ;
iR = IR >> 4 ;
jR = JR >> 4 ;
Øx = IR & 15 ;
Øy = JR & 15 ;

```

où " \gg " $N_{\text{décalage}}$ désigne un décalage binaire arithmétique de $N_{\text{décalage}}$ bits vers la droite ($N_{\text{décalage}}$ étant un entier positif), et "&" désigne une opération ET au niveau du bit.

P.5 Rééchantillonnage de facteur quatre

Le rééchantillonnage de facteur quatre, qui modifie d'un facteur 2 ou $\frac{1}{2}$ la dimension horizontale et la dimension verticale de l'image, est un cas particulier de l'algorithme de rééchantillonnage décrit au P.3. La description simplifiée de l'algorithme de rééchantillonnage pour ce cas particulier est donnée dans le présent sous-paragraphe.

La valeur du paramètre RCRPR utilisée dans les Figures P.4 à P.6 est déterminée par le bit de type d'arrondi (RTYPE) dans MPPTYPE (voir 5.1.4.3) tel qu'il est indiqué au P.3. De plus, le symbole " \lceil " dans les figures indique une division par troncature.

P.5.1 Suréchantillonnage de facteur quatre

La méthode d'interpolation des valeurs de pixel utilisée dans le cas d'un suréchantillonnage de facteur quatre pour les pixels internes est illustrée à la Figure P.4. En supposant qu'il existe des pixels à l'extérieur de l'image d'après le mode de remplissage sélectionné (voir P.2.3 et P.2.4), on utilise la même méthode d'interpolation pour les pixels périphériques. La Figure P.5 illustre la méthode d'interpolation utilisée pour les pixels périphériques lorsque *écrtage* est le mode de remplissage sélectionné. Or, pour que le suréchantillonnage de facteur quatre soit précis, les

déplacements en x et y doivent avoir une précision d'au moins $\frac{1}{4}$ de pixel; le champ de précision de déplacement de distorsion (WDA) défini au P.2.1 doit donc être mis à "11" ou bien le rééchantillonnage doit être appelé implicitement pour que cette méthode de suréchantillonnage soit utilisée.

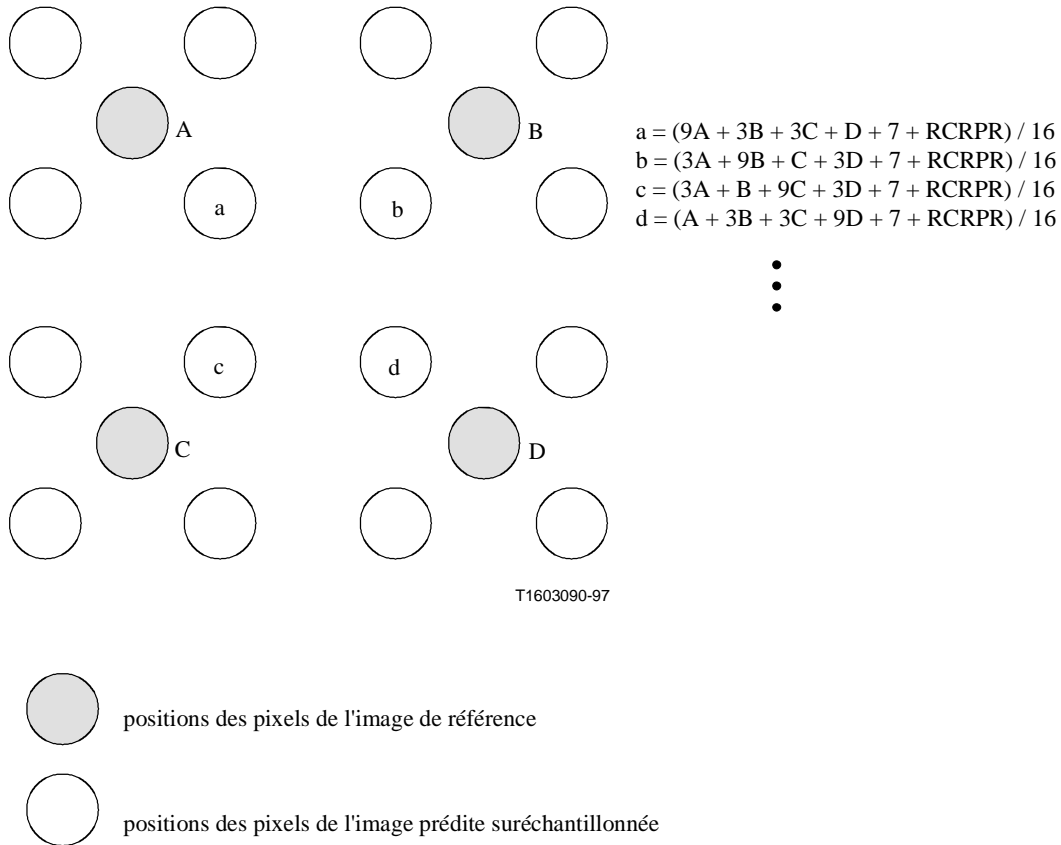


Figure P.4/H.263 – Suréchantillonnage de facteur quatre pour les pixels à l'intérieur de l'image

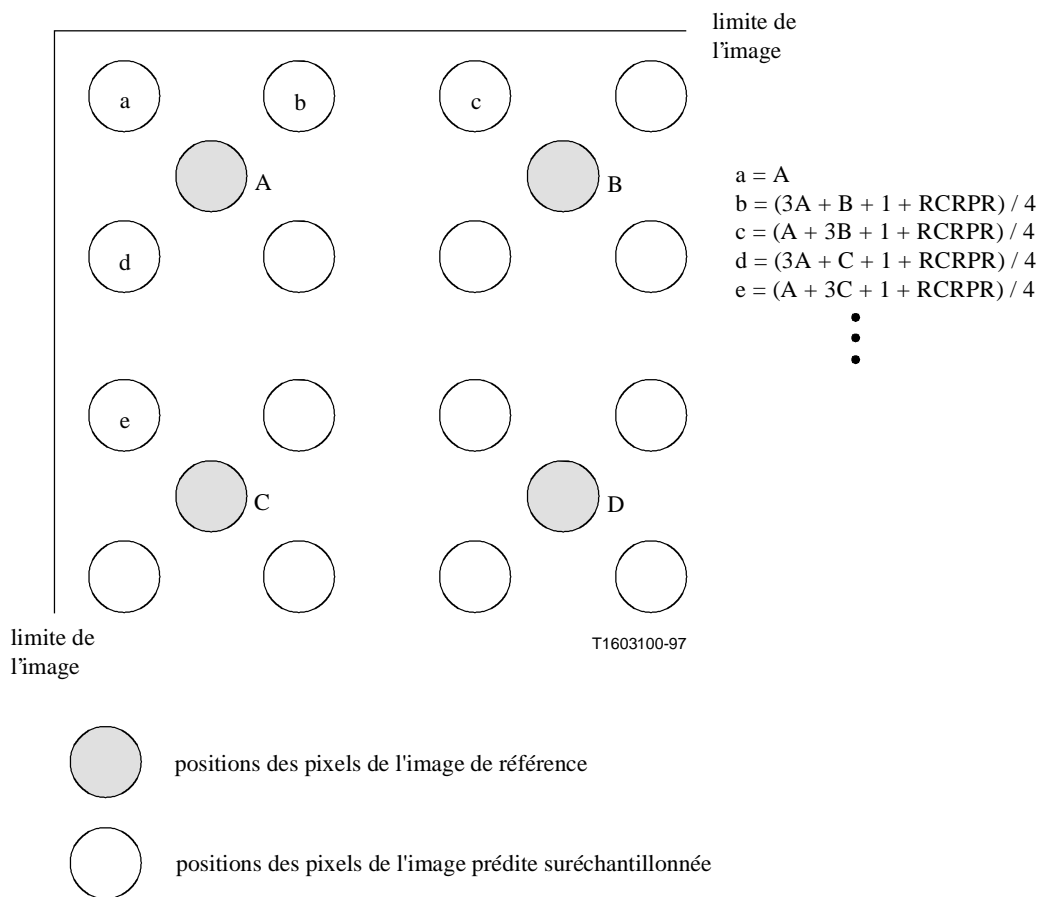


Figure P.5/H.263 – Suréchantillonnage de facteur quatre pour les pixels à la limite de l'image (mode de remplissage = *écrêtage*)

P.5.2 Sous-échantillonnage de facteur quatre

La méthode d'interpolation des valeurs de pixel utilisée dans le cas d'un sous-échantillonnage de facteur quatre est illustrée à la Figure P.6. Puisqu'une précision de $\frac{1}{2}$ pixel sur les déplacements en x et en y est suffisante pour que le sous-échantillonnage de facteur quatre soit précis, le champ de précision de déplacement (WDA) défini au P.2.1, s'il est présent, peut prendre les deux valeurs "10" et "11".

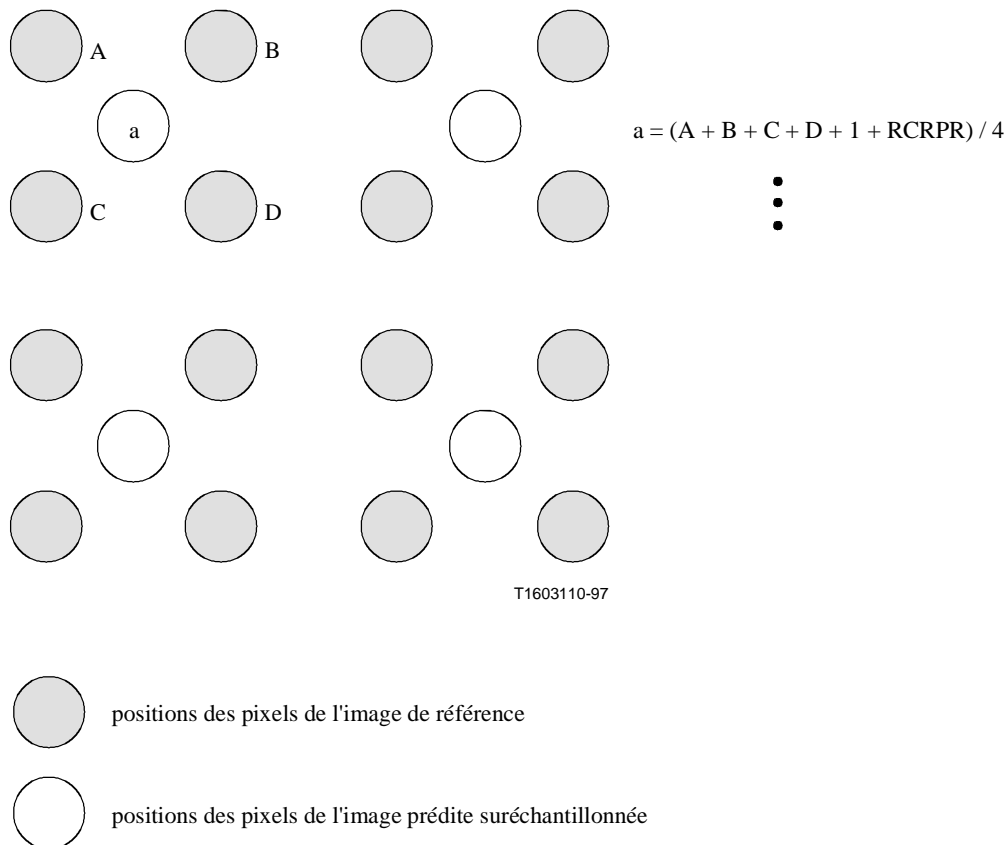


Figure P.6/H.263 – Sous-échantillonnage de facteur quatre

ANNEXE Q

Mode de rafraîchissement à résolution réduite

Q.1 Introduction

La présente annexe décrit un mode facultatif de rafraîchissement à résolution réduite de la présente Recommandation. La possibilité d'utiliser ce mode est signalée par des moyens externes (par exemple, selon la Recommandation H.245). L'utilisation de ce mode est indiquée dans le champ PLUSPTYPE de l'en-tête d'image.

Le mode de rafraîchissement à résolution réduite est prévu pour le codage de scènes très animées et son utilisation permet d'augmenter la fréquence d'image de codage tout en maintenant une qualité subjective suffisante. Ce mode permet au codeur d'envoyer des informations de rafraîchissement pour une image codée à résolution réduite tout en préservant le détail dans une image de référence à résolution plus élevée pour créer une image finale ayant la résolution plus élevée.

Dans ce mode, la syntaxe du flux binaire est identique à la syntaxe de codage sans le mode, mais la sémantique, ou interprétation du flux binaire, est assez différente et la partie de l'image couverte par un macrobloc est deux fois plus large et deux fois plus haute. Par conséquent, le nombre de macroblocs est à peu près égal au quart de ce qu'il serait sans ce mode. Les données sur les vecteurs cinétiques se rapportent aussi à des blocs dont la hauteur et la largeur sont doubles des hauteur et largeur normales, c'est-à-dire 32×32 et 16×16 au lieu de 16×16 et 8×8 . Par ailleurs, on devrait envisager les données DCT ou de texture comme décrivant des blocs 8×8 sur une version de l'image à résolution réduite. Pour produire l'image finale, les données de texture sont décodées à résolution

réduite puis suréchantillonnées jusqu'à la résolution intégrale de l'image. Après le suréchantillonnage, l'image de texture à résolution intégrale est ajoutée à l'image à compensation de mouvement (déjà à résolution intégrale) pour donner l'image qui sera affichée et servira de référence ultérieure.

Dans ce mode, l'image finale créée pour l'affichage est une image de dimension horizontale H et de dimension verticale V comme il est indiqué dans l'en-tête d'image.

Dans ce mode, une image référencée utilisée pour la prédiction et créée pour un décodage ultérieur a pour dimension horizontale H_R et pour dimension verticale V_R comme dans le mode par défaut défini au 4.1. Autrement dit, H_R et V_R s'écrivent:

$$H_R = ((H + 15) / 16) * 16$$

$$V_R = ((V + 15) / 16) * 16$$

où H et V sont la dimension horizontale et la dimension verticale indiquées dans l'en-tête d'image et où $\lceil \cdot \rceil$ représente la division par troncature.

Ensuite dans la présente annexe, la texture est codée à résolution réduite avec une largeur H_C et une hauteur V_C telles que

$$H_C = ((H_R + 31) / 32) * 32$$

$$V_C = ((V_R + 31) / 32) * 32$$

$\lceil \cdot \rceil$ étant la division par troncature.

Si H_C et H_R , ou V_C et V_R ne sont pas identiques entre eux, comme dans le cas du format QCIF, une extension de l'image référencée est effectuée et l'image est décodée de la même manière que si sa largeur et sa hauteur étaient H_C et V_C . L'image résultante qui est quadrillée de macroblocs $32 * 32$ est rognée sur la droite jusqu'à la largeur H_R et en bas jusqu'à la hauteur V_R , et cette image rognée est enregistrée comme image de référence pour décodage ultérieur.

Si H et V sont tous deux identiques aux dimensions correspondantes d'une image résultante ayant pour largeur H_C et pour hauteur V_C , cette image résultante est utilisée pour l'affichage. Sinon, elle est encore rognée jusqu'aux dimensions $H * V$ et l'image découpée ici n'est utilisée que pour l'affichage.

Si le mode d'échelonnabilité temporelle, SNR, ou spatiale (Annexe O) ou le mode de rééchantillonnage d'image de référence (Annexe P) est également utilisé avec cette option, le format de source de l'image en cours pourrait être différent de celui de l'image de référence. Dans ce cas, le rééchantillonnage d'image de référence doit être effectué avant le décodage.

NOTE – Ce mode peut être utilisé avec le mode de sélection d'image de référence (voir l'Annexe N) sans modification, puisque l'image de référence (après rééchantillonnage éventuel à l'aide du mode de rééchantillonnage d'image de référence) a les mêmes dimensions que celles qui sont indiquées dans l'en-tête de l'image en cours lorsqu'on utilise cette option.

Q.2 Procédure de décodage

La Figure Q.1 représente le schéma fonctionnel du décodage par blocs dans le mode de rafraîchissement à résolution réduite.

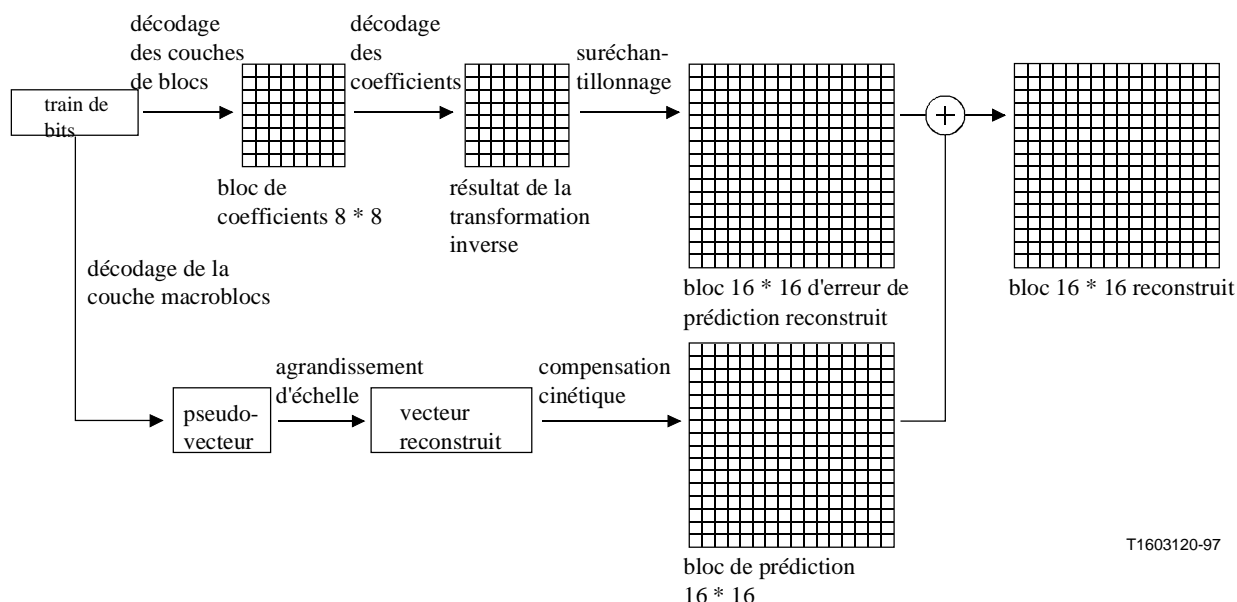


Figure Q.1/H.263 – Schéma fonctionnel du décodage par blocs dans le mode de rafraîchissement à résolution réduite

La procédure de décodage est donnée dans les sous-paragraphes qui suivent:

Q.2.1 Préparation de la référence

Dans certains cas, l'image de référence dont on dispose a des dimensions qui diffèrent de H_c et V_c . L'image de référence doit alors être convertie avant le décodage, conformément au Q.2.1.1 ou Q.2.1.2.

Q.2.1.1 Rééchantillonnage d'image de référence

Si le mode d'échelonnabilité temporelle, SNR, ou spatiale (Annexe O) ou le mode de rééchantillonnage d'image de référence (Annexe P) est également utilisé avec cette option, le format de source de l'image en cours pourrait être différent de celui de l'image de référence. Dans ce cas, le rééchantillonnage d'image de référence doit d'abord être effectué conformément à chaque annexe.

Q.2.1.2 Extension de l'image de référence

Si H_R ou V_R n'est pas divisible par 32, comme dans le cas du format QCIF, l'image de référence est soumise à une extension. La procédure détaillée de cette extension est donnée au Q.3.

Q.2.2 Décodage de la couche des macroblocs

Le décodage peut être envisagé comme un moyen de travailler sur des blocs "agrandis" de dimensions 32×32 dans la luminance et 16×16 dans la chrominance. Les données de texture et les données de mouvement pour chaque bloc agrandi sont décodées pour créer un bloc de mouvement de 32×32 et un bloc de texture de 32×32 , qui sont décrits respectivement aux Q.2.2.1. et Q.2.2.2. Ces blocs de mouvement et de texture sont ensuite ajoutés (voir Q.2.2.3).

Q.2.2.1 Compensation de mouvement

Tout d'abord, chaque composante du vecteur cinétique de macrobloc (ou des quatre vecteurs cinétiques de macrobloc) est formée à partir du champ de données de vecteur cinétique MVD (et éventuellement de MVD_{2-4}). Si l'on est dans le mode de codage par trames PB amélioré, les composantes MV_F et MV_B pour l'image B sont également formées à partir du champ de données de

vecteur cinétique pour macrobloc de type B MVDB. La procédure détaillée pour la formation des vecteurs cinétiques est donnée au Q.4. Si l'image en cours est une image B ou une image EP, les vecteurs cinétiques anticipé et différé sont également obtenus comme il est indiqué au Q.4.

Le vecteur cinétique pour les deux blocs de chrominance du macrobloc s'obtient à partir du vecteur cinétique de macrobloc comme indiqué au 6.1.1. Si l'on utilise le mode de prédiction avancé ou le mode avec filtre de dégroupage et si quatre vecteurs cinétiques sont donc définis pour le macrobloc, le vecteur cinétique pour les deux blocs de chrominance s'obtient à partir des quatre vecteurs cinétiques comme indiqué au F.2. Dans le mode de codage par trames PB amélioré, la création du vecteur de chrominance est décrite à l'Annexe M. Pour une image B ou EP, la création du vecteur de chrominance est décrite à l'Annexe O.

Une prédiction est ensuite faite à partir du vecteur cinétique pour un macrobloc INTER. Quatre blocs 16×16 de prédiction de luminance sont obtenus à partir du vecteur cinétique de macrobloc et deux blocs 16×16 de prédiction de chrominance sont obtenus à partir du vecteur cinétique de chrominance. L'interpolation pour prédire les fractions de pixel est expliquée au 6.1.2. Si l'on utilise également le mode de prédiction avancé, une compensation agrandie de mouvement par superposition de blocs permet d'obtenir quatre blocs 16×16 de prédiction de luminance à l'aide de matrices de pondération agrandies (voir la procédure détaillée au Q.5). Si le mode de l'image en cours est le mode de codage par trames PB amélioré, ou si l'on utilise des images B ou des images EP, la prédiction est obtenue selon les autres annexes correspondantes, à l'exception près que les blocs prédits sont de dimensions 16×16 au lieu de 8×8 .

Q.2.2.2 Décodage de texture

Tout d'abord, le flux binaire de la couche des blocs est décodé comme indiqué au 5.4. Ensuite, les coefficients sont décodés et les blocs 8×8 d'erreur de prédiction reconstruits à résolution réduite sont le résultat de la transformation inverse comme indiqué au 6.2.

Les blocs 16×16 d'erreur de prédiction reconstruits sont ensuite obtenus par suréchantillonnage des blocs 8×8 d'erreur de prédiction reconstruits à résolution réduite. Pour la création des pixels périphériques dans chaque bloc 16×16 d'erreur de prédiction reconstruit, seuls sont utilisés les pixels qui appartiennent au bloc correspondant. La procédure détaillée est donnée au Q.6.

Q.2.2.3 Reconstruction des blocs

Pour chaque bloc de luminance et de chrominance, la somme de la prédiction et de l'erreur de prédiction est effectuée. La procédure est identique à celle du 6.3.1, à l'exception près que les blocs sont de dimensions 16×16 au lieu de 8×8 . L'écritage est ensuite effectué comme au 6.3.2.

Un filtre de limite de bloc est ensuite appliqué aux pixels périphériques des blocs 16×16 reconstruits. La procédure détaillée est donnée au Q.7.

Q.2.3 Enregistrement de l'image

Si H_R et V_R sont tous deux divisibles par 32, comme dans le cas du format CIF, l'image reconstruite résultante décrite au Q.2.2 est enregistrée comme image de référence et pour un décodage ultérieur. Sinon, comme dans le cas du format QCIF, l'image reconstruite qui est juste couverte de macroblocs 32×32 est rognée sur la droite et en bas jusqu'à la largeur H_R et la hauteur V_R et cette image rognée est enregistrée comme image de référence pour décodage ultérieur.

Q.2.4 Affichage

Si H et V sont identiques à H_C et V_C , l'image obtenue au Q.2.2 est utilisée telle quelle pour l'affichage. Sinon, cette image résultante est encore rognée aux dimensions $H \times V$ et uniquement utilisée pour l'affichage.

Q.3 Extension de l'image de référence

Si H_r ou V_r ne sont pas divisibles par 32, comme dans le cas du format QCIF, l'extension de l'image de référence est effectuée avant le décodage de la couche des macroblocs/blocs. La largeur et la hauteur de l'image de référence étendue sont égales au multiple de 32 immédiatement supérieur pour la luminance et au multiple de 16 immédiatement supérieur pour la chrominance.

NOTE – La largeur et la hauteur de l'image de référence dans le mode par défaut sont toujours étendues pour être divisibles par 16 même si le format d'image a une largeur ou une hauteur qui n'est pas divisible par 16, parce que l'image doit être décodée comme si la largeur ou la hauteur était égale au multiple de 16 immédiatement supérieur. Voir 4.1.

Dans le cas où, ni le mode de codage par vecteurs cinétiques non restreints, ni le mode de prédiction avancé, ni le mode avec filtre de dégroupage ne sont utilisés avec cette option, les pixels étendus peuvent avoir des valeurs arbitraires car ils ne seront jamais utilisés comme pixels de référence de l'image décodée qui doit être reconstruite et affichée.

Si le mode de codage par vecteurs cinétiques non restreints, le mode de prédiction avancé ou le mode avec filtre de dégroupage est également utilisé avec cette option, l'extension de l'image de référence consiste à reproduire en double exemplaire le pixel périphérique de l'image de référence afin de s'assurer que le décodage ait lieu lorsque les vecteurs cinétiques pointent vers l'extérieur de la limite droite ou inférieure de l'image.

Par exemple, si l'on utilise le mode de rafraîchissement à résolution réduite pour un format QCIF, la largeur de l'image de référence est 176 et sa hauteur 144, deux nombres qui ne sont pas divisibles par 32. Pour couvrir une image QCIF avec des macroblocs de dimensions 32×32 , le nombre de macroblocs dans une rangée doit être égal à 6 et le nombre de macroblocs dans une colonne doit être égal à 5. Par conséquent, la largeur de l'image de référence étendue est 192 et sa hauteur est 160.

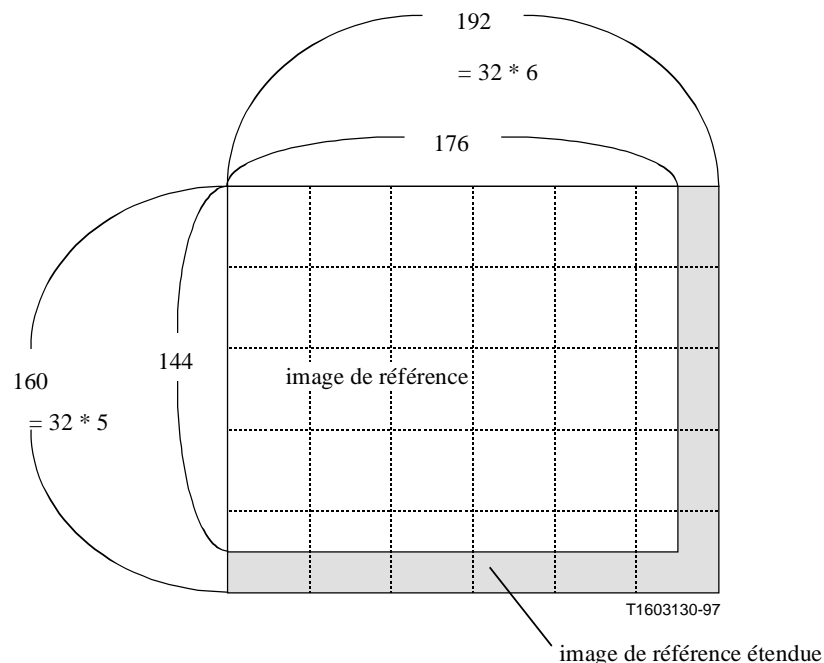


Figure Q.2/H.263 – Extension de l'image de référence pour un format d'image QCIF

La Figure Q.2 illustre l'extension de l'image de référence dans le format QCIF. L'image de référence étendue pour la luminance est donnée par la formule suivante:

$$R_{RRU}(x, y) = R(x', y')$$

où:

$$\begin{aligned}x, y &= \text{coordonnées spatiales de l'image de référence étendue dans le domaine des pixels,} \\x', y' &= \text{coordonnées spatiales de l'image de référence dans le domaine des pixels,} \\R_{RRU}(x, y) &= \text{valeur en pixels de l'image de référence étendue en } (x, y), \\R(x', y') &= \text{valeur en pixels de l'image de référence en } (x', y'), \\x' &= \begin{cases} 175 & \text{si } x > 175 \text{ et } x < 192 \\ x & \text{dans les autres cas,} \end{cases} \\y' &= \begin{cases} 143 & \text{si } y > 143 \text{ et } y < 160 \\ y & \text{dans les autres cas.} \end{cases}\end{aligned}$$

Les images de référence pour la chrominance sont également étendues de cette manière.

Q.4 Reconstruction des vecteurs cinétiques

Dans le mode de rafraîchissement à résolution réduite, la gamme des vecteurs cinétiques est pratiquement deux fois plus grande dans le sens horizontal et dans le sens vertical. Pour obtenir une gamme agrandie à l'aide du codage VLC pour MVD défini au Tableau 14, chaque composante de vecteur ne peut prendre pour valeur que zéro ou un demi-pixel. La gamme de chaque composante de vecteur cinétique est donc $[-31,5, 30,5]$ dans le mode de rafraîchissement à résolution réduite par défaut. Si l'on utilise le mode de codage par vecteurs cinétiques non restreints, la gamme $[-\text{limite}, \text{limite}]$ définie au D.2 s'applique aux pseudo-vecteurs cinétiques et se traduit par une gamme de $[-(2 * \text{limite}-0,5), 2 * \text{limite}-1,5]$ pour les vecteurs cinétiques. Pour le format CIF, cela donne une gamme de $[-32, 31,5]$ pour les pseudo-vecteurs cinétiques et une gamme de $[-63,5, 62,5]$ pour les vecteurs cinétiques. Si le champ UII est mis à "01", les vecteurs cinétiques sont illimités. Par contre, les vecteurs cinétiques (et pas seulement les pseudo-vecteurs cinétiques) sont toujours limités de sorte à ne pas pointer vers plus de 15 pixels à l'extérieur de la région codée, comme il est expliqué au D.1.1. La Figure Q.3 illustre les positions possibles du vecteur cinétique de macrobloc ou de quatre prédictions de vecteur cinétique au voisinage de la valeur vectorielle (0, 0). Les lignes en pointillés représentent les coordonnées entières.

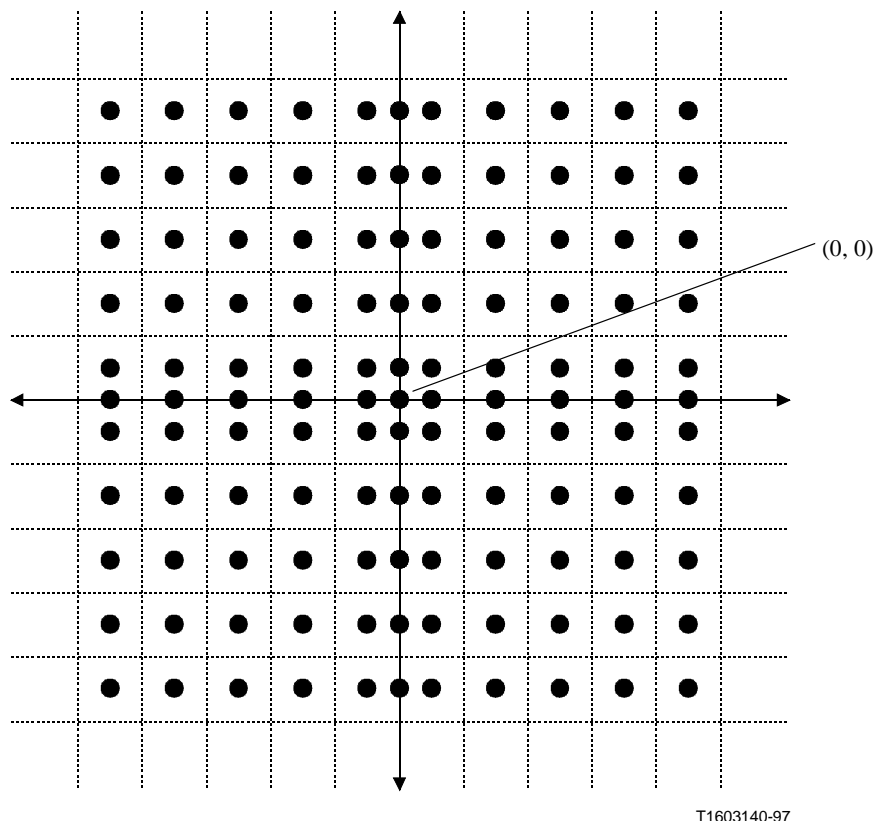


Figure Q.3/H.263 – Reconstruction du vecteur cinétique

Pour le macrobloc utilisant des vecteurs cinétiques différentiels dans une image B, les vecteurs cinétiques de prédiction différée et anticipée sont obtenus indépendamment. Dans le mode de rafraîchissement à résolution réduite, la composante de vecteur cinétique MV_C pour un bloc de luminance est reconstruite à partir de MVD et $MVD_{2.4}$ de la manière suivante:

- 1) la pseudo-composante de vecteur de prédiction **pseudo- P_C** est créée à partir de la composante P_C du vecteur de prédiction:

$$\text{pseudo-}P_C = 0 \quad \text{si } P_C = 0$$

$$\text{pseudo-}P_C = \text{sign}(P_C) * (|P_C| + 0,5) / 2,0 \quad \text{si } P_C \neq 0$$

"/" désigne la division à virgule flottante (sans perte de précision). Par définition, la composante P_C du vecteur de prédiction est la valeur médiane des composantes vectorielles MV_1 , MV_2 et MV_3 définies aux 6.1.1 et F.2;

- 2) la composante **pseudo- MV_C** du pseudo-vecteur de macrobloc s'obtient en ajoutant à pseudo- P_C les différences de vecteurs cinétiques MVD (et $MVD_{2.4}$) du Tableau 14.

Dans le mode par défaut de rafraîchissement à résolution réduite, la valeur de **pseudo- MV_C** est restreinte à la gamme $[-16, 15,5]$. Une seule des deux différences va donner pour pseudo- MV_C une valeur tombant dans la gamme permise. La procédure est similaire à celle qui est décrite au 6.1.1.

Si l'on utilise également le mode de codage par vecteurs cinétiques non restreints avec le mode de rafraîchissement à résolution réduite, on obtient **pseudo- MV_C** en additionnant les différences de vecteurs cinétiques MVD (et $MVD_{2.4}$) du Tableau D.3.

En présence de quatre vecteurs cinétiques, la procédure est similaire à celle qui est décrite au F.2;

- 3) la composante de vecteur cinétique MV_C s'obtient à partir de pseudo- MV_C à l'aide de la formule suivante:

$$MV_C = 0 \quad \text{si pseudo-}MV_C = 0$$

$$MV_C = \text{sign}(\text{pseudo-}MV_C) * (2,0 * |\text{pseudo-}MV_C| - 0,5) \quad \text{si pseudo-}MV_C \neq 0$$

Il s'ensuit que chaque composante vectorielle ne peut prendre qu'une valeur demi-entière ou nulle et la gamme de chaque composante de vecteur cinétique est presque deux fois plus grande que pour les pseudo-vecteurs;

- 4) si l'image en cours est dans le mode de codage par trames PB amélioré, ou si le champ MBTYPE indique le mode direct dans une image B, les composantes de vecteur cinétique MV_F et MV_B sont créées pour la prédiction anticipée et la prédiction différée.

En premier lieu, les composantes de pseudo-vecteurs cinétiques **pseudo- MV_F** et **pseudo- MV_B** sont calculées d'après les règles des modes de prédiction définies aux Annexes O ou M.

Dans le cas de la prédiction bidirectionnelle dans le mode de codage par trames PB amélioré (voir M.2.1) ou si le champ MBTYPE indique le mode direct dans une image B (voir O.5.2), les composantes pseudo- MV_F et pseudo- MV_B sont calculées à partir de **pseudo- MV_D** et **pseudo- MV_C** en supposant que pseudo- MV_D est nulle et que pseudo- MV_C est égale à MV , comme il est indiqué aux Annexes G et M.

Dans le cas d'une prédiction anticipée dans le mode de codage par trames PB amélioré (voir M.2.2), on obtient **pseudo- MV_{DB}** en décodant le code de longueur variable MVDB selon le Tableau 13. Ensuite, on obtient pseudo- MV_F en ajoutant pseudo- MV_{DB} au **pseudo-prédicteur**. Pour former le pseudo-prédicteur, on convertit le prédicteur obtenu par la procédure définie au M.2.2 en vecteur pseudo-prédicteur à l'aide de la formule donnée au 1) du présent sous-paragraphe.

Dans le cas d'une prédiction différée dans le mode de codage par trames PB amélioré (voir M.2.3), pseudo- MV_B est mis à zéro.

On obtient ensuite les vecteurs cinétiques MV_F ou MV_B pour la prédiction anticipée ou différée à partir de pseudo- MV_F ou pseudo- MV_B à l'aide de la formule donnée au 3) du présent sous-paragraphe.

Q.5 Matrices agrandies de compensation de mouvement par superposition de blocs pour la luminance

Si l'on utilise le mode de prédiction avancé avec le mode de rafraîchissement à résolution réduite, on utilise des matrices agrandies de facteurs de pondération pour effectuer la compensation de mouvement par superposition de blocs. A part le fait que chaque bloc et les matrices de pondération sont de dimensions 16×16 , la procédure de création de chaque bloc de prédiction est identique à celle qui est décrite en F.3.

Les matrices agrandies des facteurs de pondération pour la prédiction de luminance 16×16 sont représentées aux Figures Q.4, Q.5 et Q.6.

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

Figure Q.4/H.263 – Facteurs de pondération H_0 pour la prédiction avec le vecteur cinétique du bloc de luminance en cours 16×16

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

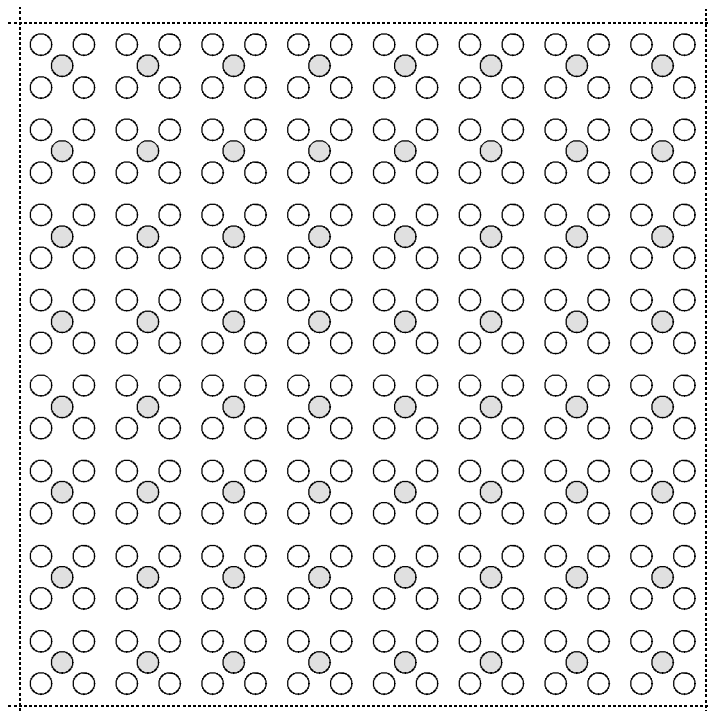
Figure Q.5/H.263 – Facteurs de pondération H_1 pour la prédiction avec vecteur cinétique des blocs de luminance en cours 16×16 en haut ou en bas du bloc de luminance en cours 16×16

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

Figure Q.6/H.263 – Facteurs de pondération H_2 pour la prédiction avec vecteur cinétique des blocs de luminance 16×16 à gauche ou à droite du bloc de luminance en cours 16×16

Q.6 Suréchantillonnage du bloc d'erreur de prédiction reconstruite à résolution réduite

On obtient le bloc 16×16 d'erreur de prédiction reconstruite en suréchantillonnant le bloc 8×8 d'erreur de prédiction reconstruite à résolution réduite. Pour la simplicité de l'opération, le filtrage est fermé à l'intérieur d'un bloc, ce qui permet d'effectuer un suréchantillonnage individuel bloc par bloc. La Figure Q.7 représente la disposition des échantillons. La procédure de suréchantillonnage pour les pixels de luminance et de chrominance qui sont à l'intérieur des blocs 16×16 d'erreur de prédiction reconstruite est décrite au Q.6.1. La procédure pour la création des pixels de luminance et de chrominance qui sont à la limite du bloc 16×16 d'erreur de prédiction reconstruit est décrite au Q.6.2. Les blocs de chrominance de même que les blocs de luminance sont suréchantillonnés. Le symbole $\lceil \rceil$ aux Figures Q.8 et Q.9 indique la division par troncature.



T1603150-97

- position des échantillons dans le bloc 8 * 8 d'erreur de prédiction reconstruite à résolution réduite
- position des échantillons dans le bloc 16 * 16 d'erreur de prédiction reconstruite
- limite de bloc

Figure Q.7/H.263 – Disposition des échantillons dans le bloc 8×8 d'erreur de prédiction reconstruite à résolution réduite et dans le bloc 16×16 d'erreur de prédiction reconstruite

Q.6.1 Procédure de suréchantillonnage pour les pixels à l'intérieur d'un bloc 16×16 d'erreur de prédiction reconstruite

La Figure Q.8 représente la création d'une erreur de prédiction reconstruite pour les pixels à l'intérieur d'un bloc. \bullet / \circ désigne la division par troncature.

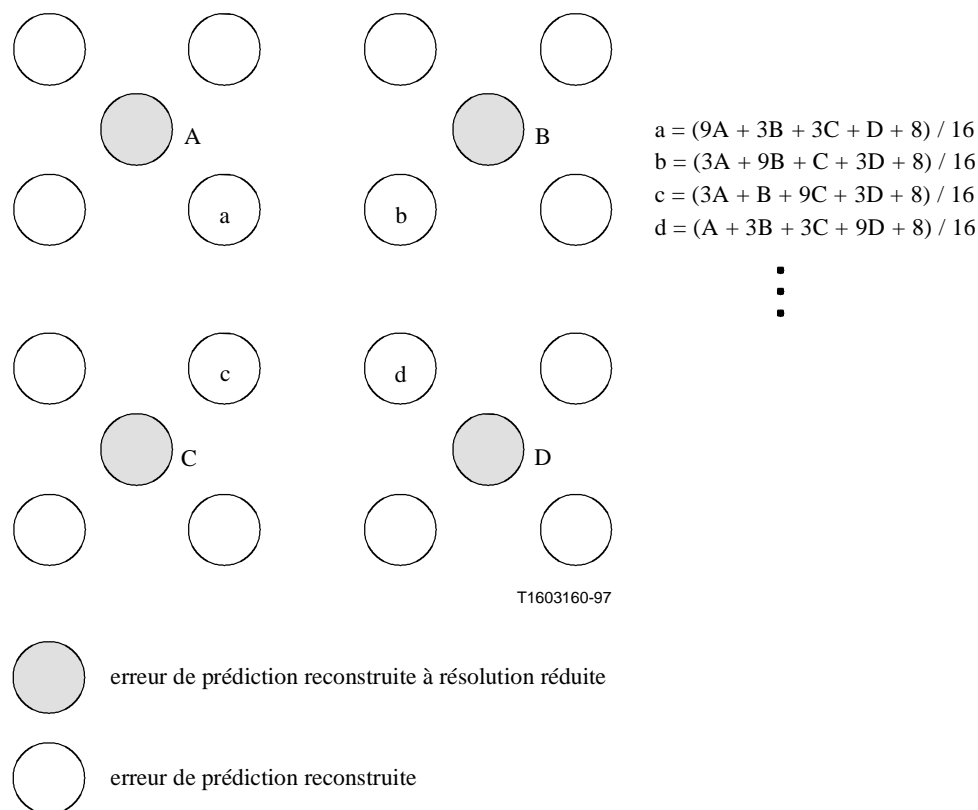


Figure Q.8/H.263 – Création de l'erreur de prédiction reconstruite pour les pixels à l'intérieur du bloc

Q.6.2 Procédure de suréchantillonnage pour les pixels à la limite du bloc 16×16 d'erreur de prédiction reconstruite

La Figure Q.9 représente la création de l'erreur de prédiction reconstruite pour les pixels d'un bloc 16×16 .

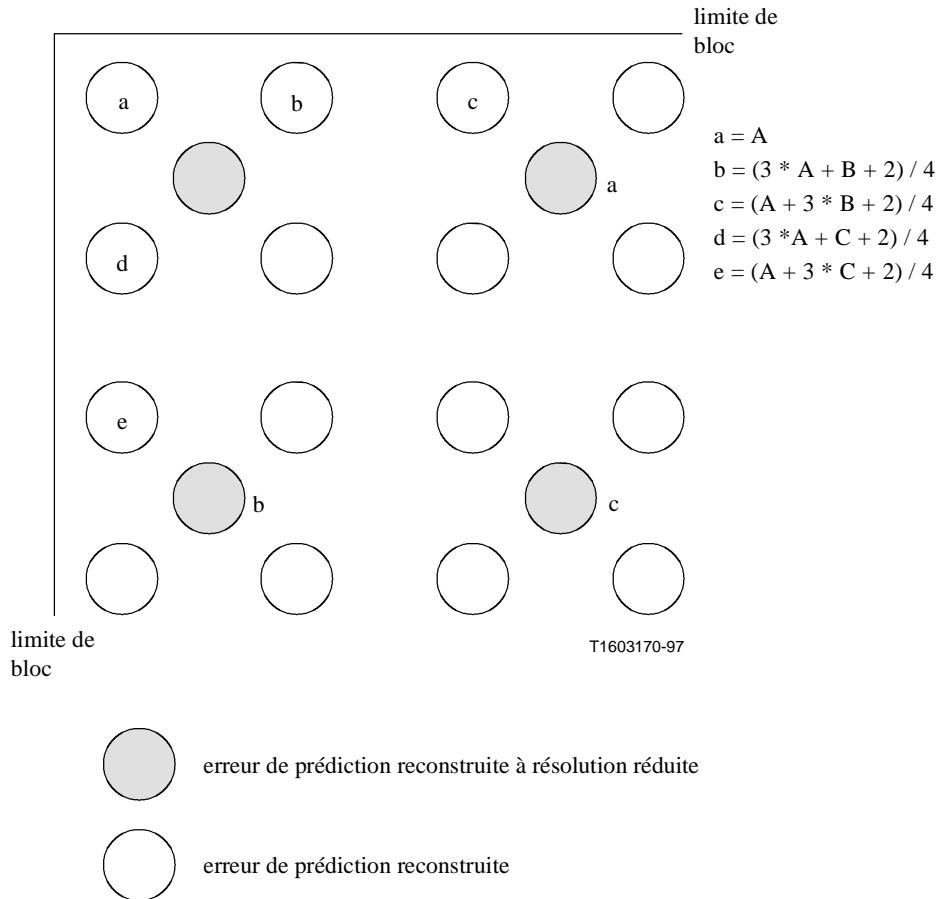


Figure Q.9/H.263 – Création d'une erreur de prédiction reconstruite pour les pixels à la limite du bloc

Q.7 Filtre de limite de bloc

Les opérations de filtrage sont effectuées le long des limites des blocs 16×16 reconstruits du côté du codeur et du côté du décodeur. Il y a deux types de filtrage, selon que le mode avec filtre de dégroupage est utilisé ou non.

Le filtrage par défaut dans le mode de rafraîchissement à résolution réduite s'effectue comme au Q.7.1.

Si l'on utilise également le mode avec filtre de dégroupage avec le mode de rafraîchissement à résolution réduite, le filtrage s'effectue comme au Q.7.2.

Dans les deux cas, le filtrage est effectué sur la totalité des données de l'image reconstruite avant que les données soient enregistrées dans la mémoire d'image en vue de prédictions ultérieures. Aucun filtrage n'est effectué à travers les limites d'image, les limites de tranche dans le mode structuré par tranches (voir l'Annexe K), ni les limites de groupe de blocs ayant des en-têtes GOB présents dans le

mode de décodage par segments indépendants (voir l'Annexe R). Les données de chrominance, de même que les données de luminance, sont filtrées.

Q.7.1 Définition du filtre par défaut de limite de bloc

Dans le mode de rafraîchissement à résolution réduite, le filtrage par défaut est effectué selon la description donnée au présent sous-paragraphe.

Soit A et B deux valeurs de pixel sur une ligne – horizontale ou verticale – de l'image reconstruite; A appartient à un bloc 16×16 appelé bloc1 alors que B appartient à un bloc 16×16 voisin appelé bloc2 qui est situé à droite ou en dessous du bloc1. La Figure Q.10 donne des exemples de disposition de ces pixels.

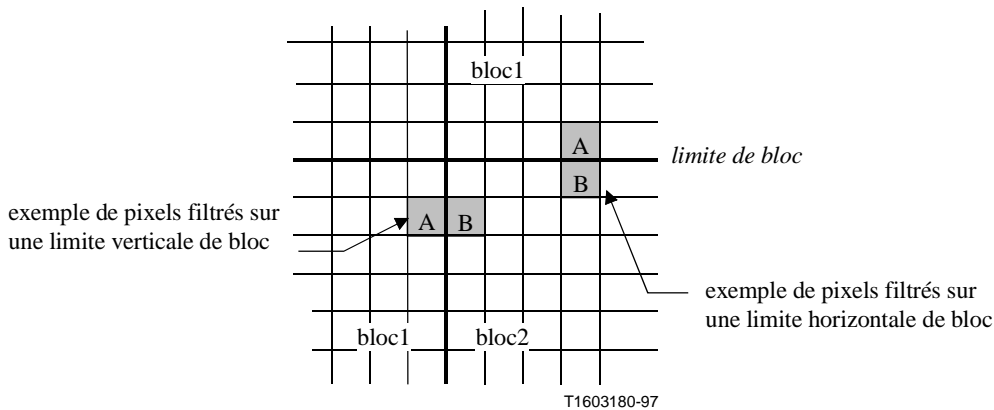


Figure Q.10/H.263 – Filtre par défaut de limite de bloc

Une des conditions suivantes doit être remplie pour actionner le filtre pour une limite particulière:

- le bloc1 appartient à un macrobloc codé (COD==0 || MB-type == INTRA);
- le bloc2 appartient à un macrobloc codé (COD==0 || MB-type == INTRA).

A sera remplacé par A1 et B par B1. $\lceil \rceil / \lfloor \rfloor$ désigne la division par troncature.

$$A1 = (3 * A + B + 2) / 4$$

$$B1 = (A + 3 * B + 2) / 4$$

L'ordre de filtrage des limites est identique à celui qui figure dans la description donnée au J.3.

Q.7.2 Définition du filtre de limite de bloc lorsque le mode avec filtre de dégroupage est utilisé

Si l'on utilise le mode avec filtre de dégroupage (voir l'Annexe J) avec le mode de rafraîchissement à résolution réduite, le filtrage défini à l'Annexe J est effectué, avec une modification, sur les pixels périphériques des blocs 16×16 de luminance et de chrominance, à la place du filtrage décrit au Q.7.1. La modification apportée au filtrage de l'Annexe J consiste à donner au paramètre STRENGTH la valeur de l'infini positif, ce qui implique que la fonction UpDownRamp(x, STRENGTH) définie au J.3 devient une fonction linéaire de x.

Il s'ensuit que la procédure de filtre de dégroupage décrite au J.3 est redéfinie de la manière suivante:

$$B1 = \text{clip}(B + d1)$$

$$C1 = \text{clip}(C - d1)$$

$$A1 = A - d2$$

$$D1 = D + d2$$

$$d1 = (A - 4B + 4C - D) / 8$$

$$d2 = \text{clipd1}((A - D) / 4, d1/2)$$

ANNEXE R

Mode de décodage par segments indépendants

R.1 Introduction

La présente annexe décrit le mode facultatif de décodage par segments indépendants de la présente Recommandation, qui permet de décoder une image sans qu'il existe de dépendance de données entre les limites de tranches ou les limites de groupes de blocs ayant des en-têtes GOB non vides. L'utilisation de ce mode est signalée dans le champ PLUSPTYPE de l'en-tête d'image. La possibilité d'utiliser ce mode facultatif est déterminée par des moyens externes (par exemple, selon la Recommandation H.245).

Lorsque l'utilisation de ce mode est indiquée, les limites de segment de l'image vidéo (correspondant aux limites des tranches, ou aux limites supérieures des groupes de blocs pour lesquels des en-têtes GOB sont envoyées, ou encore aux limites de l'image, selon les limites qui donnent la plus petite région délimitée) sont traitées comme des limites d'image lors du décodage, y compris lors du traitement des vecteurs cinétiques qui traversent ces limites (qui entraînent une extrapolation aux limites lorsque le mode de codage par vecteurs cinétiques non restreints, le mode de prédiction avancé, le mode avec filtre de dégroupage ou le mode d'échelonnabilité temporelle, SNR ou spatiale sont utilisés, et qui sont interdits lorsqu'aucun de ces modes facultatifs n'est utilisé).

R.2 Fonctionnement du mode

Un segment d'image vidéo est défini de la manière suivante:

si le mode structuré par tranches (voir l'Annexe K) n'est pas utilisé, un groupe de blocs (GOB) ou plusieurs GOB consécutifs forment un segment d'image vidéo. Le sommet de chaque segment d'image vidéo est indiqué par la présence d'un en-tête de GOB non vide pour lequel le bord du segment d'image vidéo est situé soit juste au-dessus des macroblocs dans le GOB pour lequel un en-tête est présent, soit en haut de l'image, selon la plus basse des deux positions. Le bas de chaque segment d'image vidéo est défini soit par le haut du segment suivant d'image vidéo, soit par le bas de l'image, selon la plus haute des deux positions;

si le mode structuré par tranches (voir l'Annexe K) est utilisé, chaque tranche forme un segment d'image vidéo.

Dans le mode de décodage par segments indépendants, chaque segment d'image vidéo est décodé de façon totalement indépendante de tous les autres segments d'image vidéo et il est également indépendant de toutes les données à l'extérieur de l'emplacement du même segment d'image vidéo dans l'image ou les images de référence. Ceci implique ce qui suit:

- 1) pas d'utilisation des vecteurs cinétiques à l'extérieur du segment d'image vidéo en cours pour la prédiction des vecteurs cinétiques (comme au 6.1.1);
- 2) pas d'utilisation des vecteurs cinétiques à l'extérieur du segment d'image vidéo en cours en guise de vecteurs cinétiques distants pour la compensation de mouvement par superposition de blocs, lorsque le mode de prédiction avancé est utilisé (voir F.3);
- 3) pas de fonctionnement de filtre de dégroupage à travers les limites de segments d'image vidéo (voir J.3);

- 4) pas d'utilisation des vecteurs cinétiques qui référencent des données à l'extérieur du segment d'image vidéo en cours à moins que l'on n'utilise le mode de codage par vecteurs cinétiques non restreints (voir l'Annexe D), le mode de prédiction avancé (voir l'Annexe F), le mode avec filtre de dégroupage (voir l'Annexe J) ou le mode d'échelonnabilité temporelle, SNR ou spatiale (voir l'Annexe O); dans ce cas, les bords du segment d'image vidéo en cours dans l'image précédente sont extrapolés suivant la description donnée à l'Annexe D pour former des prédictions des pixels qui référencent la région hors limite;
- 5) pas d'interpolation bilinéaire à travers les limites de la région de dimensions $\frac{1}{4}$ ou $\frac{1}{2}$ correspondant au segment d'image vidéo en cours pour la prédiction ascendante dans les images EI et EP d'échelonnabilité spatiale (définies à l'Annexe O);
- 6) pas de fonctionnement du filtre de limite de bloc à travers les limites de segments d'image vidéo lorsque le mode de rafraîchissement à résolution réduite (voir l'Annexe Q) est utilisé;
- 7) pas d'utilisation du mode de rééchantillonnage d'image de référence avec le mode de décodage par segments indépendants.

R.3 Contraintes imposées à l'utilisation du mode

L'utilisation d'autres aspects de la syntaxe de codage vidéo est soumise à certaines restrictions lorsque le mode de décodage par segments indépendants est utilisé. Ces restrictions visent à éviter deux cas "pathologiques" qui risqueraient sinon de compliquer le fonctionnement du mode de décodage par segments indépendants.

R.3.1 Contrainte sur la forme des segments

Lorsqu'on utilise le mode structuré par tranches (Annexe K) sans le sous-mode de tranches rectangulaires (voir K.1), il peut arriver dans certains cas que la forme d'un segment d'image vidéo ne soit pas convexe (il contient des "coins intérieurs" ou même deux régions distinctes et séparées de l'image).

Par conséquent, le mode de décodage par segments indépendants ne doit pas être utilisé avec le mode structuré par tranches sans l'emploi simultané du sous-mode de tranches rectangulaires du mode structuré par tranches (voir l'Annexe K). Cette condition est indispensable si l'on veut éviter la difficulté d'avoir à traiter des cas particuliers afin de déterminer la manière et le moment de faire une extrapolation de chaque segment d'image vidéo.

R.3.2 Contrainte sur les variations de forme des segments

Si la forme des segments d'image vidéo pouvait changer d'une image à l'autre dans le flux binaire, il pourrait arriver dans certains cas que le flux binaire soit difficile à décoder. En effet, dans ce genre de cas, le contenu du flux binaire n'est pas suffisant pour déterminer la forme de chaque segment d'image vidéo avant l'apparition possible des vecteurs cinétiques dans le flux binaire, ce qui demande de connaître la forme du segment d'image vidéo pour faire une interprétation correcte.

Par conséquent, lorsqu'on utilise le mode de décodage par segments indépendants, la segmentation des images vidéo pour toutes les images et les trames permettant une prédiction temporelle (c.-à-d., toutes les images P, B et EP et toutes les trames PB améliorées) sera la même que celle qui est utilisée dans leur image de référence temporelle. De plus, lorsqu'on utilise le mode de décodage par segments indépendants, la segmentation des images vidéo pour toutes les images EI sera identique à la segmentation utilisée dans leur image de référence ou bien ne s'en distinguera que par une subdivision de cette segmentation. En outre, le mode de décodage par segments indépendants ne devra pas être utilisé dans une image ou une trame qui emploie des images de référence (tous les types d'image sauf INTRA) à moins que le mode de décodage par segments indépendants ne soit aussi utilisé dans chacune des images de référence pour l'image en cours. A cause de cette contrainte,

la forme des segments d'image vidéo dans le mode de décodage par segments indépendants ne changera jamais d'une image à l'autre, sauf pour ce qui est des changements dans les images I et EI (et la manière dont les images EI peuvent changer de segmentation est elle-même soumise en quelque sorte à une contrainte).

ANNEXE S

Autre mode INTER VLC

S.1 Introduction

La présente annexe décrit un autre mode INTER VLC facultatif de la présente Recommandation, qui améliore l'efficacité du codage interimage lorsque des changements importants sont évidents dans l'image. On obtient cette augmentation d'efficacité en permettant que certains codes VLC conçus à l'origine pour les images INTRA soient utilisés pour certains coefficients d'image INTER ainsi que pour des données CBPY. L'utilisation de ce mode est signalée dans le champ PLUSPTYPE de l'en-tête d'image. La possibilité d'utiliser ce mode facultatif est négociée par des moyens externes (par exemple, selon la Recommandation H.245). Le mode contient deux altérations de syntaxe, une pour le codage des coefficients INTER et l'autre pour le codage des valeurs INTER CBPY.

S.2 Autre mode INTER VLC pour les coefficients

Le tableau INTRA VLC de l'Annexe I s'appuie sur le concept qui consiste à utiliser les mêmes mots de code que dans le mode INTER VLC original, mais avec une interprétation différente de LEVEL et RUN. Le mode INTRA VLC convient mieux dans les cas où les coefficients sont nombreux ou s'ils ont de grandes amplitudes.

Le mode INTRA VLC est construit de sorte que les mots de code aient la même valeur pour LAST (0 ou 1) dans les deux tables INTER et INTRA. La table INTRA est ainsi produite par "brassage" des significations des mots de code avec la même valeur de LAST. De plus, pour les événements ayant un |LEVEL| élevé, la table INTRA utilise un mot de code qui a un RUN élevé dans la table INTER. Dans les blocs INTER ayant un grand nombre de coefficients de grande amplitude, il vaut parfois mieux utiliser la table INTRA plutôt que la table INTER, et dans certains cas de ce type, le choix de la table VLC peut s'imposer au décodeur puisque le décodage à l'aide de la table INTER donnerait des valeurs de RUN si élevées qu'elles indiqueraient la présence de plus de 64 coefficients pour un bloc. Dans ces circonstances, la table INTRA peut servir à améliorer l'efficacité du codage INTER.

S.2.1 Mode d'action du codeur.

Le codeur peut utiliser la table INTRA VLC pour coder un bloc INTER chaque fois que le décodeur peut détecter son utilisation – autrement dit, chaque fois que des coefficients hors des 64 coefficients d'un bloc seraient adressés à cause du décodage à l'aide de la table INTER VLC.

Le codeur choisirait normalement d'utiliser la table INTRA VLC pour coder un bloc INTER uniquement si la condition ci-dessus est remplie et aussi lorsque l'utilisation de la table INTRA VLC fait intervenir moins de bits que la table INTER VLC pour les mêmes valeurs des coefficients. C'est souvent le cas lorsqu'il y a de nombreux coefficients de grande amplitude, à cause de la façon dont la table INTRA VLC a été produite (puisque, pour les mêmes mots de code, la table INTER VLC ordinaire contient de longues plages alors que la table INTRA VLC contient des coefficients de grande amplitude).

S.2.2 Mode d'action du décodeur

Le décodage se déroule de la manière suivante:

- 1) le décodeur reçoit d'abord tous les codes de coefficient d'un bloc;
- 2) les mots de code sont ensuite interprétés en supposant que la table INTER VLC est utilisée; Si l'adressage des coefficients reste à l'intérieur des 64 coefficients d'un bloc, le décodage VLC est terminé;
- 3) si des coefficients à l'extérieur du bloc sont adressés, les mots de code sont interprétés conformément à la table INTRA VLC.

S.3 Autre mode INTER VLC pour CBPY

Les mots de code INTER CBPY (Tableau 13) sont conçus dans l'hypothèse qu'il y a plus de blocs Y avec tous les coefficients nuls que de blocs ayant au moins un coefficient non nul. Lorsque les blocs C_B et C_R ont tous deux au moins un coefficient non nul, c.-à-d., $CBPC_5 = CBPC_6 = 1$, cette hypothèse n'est plus valable. C'est pourquoi, lorsqu'on utilise l'autre mode INTER VLC, les mots de code CBPY définis au Tableau 13 pour les macroblocs INTRA doivent également être utilisés pour les macroblocs INTER chaque fois que $CBPC_5 = CBPC_6 = 1$.

ANNEXE T

Mode de quantification modifié

T.1 Introduction

La présente annexe décrit un mode facultatif de quantification modifié de la présente Recommandation, qui change le fonctionnement du quantificateur. L'utilisation de ce mode est indiquée dans le champ PLUSTYPE de l'en-tête d'image. La possibilité d'utiliser ce mode facultatif est négociée par de moyens externes (par exemple, selon la Recommandation H.245).

Le mode de quantification modifié présente quatre caractéristiques essentielles:

- 1) amélioration de l'aptitude à contrôler le débit pendant le codage, en modifiant la syntaxe du champ DQUANT;
- 2) amélioration de la fidélité de la chrominance, en spécifiant pour la chrominance un échelon de quantification plus faible que pour les données de luminance;
- 3) élargissement de la gamme de valeurs représentables des coefficients, afin d'assurer la représentation, dans les limites de précision autorisées par la taille de l'échelon de quantification, de n'importe quelle valeur réelle des coefficients;
- 4) limitation de la gamme de niveaux des coefficients quantifiés aux niveaux susceptibles de se produire réellement, pour améliorer la détectabilité des erreurs et minimiser la complexité du décodage.

T.2 Mise à jour modifiée de DQUANT

Le mode de quantification modifié change la sémantique du champ DQUANT. Dans ce mode, on peut utiliser DQUANT, soit pour modifier la valeur de QUANT d'une faible quantité positive ou négative, soit pour signaler une nouvelle valeur spécifique de QUANT. Le faible niveau de variation dépend de la valeur actuelle de QUANT. Grâce à ce mode, on peut spécifier dans le champ DQUANT un contrôle plus souple de la taille de l'échelon de quantification.

En mode de quantification modifié, le mot de code associé à DQUANT n'est plus un champ de longueur fixe de deux bits, mais un champ de longueur variable pouvant comporter deux ou six bits, le choix dépendant de la valeur du premier bit du code. La description qui suit est donc divisée en deux sections, selon la valeur du premier bit.

T.2.1 Modification de QUANT par échelon de faible valeur

Lorsque le premier bit du champ DQUANT est 1, un seul bit supplémentaire est envoyé dans DQUANT. Ce bit additionnel sert à modifier QUANT d'une valeur différentielle. La variation de la valeur de QUANT est fonction du second bit de DQUANT et de la valeur antérieure de QUANT, comme l'indique le Tableau T.1.

Exemple: si la valeur antérieure de QUANT est égale à 29 et si DQUANT est signalé par le mot de code "11", la valeur différentielle sera égale à +2, ce qui résultera dans une nouvelle valeur de QUANT égale à 31.

Tableau T.1/H.263 – Sémantique de la modification de QUANT par échelon de faible valeur

Valeur antérieure de QUANT	Variation de QUANT	
	DQUANT = 10	DQUANT = 11
1	+2	+1
2-10	-1	+1
11-20	-2	+2
21-28	-3	+3
29	-3	+2
30	-3	+1
31	-3	-5

T.2.2 Sélection arbitraire de QUANT

Lorsque le premier bit du champ DQUANT est 0, cinq bits supplémentaires sont envoyés dans DQUANT. Les cinq bits suivants représentent une nouvelle valeur de QUANT, comme défini au 5.1.19.

Exemple: si, quelle que soit la valeur en cours de QUANT, DQUANT est signalé par le mot de code '001111', la nouvelle valeur de QUANT sera égale à 15.

T.3 Valeur modifiée de l'échelon de quantification pour les coefficients de chrominance

En mode de quantification modifié, le paramètre de quantification des coefficients de chrominance est différent de celui de la luminance. Le paramètre de quantification de la luminance, appelé QUANT, est signalé dans le train binaire. Le mode de quantification modifié utilise un paramètre de quantification différent, appelé QUANT_C, pour la quantification inverse des coefficients de chrominance. Le Tableau T.2 montre la relation entre QUANT et QUANT_C. Si le mode avec filtre de dégroupage (voir l'Annexe J) est utilisé, QUANT_C le sera aussi pour l'application du filtre de dégroupage aux données de chrominance. Dans le cadre de la présente Recommandation, chaque fois que QUANT sera discuté dans un contexte différent, il est convenu qu'il signifiera la taille de l'échelon de quantification de la luminance.

Tableau T.2/H.263 – Relation entre QUANT et QUANT_C

Gamme de valeurs de QUANT	Valeur de QUANT_C
1-6	QUANT_C = QUANT
7-9	QUANT_C = QUANT – 1
10-11	9
12-13	10
14-15	11
16-18	12
19-21	13
22-26	14
27-31	15

T.4 Gamme modifiée des valeurs de coefficient

En mode de quantification modifié, les coefficients DCT quantifiés dont le niveau est supérieur à 127 sont représentables dans le train binaire, ce qui offre deux avantages:

- 1) amélioration des performances du codeur en permettant la représentation de la gamme réelle complète de toutes les valeurs possibles des coefficients;
- 2) réduction de la complexité du codeur en éliminant la nécessité d'augmenter la taille de l'échelon de quantification en présence de valeurs importantes des coefficients (qui sans cela ne seraient pas représentables).

Il se peut que la valeur correcte d'un coefficient DCT avant quantification dans le codeur atteigne même 2040. Dans ce cas, une gamme de -127 à +127 pour LEVEL est insuffisante pour couvrir toute la gamme des valeurs possibles des coefficients chaque fois que le paramètre de quantification QUANT ou QUANT_C est inférieur à 8. La gamme élargie des valeurs de coefficient étend la gamme de valeurs de LEVEL pour permettre le codage correct de n'importe quelle valeur réelle des coefficients.

En mode de quantification modifié, la signification du champ LEVEL qui suit un code ESCAPE (0000 011 d'après 5.4.2) est modifiée. Dans ce mode, la séquence de bits 1000 0000, au lieu d'être interdite, est utilisée pour représenter un code EXTENDED-ESCAPE. Un coefficient AC de valeur supérieure à 127 est représenté par l'envoi d'un code EXTENDED-ESCAPE immédiatement suivi par un champ EXTENDED-LEVEL d'une longueur fixe de 11 bits. Une valeur élargie de coefficient est codée dans le champ EXTENDED-LEVEL en prélevant les 11 bits les moins significatifs de la représentation binaire du complément à 2 de LEVEL et en leur faisant subir un décalage circulaire de 5 bits vers la droite. Ce décalage circulaire, illustré à la Figure T.1, est nécessaire pour empêcher l'émulation du code de déclenchement.

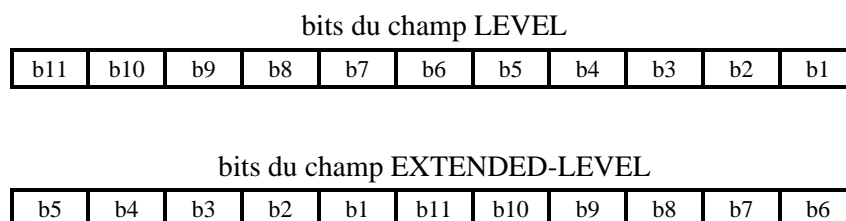


Figure T.1/H.263 – Décalage circulaire de la représentation binaire des coefficients

T.5 Restrictions d'utilisation

En mode de quantification modifié, les valeurs codées des coefficients font l'objet de certaines restrictions, ce qui offre plusieurs avantages:

- 1) amélioration de la détectabilité des erreurs sur les bits, en empêchant que certaines valeurs non réalistes des coefficients soient émises et en faisant en sorte que ces valeurs soient interprétées par le décodeur comme des erreurs sur les bits;
- 2) réduction de la complexité du décodeur, en diminuant la longueur du mot nécessaire à la quantification inverse précédant l'écritage;
- 3) empêchement de l'émulation du code de déclenchement pour les coefficients codés selon la méthode EXTENDED-ESCAPE décrite au T.4.

Les restrictions sont les suivantes. En mode de quantification modifié:

- 1) pour n'importe quel coefficient, la valeur |REC| du niveau de reconstruction générée par le processus de quantification inverse décrit au 6.2.1 (qui utilise, selon le cas, la valeur actuelle de QUANT ou de QUANT_C), ainsi que la valeur codée de LEVEL devront être inférieures à 4096. Cette restriction supplémentaire s'applique à tous les coefficients, que le coefficient soit transmis selon la méthode EXTENDED-ESCAPE ou non;
- 2) le flux binaire ne devra utiliser ni le code ESCAPE normal ni le code EXTENDED-ESCAPE pour le codage d'une combinaison de LAST, RUN et LEVEL pour laquelle une valeur du mot de code figure dans la Table VLC pertinente [c.-à-d. le Tableau 16 (voir 5.4.2) ou le Tableau I.2 (voir I.3)];
- 3) le code EXTENDED-ESCAPE ne devra être utilisé que lorsque le paramètre de quantification du coefficient (QUANT ou QUANT_C) est inférieur à huit (8);
- 4) le code EXTENDED-ESCAPE ne devra être utilisé que s'il est suivi par un champ EXTENDED-LEVEL représentant une valeur de LEVEL située en dehors de la gamme -127 à +127.

APPENDICE I

Dépistage d'erreurs

I.1 Introduction

Le présent appendice décrit une méthode de reprise efficace après des erreurs de transmission si des macroblocs erronés sont signalés au codeur via une voie de retour. La capacité d'envoyer et de traiter des informations de retour est signalée par des moyens définis ailleurs (par exemple dans la Recommandation H.245). En outre, le format et la teneur du message de retour sont aussi définis ailleurs (par exemple, dans la Recommandation H.245).

I.2 Dépistage d'erreurs

Du fait que le codage INTRA arrête la propagation temporelle des erreurs, il convient de l'utiliser pour les macroblocs à fort taux d'erreur de transmission. Pour ce faire, il faut que l'emplacement et l'ampleur des défauts de l'image puissent être communiqués au codeur. L'algorithme ci-après permet d'obtenir une estimation de la distribution des erreurs fondée sur les informations de retour reçues par le codeur. Il prend en compte la propagation spatiale des erreurs provoquée par la technique de prévision par compensation du mouvement ainsi que le temps de propagation jusqu'à la réception du message de retour. L'algorithme illustre l'une des méthodes permettant d'évaluer les messages de retour pour le dépistage d'erreurs spatio-temporel. On peut aussi utiliser d'autres algorithmes.

Supposons N macroblocs dans chaque trame $mb = 1 \dots N$ depuis le coin supérieur gauche jusqu'au coin inférieur droit. $\{n_{err}, mb_{first}, mb_{last}\}$ est le message de retour vers le codeur, où $mb_{first} \leq mb \leq mb_{last}$ indiquent un ensemble de macroblocs erronés dans la trame n_{err} .

Pour évaluer le message de retour, le codeur doit enregistrer en permanence des informations pendant le codage de chaque trame. Premièrement, l'erreur initiale $E_0(mb, n)$ qui serait provoquée par la perte du macrobloc mb dans la trame n doit être enregistrée. Si l'on part de l'hypothèse d'un simple masquage des erreurs dans le cadre duquel des macroblocs avec erreur sont traités comme n'étant pas codés, on calcule $E_0(mb, n)$ comme la différence absolue sommée (SAD, *summed absolute difference*) du macrobloc mb dans les trames n et $n - 1$. Deuxièmement, le nombre de pixels transférés du macrobloc mb_{source} dans la trame $n - 1$ au macrobloc mb_{dest} dans la trame n est enregistré dans des dépendances $d(mb_{source}, mb_{dest}, n)$. Ces dépendances sont tirées des vecteurs de mouvement.

Supposons qu'un message de retour arrive avant que la trame n_{next} ne soit codée, telle que $n_{next} > n_{err}$. Alors l'erreur estimée $E(mb, n_{err})$ dans le macrobloc mb et dans la trame n_{err} est initialisée comme suit:

$$E(mb, n_{err}) = \begin{cases} E_0(mb, n_{err}) & \text{pour } mb_{first} \leq mb \leq mb_{last} \\ 0 & \text{dans les autres cas} \end{cases}$$

Pour les trames suivantes n , où $n_{err} < n < n_{next}$, l'erreur peut être estimée comme suit:

$$E(mb, n) = \sum_{i=1}^N E(i, n-1) \frac{d(i, mb, n)}{256}$$

si l'on suppose une erreur distribuée uniformément dans chaque macrobloc après chaque itération.

L'erreur estimée $E(mb, n_{next} - 1)$ est incorporée dans la décision du mode de la trame suivante. Par exemple, le macrobloc mb est codé en mode INTRA si $E(mb, n_{next} - 1)$ dépasse un certain seuil.

Dans la pratique, les informations de dépistage des erreurs ne seront enregistrées que pour les dernières trames M . Dans ce cas, si $n_{err} < n_{next} - M$, aucune information de dépistage des erreurs n'est disponible et le codeur doit prendre une mesure spéciale. La trame suivante peut être codée en mode INTRA, par exemple. Toutefois, on peut envisager d'autres procédures qui peuvent être plus efficaces.

APPENDICE II

Amélioration facultative recommandée

II.1 Introduction

Compte tenu de la diversité des modes facultatifs présentés dans la présente Recommandation (H.263, Version 2), il est essentiel de définir, pour fins opérationnelles, plusieurs combinaisons de modes préférés, de sorte que les terminaux offrant des options améliorées puissent plus facilement communiquer entre eux au moyen d'une syntaxe plus perfectionnée que la syntaxe "de base". Le présent appendice contient une liste des combinaisons de modes préférés, groupées dans trois "niveaux" de prise en charge. Chaque niveau prévoit la prise en charge des niveaux inférieurs pour ainsi créer une structure de capacités en "pelure d'oignon". Le but premier du présent appendice est de décrire l'ordre dans lequel les modes devraient être pris en charge par les décodeurs, plutôt que d'imposer un petit ensemble précis de combinaisons de modes aux codeurs.

Aux fins de la sélection des modes à intégrer dans cette structure de niveaux, les principaux critères retenus furent des critères de performance: amélioration de la qualité subjective, incidence sur les retards et incidence sur la complexité (y compris le nombre de calculs à effectuer, la dépendance des données et la facilité de mise en œuvre). Cependant, étant donné que le présent appendice cherche à traiter une vaste gamme d'applications et de couches de transport, des fonctions qui augmentent la résistance aux erreurs et facilitent le groupage par paquets ont également été intégrées dans la même structure de niveaux. Conformément à ces objectifs, les niveaux de prise en charge décrits dans le présent appendice se veulent universels, pour en assurer l'indépendance par rapport aux moyens de transport et pour réduire la nécessité de "profils" propres à une application. L'universalité est une caractéristique clé d'un bon interfonctionnement entre les différents types de terminaux et de réseaux. Pour assurer cette universalité, plusieurs fonctions facultatives ont été omises de la structure décrite ci-dessous. Leur absence n'est pas une condamnation de leur utilisation; il s'agit plutôt du reflet d'une opinion collective concernant la probabilité d'intégration généralisée de ces fonctions dans une gamme complète de terminaux, de réseaux et d'applications.

II.2 Niveaux de prise en charge de modes préférés

Chaque mode facultatif décrit dans le présent appendice a été classé dans l'un de trois niveaux de prise en charge. Le rapport entre l'amélioration des performances et la complexité des calculs a été l'un des principaux critères retenus pour établir les niveaux, mais il a également été tenu compte de la facilité du groupage en paquets et de la résistance aux erreurs. La prise en charge d'un niveau particulier par un décodeur comporte la prise en charge de tous les niveaux inférieurs. En outre, la prise en charge d'un niveau particulier par un décodeur comporte la prise en charge de tous les sous-ensembles de combinaisons de modes de ce niveau et des niveaux inférieurs. Cette dernière exigence vise à minimiser les contraintes relatives aux choix de combinaisons de modes imposées aux codeurs. Cela est conforme à l'objectif premier du présent appendice, qui est de décrire l'ordre dans lequel les modes devraient être pris en charge par les décodeurs, plutôt que d'imposer un petit ensemble précis de combinaisons de modes aux codeurs.

Il est également important de tenir compte de la compatibilité amont. C'est pourquoi une combinaison de modes préférés comportant uniquement le mode de prédiction avancé (Annexe F) est recommandée (bien que la présente Recommandation ne soit pas incluse dans la structure de niveaux décrites ci-dessous), étant donné que le mode de prédiction avancé est considéré comme le plus avantageux des modes facultatifs présentés dans l'édition initiale (Version 1) de la présente Recommandation.

II.2.1 Modes préférés du niveau 1

Le premier niveau de prise en charge est composé des modes suivants:

- 1) **mode de codage INTRA avancé (Annexe I)** – L'utilisation de ce mode permet d'améliorer l'efficacité du codage des macroblocs INTRA (à l'intérieur d'images INTRA ou d'images à codage prédictif). Le nombre de calculs supplémentaires, tant par le codeur que par le décodeur, demandé par ce mode est minime (aussi peu qu'un maximum de 8 additions/soustractions par bloc de 8×8 pendant le processus de décodage, en plus de l'utilisation d'un tableau VLC différent mais très similaire pour accroître considérablement l'efficacité du codage). Pour ces raisons, le codage INTRA avancé est inclus à ce niveau inférieur de prise en charge;
- 2) **mode avec filtre de dégroupage (Annexe J)** – Etant donné que les filtres de dégroupage permettent d'améliorer considérablement la qualité subjective des images, ils sont déjà largement utilisés comme outils de post-traitement par les terminaux de vidéocommunications. L'Annexe J présente le mode préféré de fonctionnement d'un filtre de dégroupage, étant donné que le filtre est inséré dans la boucle de codage. Cela facilite la mise en œuvre du filtre (en réduisant les besoins en mémoire) et améliore quelque peu l'efficacité du codage par rapport à une mise en œuvre post-traitement. Comme dans le cas du mode de prédiction avancé, ce mode comporte également la caractéristique "quatre vecteurs cinétiques par macrobloc" et l'extrapolation des limites d'image pour fins de compensation de mouvement, deux caractéristiques qui permettent d'améliorer davantage l'efficacité du codage. Les besoins de calcul du filtre de dégroupage sont de plusieurs centaines d'opérations par macrobloc codé, mais l'accès mémoire et les dépendances de calcul ne sont pas complexes. Ce dernier point rend le filtre de dégroupage préférable à la prédiction avancée. En outre, les avantages de la prédiction avancée ne sont pas aussi importants quand un filtre de dégroupage est également utilisé. C'est pourquoi le filtre de dégroupage est inclus dans ce niveau inférieur de prise en charge, tandis que la prédiction avancée est renvoyée au niveau 3;
- 3) **information d'amélioration complémentaire (gel d'image complète seulement) (Annexe L, voir L.4)** – La mise en œuvre du gel d'image complète est très simple: il suffit que le décodeur puisse interrompre le transfert de données entre sa mémoire tampon de sortie et l'affichage vidéo. Ce sous-mode est utile pour prévenir l'affichage d'images basse fidélité pendant que le décodeur établit une image haute fidélité;
- 4) **mode de quantification modifié (Annexe T)** – Ce mode comporte une gamme de coefficients DCT étendue, une syntaxe DQUANT modifiée et une taille d'échelon modifiée pour la chrominance. Les deux premières caractéristiques donnent une plus grande souplesse au codeur et peuvent même réduire sa charge de calcul (en éliminant la nécessité de coder de nouveau les macroblocs en cas de saturation des niveaux de coefficients). La troisième caractéristique améliore de façon sensible la fidélité de chrominance, généralement avec un effet minime sur le débit et une augmentation à peu près nulle des calculs. Au décodeur, la seule charge de calcul importante est la capacité d'analyser plusieurs nouveaux symboles du flux binaire.

II.2.2 Modes préférés du niveau 2

Le deuxième niveau de prise en charge est composé des modes suivants:

- 1) **mode de codage par vecteurs cinétiques non restreints (avec le champ UUI = "1" suffisant) (Annexe D)** – L'Annexe D représente deux caractéristiques principales:
 - a) l'extrapolation hors des limites d'image;

b) la prise en charge de vecteurs cinétiques plus longs.

La première caractéristique est déjà prise en charge par l'inclusion de l'Annexe J dans le premier niveau de prise en charge. La prise en charge de vecteurs cinétiques plus longs peut améliorer grandement l'efficacité du codage, surtout dans le cas des grandes images, des mouvements rapides, des mouvements de caméra et des fréquences d'image peu élevées. Utilisé quand le champ PLUSTYPE est présent, ce mode permet de plus grandes différences vectorielles, ce qui peut simplifier considérablement l'opération de codage. Les vecteurs cinétiques plus longs peuvent présenter des difficultés au décodeur en ce qui a trait à l'accès mémoire, mais les limites, dépendantes de la taille d'image, visant la taille maximale des vecteurs cinétiques atténuent ce problème et l'empêchent de devenir un obstacle majeur à la mise en œuvre;

- 2) **mode structuré par tranches (Annexe K)** – Le mode structuré par tranches est inclus dans ce niveau pour fins de compatibilité avec la couche de transport en mode paquet. Tous les sous-modes de l'Annexe K doivent être pris en charge, y compris le sous-mode de tranches rectangulaires et le sous-mode d'ordre arbitraire de tranches. La charge de calcul supplémentaire produite par le mode structuré par tranches est minime et se limite principalement à l'analyse et à la génération de flux binaires;
- 3) **mode de rééchantillonnage d'image de référence (mode de facteur 4 implicite seulement) (Annexe P)** – Le mode de facteur 4 implicite du mode de rééchantillonnage d'image de référence permet le rééchantillonnage automatique de l'image de référence uniquement quand la taille de la nouvelle image bi-trame est modifiée, tel qu'indiqué dans l'en-tête d'image. Ce mode ne demande aucun bit supplémentaire dans le flux binaire. Des modifications de résolution dynamiques prédictives permettent au décodeur de faire des compromis intelligents entre la résolution spatiale et la résolution temporelle. En outre, il s'agit du mode de fonctionnement le plus simple présenté à l'Annexe P (facteur de suréchantillonnage ou de sous-échantillonnage de 4). Il n'ajoute que peu de complexité aux calculs effectués par le codeur et le décodeur, étant donné que le mode de facteur 4 ne demande qu'un simple filtre FIR fixe (filtre à réponse impulsionnelle ne demandant qu'à peu près 4 opérations par pixel, au maximum).

II.2.3 Modes préférés du niveau 3

Le troisième niveau de prise en charge est composé des modes suivants:

- 1) **mode de prédiction avancé (Annexe F)** – Ce mode est, du point de vue de l'efficacité du codage, le mode le plus important parmi les modes définis dans la première édition (Version 1) de la présente Recommandation. Il comprend la compensation de mouvement par superposition de blocs et la possibilité de quatre vecteurs cinétiques par macrobloc, et il permet que les vecteurs cinétiques pointent hors des limites d'image. L'utilisation du mode de prédiction avancé améliore grandement les performances subjectives et objectives. Ce mode accroît cependant notablement la charge de calcul; et, surtout, les dépendances des données associées à ce mode compliquent l'ordre de traitement par le décodeur. Compte tenu du fait que l'amélioration de la qualité n'est pas très marquée quand un filtre de dégroupage est également utilisé et de la complexité accrue du traitement, ce mode est inclus dans le troisième niveau de prise en charge plutôt que dans le premier. Cependant, étant donné que des mises en œuvre de la Recommandation H.263, conçues avant l'adoption des autres modes de la présente liste, peuvent comprendre la mise en œuvre du seul mode de prédiction avancé, le fonctionnement en seul mode de prédiction avancé est également recommandé pour assurer la plus grande compatibilité amont possible;

- 2) **mode de codage par trames PB amélioré (Annexe M)** – Le mode de codage par trames PB amélioré offre une nette amélioration par rapport au mode de codage par trames PB de base (Annexe G), et il a été démontré qu'il améliore considérablement l'efficacité du codage des séquences vidéo à faible degré de mouvement. La version de base du mode de codage par trames PB a de la difficulté à traiter les coupures de scènes et le mouvement irrégulier, mais le mode amélioré peut être utilisé efficacement en tout temps. Ce mode est à peu près neutre en ce qui a trait à la charge de calcul (particulièrement dans le cas des architectures pipeline pour macroblocs). Cependant, comme le mode de codage par trames PB amélioré peut augmenter les délais et demande la mise en œuvre d'une mémoire de trame supplémentaire (dans le codeur et le décodeur), il est placé dans une couche supérieure;
- 3) **mode de décodage par segments indépendants (Annexe R)** – Le mode de décodage par segments indépendants augmente notablement la complexité de la mise en œuvre, mais il améliore également considérablement la robustesse aux erreurs d'un décodeur vidéo, surtout en cas de perte de paquets. Pour cette raison, il est probable que beaucoup d'applications dans lesquelles des données peuvent être perdues en cours de transmission (soit en raison de perte de paquets ou de brouillage dans le canal) utiliseront le mode de décodage par segments indépendants, surtout de concert avec le mode structuré par tranches;
- 4) **autre mode INTER VLC (Annexe S)** – Il a été démontré que ce mode améliore l'efficacité du codage, surtout dans le cas des images à haut degré de mouvement codées en haute fidélité. Il utilise le même tableau VLC que le mode de codage INTRA avancé et il est simple en ce qui a trait aux calculs (pour chaque bloc, il demande au plus une lecture supplémentaire des symboles analysés pour les décoder et les convertir en coefficients DCT). Cependant, comme ce mode pourrait être difficile à mettre en œuvre (surtout dans le cas d'une architecture massivement pipeline), il n'est pas inclus dans les niveaux inférieurs de prise en charge.

II.3 Formats d'image et fréquences de base d'image

Pour assurer un haut degré de qualité d'interfonctionnement, les codeurs et les décodeurs qui prennent en charge un grand format d'image normalisé (QCIF, CIF, 4CIF et 16CIF) devraient également prendre en charge de plus petits formats d'image standard. (Tel qu'indiqué ailleurs dans la présente Recommandation, les décodeurs doivent prendre en charge les formats sub-QCIF et QCIF et les codeurs, le format sub-QCIF ou QCIF.) Les décodeurs devraient pouvoir fonctionner en présence de formats d'image plus petits à des fréquences de base image maximales au moins égales aux fréquences de base d'image maximales auxquelles ils peuvent traiter les grands formats d'image standard. Par exemple, un décodeur pouvant décoder 15 images 4CIF par seconde [de fait $15 \times (1000/1001)$ images par seconde] devrait pouvoir décoder au moins 15 images CIF, QCIF ou SQCIF par seconde.

Il est recommandé que les codeurs et les décodeurs qui prennent en charge des formats d'image personnalisés ou des fréquences de base d'image personnalisées, ou les deux, respectent les lignes directrices ci-dessous:

- 1) un codeur ou un décodeur qui prend en charge un format d'image personnalisé devrait également prendre en charge tous les formats d'image normalisé dont la taille est égale ou inférieure (largeur et hauteur) à celle de tout format d'image personnalisé pris en charge. Par exemple, un codeur qui prend en charge un format d'image personnalisé de 256×256 devrait également prendre en charge le codage d'image QCIF et SQCIF;

- 2) un codeur ou un décodeur qui prend en charge une fréquence de base d'image personnalisée pour un format d'image normalisé devrait prendre en charge, pour le même format d'image, une fréquence d'image (exprimée en images par seconde) égale ou supérieure à la fréquence de base d'image normalisée de 30 Hz [de fait $30 \cdot (1000/1001)$ Hz]. Par exemple, si un décodeur prend en charge des images QCIF à une fréquence de 12,5 images par seconde à une fréquence de base d'image de 25 Hz, il devrait pouvoir décoder au moins 15 images QCIF par seconde à une fréquence de base d'image de 30 Hz. Si un format d'image personnalisé est utilisé, tous les formats d'image normalisés de taille égale ou inférieure qui sont pris en charge devraient pouvoir être traités à des fréquences d'image, pour la fréquence de base d'image normalisée, au moins égales à la fréquence d'image maximale du format d'image normalisé.

SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information
Série Z	Langages de programmation