



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.800

(08/2002)

SÉRIE T: TERMINAUX DES SERVICES
TÉLÉMATIQUES

**Technologies de l'information – Système de
codage d'images JPEG 2000: système de
codage noyau**

Recommandation UIT-T T.800

**Technologies de l'information – Système de codage d'images JPEG 2000:
système de codage noyau**

Résumé

La présente Recommandation | Norme internationale définit un ensemble de méthodes de compression sans perte (au bit près) et avec perte pour codage d'images numériques fixes, monochromes, à modelé continu en échelle de gris, à palette ou à modelé continu de couleurs.

La présente Recommandation | Norme internationale:

- spécifie des processus de décodage permettant de convertir des données d'image comprimées en données d'image reconstituées;
- spécifie une syntaxe de flux codé contenant des informations permettant d'interpréter des données d'image comprimées;
- spécifie un format de fichier;
- fournit des directives sur les processus de codage permettant de convertir des données d'image de source en données d'image comprimées;
- fournit des directives sur la façon d'implémenter ces processus en pratique.

Source

La Recommandation T.800 de l'UIT-T, élaborée par la Commission d'études 16 (2001-2004) de l'UIT-T, a été approuvée le 29 août 2002. Un texte identique est publié comme Norme Internationale ISO/CEI 15444-1.

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.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude 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 1 de l'AMNT.

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 implémenter 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 2003

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives..... 1
2.1	Recommandations Normes internationales identiques..... 1
2.2	Autres références..... 2
3	Définitions..... 2
4	Abréviations et symboles 6
4.1	Abréviations..... 6
4.2	Symboles..... 7
5	Description générale..... 8
5.1	Objet..... 8
5.2	Flux codé..... 9
5.3	Principes de codage..... 9
6	Exigences relatives au codeur 11
7	Exigences relatives au décodeur..... 11
7.1	Exigences de la syntaxe du flux codé..... 11
7.2	Exigences relatives au format de fichier facultatif..... 11
8	Exigences relatives à l'implémentation..... 11
Annexe A – Syntaxe du flux codé..... 12	
A.1	Marqueurs, segments marqueurs, et en-têtes..... 12
A.2	Informations contenues dans les segments marqueurs..... 14
A.3	Construction du flux codé..... 16
A.4	Marqueurs et segments marqueurs délimitants 19
A.5	Segment marqueur d'information fixe..... 21
A.6	Segments marqueurs fonctionnels..... 22
A.7	Segments marqueurs de pointage..... 33
A.8	Marqueur et segments marqueurs dans le flux binaire..... 39
A.9	Segments marqueurs informationnels..... 40
A.10	Restrictions du flux codé conformes à la présente Recommandation Norme internationale..... 41
Annexe B – Ordonnancement des données d'image comprimées et non comprimées..... 43	
B.1	Introduction aux concepts de structure des données d'image 43
B.2	Affectation des composantes à la grille de référence 43
B.3	Subdivision de la zone d'image en pavés et composantes-pavés..... 45
B.4	Exemple du mappage de composantes à la grille de référence (pour information)..... 47
B.5	Subdivision de composante-pavé transformée en niveaux de résolution et sous-bandes..... 50
B.6	Subdivision des niveaux de résolution en districts..... 51
B.7	Subdivision des sous-bandes en blocs codés..... 52
B.8	Couches..... 53
B.9	Paquets..... 54
B.10	Codage des informations d'en-tête de paquet..... 55
B.11	Pavé et éléments de pavé..... 61
B.12	Ordre de progression..... 62
Annexe C – Codage arithmétique entropique..... 67	
C.1	Codage binaire (pour information)..... 67
C.2	Description du codeur arithmétique (pour information)..... 68
C.3	Procédure de décodage arithmétique..... 79
Annexe D – Modélisation des bits de coefficient..... 87	
D.1	Structure d'exploration des blocs codés dans les blocs codés 87
D.2	Bits de coefficient et signification..... 87
D.3	Passes de décodage sur les plans binaires 88
D.4	Initialisation et terminaison..... 92
D.5	Symbole de segmentation assurant l'élasticité aux erreurs..... 93

	<i>Page</i>	
D.6	Evitement sélectif du codage arithmétique.....	94
D.7	Formation verticale du contexte causal.....	95
D.8	Schéma fonctionnel du codage de bloc codé.....	95
Annexe E	– Quantification.....	98
E.1	Procédure de quantification inverse.....	98
E.2	Quantification de coefficient scalaire (pour information).....	100
Annexe F	– Transformation en ondelettes discrètes de composantes-pavés.....	101
F.1	Paramètres de composante-pavé.....	101
F.2	Transformations en ondelettes discrètes.....	101
F.3	Transformation inverse en ondelettes discrètes.....	101
F.4	Transformation directe (pour information).....	113
Annexe G	– Décalage au niveau aperiodique et transformations multiples de composante.....	124
G.1	Décalage au niveau aperiodique de composantes-pavés.....	124
G.2	Transformation multiple de composante réversible (RCT).....	125
G.3	Transformation multiple de composante irréversible (ICT).....	125
G.4	Sous-échantillonnage de composante de chrominance et la grille de référence.....	126
Annexe H	– Codage d'images avec régions intéressantes.....	127
H.1	Décodage d'images ROI.....	127
H.2	Description de la méthode du décalage maximal.....	127
H.3	Remarques sur le codage de région intéressante (pour information).....	128
Annexe I	– Syntaxe de format de fichier JP2.....	131
I.1	Domaine d'application du format de fichier JP2.....	131
I.2	Introduction au format de fichier JP2.....	131
I.3	Architecture de spécification d'échelle de gris/de couleur/de palette/de composantes multiples.....	133
I.4	Définition de boîte.....	135
I.5	Boîtes définies.....	138
I.6	Adjonction d'informations sur les droits de propriété intellectuelle dans JP2.....	154
I.7	Adjonction d'informations spécifiques du vendeur au format de fichier JP2.....	154
I.8	Traitement des boîtes inconnues.....	157
Annexe J	– Exemples et directives.....	158
J.1	Décodeur entropique adaptatif de conventions logicielles.....	158
J.2	Sélection de grandeurs d'échelon de quantification pour transformations irréversibles.....	160
J.3	Réponses impulsionnelles de filtre correspondant à des procédures de filtrage irréversibles en mode lissage.....	160
J.4	Exemple de transformation en ondelettes discrètes.....	161
J.5	Transformée en ondelettes par rangée.....	165
J.6	Codage par exploration.....	176
J.7	Elasticité aux erreurs.....	176
J.8	Exigence de compatibilité avec fichiers JFIF/SPIFF.....	177
J.9	Implémentation de la méthode ICC limitée à l'extérieur d'un moteur complet de gestion de couleur ICC.....	177
J.10	Exemple d'interprétation de composantes multiples.....	183
J.11	Exemple de décodage montrant les étapes intermédiaires.....	183
J.12	Pondération de fréquence visuelle.....	187
J.13	Sous-échantillonnage de composantes par le codeur.....	189
J.14	Commande de débit.....	190
J.15	Directives sur la gestion des flux codés YCC.....	194
Annexe K	– Bibliographie.....	197
K.1	Généralités.....	197
K.2	Quantification et codage entropique.....	197
K.3	Transformation en ondelettes.....	197
K.4	Codage de région intéressante.....	198
K.5	Pondération de fréquence visuelle.....	198
K.6	Elasticité aux erreurs.....	198

	<i>Page</i>
K.7 Codage par exploration	199
K.8 Couleur.....	199
Annexe L – Déclaration relative aux brevets.....	200

LISTE DES FIGURES

	<i>Page</i>
Figure 5-1 – Schéma fonctionnel de la spécification.....	9
Figure A.1 – Exemple de figures descriptives du segment marqueur	14
Figure A.2 – Construction du flux codé	16
Figure A.3 – Construction de l'en-tête principal.....	17
Figure A.4 – Construction du premier en-tête d'élément d'un pavé donné	18
Figure A.5 – Construction d'un autre en-tête d'élément de pavé que le premier	18
Figure A.6 – Syntaxe de début d'élément de pavé.....	19
Figure A.7 – Syntaxe dimensionnelle d'image et de pavé	21
Figure A.8 – Syntaxe par défaut du style de codage	23
Figure A.9 – Schéma du paramètre style de codage des paramètres SGcod et SPcod	24
Figure A.10 – Syntaxe de composante de style de codage	27
Figure A.11 – Schéma du paramètre style de codage des paramètres SPcod	27
Figure A.12 – Syntaxe de région intéressante	28
Figure A.13 – Syntaxe de quantification par défaut	29
Figure A.14 – Syntaxe de composante de quantification	31
Figure A.15 – Syntaxe de pavé de changement d'ordre de progression	32
Figure A.16 – Longueurs d'élément de pavé	33
Figure A.17 – Syntaxe de longueur d'élément de pavé.....	33
Figure A.18 – Syntaxe d'en-tête principal de longueur de paquet	35
Figure A.19 – Syntaxe d'en-tête d'élément de pavé de longueur de paquet.....	36
Figure A.20 – Syntaxe d'en-tête principal de paquet compact.....	37
Figure A.21 – Syntaxe d'en-tête d'élément de pavé d'en-tête de pavé compact.....	38
Figure A.22 – Syntaxe de début de paquet.....	39
Figure A.23 – Syntaxe d'enregistrement de composante.....	40
Figure A.24 – Syntaxe de commentaire	41
Figure B.1 – Schéma de grille de référence.....	44
Figure B.2 – Emplacements d'échantillon de composante sur la grille de référence pour différentes valeurs XRsiz et YRsiz.....	45
Figure B.3 – Exemple d'emplacements d'échantillon de composante supérieure gauche.....	45
Figure B.4 – Schéma du pavage de grille de référence.....	46
Figure B.5 – Exemple de grille de référence	48
Figure B.6 – Exemple de dimensions et d'emplacements de pavé pour composante 0	49
Figure B.7 – Exemple de dimensions et d'emplacements de pavé pour composante 1	50
Figure B.8 – Districts d'une résolution réduite	51
Figure B.9 – Blocs codés et districts dans la sous-bande <i>b</i> , issus de quatre pavés différents	53
Figure B.10 – Diagramme des districts d'un niveau de résolution d'une composante	54
Figure B.11 – Diagramme des blocs codés dans des districts à un niveau de résolution	55
Figure B.12 – Exemple d'une représentation en arbre de résolution	56
Figure B.13 – Exemple d'informations connues du codeur	60
Figure B.14 – Exemple de volume d'ordre de progression en deux dimensions	65
Figure B.15 – Exemple du placement de segments marqueurs de changement POC.....	66
Figure C.1 – Entrées et sorties de codeur arithmétique	67
Figure C.2 – Codeur pour le codeur MQ.....	69
Figure C.3 – Procédure ENCODE.....	70
Figure C.4 – Procédure CODE1	70
Figure C.5 – Procédure CODE0	71
Figure C.6 – Procédure CODELPS avec échange conditionnel de symbole MPS/LPS	72
Figure C.7 – Procédure CODEMPS avec échange conditionnel de symbole MPS/LPS	74
Figure C.8 – Procédure de renormalisation du codeur	75
Figure C.9 – Procédure OCTETOUT pour codeur.....	76
Figure C.10 – Initialisation du codeur	77
Figure C.11 – Procédure FLUSH	78
Figure C.12 – Réglage des bits finals dans le registre C	79
Figure C.13 – Entrées et sorties du décodeur arithmétique	79
Figure C.14 – Décodeur pour le codeur MQ	80
Figure C.15 – Décodage d'un symbole MPS ou d'un symbole LPS	81
Figure C.16 – Procédure d'échange conditionnel sur le trajet de symbole MPS dans le décodeur.....	82

Figure C.17 – Procédure d'échange conditionnel sur le trajet de symbole LPS dans le décodeur	83
Figure C.18 – Procédure de renormalisation du décodeur	84
Figure C.19 – Procédure BYTEIN pour décodeur	85
Figure C.20 – Initialisation du décodeur	86
Figure D.1 – Exemple de structure d'exploration du plan binaire d'un bloc codé	87
Figure D.2 – Etats de voisinage utilisés pour former le contexte	88
Figure D.3 – Organigramme pour toutes les passes de codage sur le plan binaire d'un bloc codé	96
Figure F.1 – Entrées et sorties de la procédure de transformation IDWT	101
Figure F.2 – La transformation IDWT ($N_L = 2$)	102
Figure F.3 – La procédure de transformation IDWT	103
Figure F.4 – Entrées et sorties de la procédure 2D_SR	103
Figure F.5 – Niveau unique de reconstruction de quatre sous-bandes (procédure 2D_SR) en sous-bandes	103
Figure F.6 – La procédure 2D_SR	104
Figure F.7 – Paramètres de la procédure 2D_INTERLEAVE	104
Figure F.8 – La procédure 2D_INTERLEAVE	105
Figure F.9 – Entrées et sorties de la procédure HOR_SR	106
Figure F.10 – La procédure HOR_SR	107
Figure F.11 – Entrées et sorties de la procédure VER_SR	108
Figure F.12 – La procédure VER_SR	108
Figure F.13 – Paramètres de la procédure 1D_SR	109
Figure F.14 – La procédure 1D_SR	109
Figure F.15 – Extension périodique et symétrique du signal	109
Figure F.16 – Paramètres de la procédure 1D_FILTR	110
Figure F.17 – Entrées et sorties de la procédure FDWT	113
Figure F.18 – La transformation FDWT ($N_L = 2$)	113
Figure F.19 – La procédure FDWT	114
Figure F.20 – Entrées et sorties de la procédure 2D_SD	114
Figure F.21 – Décomposition d'un niveau unique en quatre sous-bandes (procédure 2D_SD)	115
Figure F.22 – La procédure 2D_SD	115
Figure F.23 – Entrées et sorties de la procédure VER_SD	116
Figure F.24 – La procédure VER_SD	116
Figure F.25 – Entrées et sorties de la procédure HOR_SD	117
Figure F.26 – La procédure HOR_SD	118
Figure F.27 – Paramètres de la procédure 2D_DEINTERLEAVE	119
Figure F.28 – La procédure 2D_DEINTERLEAVE	120
Figure F.29 – Paramètres de la procédure 1D_SD	121
Figure F.30 – La procédure 1D_SD	121
Figure F.31 – Paramètres de la procédure 1D_FILTD	122
Figure G.1 – Placement du décalage au niveau aperiodique avec transformation de composante	124
Figure G.2 – Placement du décalage au niveau aperiodique sans transformation de composante	124
Figure H.1 – La transformation inverse d'ondelettes avec le filtre réversible 5-3	129
Figure H.2 – La transformation inverse d'ondelettes avec le filtre irréversible 9-7	130
Figure I.1 – Structure théorique d'un fichier JP2	132
Figure I.2 – Exemple des figures descriptives de boîte	135
Figure I.3 – Exemple des figures descriptives de superboîte	135
Figure I.4 – Organisation d'une boîte	135
Figure I.5 – Illustration des longueurs de boîte	136
Figure I.6 – Organisation du contenu d'une boîte de type de fichier	138
Figure I.7 – Organisation du contenu d'une boîte d'en-tête de fichier JP2	139
Figure I.8 – Organisation du contenu d'une boîte d'en-tête d'image	140
Figure I.9 – Organisation du contenu d'une boîte de bits par composante	142
Figure I.10 – Organisation du contenu d'une boîte de spécification de couleur	143
Figure I.11 – Organisation du contenu de la boîte de palette	146
Figure I.12 – Organisation du contenu d'une boîte de mappage des composantes	147
Figure I.13 – Organisation du contenu d'une boîte de définition de canal	148
Figure I.14 – Organisation du contenu de la boîte de résolution	151
Figure I.15 – Organisation du contenu de la boîte de résolution de capture	151
Figure I.16 – Organisation du contenu de la boîte de résolution d'affichage par défaut	153

Figure I.17 – Organisation du contenu de la boîte de flux codé contigu	153
Figure I.18 – Organisation du contenu de la boîte de langage XML	154
Figure I.19 – Organisation du contenu de la boîte d'identification UUID	155
Figure I.20 – Organisation du contenu d'une boîte d'informations d'identification UUID	155
Figure I.21 – Organisation du contenu d'une boîte de liste d'identificateurs UUID	156
Figure I.22 – Organisation du contenu d'une boîte d'URL d'entrée de données	156
Figure J.1 – Initialisation du décodeur de conventions logicielles	158
Figure J.2 – Décodage d'un symbole MPS ou d'un symbole LPS dans le décodeur de conventions logicielles	159
Figure J.3 – Insertion d'un nouvel octet dans le registre C du décodeur de conventions logicielles	160
Figure J.4 – La procédure de transformation FDWT_ROW	167
Figure J.5 – La procédure GET_ROW	168
Figure J.6 – La procédure INIT	169
Figure J.7 – La procédure START_VERT	170
Figure J.8 – La procédure RB_VERT_1	171
Figure J.9 – La procédure RB_VERT_2	172
Figure J.10 – La procédure END_1	173
Figure J.11 – La procédure END_2	175
Figure J.12 – Illustration des contributions de bloc codé aux couches de flux binaire	191
Figure J.13 – Format 4:2:2 (cositionné)	195
Figure J.14 – Format 4:2:2 (centré)	195
Figure J.15 – Format 4:2:0 (cositionné)	195
Figure J.16 – Format 4:2:0 (centré)	196

LISTE DES TABLEAUX

	<i>Page</i>
Tableau A.1 – Définitions des marqueurs.....	13
Tableau A.2 – Liste des marqueurs et segments marqueurs.....	14
Tableau A.3 – Informations contenues dans les segments marqueurs.....	15
Tableau A.4 – Valeurs paramétriques de début de flux codé.....	19
Tableau A.5 – Valeurs paramétriques de début d'élément de pavé.....	20
Tableau A.6 – Valeur du paramètre Nombre d'éléments de pavé, TNsot.....	20
Tableau A.7 – Valeurs paramétriques de début de données.....	20
Tableau A.8 – Valeurs paramétriques de fin de flux codé.....	20
Tableau A.9 – Valeurs paramétriques de dimensions d'image et de pavé.....	22
Tableau A.10 – Paramètre Rsiz de capacité.....	22
Tableau A.11 – Paramètre Ssiz de composante.....	22
Tableau A.12 – Valeurs paramétriques de style de codage par défaut.....	23
Tableau A.13 – Valeurs paramétriques de style de codage pour le paramètre Scod.....	24
Tableau A.14 – Valeurs paramétriques de style de codage du paramètre SGcod.....	24
Tableau A.15 – Valeurs paramétriques de style de codage des paramètres SPcod et SPcoc.....	24
Tableau A.16 – Ordre de progression pour les paramètres SGcod, SPcoc et Ppoc.....	25
Tableau A.17 – Transformation multiple de composante pour les paramètres SGcod.....	25
Tableau A.18 – Exposant de largeur ou de hauteur des blocs codés pour les paramètres SPcod et SPcoc.....	25
Tableau A.19 – Style de bloc codé pour les paramètres SPcod et SPcoc.....	26
Tableau A.20 – Transformation pour les paramètres SPcod et SPcoc.....	26
Tableau A.21 – Largeur et hauteur de district pour les paramètres SPcod et SPcoc.....	26
Tableau A.22 – Valeurs paramétriques de composant de style de codage.....	27
Tableau A.23 – Valeurs paramétriques de style de codage pour le paramètre Scoc.....	28
Tableau A.24 – Valeurs paramétriques de région intéressante.....	28
Tableau A.25 – Valeurs paramétriques de région intéressante pour le paramètre Srgn.....	29
Tableau A.26 – Valeurs de région intéressante extraites du paramètre SPRgn (Srgn = 0).....	29
Tableau A.27 – Valeurs paramétriques de quantification par défaut.....	30
Tableau A.28 – Valeurs de quantification par défaut pour les paramètres Sqcd et Sqcc.....	30
Tableau A.29 – Valeurs d'échelon réversibles pour les paramètres SPqcd et SPqcc (transformée réversible seulement).....	30
Tableau A.30 – Valeurs de quantification pour les paramètres SPqcd et SPqcc (transformation irréversible seulement).....	30
Tableau A.31 – Valeurs paramétriques de composante de quantification.....	31
Tableau A.32 – Valeurs paramétriques de pavé de changement de l'ordre de progression.....	33
Tableau A.33 – Valeurs paramétriques de longueur d'élément de pavé.....	34
Tableau A.34 – Paramètres dimensionnels pour Stlm.....	35
Tableau A.35 – Valeurs paramétriques d'en-tête principal de longueur de paquet.....	36
Tableau A.36 – Liste Iplm ou Iplt des longueurs de paquet.....	36
Tableau A.37 – Valeurs paramétriques d'en-tête d'élément de pavé de longueur de paquet.....	37
Tableau A.38 – Valeurs paramétriques d'en-tête principal de paquet compact.....	38
Tableau A.39 – Valeurs paramétriques d'en-tête d'élément de pavé d'en-tête de paquet.....	38
Tableau A.40 – Valeurs paramétriques de début de paquet.....	39
Tableau A.41 – Valeurs paramétriques de fin d'en-tête de paquet.....	40
Tableau A.42 – Valeurs paramétriques d'enregistrement de composante.....	40
Tableau A.43 – Valeurs paramétriques de commentaire.....	41
Tableau A.44 – Valeurs d'enregistrement pour le paramètre Rcom.....	41
Tableau A.45 – Restrictions du flux codé.....	42
Tableau B.1 – Grandeurs (x_{0b}, y_{0b}) pour sous-bande b.....	51
Tableau B.2 – Exemple de formation de couche (une seule composante est montrée).....	51
Tableau B.3 – Exemple de formation de paquet.....	55
Tableau B.4 – Mots de code pour le nombre de passes de codage sur chaque bloc codé.....	58
Tableau B.5 – Exemple de flux binaire d'en-tête de paquets.....	60
Tableau C.1 – Structures de registre de codeur.....	69
Tableau C.2 – Estimation de valeurs Qe et de probabilité.....	72
Tableau C.3 – Structures de registre de décodeur.....	80
Tableau D.1 – Contexte des passes de codage de propagation de signification et de nettoyage.....	89

Tableau D.2 – Contributions des voisinages verticaux (et horizontaux) au contexte de signe	89
Tableau D.3 – Contextes de signe issus des contributions verticales et horizontales	90
Tableau D.4 – Contexte des passes de codage de raffinement de grandeur	90
Tableau D.5 – Décodeur de longueur de plage pour passes de nettoyage	91
Tableau D.6 – Exemple d'ordre de codage de sous-plan binaire et de propagation de signification	92
Tableau D.7 – Etats initiaux pour tous les contextes	92
Tableau D.8 – Structures de terminaison de codeur arithmétique	92
Tableau D.9 – Evitement sélectif du codage arithmétique	94
Tableau D.10 – Décisions dans l'organigramme de modèle de contexte	97
Tableau D.11 – Décodage dans l'organigramme de modèle de contexte	97
Tableau E.1 – Gains de sous-bande	99
Tableau F.1 – Niveau de décomposition n_b pour sous-bande b	102
Tableau F.2 – Extension vers la gauche	110
Tableau F.3 – Extension vers la droite	110
Tableau F.4 – Définition des paramètres de lissage pour le filtre irréversible 9-7	112
Tableau F.5 – Définition des coefficients g_n	112
Tableau F.6 – Expressions intermédiaires (r_0, r_1, s_0, t_0)	112
Tableau F.7 – Expressions intermédiaires	113
Tableau F.8 – Extension vers la gauche	121
Tableau F.9 – Extension vers la droite	122
Tableau I.1 – Structure binaire d'une boîte	136
Tableau I.2 – Boîtes définies	137
Tableau I.3 – Valeurs de marque légale	138
Tableau I.4 – Format du contenu de la boîte de type de fichier	139
Tableau I.5 – Format du contenu de la boîte d'en-tête d'image	141
Tableau I.6 – Valeurs de bits par composante	142
Tableau I.7 – Format du contenu de la boîte de bits par composante	142
Tableau I.8 – Valeurs BPC ⁱ	143
Tableau I.9 – Valeurs légales du champ METH	144
Tableau I.10 – Valeurs légales EnumCS	145
Tableau I.11 – Format du contenu de la boîte de spécification de couleur	145
Tableau I.12 – Format du contenu de la boîte de palette	146
Tableau I.13 – Valeurs B ⁱ	147
Tableau I.14 – Valeurs de champ MTYP ⁱ	147
Tableau I.15 – Format du contenu de la boîte d'affectation des composantes	148
Tableau I.16 – Valeur de champ Typ ⁱ	149
Tableau I.17 – Valeurs de champ Assoc ⁱ	149
Tableau I.18 – Couleurs indiquées par le champ Assoc ⁱ	150
Tableau I.19 – Format de la boîte de définition de canal	151
Tableau I.20 – Format du contenu de la boîte de résolution de capture	152
Tableau I.21 – Format du contenu de la boîte de résolution d'affichage par défaut	153
Tableau I.22 – Format du contenu de la boîte de flux codé contigu	154
Tableau I.23 – Format du contenu d'une boîte d'identification UUID	155
Tableau I.24 – Valeurs de structure de données de contenu de boîte de liste d'identificateurs UUID	156
Tableau I.25 – Valeurs de structure de données de contenu de boîte d'URL d'entrée de données	157
Tableau J.1 – Définition des réponses impulsionnelles pour le banc 9-7 de filtres irréversibles d'analyse	161
Tableau J.2 – Définition des réponses impulsionnelles pour le banc 9-7 de filtres irréversibles de synthèse	161
Tableau J.3 – Echantillons de composante-pavé de source	162
Tableau J.4 – Coefficients de sous-bande 2LL (transformation irréversible en ondelettes 9-7)	162
Tableau J.5 – Coefficients de sous-bande 2HL (transformation irréversible en ondelettes 9-7)	162
Tableau J.6 – Coefficient de sous-bande 2LH (transformation irréversible en ondelettes 9-7)	163
Tableau J.7 – Coefficients de sous-bande 2HH (transformation irréversible en ondelettes 9-7)	163
Tableau J.8 – Coefficients de sous-bande 1HL (transformation irréversible en ondelettes 9-7)	163
Tableau J.9 – Coefficients de sous-bande 1LH (transformation irréversible en ondelettes 9-7)	163
Tableau J.10 – Coefficients de sous-bande 1HH (transformation irréversible en ondelettes 9-7)	164
Tableau J.11 – Coefficients de sous-bande 2LL (transformation réversible en ondelettes 5-3)	164
Tableau J.12 – Coefficients de sous-bande 2HL (transformation réversible en ondelettes 5-3)	164
Tableau J.13 – Coefficients de sous-bande 2LH (transformation réversible en ondelettes 5-3)	164

	<i>Page</i>
Tableau J.14 – Coefficients de sous-bande 2HH (transformation réversible en ondelettes 5-3)	165
Tableau J.15 – Coefficients de sous-bande 1HL (transformation réversible en ondelettes 5-3).....	165
Tableau J.16 – Coefficients de sous-bande 1LH (transformation réversible en ondelettes 5-3).....	165
Tableau J.17 – Coefficients de sous-bande 1HH (transformation réversible en ondelettes 5-3)	165
Tableau J.18 – Outils d'élasticité aux erreurs	176
Tableau J.19 – Balises de traitement utilisées par un profil ICC restreint.....	178
Tableau J.20 – Décodage du premier en-tête de paquet	185
Tableau J.21 – Décodage du second en-tête de paquet.....	185
Tableau J.22 – Décodage arithmétique du premier bloc codé	186
Tableau J.23 – Décodage arithmétique du second bloc codé	187
Tableau J.24 – Pondération de fréquence recommandée.....	189
Tableau J.25 – Pondération de fréquence recommandée pour images (en couleurs) à composantes multiples	189
Tableau J.26 – Valeurs d'enregistrement de composante (CRG)	196
Tableau L.1 – Revendications de droits de propriété intellectuelle reçues.....	200

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**Technologies de l'information – Système de codage d'images JPEG 2000:
système de codage noyau**

1 Domaine d'application

La présente Recommandation | Norme internationale définit un ensemble de méthodes de compression sans perte (au bit près) et avec perte pour codage d'images numériques fixes, monochromes, à modelé continu en échelle de gris, à palette ou à modelé continu de couleurs.

La présente Recommandation | Norme internationale:

- spécifie des processus de décodage permettant de convertir des données d'image comprimées en données d'image reconstituées;
- spécifie une syntaxe de flux codé contenant des informations permettant d'interpréter des données d'image comprimées;
- spécifie un format de fichier;
- fournit des directives sur les processus de codage permettant de convertir des données d'image de source en données d'image comprimées;
- fournit des directives sur la façon d'implémenter ces processus en pratique.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT (TSB) tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T T.81 (1992) | ISO/CEI 10918-1:1994, *Technologies de l'information – Compression numérique et codage des images fixes de nature photographique – Prescriptions et lignes directrices*.
- Recommandation UIT-T T.88 (2000) | ISO/CEI 14492:2001, *Technologies de l'information – Codage avec ou sans perte des images au trait*.
- ISO/CEI 646:1991, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'information*.
- ISO 8859-15:1999, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 15: Alphabet latin n° 9*.
- Recommandation UIT-T T.84 (1996) | ISO/CEI 10918-3:1997, *Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: extensions*.
- Recommandation UIT-T T.84 (1996)/Amd.1 (1999) | ISO/CEI 10918-3:1997/Amd.1:1999, *Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: extensions – Amendement 1: Dispositions visant à permettre l'enregistrement de nouveaux types de compression et de nouvelles versions dans l'en-tête du format SPIFF*.
- Recommandation UIT-T T.86 (1998) | ISO/CEI 10918-4:1999, *Technologies de l'information – Compression numérique et codage des images fixes à modelé continu: enregistrement des profils JPEG, des profils SPIFF, des étiquettes SPIFF, des espaces chromatiques SPIFF, des marqueurs APPn, des types de compression SPIFF et des organismes d'enregistrement (REGAUT)*.
- Recommandation UIT-T T.87 (1998) | ISO/CEI 14495-1:2000, *Technologies de l'information – Compression sans perte et quasi sans perte d'images fixes à modelé continu – Principes*.

2.2 Autres références

- Spécification ICC.1:1998-09, *Spécification de format de profil ICC*.
- CEI 61966-2-1:1999, *Mesure et gestion de la couleur dans les systèmes et appareils multimédia – Partie 2-1: Gestion de la couleur – Espace chromatique RVB par défaut – sRVB*.
- W3C REC-xml-19980210, *Langage balisé extensible (XML 1.0)*.
- IETF RFC 2279 (1998), UTF-8, *Format de transformation de l'ISO 10646*.
- ISO/CEI 11578:1996, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Appel de procédures à distance (RPC)*.
- CEI 61966-2-1:1999/Amd.1:2003, *Mesure et gestion de la couleur dans les systèmes et appareils multimédia – Partie 2-1: Gestion de la couleur – Espace chromatique RVB par défaut – sRVB*.

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

- 3.1** $\lfloor x \rfloor$, **fonction plancher (minorante)**: plus grand entier inférieur ou égal à x .
- 3.2** $\lceil x \rceil$, **fonction plafond (majorante)**: plus petit entier supérieur ou égal à x .
- 3.3** **filtre réversible 5-3**: paire particulière de filtres, utilisée dans la transformation en ondelettes. Cette paire de filtres réversibles possède 5 prises passe-bas et 3 prises passe-haut.
- 3.4** **filtre irréversible 9-7**: paire particulière de filtres, utilisée dans la transformation en ondelettes. Cette paire de filtres irréversibles possède 9 prises passe-bas et 7 prises passe-haut.
- 3.5** **AND**: opérateur logique binaire ET.
- 3.6** **codeur arithmétique**: codeur entropique qui convertit des chaînes de longueur variable en codes de longueur variable (codage) et vice versa (décodage).
- 3.7** **canal auxiliaire**: canal qui est utilisé par l'application hors du domaine d'application de la conversion dans l'espace chromatique. Par exemple, un canal d'opacité ou un canal de profondeur sera un canal auxiliaire.
- 3.8** **bit**: contraction du terme anglais "binary digit" (chiffre binaire); unité d'information représentée par zéro ou un.
- 3.9** **plan binaire**: matrice binaire dimensionnelle. Dans la présente Recommandation | Norme internationale, un plan binaire se rapporte à tous les bits de même poids d'un ensemble de coefficients ou d'échantillons. Il peut s'agir d'un plan binaire dans une composante, une composante-pavé, un bloc codé, une région intéressante, ou autre.
- 3.10** **flux binaire**: séquence binaire effective résultant du codage d'une séquence de symboles. Ce flux ne comprend pas les marqueurs ou segments marqueurs contenus dans l'en-tête principal ou l'en-tête des éléments de pavé, ou le marqueur de fin EOC. Il comprend en revanche tout en-tête de paquet et tous marqueurs et segments marqueurs du flux ne se trouvant pas dans l'en-tête principal ou l'en-tête d'élément de pavé.
- 3.11** **gros-boutiste**: représentation de valeur dans laquelle les éléments binaires se présentent en séquence du poids fort au plus faible poids.
- 3.12** **boîte**: partie du format de fichier définie par une longueur et par un type de boîte unique. Les boîtes de certains types peuvent contenir d'autres boîtes.
- 3.13** **contenu de boîte**: données enveloppées dans la structure de boîte. Le contenu d'une boîte particulière est mémorisé dans le champ DBox dans la structure de boîte de données.
- 3.14** **type de boîte**: spécifie la sorte d'information qui doit être mémorisée avec la boîte. Le type d'une boîte particulière est mémorisé dans le champ TBox dans la structure de boîte de données.
- 3.15** **octet**: huit bits.
- 3.16** **canal**: composante logique de l'image. Un canal peut être une représentation directe d'une composante extraite du flux codé, ou peut être produit par l'application d'une palette à une composante extraite du flux codé.
- 3.17** **passé de nettoyage**: passe de codage effectuée sur un même plan binaire d'un bloc codé de coefficients. La première et unique passe de codage pour le premier plan binaire significatif est une passe de nettoyage; la troisième et dernière passe de chaque plan binaire restant est une passe de nettoyage.

- 3.18 flux codé:** ensemble d'un ou de plusieurs flux binaires avec l'en-tête principal, les en-têtes d'éléments de pavé et le marqueur de fin EOC requis pour leur décodage et leur expansion en données d'image. Il s'agit des données d'image sous forme comprimée avec toute la signalisation nécessaire au décodage.
- 3.19 bloc codé:** groupement rectangulaire des coefficients issus de la même sous-bande d'une composante-pavé.
- 3.20 exploration d'un bloc codé:** ordre dans lequel les coefficients contenus dans un bloc codé sont visités pendant une passe de codage. Le bloc codé est traité par zones, consistant chacune de quatre rangées (ou de toutes les rangées restantes s'il y en a moins de quatre) et couvrant la largeur du bloc codé. Chaque zone est traitée colonne par colonne de haut en bas et de gauche à droite.
- 3.21 codeur:** matérialisation d'un procédé de codage ou de décodage.
- 3.22 passe de codage:** passe complète par un bloc codé dans laquelle les valeurs de coefficient et le contexte approprié sont appliqués. Il y a trois types de passes de codage: passe de propagation de signification, passe de raffinement de grandeur et passe de nettoyage. Le résultat de chaque passe (après codage arithmétique, si l'évitement sélectif du codage arithmétique n'est pas utilisé) est un flux de données d'image comprimées.
- 3.23 coefficient:** valeur qui résulte d'une transformation.
- 3.24 canal de couleur:** canal qui fonctionne comme entrée dans un système de transformation de couleur. Par exemple, un canal de rouge ou un canal d'échelle de gris sera un canal de couleur.
- 3.25 composante:** matrice bidimensionnelle d'échantillons. Une image se compose normalement de plusieurs composantes, par exemple la rouge, la verte et la bleue.
- 3.26 données d'image comprimées:** tout ou partie d'un flux codé. Peut également se rapporter à un ensemble de flux binaires dans tout ou partie d'un flux codé.
- 3.27 lecteur conforme:** application qui lit et interprète correctement un fichier JP2.
- 3.28 contexte:** fonction de coefficients déjà décodés, utilisée pour conditionner le décodage du coefficient en cours de traitement.
- 3.29 étiquette de contexte:** index arbitraire utilisé pour distinguer différentes valeurs contextuelles. Les étiquettes sont utilisées par commodité de notation plutôt que par notation normative.
- 3.30 vecteur de contexte:** vecteur binaire composé des états de signification des coefficients inclus dans un contexte.
- 3.31 décodeur:** matérialisation d'un processus de décodage et, facultativement, d'un processus de transformation de couleur.
- 3.32 processus de décodage:** processus qui reçoit en entrée tout ou partie d'un flux codé et génère en sortie tout ou partie d'une image reconstruite.
- 3.33 niveau de décomposition:** collection de sous-bandes d'ondelettes où tous les coefficients possèdent le même impact spatial ou la même portée par rapport aux échantillons de la composante source. Ces collections comprennent toutes les sous-bandes HL, LH et HH de la même décomposition bidimensionnelle en sous-bandes. Pour le dernier niveau de décomposition, la sous-bande LL est également incluse.
- 3.34 marqueurs et segments marqueurs délimitants:** marqueurs et segments marqueurs qui donnent des informations sur les points de début et de fin de structures contenues dans le flux codé.
- 3.35 transformation en ondelettes discrètes (DWT, *discrete wavelet transformation*):** transformation qui convertit itérativement un signal en deux ou plusieurs signaux filtrés et échantillonnés correspondant à différentes bandes de fréquences. Cette transformation opère sur des échantillons spatialement discrets.
- 3.36 codeur:** matérialisation d'un processus de codage.
- 3.37 processus de codage:** processus qui reçoit en entrée tout ou partie des données d'image de source et qui génère en sortie un flux codé.
- 3.38 format de fichier:** informations de flux codé et données supports additionnelles non explicitement requises pour le décodage d'un flux codé. Comme exemple de données supports, on peut citer les champs de texte fournissant les informations de titrage, de sécurité et d'historique, les données déterminant le placement de multiples flux codés dans un fichier de données, et les données de prise en charge de l'échange entre plates-formes ou de la conversion en d'autres formats de fichier.
- 3.39 marqueurs d'information fixe et segments marqueurs d'information fixe:** marqueurs et segments marqueurs qui donnent des informations sur l'image originale.

- 3.40 marqueurs fonctionnels et segments marqueurs fonctionnels:** marqueurs et segments marqueurs qui donnent des informations sur les procédures de codage.
- 3.41 résolution de grille:** résolution spatiale de la grille de référence, spécifiant la distance entre points voisins sur la grille de référence.
- 3.42 bits de garde:** bits de plus fort poids additionnels qui ont été ajoutés aux données d'échantillon.
- 3.43 en-tête:** partie du flux codé qui contient seulement des marqueurs et des segments marqueurs (en-tête principal et en-tête de pavé) ou partie de signalisation d'un paquet (en-tête de paquet).
- 3.44 sous-bande HH:** sous-bande obtenue par filtrages passe-haut horizontal et vertical directs. Cette sous-bande contribue à la reconstruction par filtrages passe-haut vertical et horizontal inverses.
- 3.45 sous-bande HL:** sous-bande obtenue par filtrages passe-haut horizontal et passe-bas vertical directs. Cette sous-bande contribue à la reconstruction par filtrages passe-bas vertical et passe-haut horizontal inverses.
- 3.46 image:** ensemble de toutes les composantes.
- 3.47 zone d'image:** partie rectangulaire de la grille de référence, enregistrée par décalages par rapport à l'origine et l'étendue de la grille de référence.
- 3.48 décalage de zone d'image:** nombre de points de grille de référence vers le bas et vers la droite de l'origine de la grille de référence, où l'origine de la zone d'image peut être trouvée.
- 3.49 données d'image:** échantillons de composantes constituant une image. Les données d'image peuvent désigner les données d'image de source ou les données d'image reconstituées.
- 3.50 marqueurs et segments marqueurs dans le flux binaire:** marqueurs et segments marqueurs qui offrent la capacité d'élasticité aux erreurs.
- 3.51 marqueurs informationnels et segments marqueurs informationnels:** marqueurs et segments marqueurs qui donnent des informations auxiliaires.
- 3.52 irréversible:** transformation, progression, système, quantification, ou autre processus qui, en raison d'une erreur systémique ou de quantification, interdit le rétablissement sans perte. Un processus irréversible ne peut conduire qu'à une compression avec perte.
- 3.53 fichier JP2:** nom d'un fichier dans le format de fichier décrit dans la présente Recommandation | Norme internationale. Structurellement, un fichier JP2 est une séquence contiguë de boîtes.
- 3.54 JPEG:** terme utilisé comme référence mondiale aux processus de codage et décodage dans les Recommandations | Normes internationales suivantes:
- Rec. UIT-T T.81 | ISO/CEI 10918-1, Technologies de l'information – Compression numérique et codage des images fixes de nature photographique: prescriptions et lignes directrices;
 - Rec. UIT-T T.83 | ISO/CEI 10918-2, Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: tests de conformité;
 - Rec. UIT-T T.84 | ISO/CEI 10918-3, Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: extensions;
 - Rec. UIT-T T.84/Amd.1 | ISO/CEI 10918-3/Amd.1, Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: Extensions – Amendement 1: Dispositions visant à permettre l'enregistrement de nouveaux types de compression et de nouvelles versions dans l'en-tête du format SPIFF;
 - Rec. UIT-T T.86 | ISO/CEI 10918-4, Technologies de l'information – Compression numérique et codage des images fixes à modelé continu: enregistrement des profils JPEG, des profils SPIFF, des étiquettes SPIFF, des espaces chromatiques SPIFF, des marqueurs APPn, des types de compression SPIFF et des organismes d'enregistrement (REGAUT).
- 3.55 JPEG 2000:** terme utilisé comme référence mondiale aux processus de codage et de décodage dans la présente Recommandation | Norme internationale et dans leur matérialisation en applications.
- 3.56 sous-bande LH:** sous-bande obtenue par filtrages passe-bas horizontal et passe-haut vertical directs. Cette sous-bande contribue à la reconstruction par filtrages passe-haut vertical et passe-bas horizontal inverses.
- 3.57 sous-bande LL:** sous-bande obtenue par filtrages passe-bas horizontal et vertical directs. Cette sous-bande contribue à la reconstruction par filtrages passe-bas vertical et horizontal inverses.

- 3.58 couche:** collection de données d'image comprimées issue des passes de codage d'un ou de plusieurs blocs codés d'une composante-pavé. Les couches ont un ordre de codage et de décodage qui doit être préservé.
- 3.59 sans perte:** terme qualifiant l'effet de processus globaux de codage et de décodage dans lesquels la sortie du processus de décodage est identique à l'entrée du processus de codage. Un rétablissement sans distorsion peut être assuré. Tous les processus ou toutes les étapes de codage utilisés pour le codage et le décodage sont alors réversibles.
- 3.60 avec perte:** terme qualifiant l'effet de processus globaux de codage et de décodage dans lesquels la sortie du processus de décodage n'est pas identique à l'entrée du processus de codage. Il y a distorsion (mesurée mathématiquement). Au moins un des processus ou étapes de codage ou de décodage est alors irréversible.
- 3.61 passe de raffinement de grandeur:** type de passe de codage.
- 3.62 en-tête principal:** groupe de marqueurs et de segments marqueurs au début du flux codé qui décrit les paramètres d'image et de codage qui peuvent s'appliquer à chaque pavé et composante-pavé.
- 3.63 marqueur:** code de deux octets dans lequel le premier octet est le nombre hexadécimal FF (0xFF) et où le second octet est une valeur comprise entre 1 (0x01) et le nombre hexadécimal FE (0xFE).
- 3.64 segment marqueur:** marqueur et son ensemble de paramètres (non vide) associé.
- 3.65 mod:** opérateur de congruence, tel que si $\text{mod}(y,x) = z$, alors $0 \leq z < x$ et $y-z$ est un multiple de x .
- 3.66 paquet:** partie du flux codé composé d'un en-tête de paquet et des données d'image comprimées issus d'une seule couche d'un district d'un même niveau de résolution d'une composante-pavé.
- 3.67 en-tête de paquet:** portion du paquet qui contient la signalisation nécessaire pour le décodage de ce paquet.
- 3.68 marqueurs de pointage et segments marqueurs de pointage:** marqueurs et segments marqueurs qui donnent des informations sur l'emplacement de structures contenues dans le flux codé.
- 3.69 district:** région rectangulaire d'une composante-pavé transformée, dans chaque niveau de résolution, utilisée pour limiter la taille de paquets.
- 3.70 précision:** nombre de bits alloués à un échantillon, à un coefficient, ou à une autre représentation numérique binaire.
- 3.71 progression:** ordonnancement d'un flux codé où le décodage de chaque bit successif contribue à une "meilleure" reconstruction de l'image. La mesure qui rend compte de "l'amélioration" de la reconstruction est fonction de l'application. Exemples de progressivité: résolution croissante ou meilleure fidélité d'échantillon.
- 3.72 quantification:** méthode de réduction de la précision de différents coefficients visant à réduire le nombre de bits utilisés pour les représenter. La quantification équivaut à une division lors de la compression et à une multiplication lors de la décompression. La quantification peut être réalisée par une opération explicite avec une valeur de quantification donnée ou par omission (troncature) de passes de codage dans le flux codé.
- 3.73 ordre de balayage:** ordre séquentiel particulier de données d'un type quelconque dans une matrice. L'ordre de balayage commence au coin gauche supérieur des données et se déplace vers le point de données situé immédiatement à droite et ainsi de suite jusqu'à la fin de la rangée. Une fois que la fin de la rangée est atteinte, le prochain point de données dans la séquence est celui qui est situé le plus à gauche immédiatement au-dessous de la rangée actuelle. Cet ordre est suivi jusqu'à la fin de la matrice.
- 3.74 image reconstruite:** image générée à la sortie d'un décodeur.
- 3.75 échantillon reconstruit:** échantillon reconstruit par le décodeur. Il est toujours égal à la valeur originale de l'échantillon en codage sans perte mais peut différer par rapport à cette valeur en codage avec perte.
- 3.76 grille de référence:** matrice rectangulaire régulière de points, utilisée pour définir d'autres matrices rectangulaires de données. Exemples: composantes et pavés.
- 3.77 pavé de référence:** sous-grille rectangulaire de dimensions quelconques associée à la grille de référence.
- 3.78 région intéressante (ROI, *region of interest*):** ensemble des coefficients considérés comme ayant un intérêt particulier du point de vue d'une mesure définie par l'utilisateur.
- 3.79 niveau de résolution:** équivalent au niveau de décomposition avec une seule exception: la sous-bande LL est également un niveau de résolution séparé.
- 3.80 réversible:** transformation, progression, système, ou autre processus qui n'admet pas d'erreur systémique ou de quantification et qui permet donc un rétablissement sans perte du signal.
- 3.81 échantillon:** élément de la matrice bidimensionnelle constitutive d'une composante.

- 3.82 symbole de segmentation:** symbole spécial codé avec un contexte uniforme à la fin de chaque passe de codage afin d'assurer l'élasticité aux erreurs.
- 3.83 évitement sélectif du codage arithmétique:** style de codage où certaines des passes de bloc codé ne sont pas codées par le codeur arithmétique, et où les bits à coder sont adjoints directement au flux binaire sans codage.
- 3.84 décalage:** multiplication ou division d'un nombre par des puissances de deux.
- 3.85 bit de signe:** bit qui indique si un nombre est positif (valeur 0) ou négatif (valeur 1).
- 3.86 notation en signe-magnitude:** représentation binaire d'un entier utilisant la valeur absolue (positive) de l'entier et un seul bit de signe distinct.
- 3.87 passe de propagation de signification:** passe de codage effectuée sur un même plan binaire d'un bloc codé de coefficients.
- 3.88 état de signification:** état d'un coefficient dans un plan binaire particulier. Si un coefficient, en notation en signe-magnitude, a son premier bit 1 de grandeur dans ou avant le plan binaire en cause, ce coefficient est considéré comme "significatif". Sinon, il est considéré comme "non significatif".
- 3.89 image source:** image utilisée comme entrée dans un codeur.
- 3.90 sous-bande:** groupe de coefficients de transformée résultant de la même séquence d'opérations de filtrage passe-bas et passe-haut, aussi bien verticalement qu'horizontalement.
- 3.91 coefficient de sous-bande:** coefficient de transformée dans une sous-bande donnée.
- 3.92 décomposition en sous-bandes:** transformation d'une composante-pavé d'image en sous-bandes.
- 3.93 superboîte:** boîte qui elle-même contient une séquence contiguë de boîtes (et une seule séquence contiguë de boîtes). Comme le fichier JP2 contient une seule séquence contiguë de boîtes, le fichier JP2 est lui-même considéré comme une superboîte. Lorsqu'il est utilisé comme partie d'une relation entre deux boîtes, le terme *superboîte* se rapporte à la boîte qui contient directement l'autre boîte.
- 3.94 pavé:** matrice rectangulaire de points sur la grille de référence, repérée par un décalage par rapport à l'origine de la grille de référence et définie par une largeur et une hauteur. Les pavés qui se chevauchent sont utilisés pour définir des composantes-pavés.
- 3.95 composante-pavé:** tous les échantillons d'une composante donnée dans un pavé.
- 3.96 index de pavé:** index du pavé actuel allant de zéro jusqu'au nombre de pavés moins un.
- 3.97 partie de pavé:** partie du flux codé avec des données d'image comprimées pour tout ou partie d'un pavé. Une partie de pavé comporte un ou plusieurs paquets qui constituent le pavé codé.
- 3.98 en-tête de partie de pavé:** groupe de marqueurs et de segments marqueurs situé au début de chaque partie de pavé du flux codé et qui décrivent les paramètres de codage de la partie de pavé.
- 3.99 index de partie de pavé:** index de la partie de pavé actuel allant de zéro jusqu'au nombre de parties de pavé moins un dans un pavé donné.
- 3.100 transformation:** application mathématique d'un espace de signal à un autre.
- 3.101 coefficient de transformée:** valeur qui est le résultat d'une transformation.
- 3.102 XOR:** opérateur logique "OU" exclusif.

4 Abréviations et symboles

4.1 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes s'appliquent.

CCITT	Comité consultatif international télégraphique et téléphonique, maintenant UIT-T
ICC	Consortium international de la couleur (<i>international colour consortium</i>)
ICT	Transformation chromatique irréversible (<i>irreversible component transform</i>)
CEI	Commission électrotechnique internationale
ISO	Organisation internationale de normalisation (<i>International Organization for Standardization</i>)

ITTF	Groupe d'action pour la technologie de l'information (<i>information technology task force</i>)
UIT	Union internationale des télécommunications
UIT-T	Union internationale des télécommunications – Secteur de la normalisation des télécommunications (anciennement CCITT)
JPEG	Groupe mixte d'experts sur les images demi-tons (<i>joint photographic experts group</i>) – Comité mixte ISO/UIT chargé de l'élaboration des normes relatives au codage des images fixes à modelé continu. Cet acronyme se rapporte également aux normes produites par ce comité: UIT-T T.81 ISO/CEI 10918-1, UIT-T T.83 ISO/CEI 10918-2, UIT-T T.84 ISO/CEI 10918-3 et UIT-T T.87 ISO/CEI 14495
JURA	Organisme d'enregistrement des utilitaires JPEG (<i>JPEG utilities registration authority</i>)
1D-DWT	Transformation unidimensionnelle en ondelettes discrètes (<i>one-dimensional discrete wavelet transformation</i>)
FDWT	Transformation directe en ondelettes discrètes (<i>forward discrete wavelet transformation</i>)
IDWT	Transformation inverse en ondelettes discrètes (<i>inverse discrete wavelet transformation</i>)
LSB	Bit de plus faible poids (<i>least significant bit</i>)
MSB	Bit de plus fort poids (<i>most significant bit</i>)
PCS	Espace de connexion de profil (<i>profile connection space</i>)
RCT	Transformation chromatique réversible (<i>reversible component transform</i>)
ROI	Région intéressante (<i>region of interest</i>)
SNR	Rapport signal sur bruit (<i>signal to noise ratio</i>)
UCS	Jeu de caractères universel (<i>universal character set</i>)
URI	Identificateur uniforme de ressources (<i>uniform resource identifier</i>)
URL	Localisateur uniforme de ressources (<i>uniform resource locator</i>)
UTF-8	Format de transformation UCS à 8 bits (<i>UCS transformation format 8</i>)
UUID	Identificateur universel unique (<i>universal unique identifier</i>)
XML	Langage de balisage extensible (<i>extensible markup language</i>)
W3C	Consortium mondial W3 (<i>world-wide web consortium</i>)

4.2 Symboles

Pour les besoins de la présente Recommandation | Norme internationale, les symboles suivants s'appliquent.

0x----	Indique un nombre hexadécimal
 <i>nnn</i>	Nombre à trois chiffres précédé par une barre oblique inverse indiquant la valeur d'un octet isolé dans une chaîne de caractères, où les trois chiffres spécifient la valeur octale de cet octet
ϵ_b	Exposant de la valeur de quantification pour une sous-bande définie dans les marqueurs QCD et QCC.
μ_b	Mantisse de la valeur de quantification pour une sous-bande définie dans les marqueurs QCD et QCC.
M_b	Nombre maximal de plans binaires codés dans un bloc codé.
N_L	Nombre de niveaux de décomposition comme défini dans les marqueurs COD et COC.
R_b	Etendue dynamique d'un échantillon de composante comme défini dans le marqueur SIZ.
COC	Marqueur de composante de style de codage (<i>coding style component marker</i>)
COD	Marqueur de style de codage par défaut (<i>coding style default marker</i>)
COM	Marqueur de commentaire (<i>comment marker</i>)
CRG	Marqueur d'enregistrement de composante (<i>component registration marker</i>)
EPH	Marqueur de fin d'en-tête de paquet (<i>end of packet header marker</i>)
EOC	Marqueur de fin de flux codé (<i>end of codestream marker</i>)
PLM	Marqueur d'en-tête principal de longueur de paquet (<i>packet length, main header marker</i>)

PLT	Marqueur d'en-tête de partie de pavé de longueur de paquet (<i>packet length, tile-part header marker</i>)
POC	Marqueur de changement de l'ordre de progression (<i>progression order change marker</i>)
PPM	Marqueur d'en-tête principal de paquet compact (<i>packed packet headers, main header marker</i>)
PPT	Marqueur d'en-tête de partie de pavé de paquet compact (<i>packed packet headers, tile-part header marker</i>)
QCC	Marqueur de composante de quantification (<i>quantization component marker</i>)
QCD	Marqueur de quantification par défaut (<i>quantization default marker</i>)
RGN	Marqueur de région intéressante (<i>region-of-interest marker</i>)
SIZ	Marqueur de dimensions d'image et de pavé (<i>image and tile size marker</i>)
SOC	Marqueur de début de flux codé (<i>start of codestream marker</i>)
SOP	Marqueur de début de paquet (<i>start of packet marker</i>)
SOD	Marqueur de début de données (<i>start of data marker</i>)
SOT	Marqueur de début de partie de pavé (<i>start of tile-part marker</i>)
TLM	Marqueur de longueurs de partie de pavé (<i>tile-part lengths marker</i>)

5 Description générale

La présente Recommandation | Norme internationale décrit un système de compression d'image qui offre une grande flexibilité, non seulement pour la compression d'images, mais également pour l'accès au flux codé. Le flux codé fournit un certain nombre de mécanismes pour localiser et extraire des portions des données d'image comprimées aux fins de retransmission, mémorisation, affichage, ou édition. Cet accès permet la mémorisation et l'extraction des données d'image comprimées appropriées à une application donnée, sans décodage.

La subdivision aussi bien des données d'image originales que des données d'image comprimées selon un certain nombre de méthodes se traduit par la capacité d'extraire des données d'image à partir des données d'image comprimées afin de former une image reconstruite avec une résolution inférieure ou une précision inférieure, ou de former des régions de l'image originale. Cela permet d'adapter un flux codé au canal de transmission, au dispositif de mémorisation, ou au dispositif d'affichage, quels que soient les dimensions, le nombre de composantes et la précision d'échantillonnage de l'image originale. Le flux codé peut être manipulé sans décodage afin d'obtenir un arrangement plus efficace pour une application donnée.

Donc, les caractéristiques évoluées de la présente Recommandation | Norme internationale permettent d'utiliser efficacement un simple flux codé dans un certain nombre d'applications. Les plus grands dispositifs de l'image de source peuvent fournir un flux codé qui sera facilement traité par le plus petit dispositif d'affichage d'image, par exemple.

En général, la présente Recommandation | Norme internationale traite de trois domaines: données d'image spatiales (échantillons), données d'image transformées (coefficients), et données d'image comprimées. Certaines entités (par exemple composante-pavé) peuvent exister dans ces trois domaines. D'autres entités (par exemple blocs codés ou paquets) ne peuvent exister que dans un seul domaine (par exemple transformées ou données d'image comprimées, selon le cas). La subdivision d'une entité en autre entité dans le même domaine (par exemple une composante en composantes-pavés) est décrite séparément pour chacun des domaines.

5.1 Objet

Quatre éléments principaux sont décrits dans la présente Recommandation | Norme internationale:

- codeur: matérialisation d'un processus de codage. Un codeur reçoit en entrée des données numériques d'image de source et des spécifications paramétriques, et produit en sortie un flux codé au moyen d'un ensemble de procédures.
- décodeur: matérialisation d'un processus de décodage. Un décodeur reçoit en entrée des données d'image comprimées et des spécifications paramétriques et produit en sortie, au moyen d'un ensemble spécifié de procédures, des données numériques d'image reconstituées.
- syntaxe du flux codé: représentation des données d'image comprimées qui comporte toutes les spécifications paramétriques requises par le processus de décodage.
- format de fichier facultatif: format d'échange entre environnements applicatifs. Le flux codé peut être utilisé par d'autres formats de fichier ou rester autonome sans ce format de fichier.

5.2 Flux codé

Le flux codé est un train linéaire d'éléments binaires du premier au dernier bit. Par commodité, il peut être subdivisé en octets (8 bits), à partir du premier bit du flux codé, le bit "antérieur" d'un octet étant considéré comme le bit de plus fort poids de l'octet dans le cas, par exemple, de la représentation d'un nombre hexadécimal. Ce train d'octets peut être subdivisé en groupes d'octets consécutifs. La représentation d'une valeur par un nombre hexadécimal est parfois implicite dans le texte, par exemple lors de la description d'octets ou de groupes d'octets qui n'ont pas une représentation "naturelle" de valeur numérique.

5.3 Principes de codage

Les principales procédures de la présente Recommandation | Norme internationale sont représentées dans la Figure 5-1, qui ne montre que l'ordre de décodage. Les données d'image comprimées sont déjà attribuées théoriquement à des portions des données d'image. Les procédures sont présentées dans les Annexes dans l'ordre du processus de décodage. Le processus de codage est résumé ci-dessous.

NOTE 1 – Les Annexes A à I sont considérées comme normatives dans la présente Recommandation | Norme internationale. Certains paragraphes et certaines notes, indiqués comme tels, ainsi que tous les exemples, sont toutefois informatifs.

De nombreuses images ont des composantes multiples. La présente Recommandation | Norme internationale contient une transformation multiple de composante permettant de décorréliser trois composantes. C'est la seule fonction de la présente Recommandation | Norme internationale qui associe des composantes les unes avec les autres (voir Annexe G).

Les composantes d'image peuvent être subdivisées en pavés. Ces composantes-pavés sont des matrices rectangulaires qui se rapportent à la même portion de chacune des composantes qui constituent l'image. Donc, le pavage de l'image crée en fait des composantes-pavés qui peuvent être extraites ou décodées indépendamment les unes des autres. Cette indépendance des pavés fournit une des méthodes d'extraction d'une région de l'image (voir Annexe B).

Les composantes-pavés sont décomposées en différents niveaux de décomposition au moyen d'une transformation en ondelettes. Ces niveaux de décomposition contiennent un certain nombre de sous-bandes remplies de coefficients qui décrivent les caractéristiques fréquentielles de l'espace horizontal et vertical des composantes-pavés originales. Les coefficients fournissent des informations de fréquence sur une zone locale, plutôt que sur l'image entière comme la transformation de Fourier. En d'autres termes, un petit nombre de coefficients décrit complètement un même échantillon. Un niveau de décomposition est associé au prochain niveau de décomposition par un facteur spatial de deux. En d'autres termes, chaque niveau successif de décomposition des sous-bandes possède approximativement la moitié de la résolution horizontale et la moitié de la résolution verticale du niveau précédent. Les images de résolution inférieure à l'original sont produites par décodage d'un sous-ensemble choisi de ces sous-bandes (voir Annexe F).

Bien qu'il y ait autant de coefficients qu'il y a d'échantillons, le contenu informationnel tient à être concentré sur quelques coefficients seulement. Grâce à la quantification, le contenu informationnel d'un grand nombre de coefficients de plus faible poids est encore réduit (Annexe E). Un traitement additionnel par le codeur entropique diminue le nombre de bits requis pour représenter ces coefficients quantifiés, parfois notablement par rapport à l'image originale (voir Annexes C, D et B).

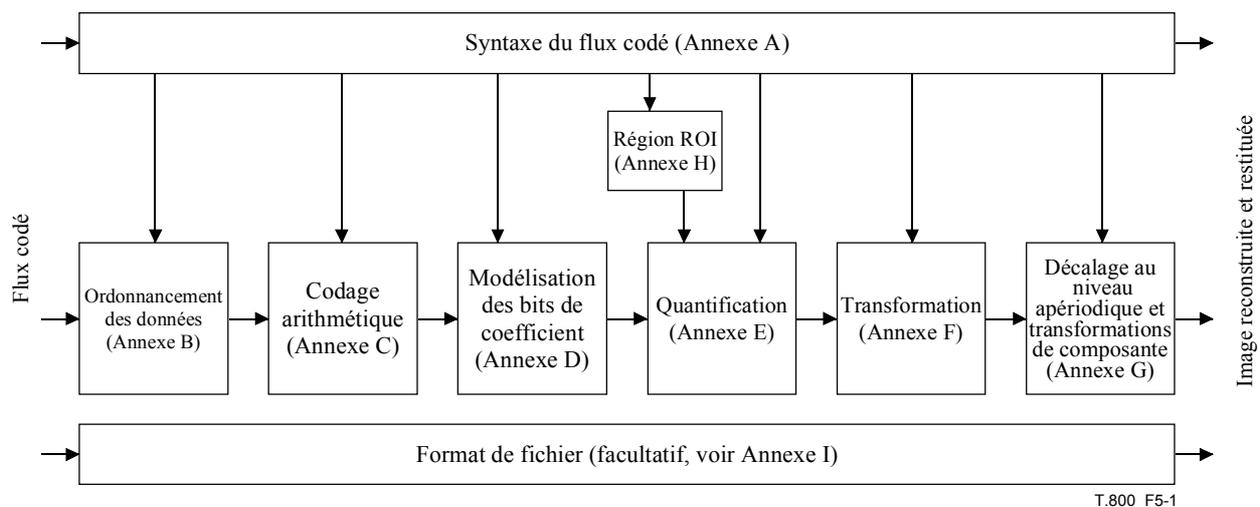


Figure 5-1 – Schéma fonctionnel de la spécification

ISO/CEI 15444-1:2004 (F)

Les sous-bandes individuelles d'une composante-pavé sont encore subdivisées en blocs codés. Ces matrices rectangulaires des coefficients peuvent être extraites indépendamment. Les plans binaires individuels de coefficients dans un bloc codé sont codés par trois passes de codage. Chacune de ces passes de codage collecte des informations contextuelles sur le plan binaire des données d'image comprimées (voir Annexe D). Un codeur arithmétique utilise ces informations contextuelles, et son état interne, pour décoder un flux binaire comprimé (voir Annexe C). Différents mécanismes de terminaison permettent différents niveaux d'extraction indépendante de ces données d'image comprimées par passe de codage.

Le flux binaire de données d'image comprimées, créé par ces passes de codage, est groupé en couches. Celles-ci sont des groupements arbitraires de passes de codage sur des blocs codés (voir Annexe B).

NOTE 2 – Bien qu'il y ait une grande flexibilité dans le groupement en couches, l'on part du principe que chaque couche successive contribue à une image de qualité supérieure.

Les coefficients de sous-bande à chaque niveau de résolution sont partitionnés en zones rectangulaires appelées *districts* (voir Annexe B).

Les paquets sont une unité fondamentale du flux codé comprimé. Un paquet contient des données d'image comprimées provenant d'une couche d'un district d'un niveau de résolution d'une composante-pavé. Les paquets fournissent une autre méthode pour extraire une région spatiale indépendamment du flux codé. Ces paquets sont entrelacés dans le flux codé au moyen de quelques méthodes différentes (voir Annexe B).

Un mécanisme est fourni qui permet de coder des données d'image comprimées correspondant à des régions intéressantes dans les composantes-pavés originales et de les placer plus tôt dans le flux binaire (voir Annexe H).

Plusieurs mécanismes sont fournis qui permettent la détection et le masquage d'erreurs binaires pouvant apparaître dans un canal de transmission bruité (voir § D.5 et J.7).

Le flux codé relatif à un pavé, organisé en paquets, est disposé en un ou plusieurs éléments de pavé. Un en-tête d'élément de pavé, constitué d'une série de marqueurs et de segments marqueurs, contient des informations sur les divers mécanismes et styles de codage qui sont nécessaires pour localiser, extraire, décoder et reconstruire chaque composante-pavé. Au début du flux codé entier se trouve un en-tête principal, constitué de marqueurs et de segments marqueurs, qui offre des informations similaires ainsi que des informations sur l'image originale (voir Annexe A).

Le flux codé est facultativement enveloppé dans un format de fichier qui permet aux applications d'interpréter la signification de l'image ainsi que d'autres informations sur celle-ci. Le format de fichier peut contenir des données en plus du flux codé (voir Annexe I).

En résumé, les procédures qui subdivisent l'image originale sont les suivantes:

- les composantes de l'image sont subdivisées en pavés rectangulaires. La composante-pavé est l'unité de base de l'image originale ou reconstruite;
- l'exécution de la transformation en ondelettes sur une composante-pavé crée des niveaux de décomposition;
- ces niveaux de décomposition sont constitués de sous-bandes de coefficients qui décrivent les caractéristiques fréquentielles de zones locales de la composante-pavé (plutôt que dans l'ensemble de celle-ci);
- les sous-bandes de coefficients sont quantifiées et réunies en matrices rectangulaires de blocs codés;
- chaque plan binaire des coefficients contenus dans un bloc codé est codé entropiquement par trois types de passe de codage;
- certains des coefficients peuvent être codés avant les autres afin de fournir une région intéressante.

A ce stade, les données d'image sont entièrement converties en données d'image comprimées. Les procédures qui réassemblent ces unités de flux binaire dans le flux codé sont les suivantes:

- des données d'image comprimées d'après les passes de codage sont réunies en couches;
- les paquets sont composés des données d'image comprimées provenant d'un district d'une même couche d'un même niveau de résolution d'une même composante-pavé. Les paquets sont l'unité de base des données d'image comprimées;
- tous les paquets provenant d'un pavé sont entrelacés selon un ordre parmi plusieurs et placés dans un ou plusieurs éléments de pavé;
- les éléments de pavé ont un en-tête d'élément de pavé descriptif et peuvent être entrelacés selon quelques ordres;
- le début du flux codé possède un en-tête principal qui décrit l'image originale et les divers styles de décomposition et de codage;

- le format facultatif de fichier décrit la signification de l'image et de ses composantes dans le contexte de l'application.

6 Exigences relatives au codeur

Un processus de codage convertit des données d'image de source en données d'image comprimées. Les Annexes A, B, C, D, E, F, G et H décrivent le processus de codage. Tous les processus de codage sont spécifiés à titre informatif.

Un codeur est une matérialisation du processus de codage. Afin d'être conforme à la présente Recommandation | Norme internationale, un codeur doit convertir des données d'image de source en données d'image comprimées, qui sont conformes à la syntaxe du flux codé spécifiée à l'Annexe A.

7 Exigences relatives au décodeur

Un processus de décodage convertit des données d'image comprimées en données d'image reconstituées. Les Annexes A à H décrivent et spécifient le processus de décodage. Tous les processus de décodage sont normatifs.

Un décodeur est une matérialisation du processus de décodage. Afin d'être conforme à la présente Recommandation | Norme internationale, un décodeur doit convertir en une image reconstruite la totalité – ou des parties spécifiques – des données d'image comprimées qui sont conformes à la syntaxe de flux codé spécifiées à l'Annexe A.

Il n'y a pas d'implémentation normative ou requise pour le codeur ou le décodeur. Dans certains cas, les descriptions utilisent des techniques d'implémentation particulières à seule fin documentaire.

7.1 Exigences de la syntaxe du flux codé

L'Annexe A décrit la syntaxe du flux codé qui définit la représentation codée des données d'image comprimées pour échange entre environnements applicatifs. Toutes les données d'image comprimées doivent être conformes aux prescriptions de syntaxe et de codage appropriées aux processus de codage définis dans la présente Recommandation | Norme internationale.

La présente Recommandation | Norme internationale ne comporte pas de définition d'observance ou de conformité. Les valeurs paramétriques de la syntaxe décrite dans l'Annexe A ne sont pas destinées à refléter les capacités requises pour indiquer la conformité.

7.2 Exigences relatives au format de fichier facultatif

L'Annexe I décrit le format facultatif de fichier contenant des métadonnées sur l'image en plus du flux codé. Ces données permettent, par exemple, un affichage sur écran ou une impression à une résolution spécifique. Le format facultatif de fichier doit, lorsqu'il est utilisé, être conforme aux prescriptions de syntaxe et de codage de format de fichier appropriées aux processus de codage définis dans la présente Recommandation | Norme internationale.

8 Exigences relatives à l'implémentation

Il y a pas d'implémentation normative ou requise pour la présente Recommandation | Norme internationale. Dans certains cas, les descriptions utilisent des techniques d'implémentation particulières à seule fin documentaire.

Annexe A

Syntaxe du flux codé

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe et dans tous ses articles, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe spécifie la syntaxe et la sémantique de marqueur et de segment marqueur définies par la présente Recommandation | Norme internationale. Ces marqueurs et segments marqueurs fournissent des informations de flux codé pour la présente Recommandation | Norme internationale. Par ailleurs, la présente annexe fournit une syntaxe de marqueur et segment marqueur qui est conçue pour être utilisée dans de futures spécifications comportant la présente Recommandation | Norme internationale en tant que référence normative.

La présente Recommandation | Norme internationale ne comporte pas de définition d'observance ou de conformité. Les valeurs paramétriques de la syntaxe décrite dans la présente annexe ne sont pas destinées à refléter les capacités requises pour assurer la conformité.

A.1 Marqueurs, segments marqueurs, et en-têtes

La présente Recommandation | Norme internationale utilise des marqueurs et segments marqueurs afin de délimiter et signaler les caractéristiques du flux codé et de l'image de source. Cet ensemble de marqueurs et de segments marqueurs constitue l'information minimale qui est nécessaire afin d'obtenir les caractéristiques de la présente Recommandation | Norme internationale; ce n'est pas un format de fichier. Un format minimal de fichier est décrit à l'Annexe I.

Les en-têtes principaux et les en-têtes d'élément de pavé sont des ensembles de marqueurs et segments marqueurs. L'en-tête principal se trouve au début du flux codé. Les en-têtes d'élément de pavé se trouvent au début de chaque élément de pavé (voir ci-dessous). Certains marqueurs et segments marqueurs sont limités à un seul des deux types d'en-têtes alors que d'autres peuvent être trouvés dans les deux types.

Chaque marqueur a une longueur de deux octets. Le premier octet se compose d'un seul octet 0xFF. Le second octet indique le marqueur spécifique et peut avoir une valeur quelconque dans l'étendue 0x01 à 0xFE. Beaucoup de ces marqueurs sont déjà utilisés dans les Recommandations UIT-T T.81 | ISO/CEI 10918-1 et UIT-T T.84 | ISO/CEI 10918-3: ils doivent être considérés comme réservés sauf utilisation spécifique.

Un seul segment marqueur comporte un marqueur et des paramètres associés, appelés paramètres de segment marqueur. Dans chaque segment marqueur, les deux premiers octets après le marqueur doivent être une valeur non signée qui indique la longueur en octets des paramètres de segment marqueur (y compris les deux octets de ce paramètre de longueur mais non compris les deux octets du marqueur proprement dit). Lorsqu'un seul segment marqueur qui n'est pas spécifié dans la présente Recommandation | Norme internationale apparaît dans un flux codé, le décodeur doit utiliser le paramètre de longueur afin de rejeter ce segment marqueur.

A.1.1 Types de marqueurs et segments marqueurs

Six types de marqueurs et segments marqueurs sont utilisés: délimitants, d'information fixe, fonctionnels, dans le flux binaire, de pointage et informationnels. Les marqueurs et segments marqueurs délimitants sont utilisés pour encadrer les en-têtes principaux et d'élément de pavé et les données du flux binaire. Les segments marqueurs d'information fixe donnent des informations requises sur l'image. L'emplacement de ces segments marqueurs, comme les marqueurs et segments marqueurs délimitants, est spécifié. Les segments marqueurs fonctionnels sont utilisés pour décrire les fonctions de codage utilisées. Les marqueurs et segments marqueurs dans le flux binaire sont utilisés pour assurer l'élasticité aux erreurs. Les segments marqueurs de pointage fournissent des décalages spécifiques dans le flux binaire. Les segments marqueurs informationnels fournissent des informations auxiliaires.

A.1.2 Similarité de syntaxe avec la Rec. UIT-T T.81 | ISO/CEI 10918-1

La syntaxe de marqueur et segment marqueur utilise la construction qui est définie dans la Rec. UIT-T T.81 | ISO/CEI 10918-1.

L'étendue de marqueur 0xFF30 à 0xFF3F est réservée par la présente Recommandation | Norme internationale pour les marqueurs sans paramètres de segment marqueur. Le Tableau A.1 montre la spécification dans laquelle ces marqueurs et segments marqueurs sont définis.

Tableau A.1 – Définitions des marqueurs

Etendue de codage de marqueur	Définition normalisée
0xFF00, 0xFF01, 0xFFFFE, 0xFFC0 to 0xFFDF	Définie dans la Rec. UIT-T T.81 ISO/CEI 10918-1
0xFFFF0 to 0xFFFF6	Définie dans la Rec. UIT-T T.84 ISO/CEI 10918-3
0xFFFF7 to 0xFFFF8	Définie dans la Rec. UIT-T T.87 ISO/CEI 14495-1
0xFF4F to 0xFF6F, 0xFF90 to 0xFF93	Définie dans la présente Recommandation Norme internationale
0xFF30 to 0xFF3F	Réservée pour définition comme marqueurs seulement (pas de segments marqueurs)
	Toutes les autres valeurs sont réservées

A.1.3 Règles applicables aux marqueurs et segments marqueurs ainsi qu'au flux codé

- Les segments marqueurs, et donc les en-têtes principaux et d'élément de pavé, sont un multiple de 8 bits (un octet). Par ailleurs, les données de flux binaire situées entre les en-têtes et avant le marqueur de fin EOC (voir § A.4.4) sont justifiées afin d'être également alignées sur un multiple de 8 bits.
- Tous les segments marqueurs contenus dans un en-tête d'élément de pavé s'appliquent seulement au pavé auquel ils appartiennent.
- Tous les segments marqueurs contenus dans l'en-tête principal s'appliquent à l'image entière sauf prise de priorité spécifique par marqueurs ou segments marqueurs dans un en-tête d'élément de pavé.
- Les marqueurs et segments marqueurs délimitants et d'information fixe doivent toujours apparaître à des points spécifiques dans le flux codé.
- Les segments marqueurs doivent correctement décrire l'image telle que représentée par le flux codé. Si une troncature, altération, ou édition du flux codé a été effectuée, les segments marqueurs doivent être mis à jour, si nécessaire.
- Toutes les valeurs paramétriques contenues dans les segments marqueurs sont gros-boutistes.
- Les segments marqueurs peuvent apparaître dans un ordre quelconque dans un en-tête donné. Exceptions: les marqueurs et segments marqueurs délimitants et les segments marqueurs d'information fixe.
- Tous les marqueurs dont le code est compris entre 0xFF30 et 0xFF3F n'ont pas de paramètres de segment marqueur. Ils doivent être omis par le décodeur.
- Certains segments marqueurs ont des valeurs attribuées à des groupes de bits dans un paramètre. Dans certains cas il y a des bits, indiqués par "x," auxquels une valeur n'est pas attribuée pour un champ quelconque dans un paramètre. Le flux codé doit contenir une valeur de zéro pour tous les bits de ce type. Le décodeur doit ignorer ces bits.

NOTE – Les marqueurs dans l'étendue 0xFF30 à 0xFF3F pourront être utilisés par de futures extensions. Ils peuvent être ou ne pas être omis par un décodeur sans ramification.

A.1.4 Légende des descriptions graphiques (pour information)

Chaque segment marqueur est décrit en termes de sa fonction, de son usage et de sa longueur. La fonction décrit les informations contenues dans le segment marqueur. L'usage décrit l'emplacement logique et la fréquence de ce segment marqueur dans le flux codé. La longueur décrit les paramètres qui déterminent la longueur du segment marqueur.

Ces descriptions sont suivies par une figure qui montre l'ordre et la relation des paramètres dans le segment marqueur. La Figure A.1 montre un exemple de ce type de figure. Les segments marqueurs sont désignés par le code à trois lettres du marqueur associé au segment marqueur. Les symboles paramétriques ont des désignations par lettre majuscule suivies du symbole de marqueur en lettres minuscules. Un rectangle est utilisé pour indiquer un emplacement de paramètre dans le segment marqueur. La largeur du rectangle est proportionnelle au nombre d'octets du paramètre. Un rectangle ombré (zones diagonales) indique que le paramètre est de longueur variable. Deux paramètres avec indices supérieurs et une zone grise intermédiaire indiquent une plage de plusieurs de ces paramètres.

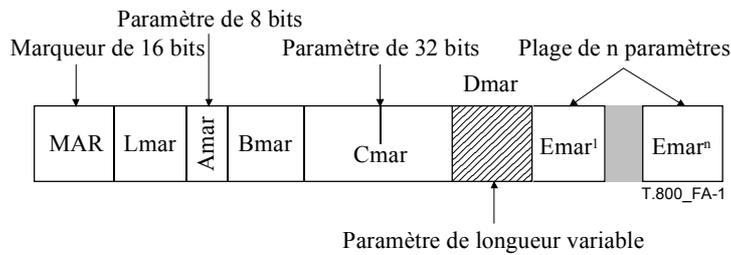


Figure A.1 – Exemple de figures descriptives du segment marqueur

La figure est suivie d'une liste qui décrit la signification de chaque paramètre dans le segment marqueur. Si des paramètres sont répétés, la longueur et la nature de la plage des paramètres sont définies. Par exemple, dans la Figure A.1, le premier rectangle représente le marqueur avec le symbole MAR. Le second rectangle représente le paramètre de longueur. Les paramètres Amar, Bmar, Cmar, et Dmar sont respectivement de longueur égale à 8, 16, 32 bits et de longueur variable. La notation Emar_i implique qu'il y a n différents paramètres, Emar_i, dans une rangée.

La liste est suivie d'un tableau qui soit décrit les valeurs paramétriques permises ou fournit des références aux autres tableaux qui décrivent ces valeurs. Les tableaux pour paramètres individuels sont présentés de façon à décrire tout paramètre sans simple valeur numérique. Dans certains cas, ces paramètres sont décrits par une valeur binaire dans un champ binaire. Dans ce cas, une lettre "x" sert à indiquer les bits qui ne sont pas inclus dans la spécification du paramètre ou sous-paramètre dans la rangée correspondante du tableau.

Certains paramètres de segment marqueur sont décrits au moyen des notations "Sxxx" et "SPxxx" (pour un symbole de marqueur, XXX). Le paramètre Sxxx choisit entre de nombreux états possibles du paramètre SPxxx. Selon cette sélection, on modifie le paramètre SPxxx ou la liste de paramètres.

A.2 Informations contenues dans les segments marqueurs

Le Tableau A.2 énumère les marqueurs spécifiés dans la présente Recommandation | Norme internationale. Le Tableau A.3 montre une liste des marqueurs et segments marqueurs ainsi que les informations qu'ils fournissent.

Tableau A.2 – Liste des marqueurs et segments marqueurs

	Symbole	Code	En-tête principal	En-tête d'élément de pavé
Marqueurs et segments marqueurs délimitants				
Début de flux codé	SOC	0xFF4F	Requis ^{a)}	Interdit
Début d'élément de pavé	SOT	0xFF90	Interdit	Requis
Début de données	SOD	0xFF93	Interdit	Dernier marqueur
Fin de flux codé	EOC	0xFFD9	Interdit	Interdit
Segments marqueurs d'information fixe				
Dimensions d'image et de pavé	SIZ	0xFF51	Requis	Interdit
Segments marqueurs fonctionnels				
Style de codage par défaut	COD	0xFF52	Requis	Facultatif
Composante de style de codage	COC	0xFF53	Facultatif	Facultatif
Région intéressante	RGN	0xFF5E	Facultatif	Facultatif
Quantification par défaut	QCD	0xFF5C	Requis	Facultatif
Composante de quantification	QCC	0xFF5D	Facultatif	Facultatif
Changement de l'ordre de progression ^{b)}	POC	0xFF5F	Facultatif	Facultatif
Segments marqueurs de pointage				
Longueurs d'élément de pavé	PLM	0xFF57	Facultatif	Interdit
En-tête principal de longueur de paquet	PLT	0xFF58	Facultatif	Interdit
En-tête d'élément de pavé de longueur de paquet	PPM	0xFF60	Interdit	Facultatif
En-tête principal de paquet compact ^{c)}	PPT	0xFF61	Facultatif	Interdit
En-tête d'élément de pavé de paquet compact ^{c)}	TLM	0xFF55	Interdit	Facultatif

Tableau A.2 – Liste des marqueurs et segments marqueurs

	Symbole	Code	En-tête principal	En-tête d'élément de pavé
Marqueurs et segments marqueurs dans le flux binaire				
Début de paquet	SOP	0xFF91	Interdit	Interdit dans un en-tête d'élément de pavé, facultatif s'ils sont dans le flux binaire
Fin d'en-tête de paquet	EPH	0xFF92	Facultatif dans un segment marqueur PPM	Facultatif dans un segment marqueur PPT ou dans le flux binaire
Segments marqueurs informationnels				
Enregistrement de composante	CRG	0xFF63	Facultatif	Interdit
Commentaire	COM	0xFF64	Facultatif	Facultatif
<p>a) "Requis" signifie que le marqueur ou segment marqueur doit se trouver dans l'en-tête considéré; "facultatif" signifie qu'il peut être utilisé.</p> <p>b) Le segment marqueur POC est requis s'il y a changement de l'ordre de progression.</p> <p>c) Le segment marqueur PPM ou PPT est requis si les en-têtes de paquet ne sont pas répartis dans le flux binaire. Si le segment marqueur PPM est utilisé, les segments marqueurs PPT ne doivent pas être utilisés, et vice versa.</p>				

Tableau A.3 – Informations contenues dans les segments marqueurs

Informations	Segment marqueur
Capacités Dimensions de zone d'image ou de grille de référence (hauteur et largeur) Dimensions de pavé (hauteur et largeur) Nombre de composantes Précision de composante Mappage des composantes à la grille de référence (sous-échantillonnage)	SIZ
Index de pavé Longueur des données d'élément de pavé	SOT, TLM
Ordre de progression Nombre de couches Transformation multiple de composante utilisée	COD
Style de codage Nombre de niveaux de décomposition Longueur de bloc codé Style de bloc codé Transformation en ondelettes Dimensions de district	COD, COC
Décalage de région intéressante	RGN
Sans quantification Quantification calculée Quantification expansée	QCD, QCC
Point de départ de la progression Point de fin de progression Ordre de progression par défaut	POC
Elasticité aux erreurs	SOP
Fin d'en-tête de paquet	EPH

Tableau A.3 – Informations contenues dans les segments marqueurs

Informations	Segment marqueur
En-tête de paquet	PPM, PPT
Longueur de paquet	PLM, PLT
Enregistrement de composante	CRG
Informations facultatives	COM

A.3 Construction du flux codé

La Figure A.2 montre la construction du flux codé. La Figure A.3 montre la construction de l'en-tête principal. Tous les traits pleins montrent les segments marqueurs requis. Les marqueurs et segments marqueurs suivants doivent se trouver à un emplacement spécifique: SOC, SIZ, SOT, SOD, et EOC. Les traits interrompus montrent les segments marqueurs facultatifs ou éventuellement non requis.

La Figure A.4 montre la construction du premier en-tête d'élément dans un pavé donné. La Figure A.5 montre la construction dans un pavé d'un en-tête d'élément de pavé autre que le premier.

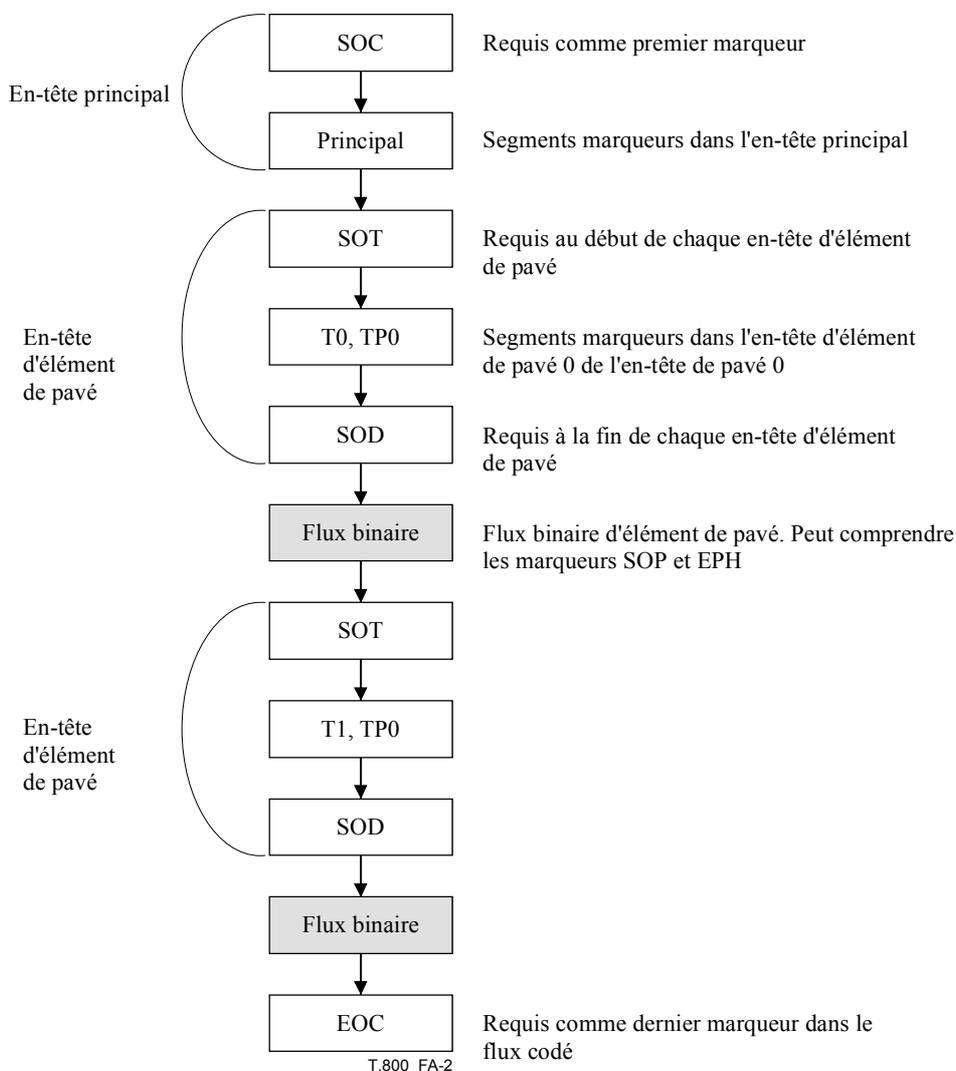


Figure A.2 – Construction du flux codé

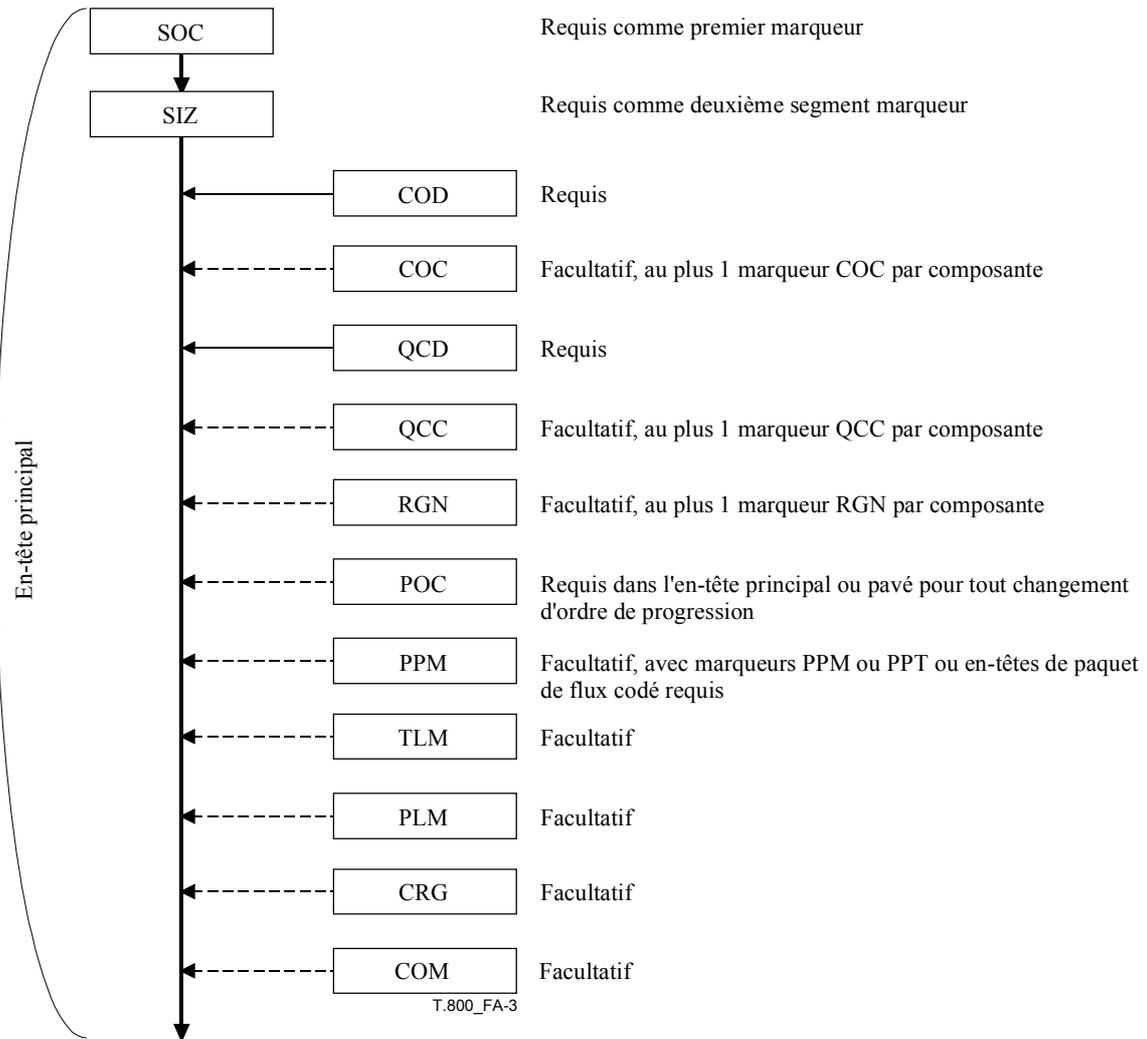


Figure A.3 – Construction de l'en-tête principal

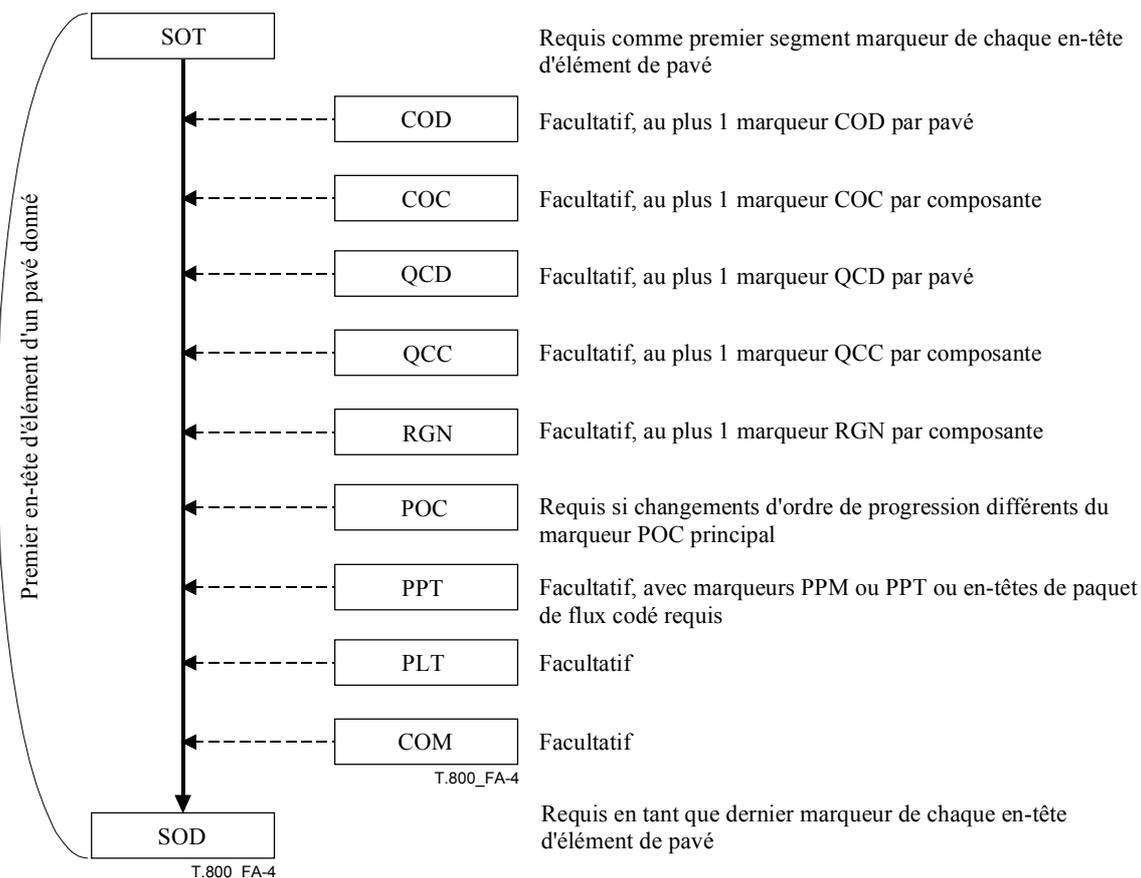


Figure A.4 – Construction du premier en-tête d'élément d'un pavé donné

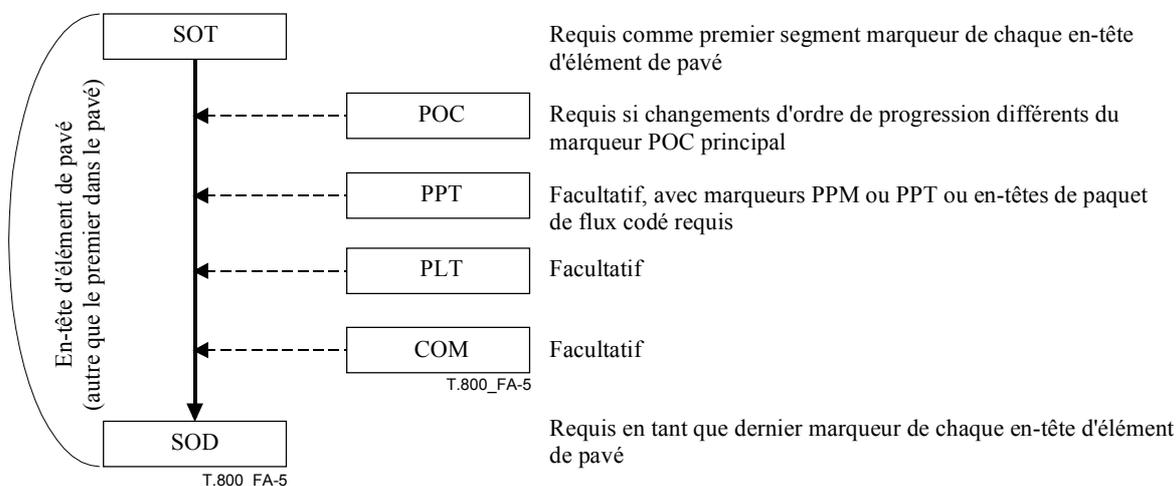


Figure A.5 – Construction d'un autre en-tête d'élément de pavé que le premier

Les segments marqueurs COD et COC et les segments marqueurs QCD et QCC ont une hiérarchie d'usage qui est conçue afin de permettre à des composantes-pavés d'avoir des caractéristiques de codage et de quantification dissemblables avec un minimum de signalisation.

Par exemple, le segment marqueur COD est requis dans l'en-tête principal. Si toutes les composantes dans tous les pavés sont codées de la même façon, c'est tout ce qui est requis. S'il y a une composante qui est codée différemment des autres (par exemple, la composante de luminance d'une image constituée de composantes de luminance et de chrominance), alors le marqueur COC peut indiquer cela dans l'en-tête principal. Si une ou plusieurs composantes sont codées différemment dans des pavés différents, alors les marqueurs COD et COC sont utilisés d'une manière similaire afin d'indiquer cela dans l'en-tête d'élément de pavé.

Le segment marqueur POC apparaissant dans l'en-tête principal est utilisé pour tous les pavés sauf si un marqueur POC différent apparaît dans l'en-tête d'élément de pavé.

A l'exception des marqueurs et segments marqueurs SOC, SOT, SOD, EOC et SIZ, les segments marqueurs peuvent apparaître dans un ordre quelconque à l'intérieur des en-têtes respectifs.

A.4 Marqueurs et segments marqueurs délimitants

Les marqueurs et segments marqueurs délimitants doivent être présents dans tous les flux codés conformes à la présente Recommandation | Norme internationale. Chaque flux codé possède un seul marqueur SOC, un marqueur de fin EOC et au moins un élément de pavé. Chaque élément de pavé possède un marqueur SOT et un marqueur SOD. Les marqueurs SOC, SOD et EOC sont délimitants et ne sont pas des segments marqueurs. Ils ne comportent pas d'informations explicites de longueur ni d'autres paramètres.

A.4.1 Début de flux codé (SOC)

Fonction: marque le début d'un flux codé spécifié dans la présente Recommandation | Norme internationale.

Usage: en-tête principal. C'est le premier marqueur contenu dans le flux codé. Il ne doit y avoir qu'un seul SOC par flux codé.

Longueur: fixe.

SOC: code de marqueur.

Tableau A.4 – Valeurs paramétriques de début de flux codé

Paramètre	Longueur (bits)	Valeurs
SOC	16	0xFF4F

A.4.2 Début d'élément de pavé (SOT)

Fonction: marque le début d'un élément de pavé, l'index de son pavé et l'index de son élément de pavé. Les éléments de pavé d'un pavé donné doivent apparaître en ordre (voir TP_{sot}) dans le flux codé. Cependant, les éléments issus d'autres pavés peuvent être entrelacés dans le flux codé. Donc, les éléments de pavé issus d'un pavé donné peuvent ne pas apparaître en séquence contiguë dans le flux codé.

Usage: chaque en-tête d'élément de pavé. Doit être le premier segment marqueur dans un en-tête d'élément de pavé. Il doit y avoir au moins un marqueur SOT dans un flux codé. Il ne doit y avoir qu'un seul SOT par élément de pavé.

Longueur: fixe.

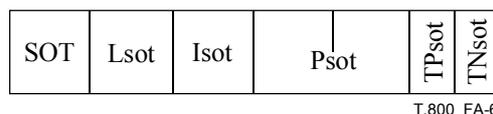


Figure A.6 – Syntaxe de début d'élément de pavé

SOT: code de marqueur. Le Tableau A.5 montre les longueurs et valeurs du symbole et les paramètres pour le segment marqueur de début d'élément de pavé.

L_{sot}: longueur de segment marqueur en octets (non compris le marqueur).

I_{sot}: index de pavé. Ce nombre se rapporte aux pavés dans l'ordre du balayage à partir du nombre 0.

P_{sot}: longueur, en octets, à partir du début du premier octet de ce segment marqueur SOT de l'élément de pavé jusqu'à la fin des données de cet élément de pavé. La Figure A.16 montre cet alignement. Seul le dernier élément de pavé contenu dans le flux codé peut contenir un 0 pour P_{sot}. Si la longueur P_{sot} est 0, cet élément de pavé est censé contenir toutes les données jusqu'au marqueur de fin EOC.

TP_{sot}: index d'élément de pavé. Il y a un ordre spécifique requis pour le décodage des éléments de pavé; cet index indique l'ordre à partir de 0. S'il y a un seul élément de pavé pour un pavé, alors cette valeur est zéro. Les éléments de ce pavé doivent apparaître dans le flux codé dans cet ordre, bien que pas forcément en séquence contiguë.

TNsot: nombre d'éléments d'un pavé contenus dans le flux codé. Deux valeurs sont permises: le nombre correct d'éléments de pavé pour ce pavé et zéro. Une valeur zéro indique que le nombre d'éléments de ce pavé n'est pas spécifié dans cet élément de pavé.

Tableau A.5 – Valeurs paramétriques de début d'élément de pavé

Paramètre	Longueur (bits)	Valeurs
SOT	16	0xFF90
Lsot	16	10
Isot	16	0 à 65 534
Psot	32	0, ou 14 à $(2^{32} - 1)$
TPsot	8	0 à 254
TNsot	8	Tableau A.6

Tableau A.6 – Valeur du paramètre Nombre d'éléments de pavé, TNsot

Valeur	Nombre d'éléments de pavé
0	Le nombre d'éléments de ce pavé dans le flux codé n'est pas défini dans cet en-tête
1 à 255	Nombre d'éléments de ce pavé contenus dans le flux codé

A.4.3 Début de données (SOD)

Fonction: indique le début des données de flux binaire pour l'élément de pavé actuel. Le marqueur SOD indique également la fin d'un en-tête d'élément de pavé.

Usage: chaque en-tête d'élément de pavé. Doit être le dernier marqueur dans un en-tête d'élément de pavé. Les données de flux binaire entre un marqueur SOD et le prochain SOT ou EOC (fin d'image) doivent être un multiple de 8 bits – le flux codé est justifié avec des bits, selon les besoins. Il doit y avoir au moins un SOD dans un flux codé. Il doit y avoir un SOD par élément de pavé.

Longueur: fixe.

SOD: code de marqueur

Tableau A.7 – Valeurs paramétriques de début de données

Paramètre	Longueur (bits)	Valeurs
SOD	16	0xFF93

A.4.4 Fin de flux codé (EOC)

Fonction: indique la fin du flux codé.

NOTE 1 – Ce marqueur partage le même code que le marqueur de fin EOI de la Rec. UIT-T T.81 | ISO/CEI 10918-1.

Usage: doit être le dernier marqueur dans un flux codé. Il doit y avoir un EOC par flux codé.

NOTE 2 – Si un fichier a été corrompu, il est possible qu'un décodeur puisse extraire beaucoup de données d'image comprimées utiles sans rencontrer de marqueur de fin EOC.

Longueur: fixe.

EOC: code de marqueur.

Tableau A.8 – Valeurs paramétriques de fin de flux codé

Paramètre	Longueur (bits)	Valeur
EOC	16	0xFFD9

A.5 Segment marqueur d'information fixe

Ce segment marqueur décrit des informations requises concernant l'image. Le segment marqueur SIZ est requis dans l'en-tête principal immédiatement après le segment marqueur SOC.

A.5.1 Dimensions d'image et de pavé (SIZ)

Fonction: fournit des informations sur l'image non comprimée comme la largeur et la hauteur de la grille de référence, la largeur et la hauteur des pavés, le nombre de composantes, la profondeur binaire des composantes et la séparation des échantillons de composante par rapport à la grille de référence (voir § B.2).

Usage: en-tête principal. Il doit y avoir qu'un et un seul marqueur SIZ dans l'en-tête principal, immédiatement après le segment marqueur SOC. Il ne doit y avoir qu'un seul marqueur SIZ par flux codé.

Longueur: variable dépendant du nombre de composantes.

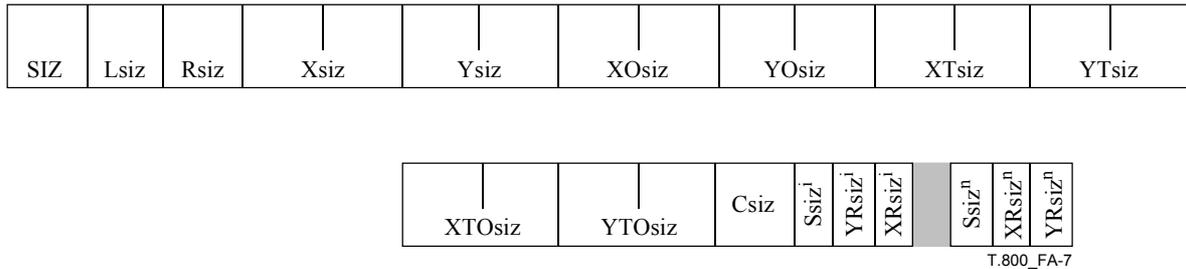


Figure A.7 – Syntaxe dimensionnelle d'image et de pavé

SIZ: code de marqueur. Le Tableau A.9 montre la longueur et les valeurs paramétriques du symbole et les paramètres pour le segment marqueur de dimensions d'image et de pavé.

Lsiz: longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lsiz = 38 + 3 \cdot Csiz \quad (\text{A-1})$$

Rsiz: indique les capacités dont un décodeur a besoin afin de décoder correctement le flux codé.

Xsiz: largeur de la grille de référence.

Ysiz: hauteur de la grille de référence.

XOsiz: décalage horizontal par rapport à l'origine de la grille de référence vers le côté gauche de la zone d'image.

YOsiz: décalage vertical par rapport à l'origine de la grille de référence jusqu'au côté supérieur de la zone d'image.

XTsiz: largeur d'un pavé de référence par rapport à la grille de référence.

YTtiz: hauteur d'un pavé de référence par rapport à la grille de référence.

XTosiz: décalage horizontal par rapport à l'origine de la grille de référence vers le côté gauche du premier pavé.

YTosiz: décalage vertical par rapport à l'origine de la grille de référence jusqu'au côté supérieur du premier pavé.

Csiz: nombre de composantes dans l'image.

Ssizⁱ: précision (profondeur) en bits et signe des i^{e} échantillons de composante. La précision est celle des échantillons de composante avant que le décalage au niveau aperiodique soit effectué (c'est-à-dire la précision des échantillons de la composante originale avant qu'un traitement quelconque soit effectué). Si les valeurs d'échantillon de composante sont signées, alors l'étendue de valeurs d'échantillon de composante est $-2^{(Ssiz+1 \text{ et } 0x7F)-1} \leq \text{valeur d'échantillon de composante} \leq 2^{(Ssiz+1 \text{ et } 0x7F)-1} - 1$. Il y a une seule occurrence de ce paramètre pour chaque composante. L'ordre correspond à l'index de composante, à partir de zéro.

XRsizⁱ: séparation horizontale d'un échantillon de i^{e} composante par rapport à la grille de référence. Il y a une seule occurrence de ce paramètre pour chaque composante.

YRsizⁱ: séparation verticale d'un échantillon de i° composante par rapport à la grille de référence. Il y a une seule occurrence de ce paramètre pour chaque composante.

Tableau A.9 – Valeurs paramétriques de dimensions d'image et de pavé

Paramètre	Longueur (bits)	Valeurs
SIZ	16	0xFF51
Lsiz	16	41 à 49 190
Rsiz	16	Tableau A.10
Xsiz	32	1 à $(2^{32} - 1)$
Ysiz	32	1 à $(2^{32} - 1)$
XOsiz	32	0 à $(2^{32} - 2)$
YOsiz	32	0 à $(2^{32} - 2)$
XTsiz	32	1 à $(2^{32} - 1)$
YTsiz	32	1 à $(2^{32} - 1)$
XTOsiz	32	0 à $(2^{32} - 2)$
YTOsiz	32	0 à $(2^{32} - 2)$
Csiz	16	1 à 16 384
Ssiz ⁱ	8	Tableau A.11
XRsiz ⁱ	8	1 à 255
YRsiz ⁱ	8	1 à 255

Tableau A.10 – Paramètre Rsiz de capacité

Valeur (bits)		Capacité
MSB	LSB	
0000	0000 0000 0000	Capacités spécifiées dans la présente Recommandation Norme internationale seulement
0000	0000 0000 0001	Flux codé limité comme décrit pour profil 0 d'après le Tableau A.45
0000	0000 0000 0010	Flux codé limité comme décrit pour profil 1 d'après le Tableau A.45
		Toutes les autres valeurs sont réservées

Tableau A.11 – Paramètre Ssiz de composante

Valeur (bits)		Précision d'échantillon de composante
MSB	LSB	
x000	0000 à x010 0101	Profondeur binaire de l'échantillon de composante = valeur + 1. D'une profondeur de 1 bit jusqu'à une profondeur de 38 bits respectivement (en comptant le bit de signe, si approprié) ^{a)} , R_1
0xxx	xxxx	Les valeurs d'échantillon de composante sont des valeurs non signées
1xxx	xxxx	Les valeurs d'échantillon de composante sont des valeurs signées
		Toutes les autres valeurs sont réservées
^{a)} La précision d'échantillon de composante est limitée par le nombre de bits de garde, par la quantification, par la croissance des coefficients à chaque niveau de décomposition et par le nombre de passes de codage qui peuvent être signalées. Toutes les combinaisons de styles de codage ne permettent pas le codage d'échantillons de 38 bits.		

A.6 Segments marqueurs fonctionnels

Ces segments marqueurs décrivent les fonctions utilisées pour coder le pavé entier, s'ils sont dans l'en-tête d'élément de pavé, ou pour coder l'image, s'ils sont dans l'en-tête principal.

A.6.1 Style de codage par défaut (COD)

Fonction: décrit le style de codage, le nombre de niveaux de décomposition et le groupement en couches qui est la méthode utilisée par défaut pour comprimer toutes les composantes d'une image (si elles sont dans l'en-tête principal) ou d'un pavé (si elles sont dans l'en-tête d'élément de pavé). Les valeurs paramétriques peuvent être outrepassées pour une composante individuelle par un seul segment marqueur COC se trouvant soit dans l'en-tête principal soit dans l'en-tête d'élément de pavé.

Usage: en-tête principal et premier en-tête d'élément d'un pavé donné. Doit être le seul marqueur dans l'en-tête principal. En outre, il peut y avoir au plus un marqueur pour chaque pavé. S'il y a de multiples éléments dans un pavé, et que ce segment marqueur soit présent, celui-ci ne doit se trouver que dans le premier élément de pavé (TPsot = 0).

Lors d'une utilisation dans l'en-tête principal, les valeurs paramétriques du segment marqueur COD sont utilisées pour toutes les composantes-pavés qui n'ont pas de segment marqueur COC correspondant dans l'en-tête principal ou dans l'en-tête d'élément de pavé. Lorsqu'il est utilisé dans l'en-tête d'élément de pavé, ce marqueur outrepassé les marqueurs COD et COC d'en-tête principal et est utilisé pour toutes les composantes contenues dans ce pavé sans segment marqueur COC correspondant dans l'élément de pavé. Donc, l'ordre de préséance est le suivant:

COC d'élément de pavé > COD d'élément de pavé > COC principal > COD principal

où le signe "supérieur à", >, signifie que le segment marqueur supérieur outrepassé l'inférieur.

Longueur: variable dépendant de la valeur de Scod.



Figure A.8 – Syntaxe par défaut du style de codage

COD: code de marqueur. Le Tableau A.12 montre la longueur et les valeurs du symbole et des paramètres pour le segment marqueur de style de codage par défaut.

Lcod: longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lcod = \begin{cases} 12 & \text{maximum_precincts} \\ 13 + \text{number_decomposition_levels} & \text{user-defined_precincts} \end{cases} \quad (\text{A-2})$$

où maximum_precincts et user-defined_precincts sont indiqués dans le paramètre Scod et où number_decomposition_levels sont indiqués dans le paramètre SPcod.

Scod: style de codage pour toutes les composantes. Le Tableau A.13 montre la valeur pour le paramètre Scod.

SGcod: paramètres du style de codage désigné dans le marqueur Scod. Ces paramètres sont indépendants des composantes et sont désignés, de haut en bas, dans le Tableau A.14. Le paramètre style de codage dans le champ SGcod apparaît dans la séquence représentée dans la Figure A.9.

SPcod: paramètres du style de codage désigné dans le marqueur Scod. Ces paramètres se rapportent à toutes les composantes et sont désignés, de haut en bas, dans le Tableau A.15. Le paramètre du style de codage dans le champ SPcod apparaît dans la séquence représentée dans la Figure A.9.

Tableau A.12 – Valeurs paramétriques de style de codage par défaut

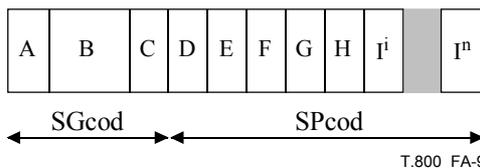
Paramètre	Longueur (bits)	Valeurs
COD	16	0xFF52
Lcod	16	12 à 45
Scod	8	Tableau A.13
SGcod	32	Tableau A.14
SPcod	Variable	Tableau A.15

Tableau A.13 – Valeurs paramétriques de style de codage pour le paramètre Scod

Valeurs (bits)		Style de codage
MSB	LSB	
xxxx	xxx0	Codeur entropique, districts avec PPx = 15 et PPy = 15
xxxx	xxx1	Codeur entropique avec districts définis ci-dessous
xxxx	xx0x	Aucun seul segment marqueur SOP n'est utilisé
xxxx	xx1x	Des segments marqueurs SOP peuvent être utilisés
xxxx	x0xx	Aucun marqueur EPH n'est utilisé
xxxx	x1xx	Un marqueur EPH doit être utilisé
		Toutes les autres valeurs sont réservées

Tableau A.14 – Valeurs paramétriques de style de codage du paramètre SGcod

Paramètres (dans l'ordre)	Longueur (bits)	Valeurs	Signification des valeurs SGcod
Ordre de progression	8	Tableau A.16	Ordre de progression
Nombre de couches	16	1 à 65535	Nombre de couches
Transformation multiple de composante	8	Tableau A.17	Usage de transformation multiple de composante



- A Ordre de progression
- B Nombre de couches
- C Transformation multiple de composante
- D Nombre de niveaux de décomposition
- E Largeur de bloc codé
- F Hauteur de bloc codé
- G Style de bloc codé
- H Transformation
- I¹ à Iⁿ Dimensions de district

Figure A.9 – Schéma du paramètre style de codage des paramètres SGcod et SPcod

Tableau A.15 – Valeurs paramétriques de style de codage des paramètres SPcod et SPcoc

Paramètres (dans l'ordre)	Longueur (bits)	Valeurs	Signification des valeurs SPcod
Nombre de niveaux de décomposition	8	0 à 32	Nombre de niveaux de décomposition, N_L , zéro implique l'absence de transformation.
Largeur de bloc codé	8	Tableau A.18	Exposant de largeur de bloc codé valeur du décalage, xcb
Hauteur de bloc codé	8	Tableau A.18	Exposant de hauteur de bloc codé valeur du décalage, ycb
Style de bloc codé	8	Tableau A.19	Style des passes de codage de bloc codé
Transformation	8	Tableau A.20	Transformation en ondelettes utilisée
Dimensions de district	Variable	Tableau A.21	Si Scod ou Scoc = xxxx xxx0, ce paramètre n'est pas présent; sinon, ce paramètre indique la largeur et la hauteur de district. Le premier paramètre (8 bits) correspond à la sous-bande N_{LL} . Chaque paramètre successif correspond à chaque niveau successif de résolution, dans l'ordre.

Tableau A.16 – Ordre de progression pour les paramètres SGcod, SPcoc et Ppoc

Valeurs (bits)		Ordre de progression
MSB	LSB	
0000	0000	Progression couche-niveau de résolution-composante-position
0000	0001	Progression niveau de résolution-couche-composante-position
0000	0010	Progression niveau de résolution-position-composante-couche
0000	0011	Progression position-composante-niveau de résolution-couche
0000	0100	Progression composante-position-niveau de résolution-couche
		Toutes les autres valeurs sont réservées

Tableau A.17 – Transformation multiple de composante pour les paramètres SGcod

Valeurs (bits)		Type de transformation multiple de composante
MSB	LSB	
0000	0000	Aucune transformation multiple de composante spécifiée
0000	0001	Transformation de composante utilisée sur les composantes 0, 1, 2 pour efficacité du codage (voir § G.2). Transformation de composante irréversible utilisée avec le filtre irréversible 9-7. Transformation de composante réversible utilisée avec le filtre réversible 5-3.
		Toutes les autres valeurs sont réservées

Tableau A.18 – Exposant de largeur ou de hauteur des blocs codés pour les paramètres SPcod et SPcoc

Valeurs (bits)		Largeur et hauteur de bloc codé
MSB	LSB	
xxxx	0000 à xxxx 1000	Valeur du décalage d'exposant de largeur et hauteur de bloc codé $xcb = valeur + 2$ ou $ycb = valeur + 2$. Largeur et hauteur de bloc codé sont limitées à des puissances de deux avec la taille minimale étant 22 et la taille maximale étant 2^{10} . Par ailleurs, la longueur de bloc codé est limitée de façon que $xcb + ycb \leq 12$.
		Toutes les autres valeurs sont réservées

Tableau A.19 – Style de bloc codé pour les paramètres SPcod et SPcoc

Valeurs (bits)		Style de bloc codé
MSB	LSB	
xxxx xxx0 xxxx xxx1		Aucun évitement sélectif du codage arithmétique Evitement sélectif du codage arithmétique
xxxx xx0x xxxx xx1x		Aucune réinitialisation de probabilités contextuelles aux limites de passe de codage Réinitialisation des probabilités contextuelles aux limites de passe de codage
xxxx x0xx xxxx x1xx		Aucune terminaison à chaque passe de codage Terminaison à chaque passe de codage
xxxx 0xxx xxxx 1xxx		Aucun contexte verticalement en relation causale Contexte verticalement en relation causale
xxx0 xxxx xxx1 xxxx		Aucune terminaison prévisible Terminaison prévisible
xx0x xxxx xx1x xxxx		Aucun symbole de segmentation n'est utilisé Des symboles de segmentation sont utilisés
		Toutes les autres valeurs sont réservées

Tableau A.20 – Transformation pour les paramètres SPcod et SPcoc

Valeurs (bits)		Transformation type
MSB	LSB	
0000	0000	Filtre irréversible 9-7
0000	0001	Filtre réversible 5-3
		Toutes les autres valeurs sont réservées

Tableau A.21 – Largeur et hauteur de district pour les paramètres SPcod et SPcoc

Valeurs (bits)		Dimensions de district
MSB	LSB	
xxxx 0000 à xxxx 1111		4 bits LSB sont l'exposant de largeur de district, $PP_x = valeur$. Cette valeur ne peut être égale qu'à zéro au niveau de résolution correspondant à la bande N_{LL} .
0000 xxxx à 1111 xxxx		4 bits MSB sont la hauteur de district $PP_y = valeur$. Cette valeur ne peut être égale qu'à zéro au niveau de résolution correspondant à la bande N_{LL} .

A.6.2 Composante de style de codage (COC)

Fonction: décrit le style de codage et le nombre de niveaux de décomposition utilisés pour comprimer une composante particulière.

Usage: en-tête principal et premier en-tête d'élément d'un pavé donné. Facultatif dans les deux en-têtes, principal et d'élément de pavé. Au plus un seul marqueur par composante donnée peut être présent dans soit l'en-tête principal ou l'en-tête d'élément de pavé. S'il y a de multiples éléments dans un pavé, et que ce segment marqueur soit présent, celui-ci ne doit être présent que dans le premier élément de pavé ($TP_{sot} = 0$).

Lorsqu'il est utilisé dans l'en-tête principal, il outrepassse le principal segment marqueur COD pour la composante spécifique. Lorsqu'il est utilisé dans l'en-tête d'élément de pavé, il outrepassse le COD principal, le COC principal, et le COD de pavé pour la composante spécifique. Donc, l'ordre de préséance est le suivant:

COC d'élément de pavé > COD d'élément de pavé > COC principal > COD principal

où le signe "supérieur à", >, signifie que le segment marqueur supérieur outrepassse l'inférieur.

Longueur: variable dépendant de la valeur de Scoc.

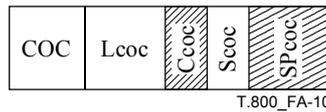


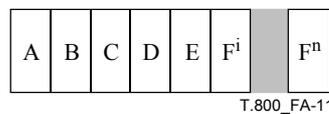
Figure A.10 – Syntaxe de composante de style de codage

- COC:** code de marqueur. Le Tableau A.22 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de composante de style de codage.
- Lcoc:** longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lcoc = \begin{cases} 9 & \text{maximum_precincts AND Csiz} < 257 \\ 10 & \text{maximum_precincts AND Csiz} \Rightarrow 257 \\ 10 + \text{number_decomposition_levels} & \text{user-defined_precincts AND Csiz} < 257 \\ 11 + \text{number_decomposition_levels} & \text{user-defined_precincts AND Csiz} \Rightarrow 257 \end{cases} \quad (\text{A-3})$$

où "maximum_precincts" et "user-defined_precincts" sont indiqués dans le paramètre Scoc et où "number_decomposition_levels" est indiqué dans le paramètre SPcoc.

- Ccoc:** index de la composante à laquelle ce segment marqueur se rapporte. Les composantes sont indexées 0, 1, 2, etc.
- Scoc:** style de codage pour cette composante. Le Tableau A.23 montre la valeur pour chaque paramètre Scoc.
- SPcoc:** paramètres du style de codage désignés dans Scoc. Les paramètres sont désignés, de haut en bas, dans le Tableau A.15. Le paramètre style de codage dans le champ SPcoc apparaît dans la séquence représentée dans la Figure A.11.



- A Nombre de niveaux de décomposition
 B Largeur de bloc codé
 C Hauteur de bloc codé
 D Style de bloc codé
 E Transformation
 Fⁱ à Fⁿ Dimensions du district

Figure A.11 – Schéma du paramètre style de codage des paramètres SPcoc

Tableau A.22 – Valeurs paramétriques de composante de style de codage

Paramètre	Longueur (bits)	Valeurs
COC	16	0xFF53
Lcoc	16	9 à 43
Ccoc	8 16	0 à 255; si Csiz < 257 0 à 16 383; Csiz ≥ 257
Scoc	8	Tableau A.23
SPcoc ⁱ	Variable	Tableau A.15

Tableau A.23 – Valeurs paramétriques de style de codage pour le paramètre Scoc

Valeurs (bits)		Style de codage
MSB	LSB	
0000	0000	Codeur entropique avec valeurs maximales de district $PPx = PPy = 15$
0000	0001	Codeur entropique avec valeurs de district définies ci-dessous
		Toutes les autres valeurs sont réservées

A.6.3 Région intéressante (RGN)

Fonction: signale la présence d'une région ROI dans le flux codé.

Usage: en-tête principal et premier en-tête d'élément d'un pavé donné. Si utilisé dans l'en-tête principal, il se rapporte à la valeur de cadrage de région ROI pour une composante de l'image entière, valide pour tous les pavés sauf ceux ayant un seul segment marqueur RGN.

Lorsque ce marqueur est utilisé dans l'en-tête d'élément de pavé, la valeur de cadrage est valide seulement pour une composante contenue dans ce pavé. Il peut y avoir au plus un seul segment marqueur RGN pour chaque composante dans soit l'en-tête principal ou l'en-tête d'élément de pavé. Le segment marqueur RGN pour une composante particulière qui apparaît dans un en-tête d'élément de pavé outrepassé tout marqueur pour cette composante dans l'en-tête principal, pour le pavé dans lequel il apparaît. S'il y a de multiples éléments dans un pavé, alors ce segment marqueur ne doit se trouver que dans le premier en-tête d'élément de pavé.

Longueur: variable.

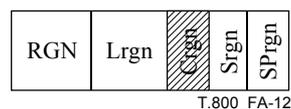


Figure A.12 – Syntaxe de région intéressante

- RGN:** code de marqueur. Le Tableau A.24 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de région intéressante.
- Lrgn:** longueur de segment marqueur en octets (non compris le marqueur).
- Crgn:** index de la composante auquel ce segment marqueur se rapporte. Les composantes sont indexées 0, 1, 2, etc.
- Srgn:** style de région ROI pour la région ROI actuelle. Le Tableau A.25 montre la valeur pour le paramètre Srgn.
- SPrgn:** paramètre du style de région ROI désigné dans Srgn.

Tableau A.24 – Valeurs paramétriques de région intéressante

Paramètre	Longueur (bits)	Valeurs
RGN	16	0xFF5E
Lrgn	16	5 à 6
Crgn	8 16	0 à 255; si $Csiz < 257$ 0 à 16 383; $Csiz \geq 257$
Srgn	8	Tableau A.25
SPrgn	8	Tableau A.26

Tableau A.25 – Valeurs paramétriques de région intéressante pour le paramètre Srgn

Valeurs	Style de région ROI (Srgn)
0	Région ROI implicite (décalage maximal)
	Toutes les autres valeurs sont réservées

Tableau A.26 – Valeurs de région intéressante extraites du paramètre SPrgn (Srgn = 0)

Paramètres (dans l'ordre)	Longueur (bits)	Valeurs	Signification de SPrgn valeur
Décalage de région ROI implicite	8	0 à 255	Décalage binaire de coefficients d'image ROI au-dessus de l'arrière-plan

A.6.4 Quantification par défaut (QCD)

Fonction: décrit la quantification par défaut utilisée pour comprimer toutes les composantes non définies par un segment marqueur QCC. Les valeurs paramétriques peuvent être outrepassées pour une composante individuelle par un segment marqueur QCC dans soit l'en-tête principal ou l'en-tête d'élément de pavé.

Usage: en-tête principal et premier en-tête d'élément d'un pavé donné. Doit être un et un seul dans l'en-tête principal. Peut être au plus un pour tous les en-têtes d'élément d'un pavé. S'il y a de multiples éléments de pavé pour un pavé, et que ce segment marqueur soit présent, celui-ci ne doit être présent que dans le premier élément de pavé (TPsot = 0).

Lorsqu'il est utilisé dans l'en-tête d'élément de pavé, il outrepassé le marqueur QCD principal et le marqueur QCC principal pour la composante spécifique. Donc, l'ordre de préséance est le suivant:

Élément de pavé QCC > Élément de marqueur QCD de pavé > Marqueur QCC principal > Marqueur QCD principal

où le signe "supérieur à", >, signifie que le segment marqueur supérieur outrepassé l'inférieur.

Longueur: variable dépendant du nombre d'éléments quantifiés.

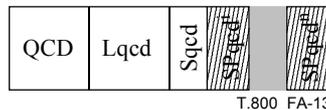


Figure A.13 – Syntaxe de quantification par défaut

QCD: code de marqueur. Le Tableau A.27 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de quantification par défaut.

Lqcd: longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lqcd = \begin{cases} 4 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization} \\ 5 & \text{scalar_quantization_derived} \\ 5 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded} \end{cases} \quad (\text{A-4})$$

où "number_decomposition_levels" est défini dans les segments marqueurs COD et COC, et où "no_quantization, scalar_quantization_derived", ou "scalar_quantization_expounded" est signalé dans le paramètre Sqcd.

NOTE – Le paramètre Lqcd peut être utilisé afin de déterminer combien de grandeurs d'échelon de quantification sont présentes dans le segment marqueur. Cependant, il n'y a pas nécessairement de correspondance avec le nombre de sous-bandes présentes parce que les sous-bandes peuvent être tronquées sans qu'il soit prescrit de corriger ce segment marqueur.

Sqcd: style de quantification pour toutes les composantes.

SPqcdi: valeur d'échelon de quantification pour la i^{e} sous-bande dans l'ordre défini (voir § F.3.1). Le nombre de paramètres est le même que le nombre de sous-bandes dans la composante-pavé ayant le plus grand nombre de niveaux de décomposition.

Tableau A.27 – Valeurs paramétriques de quantification par défaut

Paramètre	Longueur (bits)	Valeurs
QCD	16	0xFF5C
Lqcd	16	4 à 197
Sqcd	8	Tableau A.28
SPgcd ⁱ	Variable	Tableau A.28

Tableau A.28 – Valeurs de quantification par défaut pour les paramètres Sqcd et Sqcc

Valeurs (bits)		Style de quantification	Longueur de SPqcd ou SPqcc (bits)	Usage de SPqcd ou SPqcc
MSB	LSB			
xxx0	0000	Sans quantification	8	Tableau A.29
xxx0	0001	Calcul scalaire (valeurs signalées pour sous-bande N_{LL} seulement). Utiliser l'équation (E-5).	16	Tableau A.30
xxx0	0010	Expansion scalaire (valeurs signalées pour chaque sous-bande). Il y a autant d'échelons signalés qu'il y a de sous-bandes.	16	Tableau A.30
000x	xxxx à 111x xxxx	Nombre de bits de garde: 0 à 7		
		Toutes les autres valeurs sont réservées		

Tableau A.29 – Valeurs d'échelon réversibles pour les paramètres SPqcd et SPqcc (transformée réversible seulement)

Valeurs (bits)		Valeurs d'échelon réversibles
MSB	LSB	
0000	0xxx à 1111 1xxx	Exposant, ε_b , de l'étendue dynamique réversible signalée pour chaque sous-bande (voir équation (E-5))
		Toutes les autres valeurs sont réservées

Tableau A.30 – Valeurs de quantification pour les paramètres SPqcd et SPqcc (transformation irréversible seulement)

Valeurs (bits)				Valeurs d'échelon de quantification
MSB	LSB			
xxxx	x000	0000	0000 à xxxx x111 1111 1111	Mantisse, μ_b , de la valeur d'échelon de quantification (voir équation (E-3))
0000	0xxx	xxxx	xxxx à 1111 1xxx xxxx xxxx	Exposant, ε_b , de la valeur d'échelon de quantification (voir équation (E-3))

A.6.5 Composante de quantification (QCC)

Fonction: décrit la quantification utilisée pour comprimer une composante particulière.

Usage: en-tête principal et premier en-tête d'élément d'un pavé donné. Facultatif dans les deux en-têtes, principal et d'élément de pavé. Au plus un seul par composante donnée peut être présent dans soit l'en-tête principal ou l'en-tête d'élément de pavé. S'il y a de multiples éléments dans un pavé, et que ce segment marqueur soit présent, celui-ci ne doit être présent que dans le premier élément de pavé (TPsot = 0).

Facultatif dans les deux en-têtes, principal et d'élément de pavé. Lorsqu'il est utilisé dans l'en-tête principal, il outrepassa le segment marqueur QCD principal pour la composante spécifique. Lorsqu'il est utilisé dans l'en-tête d'élément de pavé, il outrepassa le marqueur QCD principal, le marqueur QCC principal, et le marqueur QCD de pavé pour la composante spécifique. Donc, l'ordre de préséance est le suivant:

Elément de pavé QCC > Elément de marqueur QCD de pavé > Marqueur QCC principal > Marqueur QCD principal
où le signe "supérieur à", >, signifie que le segment marqueur supérieur outrepassa l'inférieur.

Longueur: variable dépendant du nombre d'éléments quantifiés.

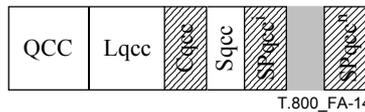


Figure A.14 – Syntaxe de composante de quantification

QCC: code de marqueur. Le Tableau A.31 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de composante de quantification.

Lqcc: longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lqcc = \begin{cases} 5 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization AND Csiz} < 257 \\ 6 & \text{scalar_quantization_derived AND Csiz} < 257 \\ 6 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded AND Csiz} < 257 \\ 6 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization AND Csiz} \Rightarrow 257 \\ 7 & \text{scalar_quantization_derived AND Csiz} \Rightarrow 257 \\ 7 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded AND Csiz} \Rightarrow 257 \end{cases} \quad (\text{A-5})$$

où "number_decomposition_levels" est défini dans les segments marqueurs COD et COC, et où "no_quantization, scalar_quantization_derived", ou "scalar_quantization_expounded" est signalé dans le paramètre Sqcc.

NOTE – Le paramètre Lqcc peut être utilisé afin de déterminer combien d'échelons sont présents dans le segment marqueur. Cependant, il n'y a pas nécessairement de correspondance avec le nombre de sous-bandes présentes parce que celles-ci peuvent être tronquées sans qu'il soit prescrit de corriger ce segment marqueur.

Cqcc: index de la composante auquel ce segment marqueur se rapporte. Les composantes sont indexées 0, 1, 2, etc. (Soit 8 ou 16 bits selon la valeur Csiz).

Sqcc: style de quantification pour cette composante.

SPqccⁱ: valeur de quantification pour chaque sous-bande dans l'ordre défini (voir § F.3.1). Le nombre de paramètres est le même que le nombre de sous-bandes dans la composante-pavé ayant le plus grand nombre de niveaux de décomposition.

Tableau A.31 – Valeurs paramétriques de composante de quantification

Paramètre	Longueur (bits)	Valeurs
QCC	16	0xFF5D
Lqcc	16	5 à 199
Cqcc	8 16	0 à 255; si Csiz < 257 0 à 16 383; Csiz ≥ 257
Sqcc	8	Tableau A.28
SPqcc ⁱ	Variable	Tableau A.28

A.6.6 Changement de l'ordre de progression (POC)

Fonction: décrit les limites et l'ordre de progression pour tout ordre de progression autre que spécifié dans les segments marqueurs COD contenus dans le flux codé.

Usage: les en-têtes principaux et d'élément de pavé. Au plus un seul segment marqueur POC peut apparaître dans un en-tête quelconque. Cependant, plusieurs progressions peuvent être décrites avec un seul segment marqueur POC. Si un seul segment marqueur POC est utilisé dans l'en-tête principal, il outrepassa l'ordre de progression dans les segments marqueurs COD principaux et d'élément de pavé. Si un marqueur POC est utilisé pour décrire la progression d'un pavé particulier, un seul segment marqueur POC doit toujours apparaître dans le premier en-tête d'élément de ce pavé. Donc, l'ordre de progression d'un pavé donné est déterminé par la présence du marqueur POC ou des valeurs du marqueur COD afin de préséance suivant:

Elément de pavé POC > POC principal > COD d'élément de pavé > COD principal

où le signe "supérieur à", >, signifie que le segment marqueur supérieur outrepassa l'inférieur.

Si un seul segment marqueur POC est utilisé, la progression de chaque paquet contenu dans le flux codé (ou pour un pavé du flux codé) doit être définie dans un ou plusieurs segments marqueurs de changement POC. Chaque ordre de progression est décrit dans un seul segment marqueur POC et doit être décrit dans un en-tête quelconque d'élément de pavé avant que d'éventuels paquets de cette progression soient trouvés.

Longueur: variable dépendant du nombre de progressions différentes.

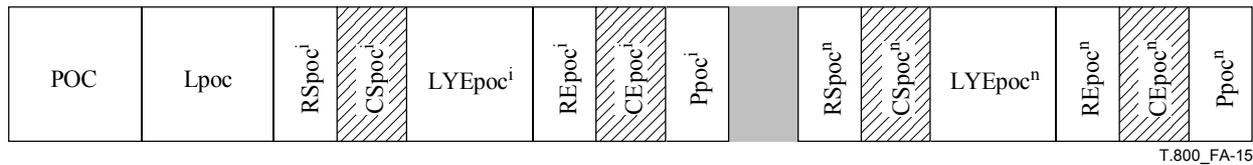


Figure A.15 – Syntaxe de pavé de changement d'ordre de progression

POC: valeur de marqueur. Le Tableau A.32 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de changement de l'ordre de progression.

Lpoc: longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Lpoc = \begin{cases} 2 + 7 \cdot \text{number_progression_order_change} & \text{Csiz} < 257 \\ 2 + 9 \cdot \text{number_progression_order_change} & \text{Csiz} \Rightarrow 257 \end{cases} \quad (\text{A-6})$$

où number_progression_order_changes est défini par le codeur.

RSpocⁱ: index de niveau de résolution (inclusif) pour le début d'une progression. Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

CSpocⁱ: index de composante (inclusif) pour le début d'une progression. Les composantes sont indexées 0, 1, 2, etc. (Soit 8 ou 16 bits selon la valeur Csiz.) Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

LYEpocⁱ: index de couche (exclusif) pour la fin d'une progression. L'index de couche commence toujours à zéro pour chaque progression. Les paquets qui ont déjà été inclus dans le flux codé ne sont pas inclus de nouveau. Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

REpocⁱ: index de niveau de résolution (exclusif) pour la fin d'une progression. Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

CEpocⁱ: index de composante (exclusif) pour la fin d'une progression. Les composantes sont indexées 0, 1, 2, etc. (Soit 8 ou 16 bits selon la valeur Csiz.) Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

Ppocⁱ: ordre de progression. Une seule valeur pour chaque changement de progression contenu dans ce pavé ou élément de pavé. Le nombre de changements de progression peut être calculé d'après la longueur du segment marqueur.

Tableau A.32 – Valeurs paramétriques de pavé de changement de l'ordre de progression

Paramètre	Longueur (bits)	Valeurs
POC	16	0xFF5F
Lpoc	16	9 à 65 535
RSpoc ⁱ	8	0 à 32
CSpoc ⁱ	8 16	0 à 255; si Csiz < 257 0 à 16 383; Csiz ≥ 257
LYEpoc ⁱ	16	1 à 65 535
REpoc ⁱ	8	(RSpoc ⁱ + 1) à 33
CEpoc ⁱ	8 16	(CSpoc ⁱ + 1) à 255, 0; si Csiz < 257 (CSpoc ⁱ + 1) à 16 384, 0; Csiz ≥ 257 (0 est interprété comme 256)
Ppoc ⁱ	8	Tableau A.16

A.7 Segments marqueurs de pointage

Les segments marqueurs de pointage fournissent soit une longueur soit un pointeur dans le flux codé. Le segment marqueur TLM décrit la longueur des éléments de pavé. Il possède les mêmes informations de longueur que le segment marqueur SOT. Le segment marqueur PLM ou PLT décrit la longueur des paquets.

NOTE – Faire apparaître tous les segments marqueurs de pointage dans l'en-tête principal permet un accès direct aux données de flux binaire. Le fait que les informations de pointage soient contenues dans l'en-tête d'élément de pavé évite au codeur d'avoir à se réinitialiser pour mémoriser ces informations.

Les paramètres TLM (Ptlm) ou SOT (Psot) pointent à partir du début de segment marqueur SOT de l'élément de pavé actuel jusqu'à la fin des données de flux binaire contenues dans cet élément de pavé. Etant donné que les éléments de pavé sont tenus d'être un multiple de 8 bits, ces valeurs ont toujours une longueur d'un octet. La Figure A.16 montre la longueur d'un élément de pavé.

Les segments marqueurs PLM et PLT sont facultatifs. Le segment marqueur PLM est utilisé dans l'en-tête principal et les segments marqueurs PLT sont utilisés dans les en-têtes d'élément de pavé. Les segments marqueurs PLM et PLT décrivent les longueurs de chaque paquet contenu dans le flux codé.

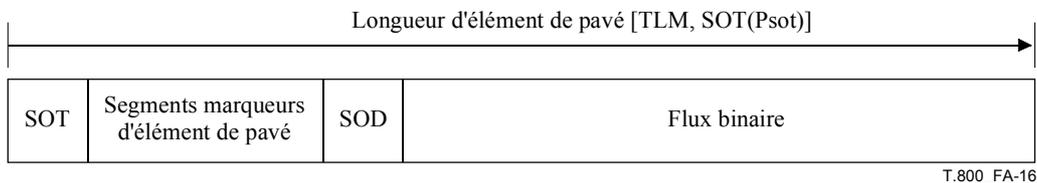


Figure A.16 – Longueurs d'élément de pavé

A.7.1 Longueurs d'élément de pavé (TLM)

Fonction: décrit la longueur de chaque élément de pavé contenu dans le flux codé. La longueur de chaque élément de pavé est mesurée depuis le premier octet du segment marqueur SOT jusqu'à la fin des données de flux binaire de cet élément de pavé. La valeur de chaque longueur d'élément de pavé individuel dans le segment marqueur TLM est la même que la valeur indiquée dans le paramètre Psot correspondant dans le segment marqueur SOT.

Usage: en-tête principal. Usage facultatif, dans l'en-tête principal seulement. Il peut y avoir de multiples segments marqueurs TLM dans l'en-tête principal.

Longueur: variable dépendant du nombre d'éléments de pavé contenus dans le flux codé.

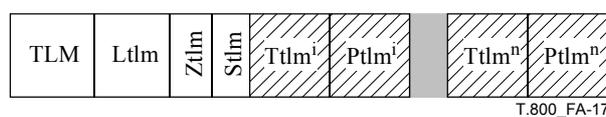


Figure A.17 – Syntaxe de longueur d'élément de pavé

- TLM:** code de marqueur. Le Tableau A.33 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de longueur d'élément de pavé.
- Ltlm:** longueur de segment marqueur en octets (non compris le marqueur). La valeur de ce paramètre est déterminée par l'équation suivante:

$$Ltlm = \begin{cases} 4 + 2 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 0 \text{ AND } SP = 0 \\ 4 + 3 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 1 \text{ AND } SP = 0 \\ 4 + 4 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 2 \text{ AND } SP = 0 \\ 4 + 4 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 0 \text{ AND } SP = 1 \\ 4 + 5 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 1 \text{ AND } SP = 1 \\ 4 + 6 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 2 \text{ AND } SP = 1 \end{cases} \quad (A-7)$$

où "number_of_tile_parts-in_marker_segment" est le nombre de longueurs d'élément de pavé indiquées dans ce segment marqueur; *ST* et *SP* sont signalés par le paramètre *Stlm*.

- Ztlm:** index de ce segment marqueur par rapport à tous les autres segments de marqueur TLM présents dans l'en-tête actuel. La séquence de paires (*Ttlm*^{*i*}, *Ptlm*^{*i*}) extraite de ce segment marqueur est concaténée, dans l'ordre croissant de *Ztlm*, avec les séquences de paires issues d'autres segments de marqueur. La *j*^o entrée dans la liste résultante contient la paire d'index de pavé et de longueur d'élément de pavé pour le *j*^o élément de pavé apparaissant dans le flux codé.
- Stlm:** longueur des paramètres *Ttlm* et *Ptlm*.
- Ttlm^{*i*}:** index de pavé du *i*^o élément de pavé. Soit aucune ou une seule valeur pour chaque élément de pavé. Le nombre d'éléments dans chaque pavé peut être calculé d'après ce segment marqueur (ou d'après la liste concaténée de tous ces marqueurs) ou d'après un paramètre *TN*sot différent de zéro, si présent.
- Ptlm^{*i*}:** longueur, en octets, à partir du début du marqueur SOT du *i*^o élément de pavé jusqu'à la fin des données de flux binaire pour cet élément de pavé. Une seule valeur pour chaque élément de pavé.

Tableau A.33 – Valeurs paramétriques de longueur d'élément de pavé

Paramètre	Longueur (bits)	Valeurs
TLM	16	0xFF55
Ltlm	16	6 à 65 535
Ztlm	8	0 à 255
Stlm	8	Tableau A.34
Ttlm ^{<i>i</i>}	0 si ST = 0 8 si ST = 1 16 si ST = 2	Pavés dans l'ordre 0 à 254 0 à 65 534
Ptlm ^{<i>i</i>}	16 si SP = 0 32 si SP = 1	14 à 65 535 14 à (2 ³² - 1)

Tableau A.34 – Paramètres dimensionnels pour Stlm

Valeurs (bits)		Longueur de paramètre
MSB	LSB	
xx00	xxxx	ST = 0; le paramètre Ttlm est bit 0, un seul élément de pavé par pavé et les pavés sont des index sans omission ou répétition
xx01	xxxx	ST = 1; Ttlm paramètre 8 bits
xx10	xxxx	ST = 2; Ttlm paramètre 16 bits
x0xx	xxxx	SP = 0; Ptlm paramètre 16 bits
x1xx	xxxx	SP = 1; Ptlm paramètre 32 bits
xx00	xxxx	Toutes les autres valeurs sont réservées

A.7.2 En-tête principal de longueur de paquet (PLM)

Fonction: fournir une liste des longueurs de paquet dans les éléments de pavé pour chaque élément de pavé en séquence.

Usage: en-tête principal. Il peut y avoir de multiples segments marqueurs PLM. Les deux segments marqueurs PLM et PLT sont facultatifs et peuvent être utilisés ensemble ou séparément.

Longueur: variable dépendant du nombre d'éléments de pavé dans l'image et du nombre de paquets dans chaque élément de pavé.

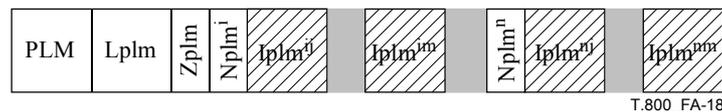


Figure A.18 – Syntaxe d'en-tête principal de longueur de paquet

PLM: code de marqueur. Le Tableau A.35 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur d'en-tête principal de longueur de paquet.

Lplm: longueur de segment marqueur en octets (non compris le marqueur).

Zplm: index de ce segment marqueur par rapport à tous les autres segments de marqueur PLM présents dans l'en-tête actuel. La séquence paramétrique ($Nplm^i$, $Iplm^i$) issue de ce segment marqueur est concaténée, dans l'ordre croissant des $Zplm$, avec les séquences paramétriques issues d'autres segments de marqueur. La k^e entrée dans la liste résultante contient la paire de nombre d'octets et d'en-tête de paquet pour le k^e élément de pavé apparaissant dans le flux codé.

Chaque segment marqueur de cette série doit se terminer par un en-tête de longueur de paquet complété. Cependant, la série de paramètres $Iplm$ décrite par le paramètre $Nplm$ n'a pas à être complète dans un segment marqueur donné. Donc, il est possible que le prochain segment marqueur PLM ne possède pas de paramètre $Nplm$ paramètre après $Zplm$, mais contienne la continuation de la série de paramètres $Iplm$ extraite du dernier segment marqueur PLM.

$Nplm^i$: nombre d'octets d'informations $Iplm$ pour le i^e élément de pavé dans l'ordre constaté dans le flux codé. Une seule valeur pour chaque élément de pavé. Si un flux codé contient un ou plusieurs éléments de pavé dépassant les limitations des marqueurs PLM, ces marqueurs ne doivent pas être utilisés.

NOTE – Cette valeur est exprimée par un nombre de 8 bits limitant le nombre d'octets $Iplm$ à 255 et le nombre de paquets dans un élément de pavé à 255, ou moins. Ce n'est pas une restriction du nombre de paquets qui peuvent se trouver dans un élément de pavé. Il s'agit seulement d'une limite à la capacité de ce segment marqueur de décrire les paquets dans un élément de pavé.

$Iplm^{ij}$: longueur du j^e paquet dans le i^e élément de pavé. Si des en-têtes de paquet sont mémorisés avec le paquet, cette longueur comporte l'en-tête de paquet. Si des en-têtes de paquet sont mémorisés dans les marqueurs PPM ou PPT, cette longueur ne comporte pas l'en-tête de longueur de paquet. Une seule étendue de valeurs pour chaque élément de pavé. Une seule valeur pour chaque paquet dans le pavé.

Tableau A.35 – Valeurs paramétriques d'en-tête principal de longueur de paquet

Paramètre	Longueur (bits)	Valeurs
PLM	16	0xFF57
Lplm	16	4 à 65 535
Zplm	8	0 à 255
Nplm ⁱ	8	0 à 255
Iplm ^{ij}	Variable	Tableau A.36

Tableau A.36 – Liste Iplm ou Iplt des longueurs de paquet

Paramètres (dans l'ordre)	Longueur (bits)	Valeurs	Signification des valeurs Iplm ou Iplt
Longueur de paquet	8 bits répétés autant de fois que nécessaire	0xxx xxxx 1xxx xxxx x000 0000 à x111 1111	7 derniers bits de longueur de paquet (nombre terminal ^{a)}) Continuer la lecture ^{b)} 7 bits de longueur de paquet
<p>a) Ce sont les 7 derniers bits qui constituent la longueur de paquet.</p> <p>b) Ce ne sont pas les 7 derniers bits qui constituent la longueur de paquet. Au contraire, ces 7 bits font partie de ceux qui constituent la longueur de paquet. Celle-ci a été subdivisée en 7 segments binaires qui sont envoyés en séquence depuis le segment de plus fort poids jusqu'au segment de plus faible poids. Par ailleurs, les bits contenus dans le segment de plus fort poids sont justifiés à droite jusqu'à la limite d'octet. Par exemple, une longueur de paquet de 128 est signalée par 1000 0001 0000 0000, alors qu'une longueur de 512 est signalée par 1000 0100 0000 0000.</p>			

A.7.3 En-tête d'élément de pavé de longueur de paquet (PLT)

Fonction: établissement d'une liste des longueurs de paquet dans l'élément de pavé.

Usage: en-tête d'élément de pavé. Il peut y avoir de multiples segments PLT de marqueur par pavé. Les deux segments marqueurs PLM et PLT sont facultatifs et peuvent être utilisés ensemble ou séparément. Doit apparaître dans un en-tête quelconque d'élément de pavé avant les paquets dont les longueurs y sont décrites.

Longueur: variable dépendant du nombre de paquets dans chaque élément de pavé.

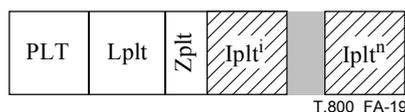


Figure A.19 – Syntaxe d'en-tête d'élément de pavé de longueur de paquet

- PLT:** code de marqueur. Le Tableau A.37 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur d'en-tête d'élément de pavé de longueur de paquet.
- Lplt:** longueur de segment marqueur en octets (non compris le marqueur).
- Zplt:** index de ce segment marqueur par rapport à tous les autres segments PLT de marqueur présents dans l'en-tête actuel. La séquence de (Ipltⁱ) paramètres extraite de ce segment marqueur est concaténée, dans l'ordre croissant des Zplt, avec les séquences paramétriques issues d'autres segments de marqueur. Chaque segment marqueur de cette série doit se terminer par un en-tête complété de longueur de paquet.
- Iplmⁱ:** longueur du i^e paquet. Si des en-têtes de paquet sont mémorisés avec le paquet, cette longueur comporte l'en-tête de paquet; si des en-têtes de paquet sont mémorisés dans un marqueur PPM ou PPT, cette longueur ne comporte pas l'en-tête de longueur de paquet.

Tableau A.37 – Valeurs paramétriques d'en-tête d'élément de pavé de longueur de paquet

Paramètre	Longueur (bits)	Valeurs
PLT	16	0xFF58
Lplt	16	4 à 65 535
Zplt	8	0 à 255
Iplt ⁱ	Variable	Tableau A.36

A.7.4 En-tête principal de paquet compact (PPM)

Fonction: ensemble des en-têtes de paquet extraits de tous les pavés.

NOTE – Son utilité est d'éviter des lectures multiples afin de décoder des en-têtes.

Usage: en-tête principal. Peut être utilisé dans l'en-tête principal pour tous les éléments de pavé sauf si un segment marqueur PPT est utilisé dans l'en-tête d'élément de pavé.

Les en-têtes de paquet doivent se trouver dans un seul de trois emplacements dans le flux codé. Si le segment marqueur PPM est présent, tous les en-têtes de paquet doivent se trouver dans l'en-tête principal. Dans ce cas, le segment marqueur PPT et les paquets répartis dans le flux binaire des éléments de pavé sont interdits.

S'il n'y a aucun segment marqueur PPM, alors les en-têtes de paquet peuvent soit être répartis dans des segments marqueurs PPT soit être répartis dans le flux codé comme défini au § B.10. L'en-tête de paquet ne doit pas être à la fois dans un segment marqueur PPT et dans le flux codé pour le même pavé. Si les en-têtes de paquet se trouvent dans des segments marqueurs PPT, ils doivent apparaître dans un en-tête d'élément de pavé avant que les données de paquet correspondantes apparaissent (c'est-à-dire dans le même en-tête d'élément de pavé ou avec une valeur TP_{sot} inférieure). Il peut y avoir de multiples segments marqueurs PPT dans un en-tête d'élément de pavé.

Longueur: variable dépendant du nombre de paquets dans chaque élément de pavé et de la longueur de l'en-tête de paquet.



Figure A.20 – Syntaxe d'en-tête principal de paquet compact

PPM: code de marqueur. Le Tableau A.38 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur d'en-tête principal de paquet compact.

Lppm: longueur de segment marqueur en octets, non compris le marqueur.

Zppm: index de ce segment marqueur par rapport à tous les autres segments de marqueur PPM présents dans l'en-tête principal. La séquence de ($Nppm^i$, $Ippm^i$) paramètres extraite de ce segment marqueur est concaténée, dans l'ordre croissant des $Zppm$, avec les séquences paramétriques issues d'autres segments de marqueur. La k^e entrée dans la liste résultante contient le nombre d'octets et les en-têtes de paquet pour le k^e élément de pavé apparaissant dans le flux codé.

Chaque segment marqueur de cette série doit se terminer par un en-tête complété de paquet. Cependant, la série de paramètres $Ippm$ décrite par les $Nppm$ n'a pas à être complète dans un segment marqueur donné. Donc, il est possible que le prochain segment marqueur PPM ne possède pas de paramètre $Nppm$ après $Zppm$, mais contienne la continuation de la série $Ippm$ extraite du dernier segment marqueur PPM.

$Nppm^i$: nombre d'octets de $Ippm$ information pour le i^e élément de pavé dans l'ordre constaté dans le flux codé. Une seule valeur pour chaque élément de pavé (et non pavé).

$Ippm^{ij}$: en-tête de paquet pour chaque paquet ordonné dans l'élément de pavé. Le contenu est exactement l'en-tête de paquet qui aurait été réparti dans le flux binaire comme décrit au § B.10.

Tableau A.38 – Valeurs paramétriques d'en-tête principal de paquet compact

Paramètre	Longueur (bits)	Valeurs
PPM	16	0xFF60
Lppm	16	7 à 65 535
Zppm	8	0 à 255
Nppm ⁱ	32	0 à (2 ³² - 1)
Ippm ^{ij}	Variable	En-têtes de paquet

A.7.5 En-tête d'élément de pavé de paquet compact (PPT, *packed packet headers, tile-part header*)

Fonction: ensemble de l'en-tête de paquet provenant d'un pavé ou élément de pavé.

Usage: en-tête d'élément de pavé. Doit apparaître dans un en-tête quelconque d'élément de pavé avant les paquets dont les en-têtes y sont décrits.

L'en-tête de paquet doit se trouver dans un seul de trois emplacements dans le flux codé. Si le segment marqueur PPM est présent, tous les en-têtes de paquet doivent se trouver dans l'en-tête principal. Dans ce cas, le segment marqueur PPT et les paquets répartis dans le flux binaire des éléments de pavé sont interdits.

S'il n'y a aucun segment marqueur PPM, alors l'en-tête de paquet peut soit être réparti dans les segments marqueurs PPT soit être réparti dans le flux codé comme défini au § B.10. Les en-têtes de paquet ne doivent pas être à la fois dans un segment marqueur PPT et dans le flux codé pour le même pavé. Si les en-têtes de paquet se trouvent dans des segments marqueurs PPT, ils doivent apparaître dans un en-tête d'élément de pavé avant les données de paquet correspondantes (c'est-à-dire dans le même en-tête d'élément de pavé ou dans un en-tête d'élément de pavé ayant une valeur TP_{pot} inférieure). Il peut y avoir de multiples segments marqueurs PPT dans un en-tête d'élément de pavé.

Longueur: variable dépendant du nombre de paquets dans chaque élément de pavé et de la compression de l'en-tête de paquet.

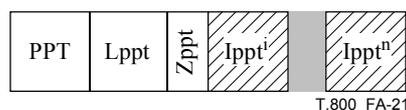


Figure A.21 – Syntaxe d'en-tête d'élément de pavé d'en-tête de pavé compact

PPT: code de marqueur. Le Tableau A.39 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur d'en-tête d'élément de pavé de paquet compact.

Lppt: longueur de segment marqueur en octets, non compris le marqueur.

Zppt: index de ce segment marqueur par rapport à tous les autres segments de marqueur PPT présents dans l'en-tête actuel. La séquence de (Ipptⁱ) paramètres issue de ce segment marqueur est concaténée, dans l'ordre croissant des Zppt, avec les séquences paramétriques issues d'autres segments de marqueur. Chaque segment marqueur de cette série doit se terminer par un en-tête complété de paquet.

Ipptⁱ: en-tête de chaque paquet ordonné dans l'élément de pavé. L'index de composante, la couche et le niveau de résolution sont déterminés par la méthode de progression ou par les segments marqueurs de changement POC. Le contenu est exactement l'en-tête de paquet qui aurait été réparti dans le flux binaire comme décrit dans les informations d'en-tête de paquet du § B.10.

Tableau A.39 – Valeurs paramétriques d'en-tête d'élément de pavé d'en-tête de paquet

Paramètre	Longueur (bits)	Valeurs
PPT	16	0xFF61
Lppt	16	4 à 65 535
Zppt	8	0 à 255
Ippt ⁱ	Variable	En-têtes de paquet

A.8 Marqueur et segments marqueurs dans le flux binaire

Ces marqueurs et segments marqueurs sont utilisés pour assurer l'élasticité aux erreurs. Ils peuvent être trouvés dans le flux binaire. (Le marqueur EPH peut également être utilisé dans les segments marqueurs PPM et PPT.)

A.8.1 Début de paquet (SOP)

Fonction: marque le début d'un paquet dans un flux codé.

Usage: facultatif. Peut être utilisé dans le flux binaire au début de chaque paquet. Ne doit pas être utilisé s'il n'est pas indiqué qu'il est permis dans le segment marqueur COD approprié (voir § A.6.1). Si les segments marqueurs PPM ou PPT sont utilisés, alors le segment marqueur SOP peut apparaître immédiatement avant les données de paquet contenues dans le flux binaire.

Si des segments marqueurs SOP sont permis (par signalisation dans le segment marqueur COD, voir § A.6.1), chaque paquet dans tout élément de pavé donné peut être ou ne pas être affixé avec un segment marqueur SOP. Cependant, que le segment marqueur SOP soit ou non utilisé, le décompte dans le champ Nsop est incrémenté pour chaque paquet. Si les en-têtes de paquet sont transférés vers des segments marqueurs PPM ou PPT (voir § A.7.4 et A.7.5), alors les segments marqueurs SOP peuvent apparaître immédiatement avant le corps de paquet dans la portion de données d'image comprimées d'élément de pavé.

Longueur: fixe.

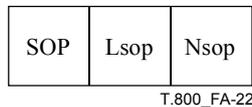


Figure A.22 – Syntaxe de début de paquet

SOP: code de marqueur. Le Tableau A.40 montre la longueur et les valeurs du symbole ainsi que les paramètres pour segment marqueur de début de paquet.

Lsop: longueur de segment marqueur en octets, non compris le marqueur.

Nsop: numéro séquentiel de paquet. Le premier paquet d'un pavé codé reçoit la valeur zéro. Pour chaque paquet contenu successivement dans ce pavé codé, ce nombre est augmenté de un. Lorsque le nombre maximal est atteint, le nombre revient à zéro.

Tableau A.40 – Valeurs paramétriques de début de paquet

Paramètre	Longueur (bits)	Valeurs
SOP	16	0xFF91
Lsop	16	4
Nsop	16	0 à 65 535

A.8.2 Fin d'en-tête de paquet (EPH)

Fonction: indique la fin de l'en-tête d'un paquet donné. Il délimite l'en-tête de paquet contenu dans le flux binaire ou dans des segments marqueurs PPM ou PPT. Ce marqueur n'indique pas les données de début de paquet. Si les en-têtes de paquet ne sont pas dans le flux binaire (c'est-à-dire si les segments marqueurs PPM ou PPT sont utilisés), ce marqueur ne doit pas être utilisé dans le flux binaire.

Usage: doit être utilisé si et seulement si indiqué dans le segment marqueur COD approprié (voir § A.6.1). Apparaît immédiatement après un en-tête de paquet.

Si des marqueurs EPH sont requis (par signalisation dans le segment marqueur COD, voir § A.6.1), chaque en-tête de paquet d'un élément de pavé donné doit être suffixé avec un segment marqueur EPH. Si les en-têtes de paquet sont transférés vers les segments marqueurs PPM ou PPT (voir § A.7.4 et A.7.5), alors les marqueurs EPH doivent apparaître après les en-têtes de paquet dans les segments marqueurs PPM ou PPT.

Longueur: fixe.

EPH: code de marqueur

Tableau A.41 – Valeurs paramétriques de fin d'en-tête de paquet

Paramètre	Longueur (bits)	Valeurs
EPH	16	0xFF92

A.9 Segments marqueurs informationnels

Ces segments marqueurs sont strictement informatifs et ne sont pas nécessaires pour un décodeur. Cependant, ces segments marqueurs peuvent faciliter le fonctionnement d'un analyseur syntaxique ou d'un décodeur. De plus amples informations sur la source et les caractéristiques de l'image peuvent être obtenues au moyen d'un format de fichier comme JP2 (voir Annexe I).

A.9.1 Enregistrement de composante (CRG)

Fonction: permet un enregistrement spécifique de composantes les unes par rapport aux autres. Aux fins du codage, les échantillons de composantes sont considérés comme étant situés à des points de grille de référence qui sont des multiples entiers de XR_{siz} et YR_{siz} (voir § A.5.1). Cependant, cela peut être inapproprié afin de restituer l'image. Le segment marqueur CRG décrit le "centre de gravité" des échantillons de chaque composante par rapport à la séparation. Ce segment marqueur n'a pas d'effet sur le décodage du flux codé.

NOTE – Ce décalage d'enregistrement de composante est calculé par rapport au décalage d'image (XO_{siz} et YO_{siz}) et à la séparation des composantes (XR_{siz}^i et YR_{siz}^i). Par exemple, le point de grille de référence horizontale pour les échantillons de composante situés le plus à gauche c est $XR_{siz}^c \lceil XO_{siz} / XR_{siz}^c \rceil$ (de même pour le sens vertical). Le décalage horizontal indiqué dans ce segment marqueur s'ajoute à ce décalage.

Usage: en-tête principal seulement. Un seul CRG peut être utilisé dans l'en-tête principal et est applicable à tous les pavés.

Longueur: variable dépendant du nombre de composantes.

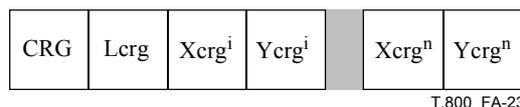


Figure A.23 – Syntaxe d'enregistrement de composante

CRG: code de marqueur. Le Tableau A.42 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur d'enregistrement de composante.

Lcrg: longueur de segment marqueur en octets (non compris le marqueur).

Xcrgⁱ: valeur du décalage horizontal, en unités de 1/65536 de la séparation horizontale XR_{siz}^i , pour la i^{e} composante. Donc, les valeurs vont de 0/65536 (l'échantillon occupe son point de grille de référence) à XR_{siz}^c (65535/65 536 (juste avant le prochain point de grille de référence de l'échantillon)). Cette valeur est répétée pour chaque composante.

Ycrgⁱ: valeur du décalage vertical, en unités de 1/65536 de la séparation verticale YR_{siz}^i , pour la i^{e} composante. Donc, les valeurs vont de 0/65536 (l'échantillon occupe son point de grille de référence) à YR_{siz}^c (65535/65536 (juste avant le prochain point de grille de référence de l'échantillon)). Cette valeur est répétée pour chaque composante.

Tableau A.42 – Valeurs paramétriques d'enregistrement de composante

Paramètre	Longueur (bits)	Valeurs
CRG	16	0xFF63
Lcrg	16	6 à 65 534
Xcrg ⁱ	16	0 à 65 535
Ycrg ⁱ	16	0 à 65 535

A.9.2 Commentaire (COM)

Fonction: permet d'insérer des données non structurées dans les en-têtes principaux et d'éléments de pavé.

Usage: les en-têtes principaux et d'élément de pavé. Répétable autant de fois que souhaité dans soit l'en-tête principal ou l'en-tête d'élément de pavé ou dans les deux. Ce segment marqueur n'a pas d'effet sur le décodage du flux codé.

Longueur: variable dépendant de la longueur du message.

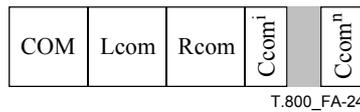


Figure A.24 – Syntaxe de commentaire

COM: code de marqueur. Le Tableau A.43 montre la longueur et les valeurs du symbole ainsi que les paramètres pour le segment marqueur de commentaire.

Lcom: longueur de segment marqueur en octets (non compris le marqueur).

Rcom: valeur d'enregistrement du segment marqueur.

Ccomⁱ: octet de données non structurées.

Tableau A.43 – Valeurs paramétriques de commentaire

Paramètre	Longueur (bits)	Valeurs
COM	16	0xFF64
Lcom	16	5 à 65 535
Rcom	16	Tableau A.44
Ccom ⁱ	8	0 à 255

Tableau A.44 – Valeurs d'enregistrement pour le paramètre Rcom

Valeurs	Valeurs d'enregistrement
0	Utilisation générale (valeurs binaires)
1	Utilisation générale (valeurs ISO/CEI 8859-15 (Latin))
	Toutes les autres valeurs sont réservées

A.10 Restrictions du flux codé conformes à la présente Recommandation | Norme internationale

Afin de faciliter une large interopérabilité du flux codé JPEG 2000, des restrictions du flux codé sont introduites. Les "restrictions du flux codé" ont deux profils: profil-0 et profil-1. Le cas "absence de restrictions" signifiant la conformité à la présente Recommandation | Norme internationale peut être appelé profil-2. Le profil-0 et le profil-1 sont définis comme suit.

Un échange maximal sera réalisé avec des flux codés correspondant à profil-0 et un échange moyen sera réalisé avec des flux codés correspondant à profil-1.

Tableau A.45 – Restrictions du flux codé

Restrictions	Profil-0	Profil-1
Segment marqueur SIZ		
Indication de profil	$R_{siz} = 1$	$R_{siz} = 2$
Dimensions d'image	$X_{siz}, Y_{siz} < 2^{31}$	$X_{siz}, Y_{siz} < 2^{31}$
Pavés	Pavés de dimensions 128×128 : $Y_{Tsiz} = X_{Tsiz} = 128$ ou un pavé pour l'image entière: $Y_{Tsiz} + Y_{TOsiz} \geq Y_{siz}$ $X_{Tsiz} + X_{TOsiz} \geq X_{siz}$	$X_{Tsiz} / \min(X_{Rsiz}^i, Y_{Rsiz}^i) \geq 1024$ $X_{Tsiz} = Y_{Tsiz}$ ou un pavé pour l'image entière: $Y_{Tsiz} + Y_{TOsiz} \geq Y_{siz}$ $X_{Tsiz} + X_{TOsiz} \geq X_{siz}$
Origine d'image et de pavé	$X_{Osiz} = Y_{Osiz} = X_{TOsiz} = Y_{TOsiz} = 0$	$X_{Osiz}, Y_{Osiz}, X_{TOsiz}, Y_{TOsiz} < 2^{31}$
Segment marqueur RGN	$SP_{rgn} \leq 37$	$SP_{rgn} \leq 37$
Sous-échantillonnage	$X_{Rsiz}^i = 1, 2, \text{ ou } 4$ $Y_{Rsiz}^i = 1, 2, \text{ ou } 4$	Absence de restriction
Blocs codés		
Longueur de bloc codé	$xcb = ycb = 5$ ou $xcb = ycb = 6$	$xcb \leq 6, ycb \leq 6$
Style de bloc codé	$SP_{cod}, SP_{coc} = 00sp$ vtra où $a = r = v = 0$, et $t, p, s = 0$ ou 1 NOTE 1 – $t = 1$ pour terminaison à chaque passe de codage, $p = 1$ pour terminaison prédictive, $s = 1$ pour symboles de segmentation	Absence de restriction
Emplacements de marqueur		
En-têtes compacts (PPM, PPT)	Interdits	Absence de restriction
COD, COC, QCD, QCC	En-tête principal seulement	Absence de restriction
Exigences de sous-ensemble		
Résolution LL	Si un pavé est utilisé pour l'image entière, $(X_{siz} - X_{Osiz}) / D(I) \leq 128$ et $(Y_{siz} - Y_{Osiz}) / D(I) \leq 128$ où $D(I) = 2^{\text{nombre_de_niveaux_de_décomposition}}$ dans SP_{cod} ou SP_{coc} , pour $I =$ composante 0 à 3	Pour chaque pavé dans l'image, $\lfloor x1 / D(i) \rfloor - \lfloor tx0 / D(i) \rfloor \leq 128$ et $\lfloor ty1 / D(i) \rfloor - \lfloor ty0 / D(i) \rfloor \leq 128$ où $D(I) = 2^{\text{nombre_de_niveaux_de_décomposition}}$ dans SP_{cod} ou SP_{coc} , pour $I =$ composante 0 à 3. NOTE 2 – $tx0, tx1, ty0$ et $ty1$ sont comme définis par les équations (B-7) à (B-10).
Analysabilité	Si le marqueur POC est présent, le marqueur POC doit avoir $RSPOC0 = 0$ et $CSPOC0 = 0$. NOTE 3 – Certains décodeurs conformes peuvent ne décoder que les paquets associés à la première progression.	Absence de restriction
Éléments de pavé	Éléments de pavé avec $TP_{spot} = 0$ dans chaque pavé avant d'éventuels éléments de pavé avec $TP_{spot} > 0$. Éléments de pavé avec $Isot = 0$ à $Isot = \text{number_of_tiles} - 1$, dans l'ordre séquentiel pour tous les éléments de pavé avec $TP_{spot} = 0$	Absence de restriction
Dimensions de district	Les "dimensions de district" définies par SP_{cod} ou par SP_{coc} (Tableau A.15 et A.21) doivent toujours être assez grandes pour qu'il y ait un seul district à tous les niveaux de résolution ayant des dimensions inférieures ou égales à 128 par 128. NOTE 4 – Les dimensions de district $PP_x \geq 7$ et $PP_y \geq 7$ sont suffisantes pour garantir un seul district par sous-bande lorsque $X_{Osiz} = 0$ et $Y_{Osiz} = 0$.	Absence de restriction

Annexe B

Ordonnancement des données d'image comprimées et non comprimées

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe décrit les diverses entités structurelles, et leur organisation dans le flux codé: composantes, pavés, sous-bandes, et leurs subdivisions.

B.1 Introduction aux concepts de structure des données d'image

La grille de référence fournit un mécanisme pour coenregistrer des composantes et pour définir des sous-ensembles de la grille de référence, par exemple, la zone d'image et les pavés.

Les composantes consistent en matrices bidimensionnelles d'échantillons. Chaque composante c possède les paramètres XR_{siz}^c , YR_{siz}^c (voir § A.5.1) qui définissent l'affectation entre échantillons de composante et points de grille de référence. Chaque échantillon de composante est associé à un point de grille de référence (quoique non inversement). Cette affectation induit un enregistrement de composante avec un autre qui est utilisé pour le codage seulement.

Chaque composante est subdivisée en pavés correspondant au pavage de la grille de référence. Ces composantes-pavés sont codées indépendamment. Chaque composante-pavé est transformée par ondelettes en plusieurs niveaux de décomposition qui sont associés à des niveaux de résolution (voir Annexe F). Chaque niveau de résolution se compose soit des sous-bandes HL, LH et HH issues d'un niveau de décomposition soit de la sous-bande N_{LL} . Il y a donc un niveau de résolution de plus qu'il y a de niveaux de décomposition.

Chaque sous-bande possède sa propre origine. Les conditions limites de sous-bande sont uniques pour chaque sous-bande HL, LH et HH.

NOTE – Cette convention diffère des diagrammes d'ondelette habituels, qui placent dans un même espace toutes les sous-bandes pour une composante.

Les districts et les blocs codés sont définis à un niveau de résolution et dans une sous-bande. Par conséquent, ils peuvent varier selon les composantes-pavés. Les districts sont définis de façon que les blocs codés s'adaptent correctement, c'est-à-dire qu'ils "s'alignent" les uns avec les autres.

Dans les figures associées, des limites et des axes de coordonnées sont représentés. Dans chaque cas, les échantillons ou coefficients qui coïncident avec les limites du bord gauche et du bord supérieur sont inclus dans une région donnée, alors que les échantillons ou coefficients situés aux limites du bord droit et/ou du bord inférieur ne sont pas inclus dans cette région.

De même, dans les formules associées, un grand nombre de variables ont des valeurs qui peuvent changer en fonction d'une composante, d'un pavé, ou d'un niveau de résolution. Ces valeurs peuvent changer explicitement (au moyen de la syntaxe décrite dans l'Annexe A) ou implicitement (par propagation). Par commodité de notation, certaines dépendances sont supprimées dans l'analyse qui suit.

B.2 Affectation des composantes à la grille de référence

Toutes les composantes (et de nombreuses autres structures dans la présente annexe) sont définies par rapport à la grille de référence. Les divers paramètres définissant la grille de référence apparaissent dans la Figure B.1. La grille de référence est une grille de points rectangulaire avec les indices de $(0, 0)$ à $(X_{siz} - 1, Y_{siz} - 1)$. Une "zone d'image" est définie sur la grille de référence par les paramètres dimensionnels, (X_{siz}, Y_{siz}) et (XO_{siz}, YO_{siz}) . Spécifiquement, la zone d'image sur la grille de référence est définie par son point supérieur gauche de grille de référence à l'emplacement (XO_{siz}, YO_{siz}) , et par son point inférieur droit de grille de référence à l'emplacement $(X_{siz} - 1, Y_{siz} - 1)$.

Les échantillons de composante c sont situés à des multiples entiers de (XR_{siz}^c, YR_{siz}^c) sur la grille de référence. Chaque domaine de composante est une version sous-échantillonnée de la grille de référence avec les coordonnées $(0, 0)$ comme point commun pour chaque composante. Les échantillons d'une rangée sont situés à des points de grille de référence qui sont à des multiples entiers de XR_{siz}^c et les échantillons d'une colonne sont situés à des points de grille de référence qui sont à des multiples entiers de YR_{siz}^c . Seuls les échantillons qui s'inscrivent dans la zone d'image appartiennent en fait à la composante d'image. Donc, les échantillons de composante c sont affectés à un rectangle dont l'échantillon supérieur gauche possède les coordonnées (x_0, y_0) et dont l'échantillon inférieur droit possède les coordonnées $(x_1 - 1, y_1 - 1)$,

où:

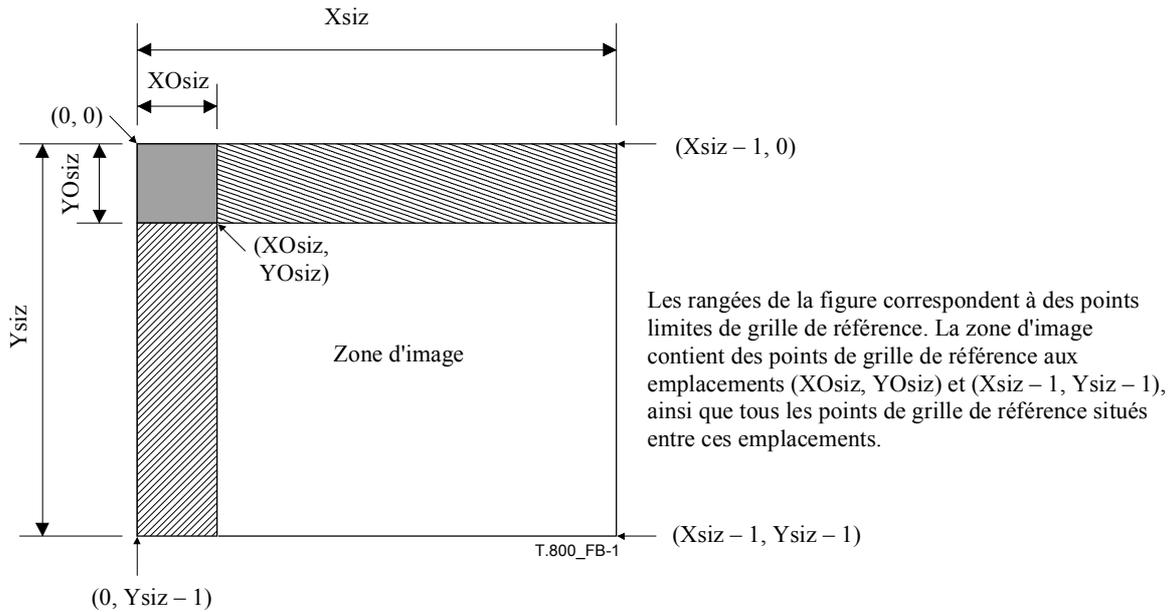


Figure B.1 – Schéma de grille de référence

$$x_0 = \left\lceil \frac{XOsiz}{XRsiz^c} \right\rceil \quad x_1 = \left\lceil \frac{Xsiz}{XRsiz^c} \right\rceil \quad y_0 = \left\lceil \frac{YOsiz}{YRsiz^c} \right\rceil \quad y_1 = \left\lceil \frac{Ysiz}{YRsiz^c} \right\rceil \quad (B-1)$$

Donc, les dimensions de composante c sont données par:

$$(largeur, hauteur) = (x_1 - x_0, y_1 - y_0) \quad (B-2)$$

Les paramètres, $Xsiz$, $Ysiz$, $XOsiz$, $YOsiz$, $XRsiz^c$ et $YRsiz^c$ sont tous définis dans le segment marqueur SIZ (voir § A.5.1).

NOTE 1 – Le fait que toutes les composantes partagent le décalage d'image ($XOsiz$, $YOsiz$) et les dimensions d'image ($Xsiz$, $Ysiz$) induit un enregistrement des composantes.

NOTE 2 – La Figure B.2 montre un exemple de trois composantes affectées à la grille de référence. La Figure B.3 montre la zone d'image à partir d'un décalage d'image particulier avec différentes valeurs ($XRsiz$, $YRsiz$). La coordonnée de l'échantillon supérieur gauche, dans le domaine de la composante d'image inclus dans la zone d'image, est également illustrée.

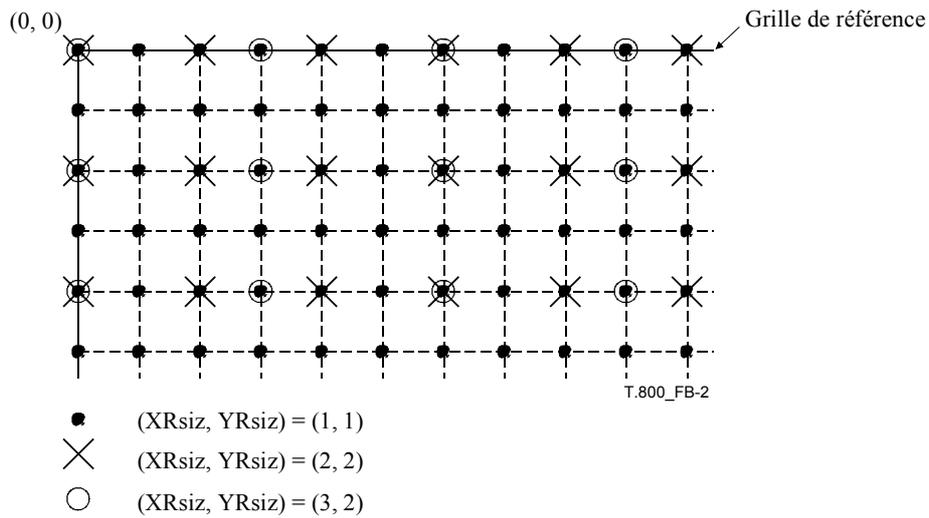


Figure B.2 – Emplacements d'échantillon de composante sur la grille de référence pour différentes valeurs XRsize et YRsize

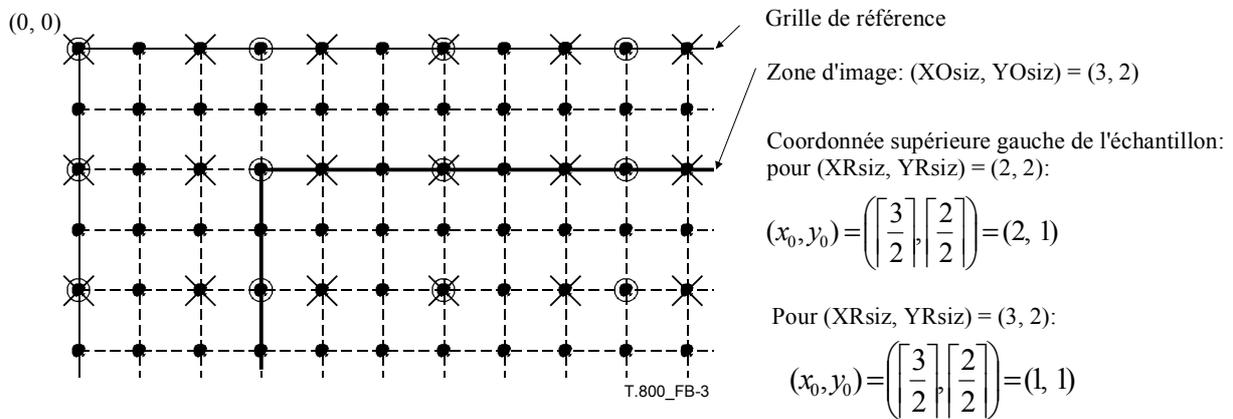


Figure B.3 – Exemple d'emplacements d'échantillon de composante supérieure gauche

B.3 Subdivision de la zone d'image en pavés et composantes-pavés

La grille de référence est partitionnée en matrices rectangulaires de pavés dont les dimensions sont régulières. Les dimensions de pavé et le décalage de pavage sont respectivement définis, dans la grille de référence, par les paires dimensionnelles (XTsize, YTsize) et (XTOsize, YTOsize). Ce sont tous des paramètres extraits du segment marqueur SIZ (voir § A.5.1).

Chaque pavé a une largeur de XTsize points de grille de référence et une hauteur de YTsize points de grille de référence. Le coin supérieur gauche du premier pavé (pavé 0) est le décalage (XTOsize, YTOsize) du coin supérieur gauche de la grille de référence. Les pavés sont numérotés dans l'ordre de balayage indiqué par l'index de pavé situé dans paramètre Isot extrait du segment marqueur SOT au § A.4.2. Donc, les coordonnées du coin supérieur gauche du premier pavé par rapport à la grille de référence sont (XTOsize, YTOsize). La Figure B.4 montre cette relation.

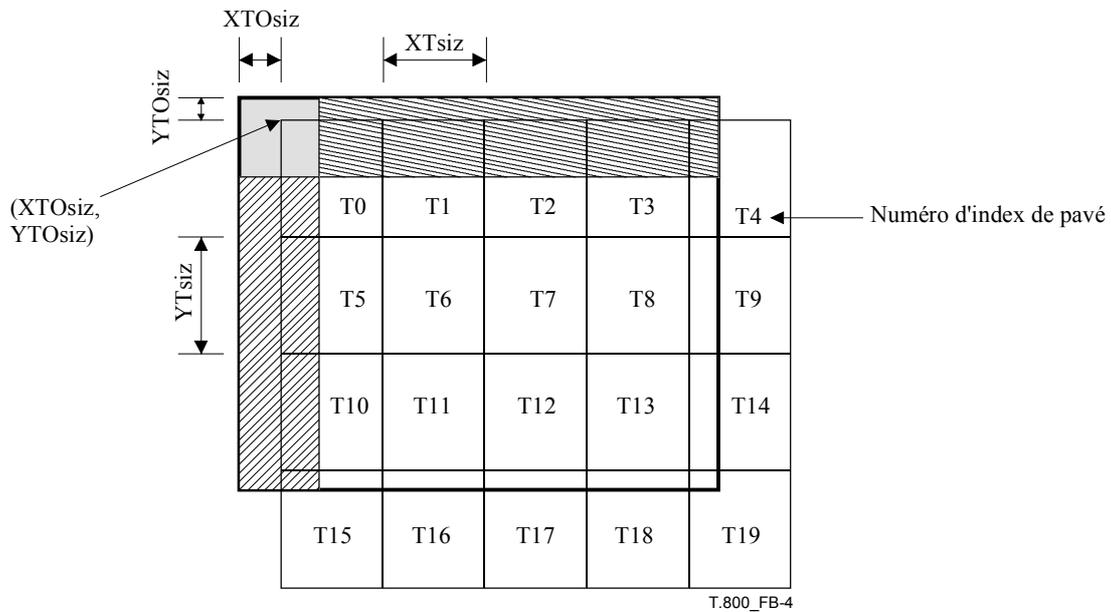


Figure B.4 – Schéma du pavage de grille de référence

Les décalages de grille de pavé ($XTOsiz$, $YTOsiz$) sont contraints à ne pas être supérieurs aux décalages de zone d'image, ce qui est exprimé par les étendues suivantes:

$$0 \leq XTOsiz \leq XOsiz \qquad 0 \leq YTOsiz \leq YOsiz \qquad (B-3)$$

De même, les dimensions de pavé plus le décalage de pavé doivent être supérieures au décalage de zone d'image. Cela garantit que le premier pavé (pavé 0) contient au moins un point de grille de référence extrait de la zone d'image, ce qui est exprimé par les étendues suivantes:

$$XTsiz + XTOsiz > XOsiz \qquad YTsiz + YTOsiz > YOsiz \qquad (B-4)$$

Le nombre de pavés dans le sens X ($numXtiles$) et dans le sens Y ($numYtiles$) est le suivant:

$$numXtiles = \left\lceil \frac{Xsiz - XTOsiz}{XTsiz} \right\rceil \qquad numYtiles = \left\lceil \frac{Ysiz - YTOsiz}{YTsiz} \right\rceil \qquad (B-5)$$

Aux fins de cette description, il est utile que les pavés soient indexés en termes de position horizontale et de position verticale. Soit p l'indice horizontal d'un pavé, allant de 0 à $numXtiles - 1$, et q l'index vertical d'un pavé, allant de 0 à $numYtiles - 1$, déterminés par l'index de pavé comme suit:

$$p = \text{mod} (t, numXtiles) \qquad q = \left\lceil \frac{t}{numXtiles} \right\rceil \qquad (B-6)$$

où t est l'index du pavé dans la Figure B.4.

Les coordonnées d'un pavé particulier sur la grille de référence sont décrites par les équations suivantes:

$$tx_0(p, q) = \max(XTOsiz + p \cdot XTsiz, XOsiz) \quad (\text{B-7})$$

$$ty_0(p, q) = \max(YTOsiz + q \cdot YTsiz, YOsiz) \quad (\text{B-8})$$

$$tx_1(p, q) = \min(XTOsiz + (p + 1) \cdot XTsiz, Xsiz) \quad (\text{B-9})$$

$$ty_1(p, q) = \min(YTOsiz + (q + 1) \cdot YTsiz, Ysiz) \quad (\text{B-10})$$

où $tx_0(p, q)$ et $ty_0(p, q)$ sont les coordonnées du coin supérieur gauche du pavé, $tx_1(p, q) - 1$ et $ty_1(p, q) - 1$ sont les coordonnées du coin inférieur droit du pavé. L'on omettra souvent les coordonnées du pavé lors de la référence à un pavé spécifique et l'on se rapportera aux coordonnées (tx_0, ty_0) et (tx_1, ty_1) .

Donc les dimensions d'un pavé dans la grille de référence sont:

$$(tx_1 - tx_0, ty_1 - ty_0) \quad (\text{B-11})$$

Dans le domaine de composante d'image i , les coordonnées de l'échantillon supérieur gauche sont données par (tcx_0, tcy_0) et les coordonnées de l'échantillon inférieur droit sont données par $(tcx_1 - 1, tcy_1 - 1)$, où:

$$tcx_0 = \left\lceil \frac{tx_0}{XRsiz^i} \right\rceil \quad tcx_1 = \left\lceil \frac{tx_1}{XRsiz^i} \right\rceil \quad tcy_0 = \left\lceil \frac{ty_0}{YRsiz^i} \right\rceil \quad tcy_1 = \left\lceil \frac{ty_1}{YRsiz^i} \right\rceil \quad (\text{B-12})$$

de sorte que les dimensions de la composante-pavé sont:

$$(tcx_1 - tcx_0, tcy_1 - tcy_0) \quad (\text{B-13})$$

B.4 Exemple du mappage de composantes à la grille de référence (pour information)

L'exemple suivant vise à illustrer l'affectation de composantes d'image à la grille de référence et la zone induite par pavage entre composantes avec différents facteurs de sous-échantillonnage. Cet exemple implique qu'une application dans laquelle une image originale avec rapport hauteur/largeur 16:9 doit être comprimée avec la présente Recommandation | Norme internationale. Les dimensions d'image, le décalage d'image, les dimensions de pavé et le décalage de pavé sont utilisés de façon qu'une image de rapport hauteur/largeur 4:3 puisse être recadrée depuis le centre de l'image originale. La Figure B.5 montre la grille de référence et les zones d'image ainsi que la structure de pavage qui sera imposée dans cet exemple.

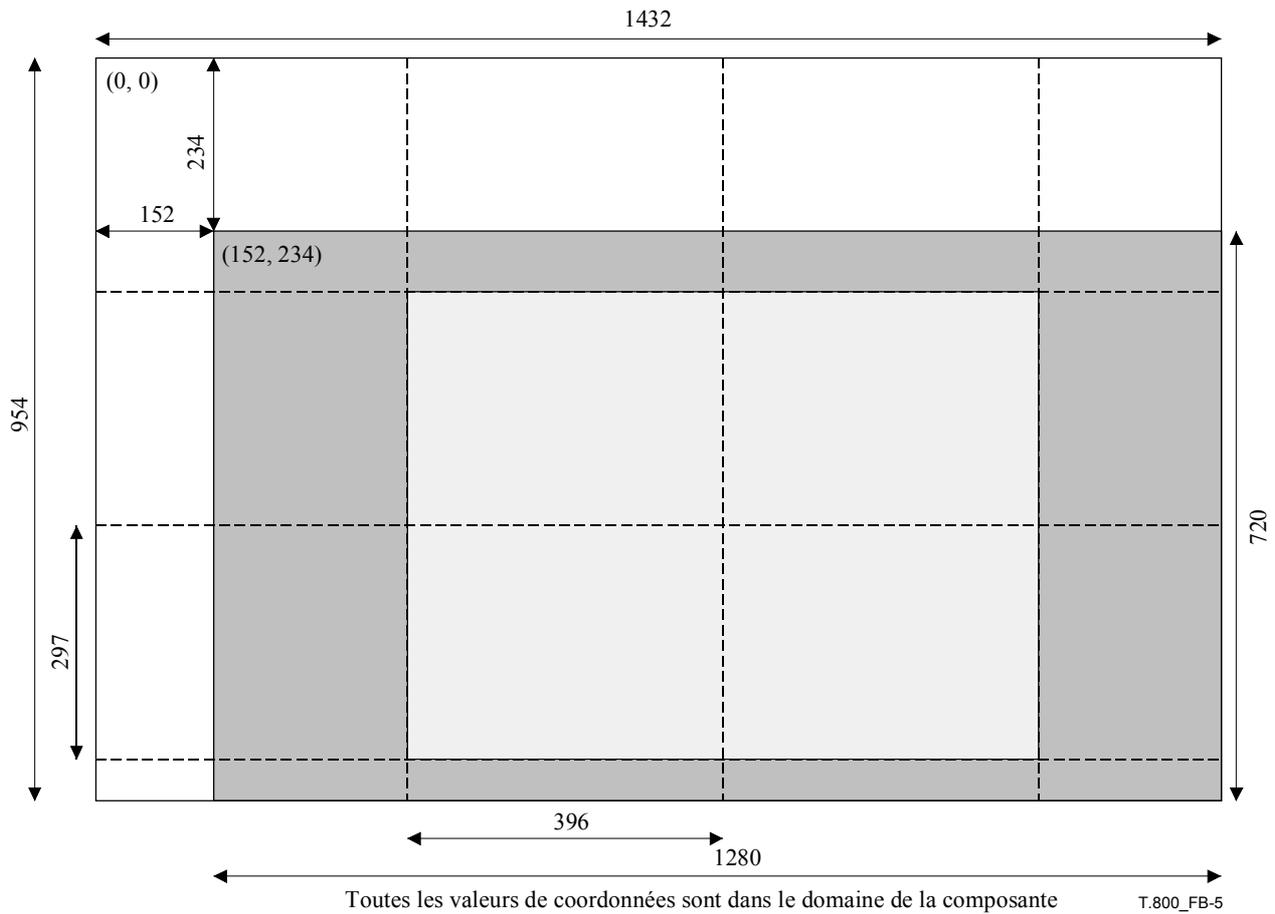
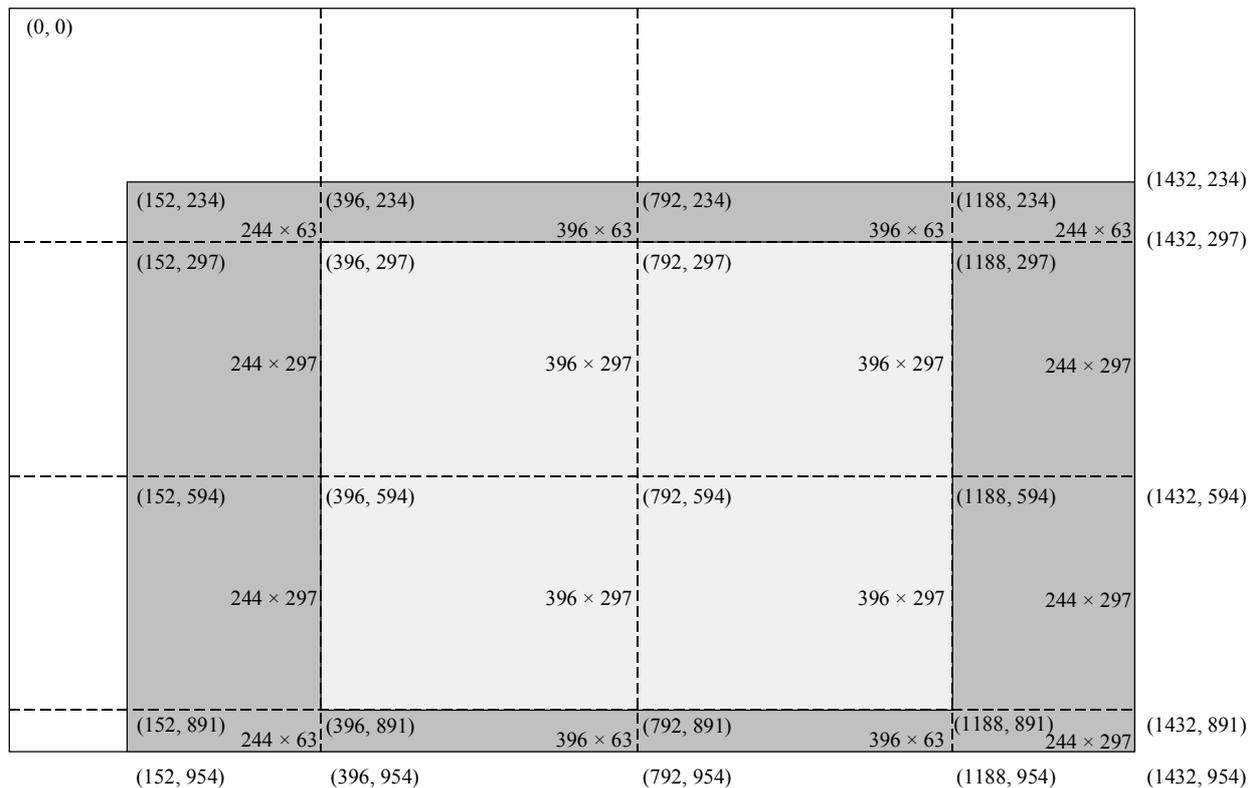


Figure B.5 – Exemple de grille de référence

Admettons que les dimensions de grille de référence (X_{siz} , Y_{siz}) soient (1432, 954). Dans cet exemple, l'image contiendra deux composantes (les indices de composante seront représentés par $i=0, 1$). Les facteurs de sous-échantillonnage XR_{siz}^i et YR_{siz}^i des deux composantes par rapport à la grille de référence seront $XR_{siz}^0 = YR_{siz}^0 = 1$ et $XR_{siz}^1 = YR_{siz}^1 = 2$. Le décalage d'image est fixé à $(XO_{siz}, YO_{siz}) = (152, 234)$. En fonction de ces paramètres, les longueurs des deux composantes d'image peuvent être déterminées d'après l'équation (B-1). Le coin supérieur gauche de la composante 0 se trouve à $(\lceil 152/1 \rceil, \lceil 234/1 \rceil) = (152, 234)$. Le coin inférieur droit de la composante 0 se trouve à $(\lceil 1432/1 \rceil - 1, \lceil 954/1 \rceil - 1) = (1431, 953)$. La taille réelle de la composante 0 est donc de 1280 échantillons en largeur par 720 échantillons en hauteur. Le coin supérieur gauche de la composante 1 se trouve à $(\lceil 152/2 \rceil, \lceil 234/2 \rceil) = (76, 117)$, alors que le coin inférieur droit de cette composante se trouve à $(\lceil 1432/2 \rceil - 1, \lceil 954/2 \rceil - 1) = (715, 476)$. La taille réelle de la composante 1 est donc de 640 échantillons en largeur par 360 échantillons en hauteur.

Les pavés sont choisis de façon à avoir un rapport hauteur/largeur de 4:3. Dans cet exemple, (XT_{siz}, YT_{siz}) sera mis à (396, 297) et les décalages de pavé (XTO_{siz}, YTO_{siz}) seront mis à (0, 0). Le nombre de pavés dans les sens X et Y est alors déterminé d'après l'équation (B-5) $numX_{tiles} = \lceil 1432/396 \rceil = 4$, $numY_{tiles} = \lceil 954/297 \rceil = 4$. Les composantes d'image repavées contiendront donc un total de $t = 16$ pavés, avec les indices de grille de pavé p et q dans l'étendue $0 \leq p, q < 4$. Il est alors possible de calculer les emplacements des pavés dans chaque composante d'image. A cette fin, les valeurs de tx_0 , tx_1 , ty_0 , et ty_1 sont déterminées d'après l'équation (B-7), l'équation (B-8), l'équation (B-9) et l'équation (B-10). Comme p et q partagent le même ensemble de valeurs admissibles, la notation '0:3' sera utilisée pour se rapporter à la séquence de valeurs $\{0, 1, 2, 3\}$, et la notation '*' sera utilisée pour indiquer que le résultat est valide pour toutes les valeurs admissibles. Les valeurs de tx_0 sont trouvées à $tx_0(0:3, *) = \{152, 396, 792, 1188\}$, et les valeurs de tx_1 sont données par $tx_1(*, 0:3) = \{396, 792, 1188, 1432\}$. Les valeurs de ty_0 sont $ty_0(*, 0:3) = \{234, 297, 594, 891\}$ et les valeurs de ty_1 sont $ty_1(*, 0:3) = \{297, 594, 891, 954\}$.



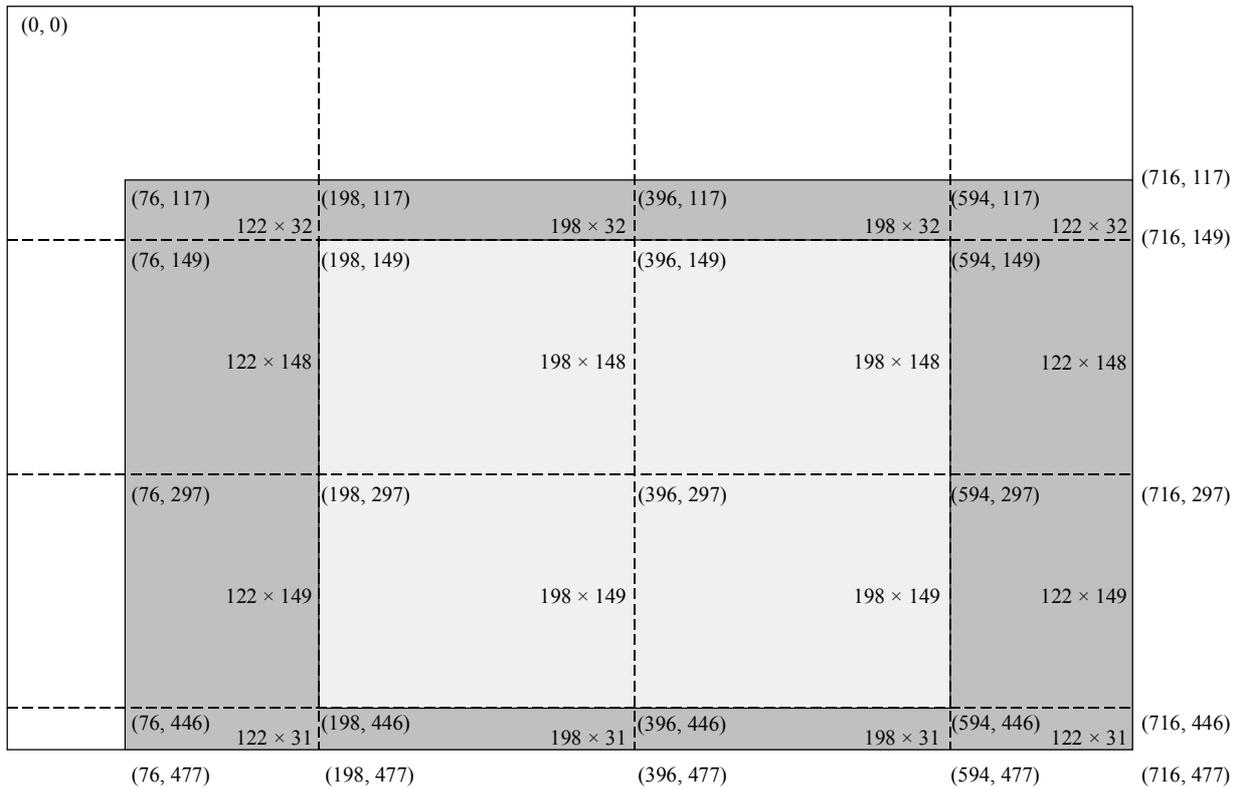
Pavage pour composante 0. Toutes les coordonnées de cette figure sont dans le domaine de la composante

T.800_FB-6

Figure B.6 – Exemple de dimensions et d'emplacements de pavé pour composante 0

Avec les valeurs de tx_0 , tx_1 , ty_0 , et ty_1 maintenant connues, les emplacements et tailles de tous les pavés peuvent être déterminés pour chacune des composantes. A cette fin, l'équation (B-12) est utilisée. Les emplacements et tailles applicables à la composante 0 sont représentés dans la Figure B.6, alors que la même information est fournie pour la composante 1 dans la Figure B.7. Relever plus particulièrement les pavés 'intérieurs' dans les figures [pavés (1, 1), (1, 2), (2, 1), et (2, 2)]. Ces pavés ne sont pas limités dimensionnellement par la zone d'image. Dans la composante 0, tous ces pavés ont la même taille. Cette régularité résulte du fait que les facteurs de sous-échantillonnage pour cette composante sont $(XR_{siz}^0, YR_{siz}^0) = (1, 1)$. Cependant, dans la composante 1, ces pavés ne sont pas tous de même taille parce que $(XR_{siz}^1, YR_{siz}^1) = (2, 2)$. Noter que les pavés (1, 1) et (2, 1) sont tous les deux de taille 198 par 148, alors que les pavés (1, 2) et (2, 2) sont tous les deux de taille 198 par 149. Cela montre que le nombre des échantillons dans les pavés intérieurs d'une composante peuvent varier selon la combinaison particulière de dimensions de pavé et de facteurs de sous-échantillonnage de composante.

Avec ces choix de grille de référence, de décalage d'image, de dimensions de pavé, et de décalage de pavé, l'image codée peut être recadrée directement dans la région intérieure recherchée. Les quatre pavés intérieurs issus de chaque composante peuvent être conservés et représenteront une image recadrée de dimensions de grille de référence (792, 594). Lorsqu'un tel recadrage est effectué, il n'est pas nécessaire de recoder les pavés, mais les valeurs de certains paramètres de la grille de référence doivent toujours changer. Les décalages d'image doivent toujours être mis aux coordonnées des emplacements de recadrage, de façon que $(XO_{siz}', YO_{siz}') = (396, 297)$ où (XO_{siz}', YO_{siz}') sont les décalages d'image de l'image recadrée. De même, les dimensions d'image doivent toujours être réglées de façon à refléter la taille recadrée $(Xs_{siz}', Ys_{siz}') = (1\ 188, 891)$ où (Xs_{siz}', Ys_{siz}') sont les longueurs de la grille de référence recadrée. Finalement, les décalages de pavé ne sont plus zéro et au contraire doivent toujours être mis à $(XTO_{siz}', YTO_{siz}') = (396, 297)$ où (XTO_{siz}', YTO_{siz}') sont les décalages de pavé de la grille de référence recadrée.



Pavage pour composante 1. Toutes les coordonnées de cette figure sont dans le domaine de la composante

T.800_FB-7

Figure B.7 – Exemple de dimensions et d'emplacements de pavé pour composante 1

B.5 Subdivision de composante-pavé transformée en niveaux de résolution et sous-bandes

Chaque composante-pavé est transformée en ondelettes avec N_L niveaux de décomposition comme expliqué dans l'Annexe F. Donc, il y a $N_L + 1$ niveaux de résolution distincts, indiqués $r = 0, 1, \dots, N_L$. Le niveau de résolution le plus bas, $r = 0$, est représenté par la bande N_L LL. En général, une version en résolution réduite d'une composante-pavé avec niveau de résolution, r , est la sous-bande n LL, où $n = N_L - r$. Le présent paragraphe décrit les dimensions de cette résolution réduite.

Les coordonnées d'une composante-pavé donnée par rapport à la grille de référence à niveau de résolution particulier, r , fournissent les coordonnées de l'échantillon supérieur gauche, (trx_0, try_0) et les coordonnées de l'échantillon inférieur droit, $(trx_1 - 1, try_1 - 1)$, où:

$$trx_0 = \left\lceil \frac{tcx_0}{2^{N_L - r}} \right\rceil \quad try_0 = \left\lceil \frac{tcy_0}{2^{N_L - r}} \right\rceil \quad trx_1 = \left\lceil \frac{tcx_1}{2^{N_L - r}} \right\rceil \quad try_1 = \left\lceil \frac{tcy_1}{2^{N_L - r}} \right\rceil \quad (\text{B-14})$$

De façon analogue, les coordonnées de pavé peuvent être affectées à toute sous-bande particulière, b , fournissant les coordonnées d'échantillon du coin supérieur gauche (tbx_0, tby_0) et les coordonnées d'échantillon du coin inférieur droit $(tbx_1 - 1, tby_1 - 1)$ où:

$$tbx_0 = \left\lceil \frac{tcx_0 - (2^{n_b} - 1) \cdot xo_b}{2^{n_b}} \right\rceil \quad tby_0 = \left\lceil \frac{tcy_0 - (2^{n_b} - 1) \cdot yo_b}{2^{n_b}} \right\rceil$$

$$tbx_1 = \left\lceil \frac{tcx_1 - (2^{n_b} - 1) \cdot xo_b}{2^{n_b}} \right\rceil \quad tby_1 = \left\lceil \frac{tcy_1 - (2^{n_b} - 1) \cdot yo_b}{2^{n_b}} \right\rceil \quad (\text{B-15})$$

où n_b est le niveau de décomposition associé à la sous-bande b , comme exposé dans l'Annexe F, et où les grandeurs (xo_b, yo_b) sont données par le Tableau B.1.

Tableau B.1 – Grandeurs (x_{o_b}, y_{o_b}) pour sous-bande b

Sous-bande	x_{o_b}	y_{o_b}
n_bLL	0	0
n_bHL (horizontalement passe-haut)	1	0
n_bLH (verticalement passe-haut)	0	1
n_bHH	1	1

NOTE – Chaque sous-bande est différente, comme indiqué au § B.1.

Pour chaque sous-bande, ces coordonnées définissent des limites de pavé dans des domaines de sous-bande distincts. Par ailleurs, la largeur de chaque sous-bande dans son domaine (au niveau de décomposition actuel) est donnée par $tbx_1 - tbx_0$, et la hauteur est donnée par $tby_1 - tby_0$.

B.6 Subdivision des niveaux de résolution en districts

Soit une composante particulière de pavé et un niveau de résolution dont les coordonnées limites d'échantillon dans le domaine d'image à résolution réduite sont (trx_0, try_0) et $(trx_1 - 1, try_1 - 1)$, comme déjà décrit. La Figure B.8 montre la subdivision de ce niveau de résolution de composante-pavé en districts. Le district est ancré à l'emplacement $(0, 0)$, de façon que le coin supérieur gauche de tout district se trouvant dans la partition soit situé à des multiples entiers de $(2^{PPx}, 2^{PPy})$ où PPx et PPy sont signalés dans le segment marqueur COD ou COC (voir § A.6.1 et A.6.2). Les valeurs PPx et PPy peuvent être différentes pour chaque composante-pavé et pour chaque niveau de résolution. Les valeurs PPx et PPy doivent toujours être au moins 1 pour tous les niveaux de résolution sauf $r = 0$ où elles sont autorisées à être zéro.

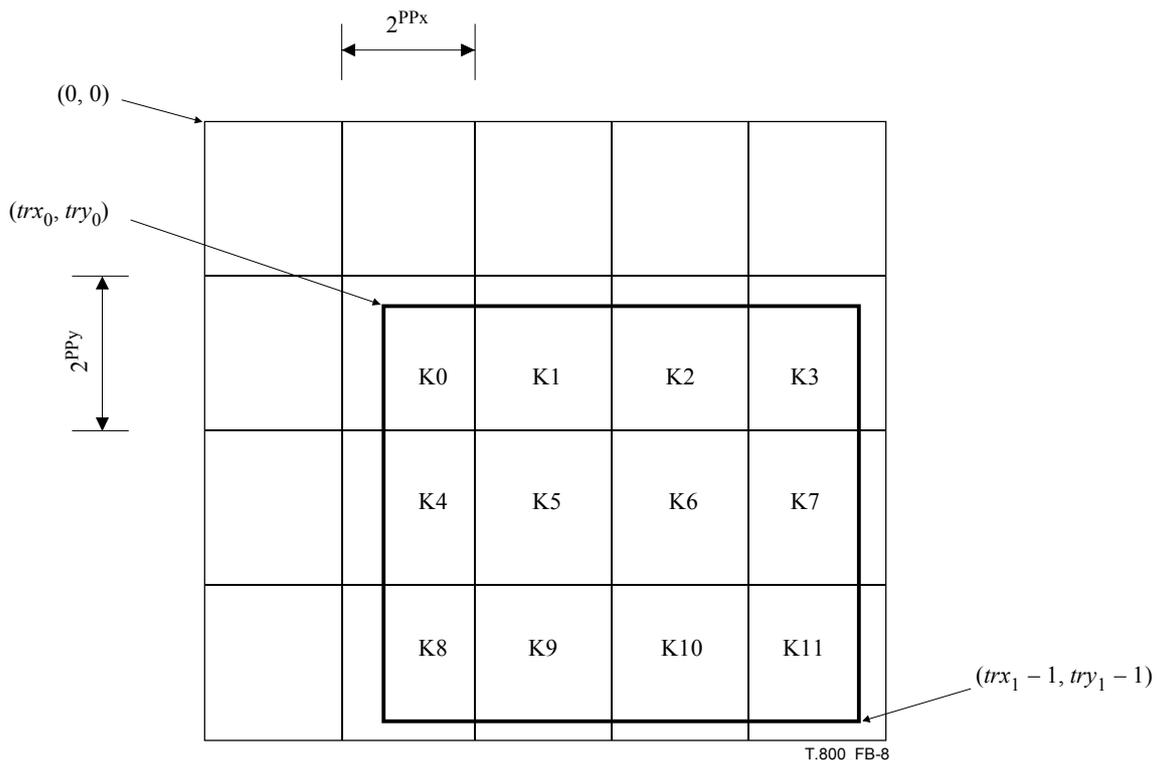


Figure B.8 – Districts d'une résolution réduite

Le nombre de districts qui couvrent la composante-pavé au niveau de résolution, r , est donné par:

$$numprecinctswide = \begin{cases} \left\lceil \frac{trx_1}{2^{PPx}} \right\rceil - \left\lfloor \frac{trx_0}{2^{PPx}} \right\rfloor & trx_1 > trx_0 \\ 0 & trx_1 = trx_0 \end{cases} \quad numprecinctshigh = \begin{cases} \left\lceil \frac{try_1}{2^{PPy}} \right\rceil - \left\lfloor \frac{try_0}{2^{PPy}} \right\rfloor & try_1 > try_0 \\ 0 & try_1 = try_0 \end{cases} \quad (B-16)$$

Même si l'équation (B-16) indique que les deux valeurs *numprecinctswide* et *numprecinctshigh* sont différentes de zéro, certains, ou la totalité, des districts peuvent être encore vides comme expliqué ci-dessous. L'index de district s'étend de 0 à *numprecincts* - 1 où *numprecincts* = *numprecinctswide* * *numprecinctshigh* dans l'ordre du balayage (voir Figure B.8). Cet index est utilisé afin de déterminer l'ordre d'apparition, dans le flux codé, de paquets correspondant à chaque district, comme expliqué au § B.12.

Il peut se produire que *numprecincts* soit égal à 0 pour une composante-pavé et un niveau de résolution particuliers. Lorsque cela se produit, il n'y a aucun paquet pour cette composante-pavé ou niveau de résolution.

Il peut se produire qu'un district soit vide, indiquant qu'aucun coefficient de sous-bande issu du niveau de résolution applicable ne contribue en fait au district. Cela peut se produire, par exemple, au coin inférieur droit d'une composante-pavé en raison de l'échantillonnage par rapport à la grille de référence. Lorsque cela se produit, chaque paquet correspondant à ce district doit continuer à apparaître dans le flux codé (voir § B.9).

B.7 Subdivision des sous-bandes en blocs codés

Les sous-bandes sont partitionnées en blocs codés rectangulaires aux fins de la modélisation en coefficients et du codage. La longueur de chaque bloc codé est déterminée d'après deux paramètres, *xcb* et *ycb*, qui sont signalés dans les segments marqueurs COD ou COC (voir § A.6.1 et A.6.2). La longueur de bloc codé est le même à partir de tous les niveaux de résolution. Cependant, à chaque niveau de résolution, la longueur de bloc codé est limitée par les dimensions de district. La longueur de bloc codé pour chaque sous-bande à un niveau de résolution particulier est déterminée comme étant $2^{xcb'}$ par $2^{ycb'}$ où

$$xcb' = \begin{cases} \min(xcb, PPx - 1), & \text{pour } r > 0 \\ \min(xcb, PPx), & \text{pour } r = 0 \end{cases} \quad (\text{B-17})$$

et:

$$ycb' = \begin{cases} \min(ycb, PPy - 1), & \text{pour } r > 0 \\ \min(ycb, PPy), & \text{pour } r = 0 \end{cases} \quad (\text{B-18})$$

Ces équations reflètent le fait que la longueur de bloc codé est contrainte à la fois par les dimensions de district et par la longueur de bloc codé, dont les paramètres, *xcb* et *ycb*, sont identiques pour toutes les sous-bandes dans la composante-pavé. Comme le district, la partition de bloc codé est ancrée à (0, 0), comme illustré dans la Figure B.9. Donc, toutes les premières rangées de blocs codés dans la partition de bloc codé sont situées à $y = m2^{ycb'}$ et toutes les premières colonnes de blocs codés sont situées à $x = n2^{xcb'}$, où *m* et *n* sont des entiers.

NOTE – Les blocs codés contenus dans la partition peuvent s'étendre au-delà des limites des coefficients de sous-bande. Lorsque cela se produit, seuls les coefficients se trouvant dans la sous-bande sont codés au moyen de la méthode décrite à l'Annexe D. La première zone codée au moyen de cette méthode correspond aux quatre premières rangées de coefficients de sous-bande dans le bloc codé ou à autant de telles rangées qu'il y en a.

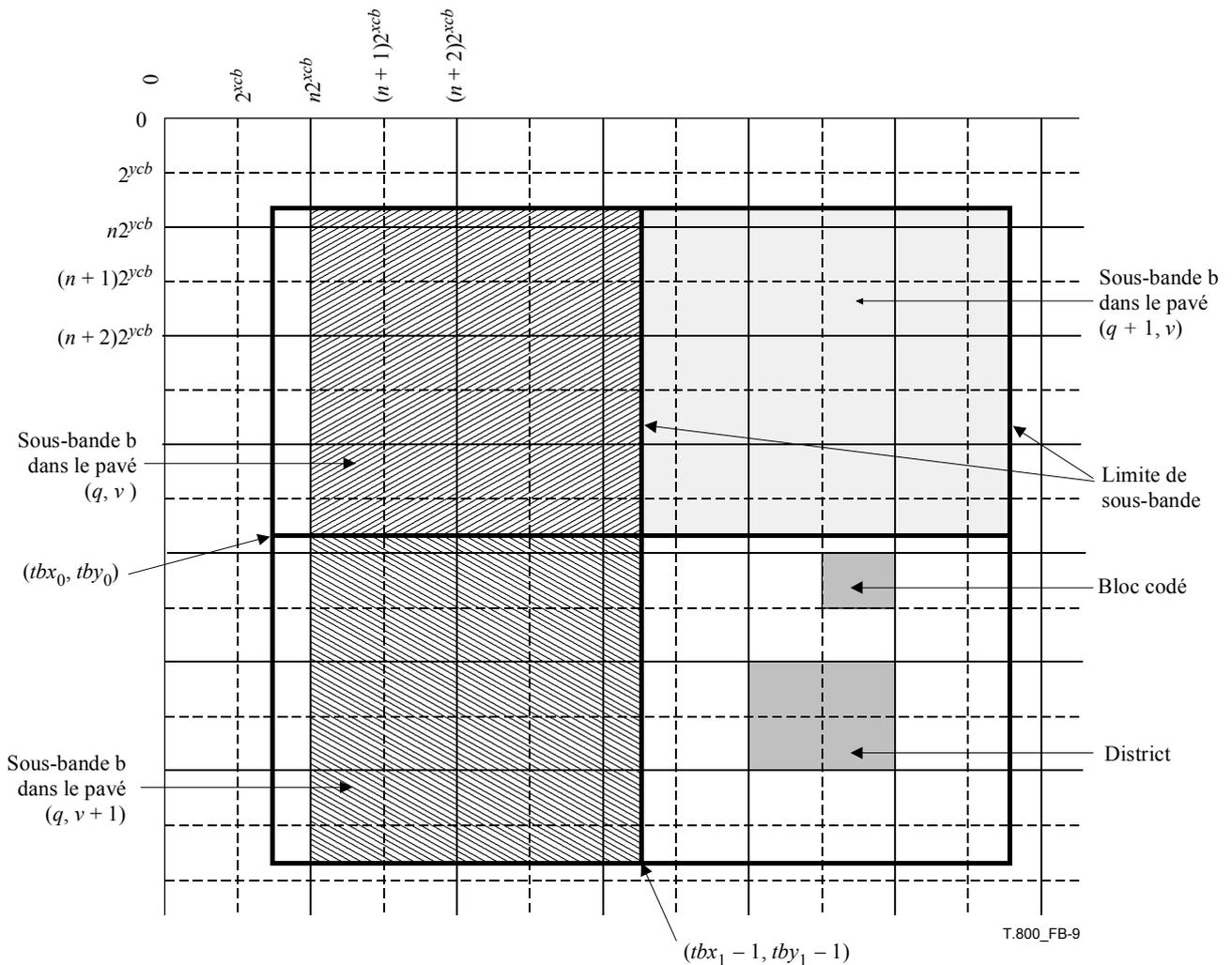


Figure B.9 – Blocs codés et districts dans la sous-bande b , issus de quatre pavés différents

B.8 Couches

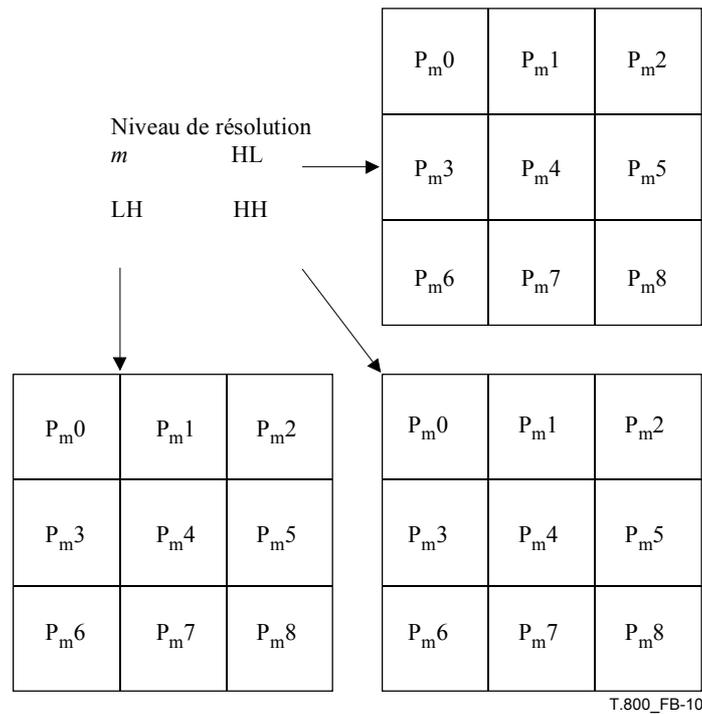
Les données d'image comprimées de chaque bloc codé sont réparties entre une ou plusieurs couches contenues dans le flux codé. Chaque couche se compose d'un certain nombre de passes de codage consécutives de plans binaires issus de chaque bloc codé dans le pavé, y compris toutes les sous-bandes de toutes les composantes pour ce pavé. Le nombre de passes de codage dans la couche peut varier de bloc codé à bloc codé et peut être aussi petit que zéro pour un seul ou la totalité des blocs codés. Le nombre de couches pour le pavé est signalé dans le segment marqueur COD (voir § A.6.1).

Pour un certain bloc codé, la première passe de codage, si elle existe dans la couche n , est celle qui suit immédiatement la dernière passe de codage pour le bloc codé dans la couche $n - 1$, si elle existe.

NOTE 1 – Chaque couche améliore successivement et de manière monotone la qualité d'image.

Les couches sont indexées de 0 à $L - 1$, où L est le nombre de couches dans chaque composante-pavé.

NOTE 2 – La Figure B.10 montre un exemple de neuf districts de niveau de résolution m . Le Tableau B.2 montre la formation de couche.



T.800_FB-10

Figure B.10 – Diagramme des districts d'un niveau de résolution d'une composante

Tableau B.2 – Exemple de formation de couche (une seule composante est montrée)

Niveau de résolution	0			...	m				...	N_L		
District	P_{00}	P_{01}	P_{m0}	P_{m1}	...	P_{m8}	...	P_{Nl0}	P_{Nl1}	...
Couche 0	Paquet 0	Paquet 0	Paquet 0	Paquet 0	...	Paquet 0	...	Paquet 0	Paquet 0	...
Couche 1	Paquet 1	Paquet 1	Paquet 1	Paquet 1	...	Paquet 1	...	Paquet 1	Paquet 1	...
...

Les modules élémentaires des couches sont les paquets, qui sont créés d'après les données d'image comprimées de bloc codé d'après les districts de différents niveaux de résolution (pour une composante-pavé donnée).

B.9 Paquets

Toutes les données d'image comprimées représentant un pavé, une couche, une composante, un niveau de résolution ou un district spécifique apparaissent dans le flux codé sous la forme d'un segment contigu appelé *paquet*. Les données de paquet sont alignées sur des limites de 8 bits (un octet).

Comme défini au § F.3.1, le niveau de résolution $r = 0$ contient les coefficients de sous-bande d'après la bande $N_L LL$, où N_L est le nombre de niveaux de décomposition. Chaque niveau de résolution subséquent, $r > 0$, contient les coefficients de sous-bande d'après les sous-bandes nHL , nLH , et nHH , comme défini dans l'Annexe F, où $n = N_L - r + 1$. Il y a $N_L + 1$ niveaux de résolution pour une composante-pavé avec N_L niveaux de décomposition.

Les données d'image comprimées dans un paquet sont ordonnées de façon que la contribution des sous-bandes LL , HL , LH et HH apparaisse dans cet ordre. Cet ordre de sous-bande est identique à celui qui est défini au § F.3.1. Dans chaque sous-bande, les contributions de bloc codé apparaissent dans l'ordre du balayage, confinées aux limites établies par le district applicable. Le niveau de résolution $r = 0$ contient seulement la bande $N_L LL$ et les niveaux de résolution $r > 0$ contiennent seulement les bandes HL , LH et HH . Seuls les blocs codés qui contiennent des échantillons issus de la sous-bande applicable, confinés au district, ont une représentation dans le paquet.

NOTE 1 – La Figure B.11 montre l'organisation des blocs codés dans un district qui forme un paquet. Le Tableau B.3 montre un exemple de passes de codage de bloc qui forment des paquets. Dans le Tableau B.3, les variables a, b, et c sont des passes de codage de bloc codé où a = passe de propagation de signification, b = passe de raffinement de grandeur, et c = passe de nettoyage (voir Annexe D).

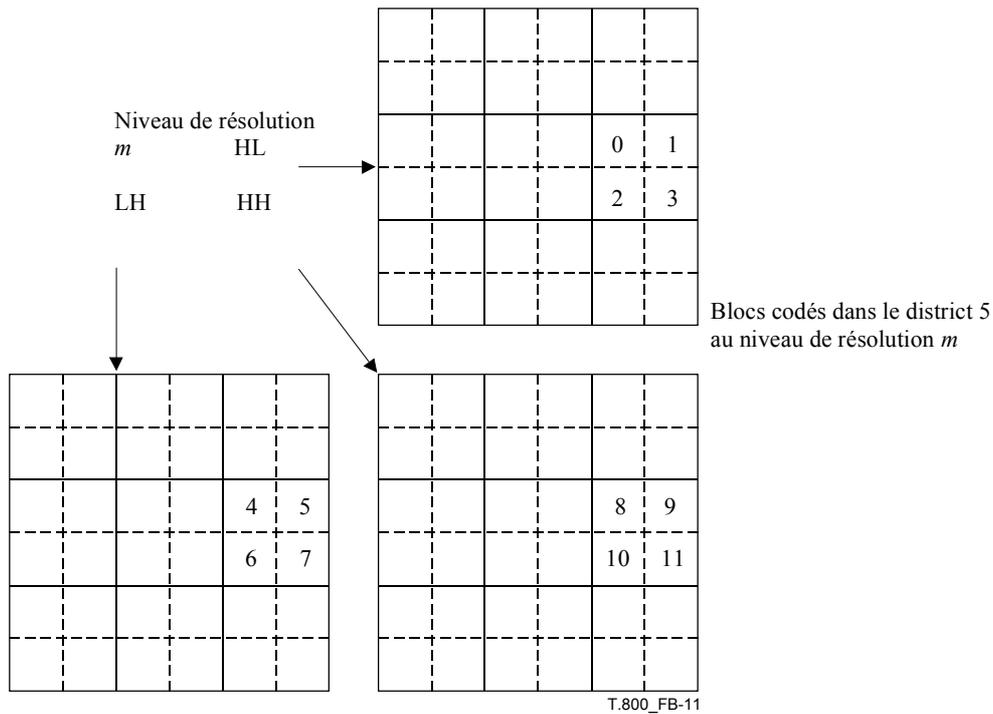


Figure B.11 – Diagramme des blocs codés dans des districts à un niveau de résolution

Tableau B.3 – Exemple de formation de paquet

	Bloc codé 0	Bloc codé 1	Bloc codé 2	...	Bloc codé 10	Bloc codé 11	
MSB	c	0	0	...	c	0	Paquet 0
	a	0	0	...	a	0	
	b	0	0	...	b	0	
	c	c	0	...	c	0	
...	a	a	0	...	a	0	Paquet 1
	b	b	0	...	b	0	
	c	c	c	...	c	c	
LSB	etc.
	a	a	a	...	a	a	
	b	b	b	...	b	b	
	c	c	c	...	c	c	

Les données de paquet sont introduites par un en-tête de paquet dont la syntaxe est décrite au § B.10. Elles sont suivies par un corps de paquet contenant les octets codés réellement apportés par chacun des blocs codés applicables. L'ordre défini ci-dessus est suivi lors de la construction de l'en-tête comme du corps de paquet.

Comme décrit au § B.6, il peut se produire qu'un district ne contienne aucun bloc codé en provenance d'une des sous-bandes à un certain niveau de résolution. Lorsque cela se produit, tous les paquets correspondant à ce district doivent apparaître dans le flux codé comme étant vides, conformément à l'en-tête de paquet décrit au § B.10.

NOTE 2 – Même lorsqu'un district contient des blocs codés utiles, un codeur peut choisir de n'inclure aucune passe de codage que ce soit dans le paquet correspondant à une couche donnée. Dans ce cas, un paquet vide doit continuer à apparaître dans le flux codé.

B.10 Codage des informations d'en-tête de paquet

Les paquets ont des en-têtes contenant les informations suivantes:

- paquet de longueur nulle;
- inclusion de bloc codé ;

- informations de plan binaire zéro;
- nombre de passes de codage;
- longueur des données d'image comprimées d'un certain bloc codé.

Deux éléments dans l'en-tête sont codés selon un schéma appelé *arbre de résolution*, qui est décrit ci-dessous. Les bits de l'en-tête de paquet sont compactés en un nombre entier d'octets par la routine de bourrage binaire décrite au § B.10.1.

Les en-têtes de paquet apparaissent dans le flux codé immédiatement avant les données de paquet, sauf si un des segments marqueurs PPM ou PPT a été utilisé. Si le segment marqueur PPM est utilisé, tous les en-têtes de paquet sont repositionnés dans l'en-tête principal (voir § A.7.4). Si le marqueur PPM n'est pas utilisé, alors un segment marqueur PPT peut être utilisé. Dans ce cas, tous les en-têtes de paquet contenus dans ce pavé sont repositionnés dans les en-têtes d'élément de pavé (voir § A.7.5).

B.10.1 Routine de bourrage binaire

Les éléments binaires sont compactés en octets du bit MSB au bit LSB. Une fois qu'un octet complet est assemblé, il est adjoint à l'en-tête de paquet. Si la valeur de l'octet est 0xFF, le prochain octet comporte un bit zéro supplémentaire inséré dans le bit MSB. Une fois que tous les bits de l'en-tête de paquet ont été assemblés, le dernier octet est compacté jusqu'à la limite d'octet et émis. Le dernier octet dans l'en-tête de paquet ne doit pas être une valeur 0xFF (donc l'unique bit zéro inséré après un octet avec 0xFF doit toujours être inclus même si la valeur 0xFF aurait sinon été dans le dernier octet).

B.10.2 Arbres de résolution

Un arbre de résolution est un mode de représentation hiérarchique d'une matrice bidimensionnelle d'entiers non négatifs. Il crée successivement des niveaux de résolution réduits de cette matrice bidimensionnelle, formant ainsi un arbre. A chaque nœud de cet arbre, on enregistre l'entier minimal des (jusqu'à quatre) nœuds hiérarchiquement inférieurs. La Figure B.12 montre un exemple de cette représentation. La notation, $q_i(m, n)$, est la valeur du nœud qui est le m^e à partir de la gauche et le n^i ème à partir du haut, au i^e niveau. Le niveau 0 est le plus bas de l'arbre de résolution; il contient le nœud sommital.

1 $q_3(0, 0)$	3 $q_3(1, 0)$	2 $q_3(2, 0)$	3	2	3
2	2	1	4	3	2
2	2	2	2	1	2

a) Série tabulaire de nombres originale – Niveau 3

1 $q_2(0, 0)$	1 $q_2(1, 0)$	2
2	2	1

b) Minimum de quatre nœuds (ou moins) – Niveau 2

1 $q_1(0, 0)$	1
------------------	---

c) Minimum de quatre nœuds (ou moins) – Niveau 1

1 $q_0(0, 0)$

d) Minimum de quatre nœuds (ou moins) – Niveau 0

T.800_FB-12

Figure B.12 – Exemple d'une représentation en arbre de résolution

Les éléments de la matrice sont traversés dans l'ordre du balayage pour le codage. Le codage est la réponse à une série de questions. Chaque nœud possède une valeur actuelle associée, qui est initialisée à zéro (le minimum). Un bit 0 dans l'arbre de résolution signifie que le minimum (ou la valeur dans le cas du niveau le plus élevé) est plus grand que la valeur actuelle et un bit 1 signifie que le minimum (ou la valeur dans le cas du niveau le plus élevé) est égal à la valeur actuelle. Pour chaque bit 0 contigu dans l'arbre de résolution, la valeur actuelle est augmentée de un. Les nœuds situés

aux niveaux supérieurs ne peuvent pas être codés tant que les valeurs des nœuds inférieurs ne sont pas déterminées (c'est-à-dire qu'un bit 1 est codé). Le nœud sommital au niveau 0 (le plus bas niveau) est interrogé en premier. Le prochain nœud, correspondant au niveau 1, est alors interrogé, et ainsi de suite.

Seules les informations nécessaires au bloc codé actuel sont mémorisées au point actuel dans l'en-tête de paquet. Le décodage de bits est arrêté lorsque des informations suffisantes ont été obtenues. De même, la nature hiérarchique des arbres de résolution signifie que les réponses à de nombreuses questions seront déjà formées lors du codage des blocs codés et/ou couches adjacents. Ces informations ne sont pas codées de nouveau. Il existe donc une relation déterministe avec les informations contenues dans l'en-tête de paquet.

NOTE – Par exemple, dans la Figure B.12, le codage du nombre situé à $q_3(0,0)$ sera 01111. Les deux bits, 01, impliquent que le nœud sommital à $q_0(0,0)$ est supérieur à zéro et est, en fait un. Le troisième bit, 1, implique que le nœud à $q_1(0,0)$ est également un. Le quatrième bit, 1, implique que le nœud à $q_2(0,0)$ est également un. Et le bit final, 1, implique que le nœud cible à $q_3(0,0)$ est également un. Pour décoder le prochain nœud $q_3(1,0)$, les nœuds à $q_0(0,0)$, $q_1(0,0)$ et $q_2(0,0)$ sont déjà connus. Donc, les bits codés sont 001, où le zéro indique que le nœud à $q_3(1,0)$ est supérieur à 1, la seconde zéro indique qu'il est supérieur à 2, et bit un implique que la valeur est 3. Maintenant que $q_3(0,0)$ et $q_3(1,0)$ sont connus, les bits de code pour $q_3(2,0)$ seront 101. Le premier 1 indique que $q_2(1,0)$ est un. Les bits 01 suivants indiquent ensuite que $q_3(2,0)$ est 2. Ce processus continue pour toute la matrice représentée dans la Figure B.12a.

B.10.3 Paquet de longueur nulle

Le premier bit dans l'en-tête de paquet indique si le paquet possède une longueur de zéro (paquet vide). La valeur 0 indique une longueur nulle; aucun bloc codé n'est inclus dans ce cas. La valeur 1 indique une longueur non nulle; ce cas est considéré exclusivement ci-après.

NOTE – Si un paquet est marqué comme étant vide, alors aucun bloc codé ne peut contribuer à la couche correspondante. Si le prochain paquet n'est pas marqué comme vide, l'inclusion d'informations de bloc codé (définies au § B.10.4) pour la précédente couche avec le bit de paquet vide activé doit être incluse. L'inclusion d'informations de bloc codé pour des blocs codés qui n'ont pas encore été inclus dans un quelconque paquet est codée au moyen d'un arbre de résolution dont les entrées sont initialisées avec le numéro de la première couche à laquelle le bloc codé contribue. Donc, l'arbre de résolution comportera des informations redondantes indiquant si le bloc codé contribue ou non aussi bien à la couche actuelle et qu'à la couche dans laquelle le paquet était marqué comme vide.

B.10.4 Inclusion de bloc codé

Les informations relatives au fait de savoir si d'éventuelles données d'image comprimées issues de chaque bloc codé sont ou non incluses dans le paquet sont signalées de l'une ou de l'autre de deux façons différentes selon que le même bloc codé a ou non été déjà inclus dans un précédent paquet (c'est-à-dire dans une précédente couche).

Pour les blocs codés qui ont été inclus dans un précédent paquet, un seul bit est utilisé pour représenter les informations, où un 1 signifie que le bloc codé est inclus dans cette couche et un 0 signifie qu'il ne l'est pas.

Pour les blocs codés qui n'ont pas été déjà inclus dans un paquet quelconque, ces informations sont signalées avec un code distinct d'arbre de résolution pour chaque district se trouvant confiné à une sous-bande. Les valeurs contenues dans cet arbre de résolution sont le numéro de la couche dans laquelle le bloc codé actuel est inclus en premier. Bien que la séquence exacte des bits qui représentent l'arbre de résolution permettant l'inclusion apparaisse dans le flux binaire, seuls les bits nécessaires afin de déterminer si le bloc codé est inclus sont placés dans l'en-tête de paquet. Si une partie de l'arbre de résolution est déjà connue d'après de précédents blocs codés ou de précédentes couches, cette partie n'est pas répétée. De même, seule la partie de l'arbre de résolution qui est nécessaire afin de déterminer l'inclusion dans la couche actuelle est présente. Si un bloc codé n'est pas inclus avant une couche ultérieure, alors seul un arbre de résolution partiel est inclus à ce point dans le flux binaire.

B.10.5 Informations de plan binaire zéro

Si un bloc codé est inclus pour la première fois, l'en-tête de paquet contient des informations indiquant le nombre réel de plans binaires utilisés pour représenter les coefficients issus du bloc codé. Le nombre maximal de plans binaires disponibles pour la représentation des coefficients dans une sous-bande quelconque, b , est donné par M_b comme défini dans l'équation (E-2). En général, cependant, le nombre de plans binaires réels pour lesquels des passes de codage sont produites est $M_b - P$, où le nombre de plans binaires de plus fort poids absents, P , peut varier de bloc codé à bloc codé; ces plans binaires absents sont tous censés être zéro. La valeur de P est codée dans l'en-tête de paquet avec un arbre de résolution distinct pour chaque district, de la même façon que l'inclusion d'informations de bloc codé.

B.10.6 Nombre de passes de codage

Le nombre de passes de codage inclus dans ce paquet à partir de chaque bloc codé est indiqué dans l'en-tête de paquet au moyen des mots de code représentés dans le Tableau B.4. Ce tableau permet la signalisation d'un maximum de 164 passes de codage.

Tableau B.4 – Mots de code pour le nombre de passes de codage sur chaque bloc codé

Nombre de passes de codage	Mot de code dans un en-tête de paquet
1	0
2	10
3	1100
4	1101
5	1110
6 à 36	1111 0000 0 à 1111 1111 0
37 à 164	1111 11111 0000 000 à 1111 11111 1111 111

NOTE – Comme la valeur de M_b est limitée à une valeur maximale de 37 par les contraintes imposées par la syntaxe des segments marqueurs QCD et QCC (voir § A.6.4, A.6.5, et équation (E-4)), il n'est pas possible d'employer plus que 109 passes de codage avec l'algorithme de codage de bloc codé décrit à l'Annexe D.

B.10.7 Longueur des données d'image comprimées d'après un certain bloc codé

L'en-tête de paquet indique le nombre d'octets apportés par chaque bloc codé inclus. La séquence d'octets en fait incluse pour tout bloc codé ne doit jamais se terminer par une valeur 0xFF. Donc, si une valeur 0xFF apparaissait à la fin d'une contribution de bloc codé à un certain paquet, cette valeur 0xFF pourrait être transférée sans risque vers le paquet suivant qui contient des contributions issues du bloc codé, ou être négligée s'il n'y a aucun paquet de ce type. L'algorithme de calcul de longueur de passe de codage, décrit à titre d'exemple dans l'Annexe D, garantit qu'aucune passe de codage ne sera jamais considérée comme se terminant par une valeur 0xFF.

NOTE – Cela n'est pas, en fait, une exigence gênante, car les valeurs 0xFF sont toujours synthétisées selon les nécessités par le codeur arithmétique décrit à l'Annexe C.

Lors de la signalisation du nombre d'octets apportés par le bloc codé, il y a deux cas: la contribution du bloc codé contient un seul segment de mot de code; ou la contribution du bloc codé contient de multiples segments de mot de code. De multiples segments de mot de code apparaissent lorsqu'une terminaison apparaît entre des passes de codage incluses dans le paquet, comme représenté dans les Tableaux D.8 et D.9.

B.10.7.1 Segment de mot de code unique

Un segment de mot de code est le nombre d'octets apportés à un paquet par un bloc codé. La longueur d'un segment de mot de code est représentée par un nombre binaire de longueur:

$$bits = Lblock + \lfloor \log_2(\text{coding passes added}) \rfloor \quad (\text{B-19})$$

où $Lblock$ est une variable d'état de bloc codé. Un bloc $Lblock$ distinct est utilisé pour chaque bloc codé contenu dans le district.

La valeur de $Lblock$ est initialement mise à trois. Le nombre d'octets apportés par chaque bloc codé est précédé par des bits de signalisation qui augmentent la valeur de $Lblock$ selon les besoins. Un bit de signalisation de valeur zéro indique que la valeur actuelle du bloc $Lblock$ est suffisante. S'il y a k valeurs "1" suivies par un zéro, la valeur de $Lblock$ est augmentée de k . Alors que $Lblock$ ne peut qu'augmenter, le nombre de bits utilisés afin de signaler la longueur de la contribution du bloc codé peut augmenter ou diminuer selon le nombre de passes de codage incluses.

NOTE 1 – Par exemple, considérons que dans des couches successives un bloc codé possède 6 octets, 31 octets, 44 octets et 134 octets. Supposons en outre que le nombre de passes de codage est de 1, 9, 2 et 5. Les codes respectifs seront alors de 0 110 (0 pour délimiter et 110 = 6), 0011111 (0 pour délimiter, $\log_2 9 = 3$ bits pour les 9 passes de codage, 011111 = 31), 11 0 101100 (110 ajoute deux bits à $Lblock$, $\log_2 2 = 1$, 101100 = 44), et 10 10000110 (10 ajoute un bit à $Lblock$, $\log_2 5 = 2$, 10000110 = 134).

NOTE 2 – Il n'est pas prescrit d'utiliser le nombre minimal de bits pour signaler la longueur (tout nombre est valide).

B.10.7.2 Multiples segments de mot de code

Soit T l'ensemble des indices de passes de codage terminées qui sont incluses pour le bloc codé contenu dans le paquet comme indiqué dans les Tableaux D.8 et D.9. Si la passe de codage finale d'index incluse dans le paquet ne fait pas partie de T , alors elle est ajoutée à T . Soit $n_1 < \dots < n_K$ les indices contenus dans l'ensemble T . K longueurs sont signalées en séquence contiguë, chaque longueur utilisant le mécanisme décrit au § B.10.7.1. La première longueur est le nombre d'octets depuis le début de la contribution de bloc codé dans ce paquet jusqu'à la fin de la passe de codage n_1 . Le nombre de passes de codage ajoutées aux fins de l'équation (B-19) est le nombre de passes contenu dans le paquet

jusqu'à n_1 . La seconde longueur est le nombre d'octets depuis la fin de la passe de codage, n_1 , jusqu'à la fin de la passe de codage, n_2 . Le nombre de passes de codage ajoutées aux fins de l'équation (B-19) est $n_2 - n_1$. Cette procédure est répétée pour toutes les longueurs K .

NOTE – Soit l'évitement sélectif du codage arithmétique (voir § D.6). Considérons que les passes incluses dans un paquet pour un certain bloc codé sont la passe de nettoyage de plan binaire numéro 4 jusqu'à la passe de propagation de signification de plan binaire numéro 6 (voir Tableau D.9). Ces passes sont indexées comme suit: {0, 1, 2, 3, 4} et les longueurs sont données par {6, 31, 44, 134, 192} respectivement. Alors l'ensemble $T = \{0, 2, 3, 4\}$ et $K = 4$ longueurs sont signalées. L'ensemble des longueurs à signaler est {6, 75, 134, 192} et le nombre correspondant des passes de codage qui sont ajoutées est {1, 2, 1, 1}. Une séquence binaire codée valide est 11 1110 (*Lblock* augmenté jusqu'à 8), 0000 0110 ($\log_2 1 = 0$, 8 bits utilisés pour coder la longueur de 6), 0 0100 1011 ($\log_2 2 = 1$, 9 bits utilisés pour coder la longueur de 75), 1000 0110 ($1 \log_2 1 = 0$, 8 bits utilisés pour coder la longueur de 134), et 1100 0000 ($\log_2 1 = 0$, 8 bits utilisés pour coder la longueur de 192). Noter que la valeur de *Lblock* n'est incrémentée qu'au début de la séquence.

B.10.8 Ordre des informations dans l'en-tête de paquet

Ce qui suit décrit l'ordre des informations d'en-tête de paquet pour un paquet d'une couche, d'une composante-pavé, d'un niveau de résolution ou d'un district spécifique.

bit pour paquet de longueur nulle ou non nulle

pour chaque sous-bande (LL ou HL, LH et HH)

pour tous blocs codés dans cette sous-bande confinée au district applicable, dans l'ordre du balayage

bits d'inclusion de bloc codé (si non déjà inclus alors arbre de résolution, sinon un bit)

si bloc codé inclus

si première instance de bloc codé

informations sur plans binaires zéro

nombre de passes de codage incluses

indicateur d'augmentation de longueur de bloc codé (*Lblock*)

pour chaque segment de mot de code

longueur de segment de mot de code

L'en-tête de paquet peut être immédiatement suivi par le marqueur EPH comme décrit au § A.8.2. Le marqueur EPH peut apparaître sans qu'il soit nécessaire de tenir compte du fait que le paquet contient d'éventuelles contributions de bloc codé. Si l'en-tête de paquet apparaît dans un segment marqueur PPM ou PPT, le marqueur EPH (si utilisé) doit toujours apparaître en même temps que l'en-tête de paquet.

NOTE – La Figure B.13 et le Tableau B.5 montrent un bref exemple de construction d'en-tête de paquet. La Figure B.13 montre les informations connues du codeur. En particulier, les "informations d'inclusion" montrent la couche où chaque bloc codé apparaît en premier dans un paquet. Le décodeur recevra ces informations via l'arbre de résolution d'inclusion dans plusieurs en-têtes de paquet. Le Tableau B.5 montre le flux binaire résultant (partiellement) de ces informations.

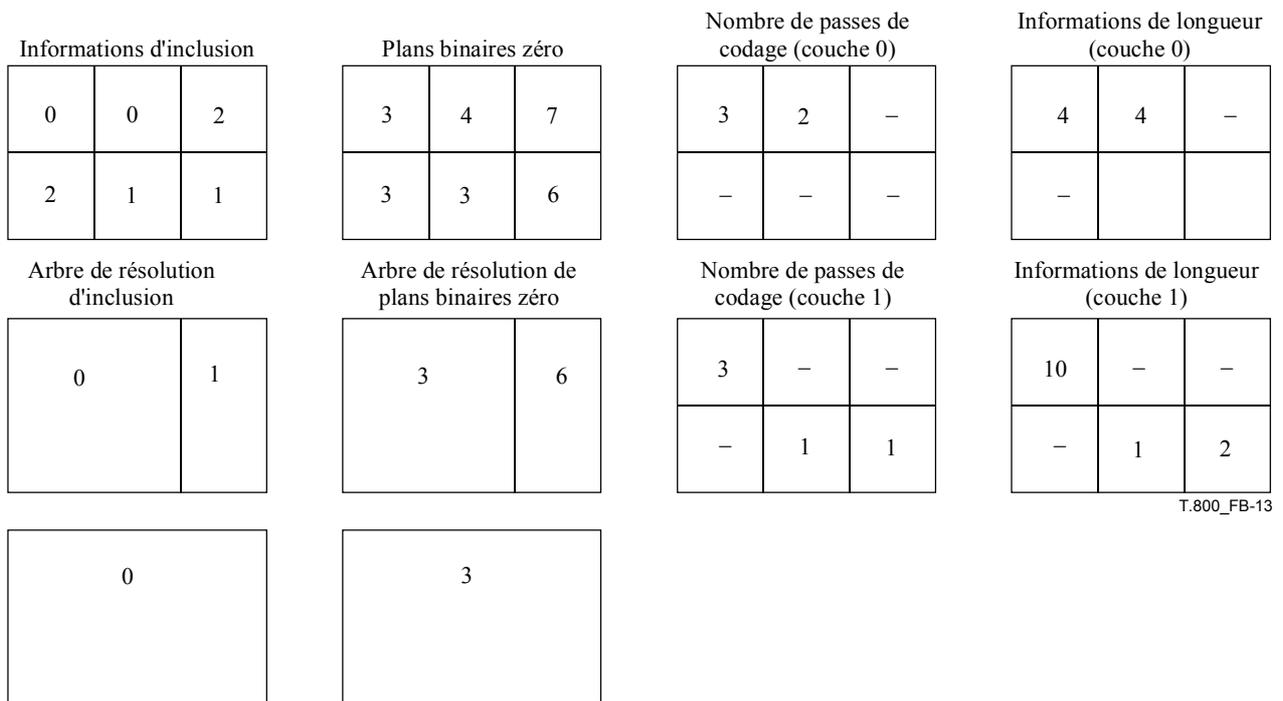


Figure B.13 – Exemple d'informations connues du codeur

Tableau B.5 – Exemple de flux binaire d'en-tête de paquets

Flux binaire (dans l'ordre)	Signification déduite
1	Paquet de longueur non nulle
111	Bloc codé 0, 0 inclus pour la première fois (arbre partiel de résolution d'inclusion)
000111	Bloc codé 0, 0 non significatif pour 3 plans binaires
1100	Bloc codé 0, 0 incluant 3 passes de codage
0	Indicateur de longueur de bloc codé 0, 0 inchangé
0100	Bloc codé 0, 0 comportant 4 octets, dont 4 bits sont utilisés, 3 + fonction plancher($\log_2 3$)
1	Bloc codé 1, 0 inclus pour la première fois (arbre partiel de résolution d'inclusion)
01	Bloc codé 1, 0 non significatif pour 4 plans binaires
10	Bloc codé 1, 0 incluant 2 passes de codage
10	Indicateur de longueur de bloc codé 1, 0 augmenté de 1 bit (3 à 4)
00100	Bloc codé 1, 0 comportant 4 octets, dont 5 bits sont utilisés, 4 + fonction plancher($\log_2 2$), (Noter que, bien qu'il s'agisse d'une entrée légitime, celle-ci n'est pas minimale en longueur codée.)
0	Bloc codé 2, 0 pas encore inclus (arbre partiel de résolution)
0	Bloc codé 0, 1 pas encore inclus
0	Bloc codé 1, 1 pas encore inclus
	Bloc codé 2, 1 pas encore inclus (pas de données requises, déjà acheminées par arbre partiel de résolution pour bloc codé 2, 0)
...	En-tête de données de paquet pour les autres sous-bandes, données de paquet
Paquet pour la prochaine couche	
1	Paquet de longueur non nulle
1	Bloc codé 0, 0 inclus de nouveau
1100	Bloc codé 0, 0 incluant 3 passes de codage
0	Indicateur de longueur de bloc codé 0, 0 inchangé
1010	Bloc codé 0, 0 comportant 10 octets, 3 + $\log_2(3)$ bits utilisés
0	Bloc codé 1, 0 non inclus dans cette couche

Tableau B.5 – Exemple de flux binaire d'en-tête de paquets

Flux binaire (dans l'ordre)	Signification déduite
10	Bloc codé 2, 0 pas encore inclus
0	Bloc codé 0, 1 pas encore inclus
1	Bloc codé 1, 1 inclus pour la première fois
1	Bloc codé 1, 1 non significatif pour 3 plans binaires
0	Bloc codé 1, 1 incluant 1 passe de codage
0	Informations de longueur de bloc codé 1, 1 inchangées
001	Bloc codé 1, 1 comportant 1 octet, $3 + \log_2(1)$ bits utilisés
1	Bloc codé 2, 1 inclus pour la première fois
00011	Bloc codé 2, 1 non significatif pour 6 plans binaires
0	Bloc codé 2, 1 incluant 1 passe de codage
0	Indicateur de longueur de bloc codé 2, 1 inchangé
010	Bloc codé 2, 1 comportant 2 octets, $3 + \log_2$ bits 1 utilisés
...	En-tête de données de paquet pour les autres sous-bandes, données de paquet

B.11 Pavé et éléments de pavé

Chaque pavé codé est représenté par une séquence de paquets. Les règles régissant l'ordre dans lequel les paquets d'un pavé apparaissent dans le flux codé sont spécifiées dans le § B.12. Il est possible qu'un pavé ne contienne aucun paquet, si aucun échantillon issu d'une composante d'image n'est affecté à la région occupée par le pavé sur la grille de référence.

Toute représentation de pavé peut être tronquée par rejet d'un ou de plusieurs octets finals. De même, un nombre quelconque de paquets entiers (dans l'ordre) peut être négligé et le paquet final apparaissant dans le pavé peut être partiellement tronqué. Les paramètres de segment marqueur de longueur de pavé doivent refléter cela.

La séquence de paquets représentant un pavé particulier peut être subdivisée en segments contigus, appelés *éléments de pavé*. Un nombre quelconque de paquets (y compris zéro) peut être contenu dans un élément de pavé. Chaque pavé doit toujours contenir au moins un élément de pavé. Les divisions entre éléments de pavé doivent toujours apparaître aux limites de paquet. Alors que les pavés sont des zones géométriques cohérentes, les éléments de pavé peuvent être répartis dans tout le flux codé de toute façon souhaitée, à condition que les éléments de pavé issus du même pavé apparaissent dans l'ordre qui préserve la séquence originale de paquets. Chaque élément de pavé commence par un seul segment marqueur SOT (voir § A.4.2) contenant l'index du pavé auquel l'élément de pavé appartient.

NOTE – Il est possible d'entrelacer des éléments de pavé issus de pavés différents, du moment que l'ordre des éléments de pavé issus de chaque pavé est préservé. Par exemple, un flux codé correct pourrait avoir l'ordre suivant:

- pavé numéro 0, élément de pavé numéro 0;
- pavé numéro 1, élément de pavé numéro 0;
- pavé numéro 0, élément de pavé numéro 1;
- pavé numéro 1, élément de pavé numéro 1;
- etc.

Si des segments marqueurs SOP sont permis (par signalisation dans le segment marqueur COD, voir § A.6.1), chaque paquet contenu dans un élément de pavé donné peut être affixé avec un segment marqueur SOP (voir § A.8.1). Cependant, que le segment marqueur SOP soit ou non utilisé, le décompte dans le paramètre N_{sop} est incrémenté pour chaque paquet. Si les en-têtes de paquet sont transférés vers les segments marqueurs PPM ou PPT (voir § A.7.4 et A.7.5), alors les segments marqueurs SOP peuvent apparaître immédiatement avant le corps de paquet dans la partie d'élément de pavé contenant les données d'image comprimées.

Si des marqueurs EPH sont requis (par signalisation dans le segment marqueur COD, voir § A.6.1), chaque en-tête de paquet dans tout élément de pavé donné doit être suffixé avec un segment marqueur EPH (voir § A.8.2). Si les en-têtes de paquet sont transférés vers les segments marqueurs PPM ou PPT (voir § A.7.4 et A.7.5), alors les marqueurs EPH doivent apparaître après les en-têtes de paquet dans les segments marqueurs PPM ou PPT.

B.12 Ordre de progression

Pour un élément de pavé donné, les paquets contiennent toutes les données d'image comprimées issues d'une couche spécifique, d'une composante spécifique, d'un niveau spécifique de résolution, et d'un district spécifique. L'ordre dans lequel ces paquets se trouvent contenus dans le flux codé est appelé *ordre de progression*. L'ordonnement des paquets peut progresser dans quatre axes: couche, composante, niveau de résolution et district.

Il est possible que des composantes aient un nombre différent de niveaux de résolution. Dans ce cas, le niveau de résolution qui correspond à la sous-bande N_{LL} est le premier niveau de résolution ($r = 0$) pour toutes les composantes. Les indices sont synchronisés à partir de ce point.

NOTE – Par exemple, si l'on prend le cas d'une progression niveau de résolution-position-composante-couche et de deux composantes avec respectivement 7 niveaux de résolution (6 niveaux de décomposition) et 3 niveaux de résolution (2 niveaux de décomposition). Le niveau $r = 0$ correspondra à la sous-bande N_{LL} des deux composantes. De $r = 0$ à $r = 2$, les composantes seront entrelacées comme décrit ci-dessous. De $r = 3$ à $r = 6$, seule la composante 0 comportera des paquets.

B.12.1 Détermination de l'ordre de progression

Les segments marqueurs COD signalent quels ordres de progression sont utilisés, sur les cinq qui existent (voir § A.6.1). L'ordre de progression peut également être outrepassé avec le segment marqueur POC (voir § A.6.6) dans un en-tête quelconque d'élément de pavé. Pour chacun des ordres de progression possibles, le mécanisme permettant de déterminer l'ordre dans lequel les paquets sont inclus est décrit ci-dessous.

B.12.1.1 Progression couche-niveau de résolution-composante-position

La progression couche-niveau de résolution-composante-position est définie comme l'entrelacement des paquets dans l'ordre suivant:

pour chaque $l = 0, \dots, L - 1$
 pour chaque $r = 0, \dots, N_{max}$
 pour chaque $i = 0, \dots, Csiz - 1$
 pour chaque $k = 0, \dots, numprecincts - 1$
 paquet pour composante i , niveau de résolution r , couche l , et district k .

Ici, L est le nombre de couches et N_{max} est le nombre maximal de niveaux de décomposition, N_L , utilisés dans toute composante du pavé. Une progression de ce type peut être utile lorsqu'une basse précision d'échantillon est très souhaitable, mais des informations sont nécessaires pour toutes les composantes.

B.12.1.2 Progression niveau de résolution-couche-composante-position

La progression niveau de résolution-couche-composante-position est définie comme l'entrelacement des paquets dans l'ordre suivant:

pour chaque $r = 0, \dots, N_{max}$
 pour chaque $l = 0, \dots, L - 1$
 pour chaque $i = 0, \dots, Csiz - 1$
 pour chaque $k = 0, \dots, numprecincts - 1$
 paquet pour composante i , niveau de résolution r , couche l , et district k .

Une progression de ce type peut être utile afin de fournir des versions à bas niveau de résolution de toutes les composantes d'image.

B.12.1.3 Progression niveau de résolution-position-composante-couche

La progression niveau de résolution-position-composante-couche est définie comme l'entrelacement des paquets dans l'ordre suivant:

- pour chaque $r = 0, \dots, N_{max}$
 - pour chaque $y = ty_0, \dots, ty_1 - 1$,
 - pour chaque $x = tx_0, \dots, tx_1 - 1$,
 - pour chaque $i = 0, \dots, Csiz - 1$
 - si ((y divisible par $YRsiz(i) \cdot 2^{PPy(r,i) + N_L(i) - r}$) OU (($y = ty_0$) ET ($try_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPy(r,i) + N_L(i) - r}$))
 - si ((x divisible par $XRsiz(i) \cdot 2^{PPx(r,i) + N_L(i) - r}$) OU (($x = tx_0$) ET ($trx_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPx(r,i) + N_L(i) - r}$))
 - pour le prochain district, k , s'il en existe un,
 - pour chaque $l = 0, \dots, L - 1$
 - paquet pour composante i , niveau de résolution r , couche l , et district k .

Dans ce qui précède, k peut être obtenu par:

$$k = \left\lfloor \left\lfloor \frac{x}{XRsiz(i) \cdot 2^{N_L - r}} \right\rfloor - \left\lfloor \frac{trx_0}{2^{PPx(r,i)}} \right\rfloor + numprecinctswide(r,i) \cdot \left(\left\lfloor \frac{y}{YRsiz(i) \cdot 2^{N_L - r}} \right\rfloor - \left\lfloor \frac{try_0}{2^{PPy(r,i)}} \right\rfloor \right) \right\rfloor \quad (B-20)$$

Pour utiliser cette progression, les valeurs XRsiz et YRsiz doivent toujours être des puissances de deux pour chaque composante. Une progression de ce type peut être utile afin de fournir des versions à bas niveau de résolution de toutes les composantes d'image à un emplacement spatial particulier.

NOTE – L'itération des variables x et y dans la formulation ci-dessus n'est donnée que pour simplifier l'expression, et non l'implémentation. La plupart des paires (x, y) produites par cette boucle se traduiront généralement par l'inclusion d'aucun paquet. Des itérations plus efficaces peuvent être trouvées sur la base des dimensions minimales des divers districts, mappés à la grille de référence. Cette note s'applique également aux boucles présentées pour les deux progressions suivantes.

B.12.1.4 Progression position-composante-niveau de résolution-couche

La progression position-composante-niveau de résolution-couche est définie comme l'entrelacement des paquets dans l'ordre suivant:

- pour chaque $y = ty_0, \dots, ty_1 - 1$,
 - pour chaque $x = tx_0, \dots, tx_1 - 1$,
 - pour chaque $i = 0, \dots, Csiz - 1$
 - pour chaque $r = 0, \dots, N_L$ où N_L est le nombre de niveaux de décomposition pour composante i ,
 - si ((y divisible par $YRsiz(i) \cdot 2^{PPy(r,i) + N_L(i) - r}$) OU (($y = ty_0$) ET ($try_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPy(r,i) + N_L(i) - r}$))
 - si ((x divisible par $XRsiz(i) \cdot 2^{PPx(r,i) + N_L(i) - r}$) OU (($x = tx_0$) ET ($trx_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPx(r,i) + N_L(i) - r}$))
 - pour le prochain district, k , s'il en existe un, dans la séquence représentée dans la Figure B.8
 - pour chaque $l = 0, \dots, L - 1$
 - paquet pour composante i , niveau de résolution r , couche l , et district k .

ISO/CEI 15444-1:2004 (F)

Dans ce qui précède, k peut être obtenu d'après l'équation (B-20). Afin d'utiliser cette progression, les valeurs XR_{siz} et YR_{siz} doivent être des puissances de deux pour chaque composante. Une progression de ce type peut être utile afin de fournir une haute précision d'échantillon pour un emplacement spatial particulier dans toutes les composantes.

B.12.1.5 Progression composante-position-niveau de résolution-couche

La progression composante-position-niveau de résolution-couche est définie comme l'entrelacement des paquets dans l'ordre suivant:

pour chaque $i = 0, \dots, Csiz - 1$

pour chaque $y = ty_0, \dots, ty_1 - 1$,

pour chaque $x = tx_0, \dots, tx_1 - 1$,

pour chaque $r = 0, \dots, N_L$ où N_L est le nombre de niveaux de décomposition pour composante i ,

si ((y divisible par $YR_{siz}(i) \cdot 2^{PPy(r,i) + N_L(i) - r}$) OU (($y = ty_0$) ET ($try_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPy(r,i) + N_L(i) - r}$))

si ((x divisible par $XR_{siz}(i) \cdot 2^{PPx(r,i) + N_L(i) - r}$) OU (($x = tx_0$) ET ($trx_0 \cdot 2^{N_L(i) - r}$ NON divisible par $2^{PPx(r,i) + N_L(i) - r}$))

pour le prochain district, k , s'il en existe un, dans la séquence représentée dans la Figure B.8

pour chaque $l = 0, \dots, L - 1$

paquet pour composante i , niveau de résolution r , couche l , et district k .

Dans ce qui précède, k peut être obtenu d'après l'équation (B-20). Une progression de ce type peut être utile afin de fournir une haute précision pour un emplacement spatial particulier dans une composante d'image particulière.

B.12.2 Volumes d'ordre de progression

L'ordre de progression par défaut est signalé dans le segment marqueur COD de l'en-tête principal ou des en-têtes de pavé (voir § A.6.1). Les boucles de progression du § B.12.1 vont toutes de zéro à la valeur maximale.

Si cet ordre de progression doit être modifié, le segment marqueur POC est utilisé (voir § A.6.6). Dans ce cas, les boucles "pour" décrites au § B.12.1 sont limitées par des points de début (CSpoc, RSpoc, Couche = 0, inclusif) et des points de fin (CEpoc, REpoc et LEpoc, exclusif). Cela crée un volume d'ordre de progression de paquets. Tous les paquets inclus dans le volume entier d'ordre de progression se trouvent ordonnés dans le flux codé avant que le prochain changement de l'ordre de progression ne prenne effet. Aucun paquet n'est jamais répété dans le flux codé. Donc, la couche commence toujours par le prochain paquet pour une composante-pavé, un niveau de résolution, et un district donné. Le décodeur est requis afin de déterminer la prochaine couche.

Donc, les variables contenues dans les boucles ci-dessus sont limitées par le volume d'ordre de progression comme décrit dans l'équation (B-21).

$$\begin{aligned} CSpod &\leq i < CEpod \\ RSpod &\leq r < REpod \\ 0 &\leq l < LEpod \end{aligned} \tag{B-21}$$

NOTE – La Figure B.14 montre un exemple de deux volumes de progression pour une même composante d'image. Les premiers paquets sont envoyés dans l'ordre de progression suivant: niveau de résolution-couche-composante-position jusqu'à ce que la boîte étiquetée "Premier" dans la figure soit complète; puis les paquets sont envoyés en progression couche-niveau de résolution-composante-position pour les couches de tous les niveaux de résolution qui n'ont pas été déjà envoyés.

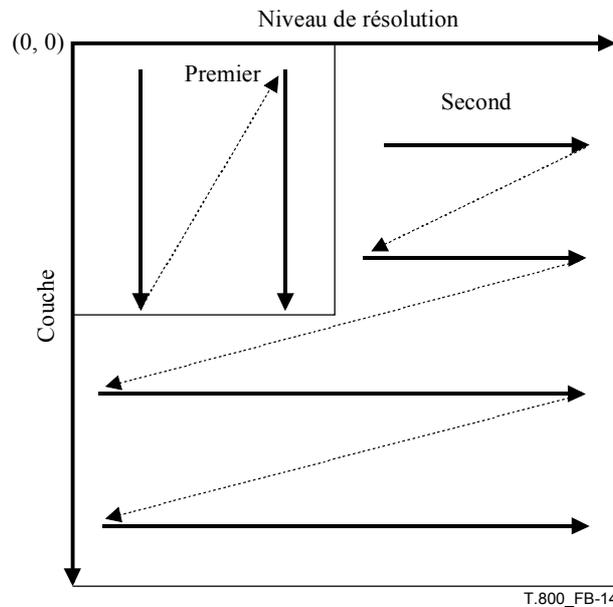


Figure B.14 – Exemple de volume d'ordre de progression en deux dimensions

B.12.3 Signalisation de changement de l'ordre de progression

S'il y a changement de l'ordre de progression, au moins un segment marqueur POC doit être utilisé dans le flux codé (voir § A.6.6). Il ne peut y avoir qu'un seul segment marqueur POC dans un en-tête donné (principal ou d'élément de pavé) mais ce segment marqueur peut décrire de nombreux changements de l'ordre de progression.

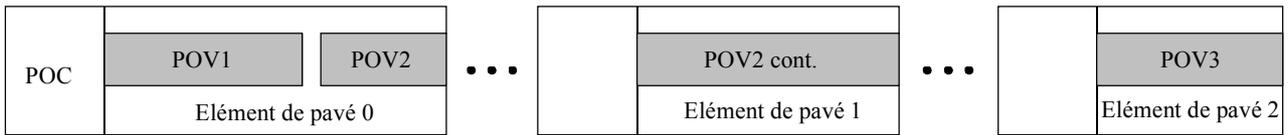
Si le segment marqueur POC se trouve dans l'en-tête principal, il outrepassse la progression trouvée dans le marqueur COD pour tous les pavés. Le segment marqueur POC d'en-tête principal est utilisé pour les pavés qui n'ont pas de segment marqueur de changement POC dans leur en-tête d'élément de pavé.

Si un segment marqueur POC est utilisé pour un pavé individuel, il doit y avoir un marqueur POC dans le premier en-tête d'élément de ce pavé et tous les changements de l'ordre de progression doivent être signalés dans l'en-tête d'élément de ce pavé. l'ordre de progression du marqueur COD et le segment marqueur POC d'en-tête principal (s'il y a un) sont outrepassés.

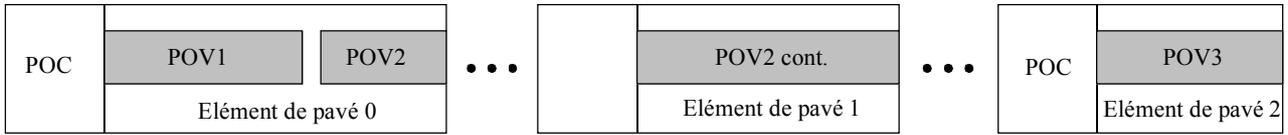
S'il y a des changements d'ordre de progression signalés par des segments marqueurs de changement POC (que ce soit dans l'en-tête principal ou dans les en-têtes d'élément de pavé), alors tout l'ordre de tous les paquets contenus dans le flux codé, ou des éléments affectés de pavé contenus dans le flux codé, doit être décrit par des volumes d'ordre de progression dans les segments marqueurs de changement POC. Le cas où un volume d'ordre de progression est rempli et où le prochain n'est pas défini n'apparaîtra jamais. D'autre part, les segments marqueurs de changement POC peuvent décrire plus de volumes d'ordre de progression qu'il n'y en a dans le flux codé. De même, le dernier volume d'ordre de progression dans chaque pavé peut être incomplet.

Les segments marqueurs de changement POC doivent décrire les volumes d'ordre de progression dans l'ordre d'un en-tête quelconque d'élément de pavé avant que le premier paquet inclus apparaisse. Cependant, le marqueur POC peut se trouver, mais n'est pas tenu de se trouver, dans l'en-tête d'élément de pavé apparaissant immédiatement avant que le volume d'ordre de progression soit utilisé. Il est possible de décrire de nombreux volumes d'ordre de progression dans un en-tête d'élément de pavé même si ces volumes d'ordre de progression n'apparaissent pas avant des éléments de pavé ultérieurs.

NOTE – Par exemple, tous les volumes d'ordre de progression peuvent être décrits comme un seul segment marqueur POC dans le premier en-tête d'élément de pavé. La Figure B.15 a) montre ce scénario. Est également acceptable, dans ce cas, la description de deux volumes d'ordre de progression dans le premier en-tête d'élément de pavé et d'un autre volume dans le troisième en-tête d'élément de pavé, comme représenté dans la Figure B.15 b).



a) Tous les volumes d'ordre de progression sont décrits dans le segment marqueur POC situé dans le premier en-tête d'élément de pavé



T.800_FB-15

b) Les volumes d'ordre de progression 1 et 2 sont décrits dans le segment marqueur POC situé dans le premier en-tête d'élément de pavé; le volume d'ordre de progression 3 est décrit dans le troisième en-tête d'élément de pavé

Figure B.15 – Exemple du placement de segments marqueurs de changement POC

Annexe C

Codage arithmétique entropique

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

C.1 Codage binaire (pour information)

La Figure C.1 montre a simple schéma fonctionnel du codeur arithmétique adaptatif binaire. Les paires décision (D) et contexte (CX) sont traitées ensemble afin de produire des données d'image comprimées (CD) en sortie. Les deux éléments D et CX sont fournis par l'unité modélisée (non représentée). Le contexte CX choisit l'estimation de probabilité à utiliser pendant le codage de D. Dans la présente Recommandation | Norme internationale, CX est une étiquette de contexte.

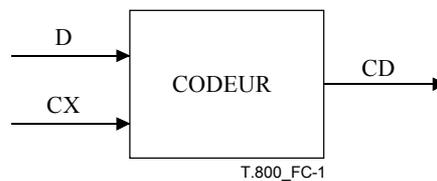


Figure C.1 – Entrées et sorties de codeur arithmétique

C.1.1 Subdivision d'intervalle récurrente (pour information)

La probabilité de subdivision récurrente d'intervalle de probabilité du codage d'Elias est la base du processus de codage arithmétique binaire. A chaque décision binaire, l'intervalle de probabilité actuel est subdivisé en deux sous-intervalles et la chaîne codée est modifiée (si nécessaire) de façon qu'elle pointe vers la base (la limite inférieure) du sous-intervalle de probabilité attribué au symbole apparu.

Dans la subdivision de l'intervalle actuel en deux sous-intervalles, le sous-intervalle pour le symbole plus probable (MPS) est ordonné au-dessus du sous-intervalle pour le symbole moins probable (LPS, *less probable symbol*). Donc, lorsque le symbole MPS est codé, le sous-intervalle du symbole LPS est ajouté à la chaîne codée. Cette convention de codage nécessite que les symboles soient reconnus comme étant soit MPS ou LPS, plutôt que 0 ou 1. Par conséquent, la longueur de l'intervalle LPS et le sens du symbole MPS pour chaque décision doivent toujours être connus afin de coder cette décision.

Comme la chaîne codée pointe toujours sur la base de l'intervalle actuel, le processus de décodage revient à déterminer, pour chaque décision, sur quel sous-intervalle les données d'image comprimées pointent. Cela est également effectué de façon récurrente, au moyen du même processus de subdivision d'intervalle que dans le codeur. Chaque fois qu'une décision est décodée, le décodeur soustrait tout intervalle que le codeur a ajouté à la chaîne codée. Donc, la chaîne codée dans le décodeur est un pointeur sur l'intervalle actuel par rapport à la base de l'intervalle actuel. Comme le processus de codage implique l'addition de fractions binaires plutôt que la concaténation de mots de code entiers, les décisions binaires plus probables peuvent souvent être codées au moyen de bien moins qu'un seul bit par décision.

C.1.2 Conventions et approximations de codage (pour information)

Les opérations de codage sont effectuées par arithmétique d'entiers à précision fixe et par représentation de valeurs fractionnaires par des entiers où 0x8000 équivaut à la valeur décimale 0,75. L'intervalle A est maintenu dans l'étendue $0,75 \leq A < 1,5$ en le doublant chaque fois que la valeur d'entier tombe au-dessous de 0x8000.

Le registre de code C est également doublé chaque fois que A est doublé. Périodiquement – afin d'éviter un débordement du registre C – un octet des données d'image comprimées est supprimé des bits d'ordre supérieur du registre C et placés dans un tampon externe de données d'image comprimées. Le report dans le tampon externe est empêché par une procédure de bourrage binaire.

ISO/CEI 15444-1:2004 (F)

Le fait de maintenir A dans l'étendue $0,75 \leq A < 1,5$ permet d'utiliser une simple approximation arithmétique dans la subdivision d'intervalle. L'intervalle est A et l'estimation actuelle de la probabilité du symbole LPS est Q_e ; un calcul précis des sous-intervalles nécessiterait:

$$A - (Q_e * A) = \text{sous-intervalle pour le symbole MPS} \quad (\text{C-1})$$

$$Q_e * A = \text{sous-intervalle pour le symbole LPS} \quad (\text{C-2})$$

Comme la valeur de A est de l'ordre de l'unité, les sous-intervalles sont approchés par:

$$A - Q_e = \text{sous-intervalle pour le symbole MPS} \quad (\text{C-3})$$

$$Q_e = \text{sous-intervalle pour le symbole LPS} \quad (\text{C-4})$$

Chaque fois que le symbole MPS est codé, la valeur de Q_e est ajoutée au registre de code et l'intervalle est réduit à $A - Q_e$. Chaque fois que le symbole LPS est codé, le registre de code est laissé inchangé et l'intervalle est réduit à Q_e . L'étendue de précision requise pour A est alors restaurée, si nécessaire, par renormalisation des deux intervalles A et C .

Avec le processus illustré ci-dessus, les approximations dans le processus de subdivision d'intervalle peuvent parfois rendre le sous-intervalle du symbole LPS plus grand que le sous-intervalle du symbole MPS. Si, par exemple, la valeur de Q_e est 0,5 et que A soit à la valeur minimale permise de 0,75, la normalisation approchée donne 1/3 de l'intervalle au symbole MPS et 2/3 au symbole LPS. Afin d'éviter cette inversion de taille, les intervalles de symbole MPS et de symbole LPS sont échangés chaque fois que l'intervalle de symbole LPS est plus grand que l'intervalle de symbole MPS. Cet échange conditionnel de symboles MPS/LPS ne peut apparaître que lorsqu'une renormalisation est requise.

Chaque fois qu'une renormalisation se produit, un processus d'estimation de probabilité est invoqué: ce processus détermine une nouvelle estimation de probabilité pour le contexte en cours de codage. Aucun décompte de symboles explicite n'est nécessaire pour l'estimation. Les probabilités relatives de renormalisation après codage d'un symbole LPS ou MPS fournissent un mécanisme de comptage approché des symboles qui est utilisé pour estimer directement les probabilités.

C.2 Description du codeur arithmétique (pour information)

L'élément ENCODER (Figure C.2) initialise le codeur par la procédure INITENC. Les paires CX et D sont lues et transmises à la procédure coder jusqu'à ce que toutes les paires aient été lues. Les procédures d'estimation de probabilité qui fournissent des estimations adaptatives de la probabilité pour chaque contexte sont intégrées dans la procédure ENCODE. Les octets des données d'image comprimées sont émis en sortie lorsque nécessaire. Lorsque toutes les paires CX et D ont été lues, la procédure FLUSH met le contenu du registre C à autant de bits 1 que possible puis émet en sortie les octets finals. La procédure FLUSH termine également le codage et produit le marqueur de terminaison requis.

NOTE – Alors que la procédure FLUSH est requise dans la Rec. UIT-T T.88 | ISO/CEI 14492, elle est informative dans la présente Recommandation | Norme internationale. D'autres méthodes, comme celle qui est définie au § D.4.2, sont acceptables.

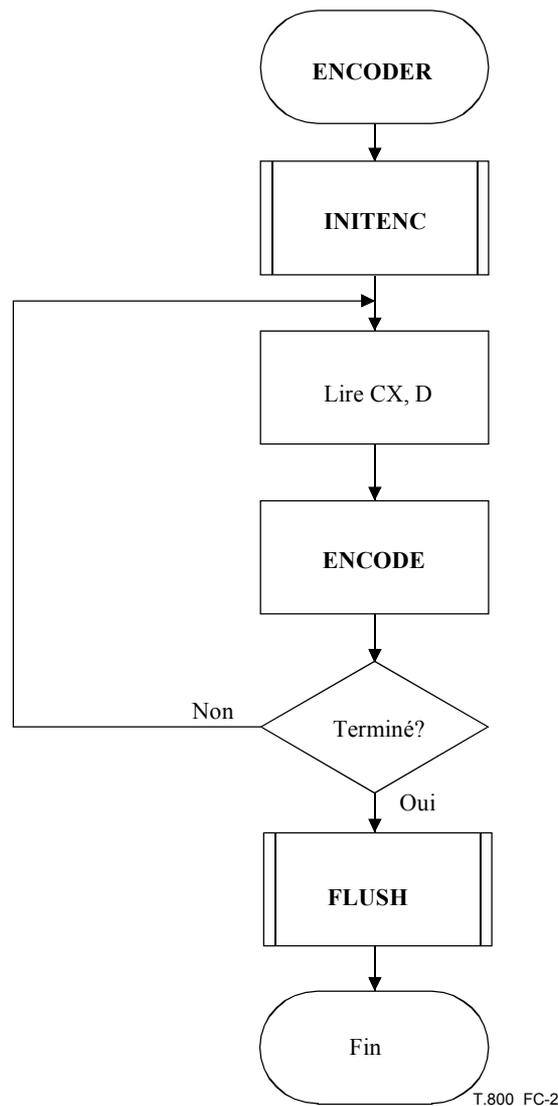


Figure C.2 – Codeur pour le codeur MQ

C.2.1 Conventions relatives au registre de code du codeur (pour information)

Les organigrammes donnés dans la présente annexe impliquent les structures de registre pour le codeur représentées dans le Tableau C.1.

Tableau C.1 – Structures de registre de codeur

	MSB			LSB
Registre C	0000 cbbb	bbbb bsss	xxxx xxxx	xxxx xxxx
Registre A	0000 0000	0000 0000	1aaa aaaa	aaaa aaaa

Les bits "a" sont les bits fractionnaires dans le registre A (la valeur actuelle de l'intervalle) et les bits "x" sont les bits fractionnaires dans le registre de code. Les bits "s" sont des bits espaceurs qui appliquent d'utiles contraintes au report, et les bits "b" indiquent les positions binaires à partir desquelles les octets complétés des données d'image comprimées sont supprimés du registre C. Le bit "c" est un bit de report. La description détaillée du bourrage binaire et de la gestion des reports sera donnée dans une partie ultérieure de la présente annexe.

C.2.2 Codage d'une décision (ENCODE) (pour information)

La procédure coder détermine si la décision D est ou non un 0. Ensuite, une procédure CODE0 ou CODE1 est appelée de façon appropriée. Il arrivera souvent que les matérialisations n'aient pas de procédure ENCODE, mais qu'elles appellent la procédure CODE0 ou CODE1 afin de coder directement une décision 0 ou une décision 1. La Figure C.3 montre cette procédure.

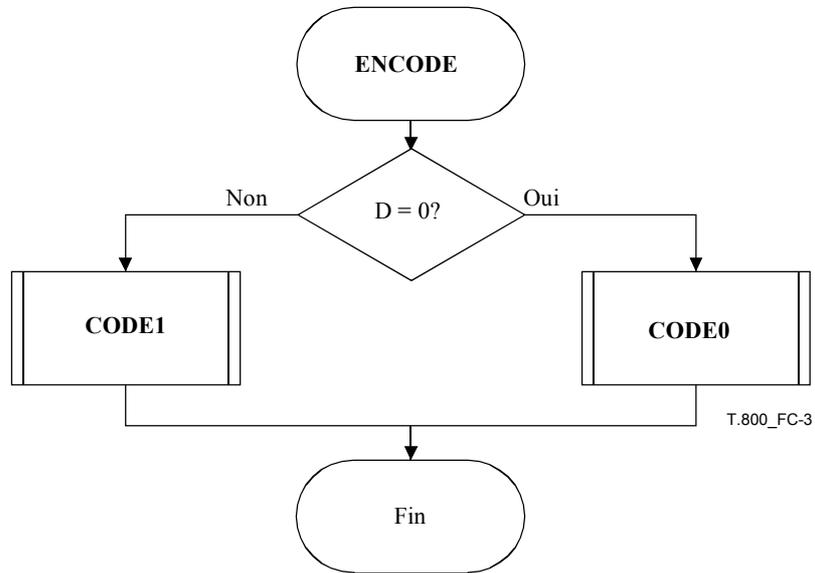


Figure C.3 – Procédure ENCODE

C.2.3 Codage d'un 1 ou d'un 0 (CODE1 et CODE0) (pour information)

Lorsqu'une décision binaire donnée est codée, une des deux possibilités suivantes apparaît: le symbole est soit le symbole plus probable soit le symbole moins probable. Les procédures CODE1 et CODE0 sont illustrées dans les Figures C.4 et C.5. Dans ces figures, CX est le contexte. Pour chaque contexte, l'index de l'estimation de probabilité qui doit être utilisé dans les opérations de codage et la valeur du symbole MPS sont mémorisés. Le terme MPS(CX) est le sens (0 ou 1) du symbole MPS pour le contexte CX.

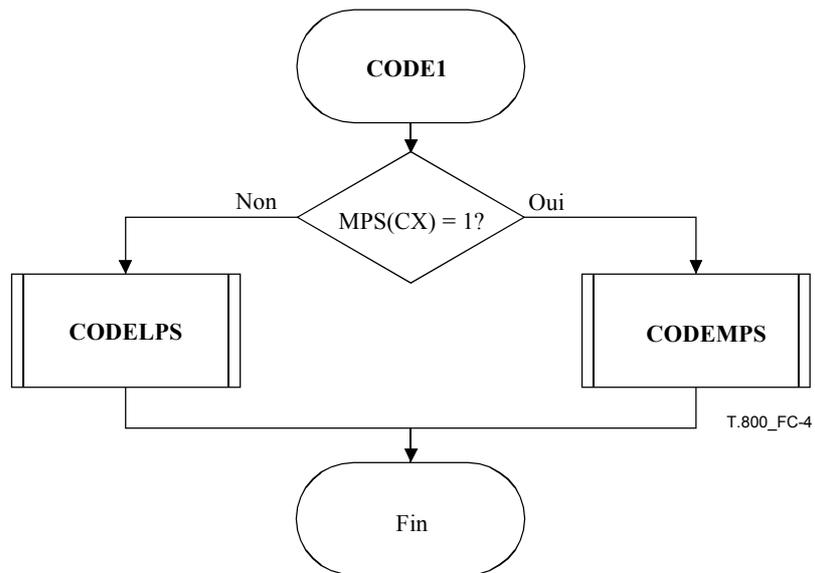


Figure C.4 – Procédure CODE1

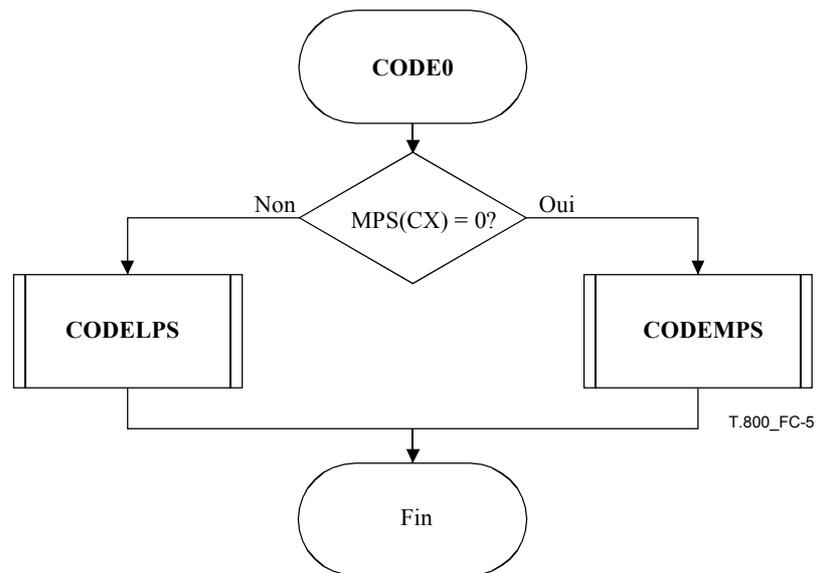


Figure C.5 – Procédure CODE0

C.2.4 Codage d'un symbole MPS ou LPS (CODEMPS et CODELPS) (pour information)

La procédure CODELPS (Figure C.6) se compose habituellement d'une normalisation de l'intervalle à $Q_e(I(CX))$, qui est l'estimation de probabilité du symbole LPS déterminée d'après l'index I mémorisé pour le contexte CX. L'intervalle supérieur est calculé en premier de façon qu'il puisse être comparé à l'intervalle inférieur afin de confirmer que Q_e possède la plus petite longueur. Cette procédure est toujours suivie par une renormalisation (RENORME). Si les longueurs d'intervalle sont inversées, cependant, l'échange conditionnel de symbole MPS/LPS apparaît et l'intervalle supérieur est codé. Dans un cas comme dans l'autre, l'estimation de probabilité est mise à jour. Si le fanion SWITCH pour l'index I(CX) est activé, alors le symbole MPS(CX) est inversé. Un nouvel index I est sauvegardé dans le contexte CX déterminé d'après la colonne du Tableau C.2 indiquant le prochain index de symbole LPS (NLPS).

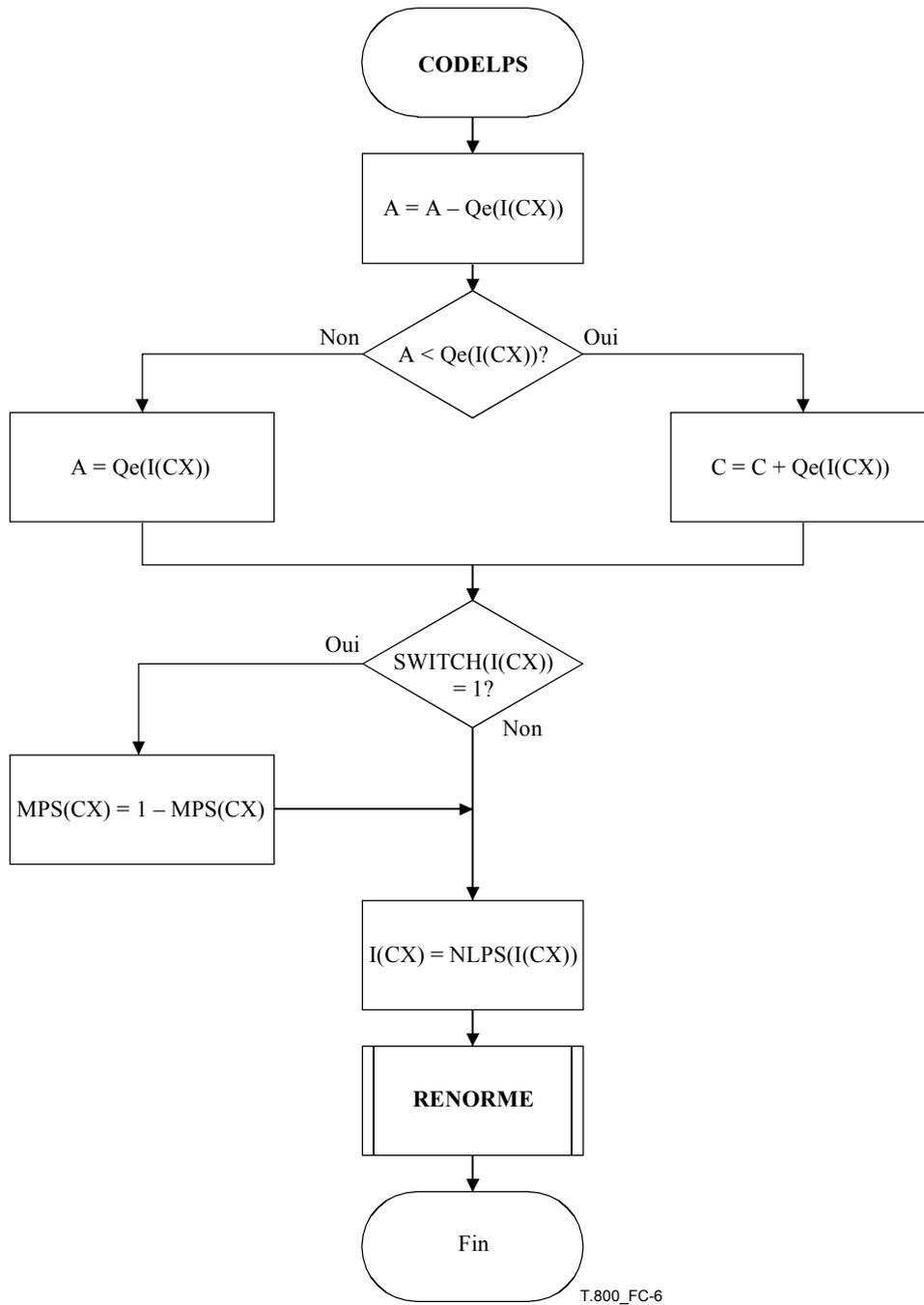


Figure C.6 – Procédure CODELPS avec échange conditionnel de symbole MPS/LPS

Tableau C.2 – Estimation de valeurs Qe et de probabilité

Index	Valeur Qe			NMPS	NLPS	SWITCH
	(nombre hexadécimal)	(binaire)	(décimal)			
0	0x5601	0101 0110 0000 0001	0,503 937	1	1	1
1	0x3401	0011 0100 0000 0001	0,304 715	2	6	0
2	0x1801	0001 1000 0000 0001	0,140 650	3	9	0
3	0x0AC1	0000 1010 1100 0001	0,063 012	4	12	0
4	0x0521	0000 0101 0010 0001	0,030 053	5	29	0
5	0x0221	0000 0010 0010 0001	0,012 474	38	33	0

Tableau C.2 – Estimation de valeurs Qe et de probabilité

Index	Valeur Qe			NMPS	NLPS	SWITCH
	(nombre hexadécimal)	(binaire)	(décimal)			
6	0x5601	0101 0110 0000 0001	0,503 937	7	6	1
7	0x5401	0101 0100 0000 0001	0,492 218	8	14	0
8	0x4801	0100 1000 0000 0001	0,421 904	9	14	0
9	0x3801	0011 1000 0000 0001	0,328 153	10	14	0
10	0x3001	0011 0000 0000 0001	0,281 277	11	17	0
11	0x2401	0010 0100 0000 0001	0,210 964	12	18	0
12	0x1C01	0001 1100 0000 0001	0,164 088	13	20	0
13	0x1601	0001 0110 0000 0001	0,128 931	29	21	0
14	0x5601	0101 0110 0000 0001	0,503 937	15	14	1
15	0x5401	0101 0100 0000 0001	0,492 218	16	14	0
16	0x5101	0101 0001 0000 0001	0,474 640	17	15	0
17	0x4801	0100 1000 0000 0001	0,421 904	18	16	0
18	0x3801	0011 1000 0000 0001	0,328 153	19	17	0
19	0x3401	0011 0100 0000 0001	0,304 715	20	18	0
20	0x3001	0011 0000 0000 0001	0,281 277	21	19	0
21	0x2801	0010 1000 0000 0001	0,234 401	22	19	0
22	0x2401	0010 0100 0000 0001	0,210 964	23	20	0
23	0x2201	0010 0010 0000 0001	0,199 245	24	21	0
24	0x1C01	0001 1100 0000 0001	0,164 088	25	22	0
25	0x1801	0001 1000 0000 0001	0,140 650	26	23	0
26	0x1601	0001 0110 0000 0001	0,128 931	27	24	0
27	0x1401	0001 0100 0000 0001	0,117 212	28	25	0
28	0x1201	0001 0010 0000 0001	0,105 493	29	26	0
29	0x1101	0001 0001 0000 0001	0,099 634	30	27	0
30	0x0AC1	0000 1010 1100 0001	0,063 012	31	28	0
31	0x09C1	0000 1001 1100 0001	0,057 153	32	29	0
32	0x08A1	0000 1000 1010 0001	0,050 561	33	30	0
33	0x0521	0000 0101 0010 0001	0,030 053	34	31	0
34	0x0441	0000 0100 0100 0001	0,024 926	35	32	0
35	0x02A1	0000 0010 1010 0001	0,015 404	36	33	0
36	0x0221	0000 0010 0010 0001	0,012 474	37	34	0
37	0x0141	0000 0001 0100 0001	0,007 347	38	35	0
38	0x0111	0000 0001 0001 0001	0,006 249	39	36	0
39	0x0085	0000 0000 1000 0101	0,003 044	40	37	0
40	0x0049	0000 0000 0100 1001	0,001 671	41	38	0
41	0x0025	0000 0000 0010 0101	0,000 847	42	39	0
42	0x0015	0000 0000 0001 0101	0,000 481	43	40	0
43	0x0009	0000 0000 0000 1001	0,000 206	44	41	0
44	0x0005	0000 0000 0000 0101	0,000 114	45	42	0
45	0x0001	0000 0000 0000 0001	0,000 023	45	43	0
46	0x5601	0101 0110 0000 0001	0,503 937	46	46	0

C.2.5 Estimation de probabilité

Le Tableau C.2 montre la valeur Q_e associée à chaque index Q_e . Les valeurs Q_e sont exprimées en entiers hexadécimaux, en entiers binaires, et en fractions décimales. Afin de convertir la représentation de Q_e par entiers de 15 bits en probabilité décimale, les valeurs Q_e sont divisées par $(4/3) * (0x8000)$.

L'estimateur peut être défini comme un automate à états finis – un tableau d'index Q_e avec les états suivants qui sont associés à chaque type de renormalisation (c'est-à-dire les nouvelles positions tabulaires) – comme représenté dans le Tableau C.2. La transition d'état n'apparaît que lorsque le registre d'intervalle du codeur arithmétique est renormalisé. Cette opération est toujours effectuée après codage du symbole LPS, et chaque fois que le registre d'intervalle est inférieur à $0x8000$ (0,75 en notation décimale) après codage du symbole MPS.

Après une renormalisation LPS, la valeur NLPS donne le nouvel index pour l'estimation de probabilité du symbole LPS. Si le fanion est 1, le sens du symbole MPS est inversé.

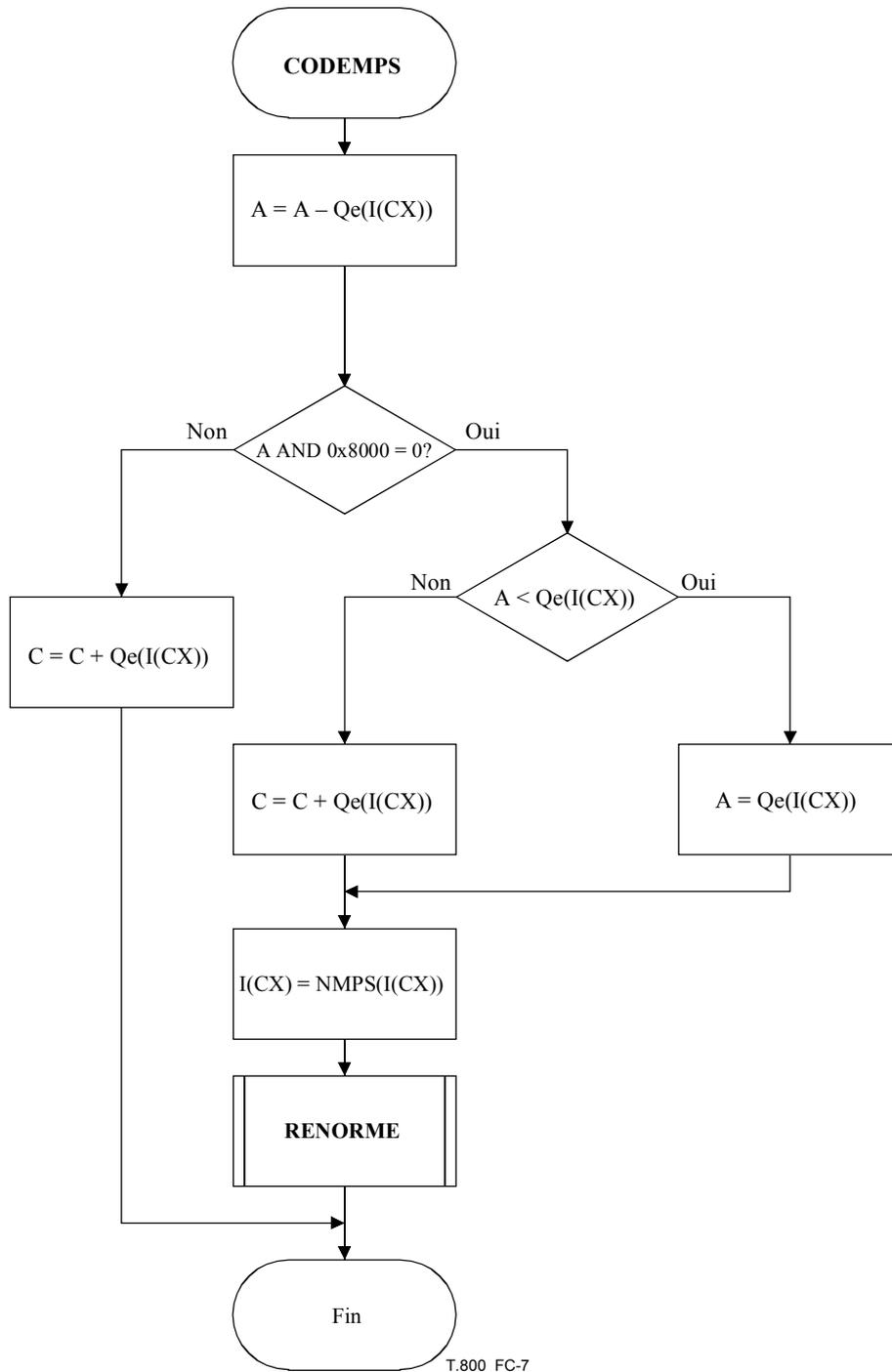


Figure C.7 – Procédure CODEMPS avec échange conditionnel de symbole MPS/LPS

L'index pointant sur l'estimation actuelle fait partie des informations mémorisées pour le contexte CX. Cet index sert à pointer sur le tableau de valeurs contenu dans le symbole NMPS, qui donne le prochain index d'une renormalisation de symbole MPS. Cet index est sauvegardé dans la mémoire de contextes à CX. La valeur MPS(CX) ne change pas.

La procédure d'estimation de la probabilité sur le trajet de renormalisation du symbole LPS est semblable à une renormalisation MPS sauf que, lorsque le fanion SWITCH(I(CX)) est 1, le sens de MPS(CX) est inversé.

L'état final d'index 46 peut servir à établir une estimation de probabilité fixe de 0,5.

C.2.6 Renormalisation dans le codeur (RENORME) (pour information)

La renormalisation est très semblable dans le codeur et dans le décodeur sauf que, dans le codeur, elle produit des bits comprimés et que, dans le décodeur, elle consomme des bits comprimés.

La procédure RENORME pour la renormalisation du codeur est illustrée dans la Figure C.8. Le registre d'intervalle A et le registre de code C sont tous deux décalés, d'un bit à la fois. Le nombre de décalages est décompté dans le compteur CT, et lorsque CT est décompté vers le bas jusqu'à zéro, un octet des données d'image comprimées est supprimé de C par la procédure OCTETOUT. La renormalisation continue jusqu'à ce que A ne soit plus inférieur à 0x8000.

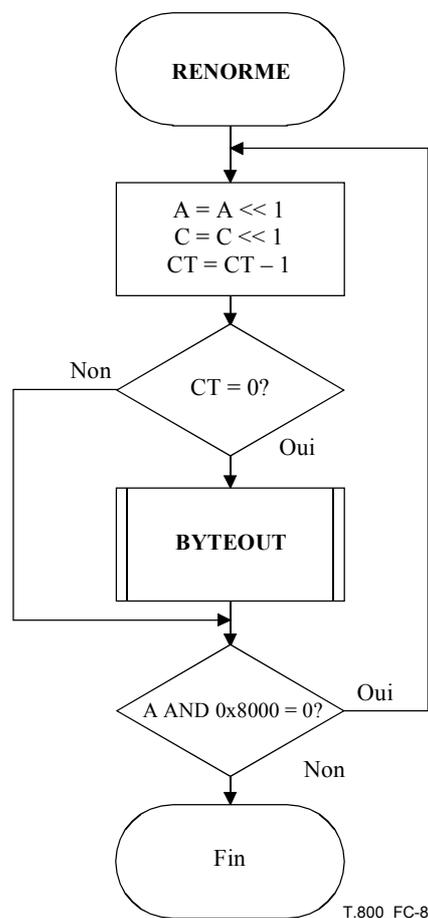


Figure C.8 – Procédure de renormalisation du codeur

C.2.7 Données d'image comprimées en sortie (OCTETOUT) (pour information)

La routine OCTETOUT appelée par la procédure RENORME est illustrée dans la Figure C.9. Cette routine contient les procédures de bourrage binaire qui sont nécessaires afin de limiter la propagation de report dans les octets complétés des données d'image comprimées. Les conventions utilisées rendent impossible qu'un report se propage au-delà de l'octet écrit le plus récemment dans le tampon de données d'image comprimées.

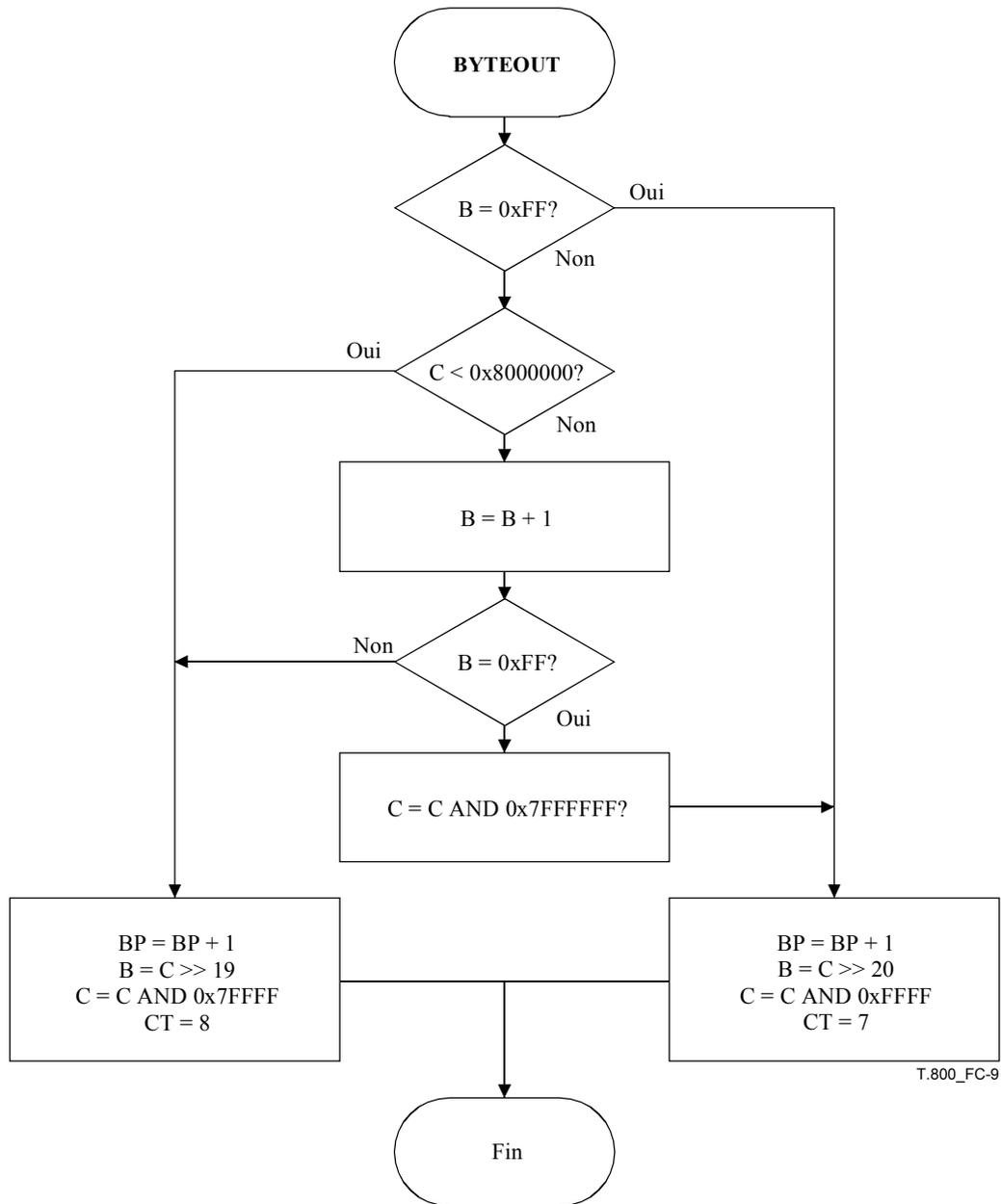


Figure C.9 – Procédure OCTETOUT pour codeur

La procédure indiquée dans le bloc du coin inférieur droit effectue un bourrage binaire après un octet 0xFF; la procédure similaire de gauche est pour le cas où le bourrage binaire n'est pas nécessaire.

B est l'octet désigné par le pointeur BP du tampon de données d'image comprimées. Si l'octet B n'est pas un octet 0xFF, le bit de report est contrôlé. Si le bit de report est activé, il est ajouté à B et B est de nouveau contrôlé afin de voir si un bit doit être inséré dans le prochain octet. Une fois la nécessité d'un bourrage binaire déterminée, le trajet approprié est choisi, BP est incrémenté et la nouvelle valeur de B est supprimée des bits "b" du registre de code.

C.2.8 Initialisation du codeur (INITENC) (pour information)

La procédure INITENC sert à lancer le codeur arithmétique. Une fois que MPS et I sont initialisés, les étapes de base sont représentées dans la Figure C.10.

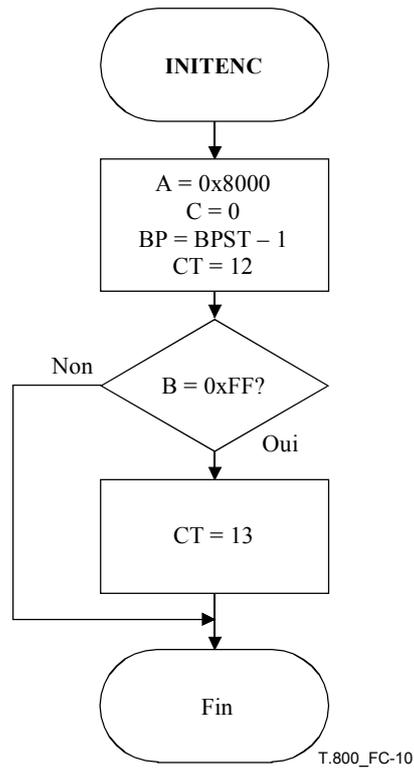


Figure C.10 – Initialisation du codeur

Le registre d'intervalle et le registre de code sont mis à leurs valeurs initiales, et le compteur de bits est activé. Le réglage $CT = 12$ reflète le fait qu'il y a trois bits espaceurs dans le registre qui doivent être remplis avant que le champ dont les octets sont supprimés soit atteint. BP pointe toujours sur l'octet précédant la position BPST où le premier octet est placé. Donc, si le précédent octet est un octet 0xFF, un bourrage binaire parasite se produira mais pourra être compensé par une augmentation de CT. Les réglages initiaux pour MPS et I sont représentés dans le Tableau D.7.

C.2.9 Terminaison de codage (FLUSH) (pour information)

La procédure FLUSH représentée dans la Figure C.11 sert à terminer les opérations de codage et à produire le marqueur de terminaison requis. La procédure garantit que le préfixe 0xFF du code de marqueur recouvre les bits finals des données d'image comprimées. Cela garantit que tout code de marqueur situé à la fin des données d'image comprimées sera reconnu et interprété avant que le décodage soit achevé.

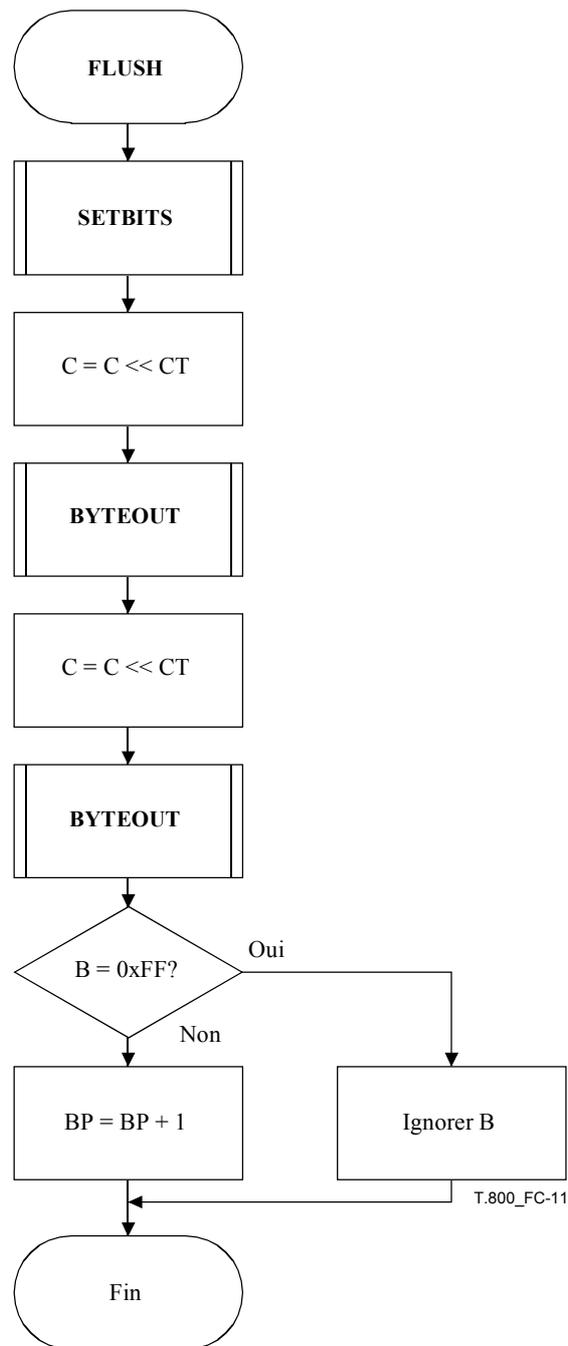


Figure C.11 – Procédure FLUSH

La première partie de la procédure FLUSH met autant de bits que possible dans le registre C jusqu'à 1, comme représenté dans la Figure C.12. La limite supérieure exclusive pour le registre C est la somme du registre C et du registre d'intervalle. Les 16 bits de plus faible poids du registre C sont forcés to 1, et le résultat est comparé à la limite supérieure. Si C est trop grand, le bit initial 1 est supprimé, ce qui réduit C à une valeur qui est dans l'intervalle.

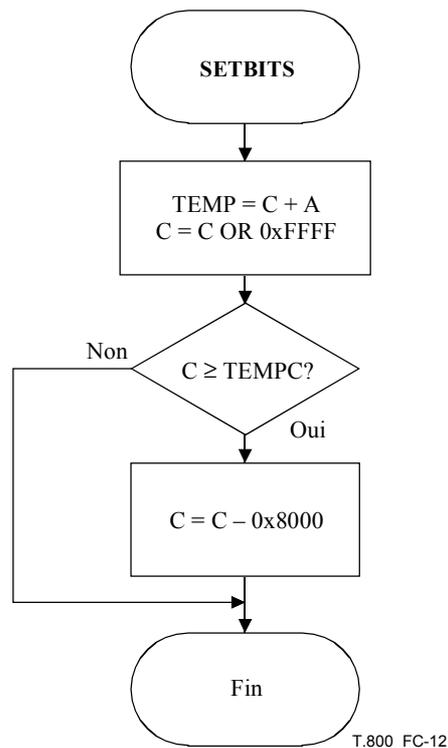


Figure C.12 – Réglage des bits finals dans le registre C

L'octet contenu dans le registre C est alors complété par décalage du registre C, et deux octets sont alors supprimés. Si l'octet contenu dans le tampon, B, est une valeur 0xFF alors il est ignoré. Sinon, le tampon B est envoyé en sortie dans le flux binaire.

NOTE – C'est la seule option normative de terminaison dans la Rec. UIT-T T.88 | ISO/CEI 14492. Cependant, une réduction supplémentaire du flux binaire est permise dans la présente Recommandation | Norme internationale à condition qu'un décodage correct soit assuré (voir § D.4.2).

C.3 Procédure de décodage arithmétique

La Figure C.13 montre un simple schéma fonctionnel d'un décodeur arithmétique adaptatif binaire. Les données d'image comprimées CD et un contexte CX issu de l'unité modélisée du décodeur (non représentée) sont introduits dans le décodeur arithmétique. La sortie du décodeur est la décision D. Les unités modélisées du codeur et du décodeur doivent fournir exactement le même contexte CX pour chaque décision.

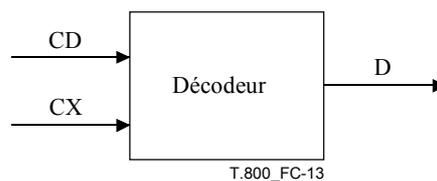


Figure C.13 – Entrées et sorties du décodeur arithmétique

Le décodeur (Figure C.14) initialise le décodeur par INITDEC. Les contextes, CX, et les octets des données d'image comprimées (selon les besoins) sont lus et communiqués à la routine DECODE jusqu'à ce que tous les contextes aient été lus. La routine DECODE décode la décision binaire D et renvoie une valeur 0 ou 1. Les procédures d'estimation de probabilité qui fournissent des estimations adaptatives de la probabilité pour chaque contexte sont imbriquées dans la routine DECODE. Lorsque tous les contextes ont été lus, les données d'image comprimées ont été décomprimées.

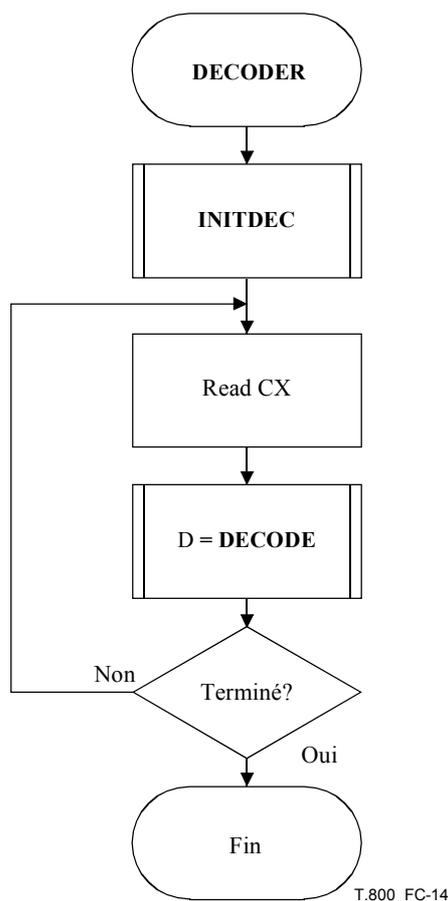


Figure C.14 – Décodeur pour le codeur MQ

C.3.1 Conventions relatives au registre de code du codeur

Les organigrammes reproduits dans la présente annexe partent du principe que les structures de registre pour le décodeur sont représentées dans le Tableau C.3.

Tableau C.3 – Structures de registre de décodeur

	MSB	LSB
Registre Haut	xxxx xxxx	xxxx xxxx
Registre Cbas	bbbb bbbb	0000 0000
Registre A	aaaa aaaa	aaaa aaaa

Les registres Haut et Cbas peuvent être considérés comme un registre C de 32 bits dont la renormalisation décale un bit de nouvelles données du bit MSB de Cbas au bit LSB de Haut. Cependant, les comparaisons de décodage utilisent le registre Haut seul. De nouvelles données sont insérées dans les bits "b" de Cbas, octet par octet.

La description détaillée de la gestion de données avec bits de bourrage sera donnée plus loin dans la présente annexe.

Noter que les comparaisons représentées dans les diverses procédures de ce paragraphe partent du principe que les précisions sont supérieures à 16 bits. Des comparaisons logiques peuvent être utilisées avec une précision de 16 bits.

C.3.2 Décodage d'une décision (DECODE)

Le décodeur décode une décision binaire à la fois. Après le décodage de la décision, le décodeur soustrait une quantité quelconque des données d'image comprimées que le codeur a ajoutées. La quantité laissée dans les données d'image comprimées est le décalage entre la base de l'intervalle actuel jusqu'au sous-intervalle attribué à toutes les décisions binaires non encore décodées. Dans le premier essai de la procédure DECODE illustrée dans la Figure C.15, le registre Haut est comparé à la longueur du sous-intervalle du symbole LPS. Sauf si un échange conditionnel est requis, cet essai détermine si un symbole MPS ou LPS est décodé. Si le registre Haut est logiquement supérieur à ou égal à

l'estimation de probabilité Q_e du symbole LPS pour l'index actuel I mémorisé à CX , alors le registre Chaut est décrémenté de cette quantité. Si A n'est pas inférieure à $0x8000$, alors le sens du symbole MPS mémorisé à CX sert à activer la décision D décodée.

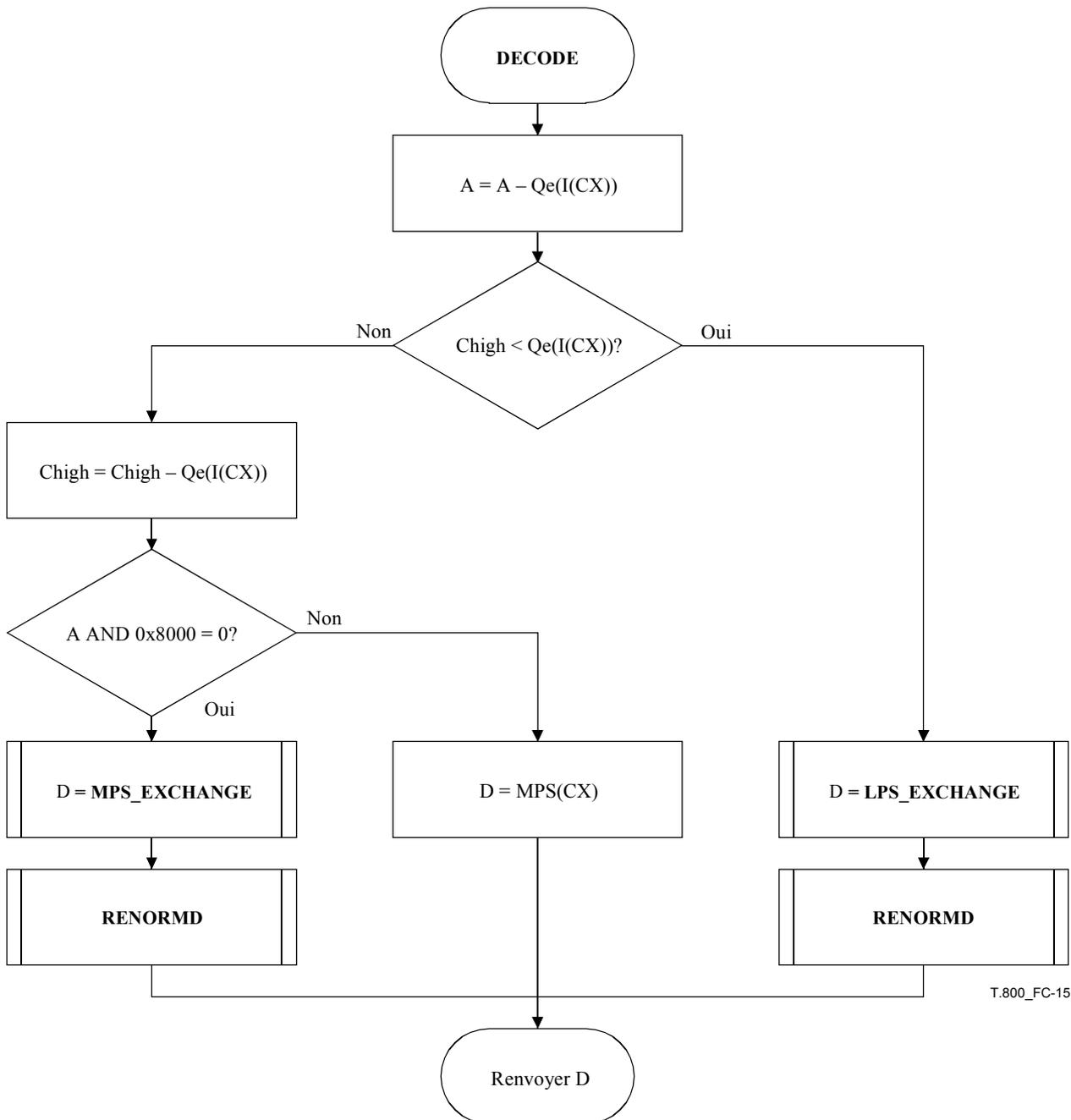
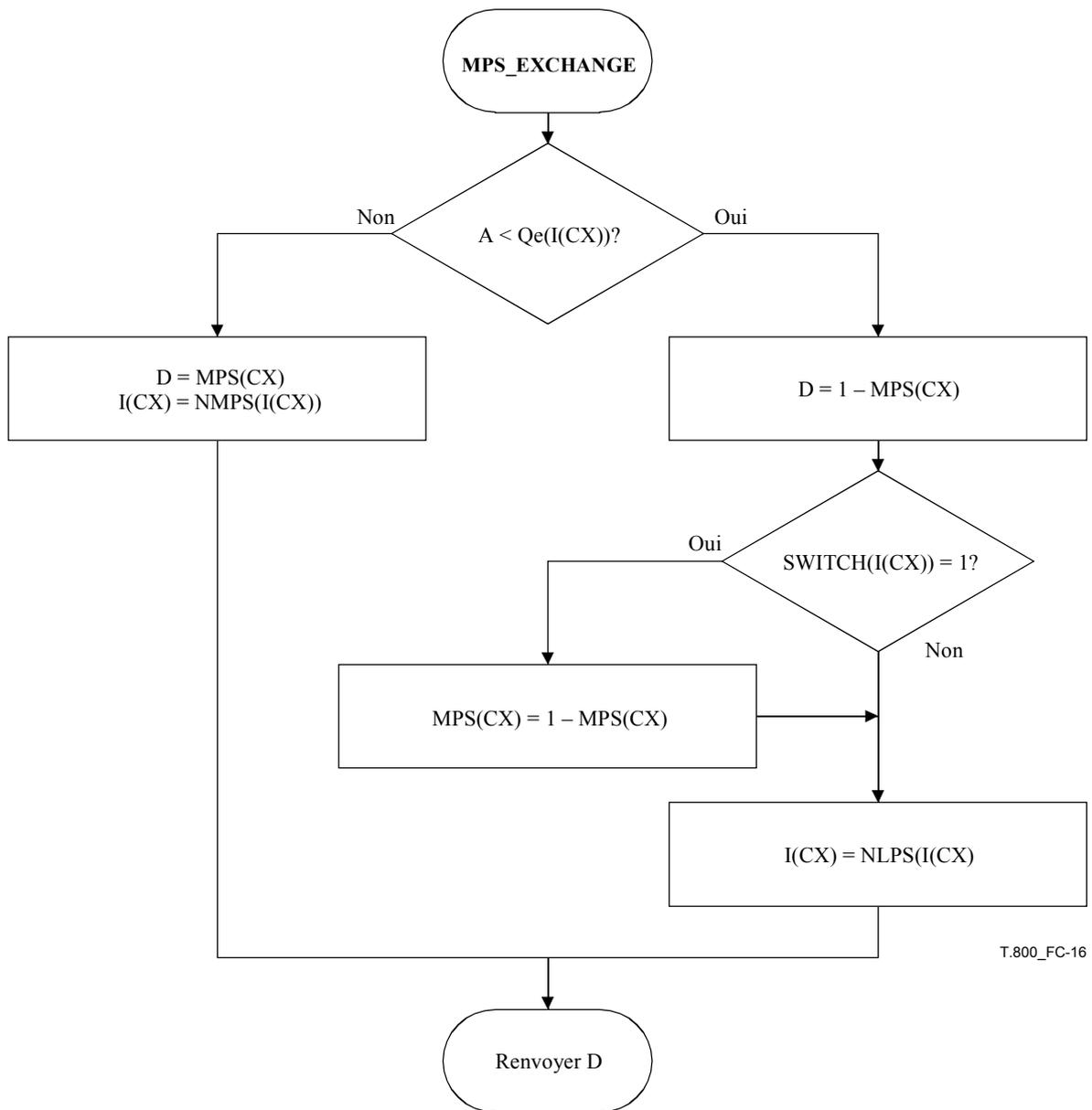


Figure C.15 – Décodage d'un symbole MPS ou d'un symbole LPS

Lorsqu'une renormalisation est requise, l'échange conditionnel de symbole MPS/LPS peut s'être produit. Pour le trajet de symbole MPS, la procédure d'échange conditionnel est représentée dans la Figure C.16. Du moment que la longueur A du sous-intervalle du symbole MPS calculée en première étape dans la Figure C.16 n'est pas logiquement inférieure à l'estimation de probabilité $Q_e(I(CX))$ du symbole LPS, un symbole MPS a bien apparu et la décision peut être prise d'après $MPS(CX)$. Alors l'index $I(CX)$ est mis à jour d'après la colonne du prochain index MPS (NMPS) dans le Tableau C.2. Si, cependant, le sous-intervalle du symbole LPS est plus grand, l'échange conditionnel s'est produit et un symbole LPS est apparu. La décision D est activée par inversion de $MPS(CX)$. La mise à jour de probabilité commute le sens du symbole MPS si la colonne SWITCH possède un "1" et met à jour l'index $I(CX)$ d'après la colonne du prochain index de symbole LPS (NLPS) dans le Tableau C.2. L'estimation de probabilité dans le décodeur doit être identique à l'estimation de probabilité dans le codeur.



T.800_FC-16

Figure C.16 – Procédure d'échange conditionnel sur le trajet de symbole MPS dans le décodeur

Pour le trajet de symbole LPS du décodeur, la procédure d'échange conditionnel reçoit la procédure LPS_EXCHANGE représentée dans la Figure C.17. La même comparaison logique entre le sous-intervalle du symbole MPSA et le sous-intervalle du symbole LPSQe(I(CX)) détermine si un échange conditionnel est apparu. Dans les deux trajets, le nouveau sous-intervalle A est mis à Qe(I(CX)). Dans le trajet de gauche l'échange conditionnel est apparu, de sorte que la décision et la mise à jour sont pour le cas du symbole MPS. Dans le trajet de droite, la décision et la mise à jour du symbole LPS sont suivies.

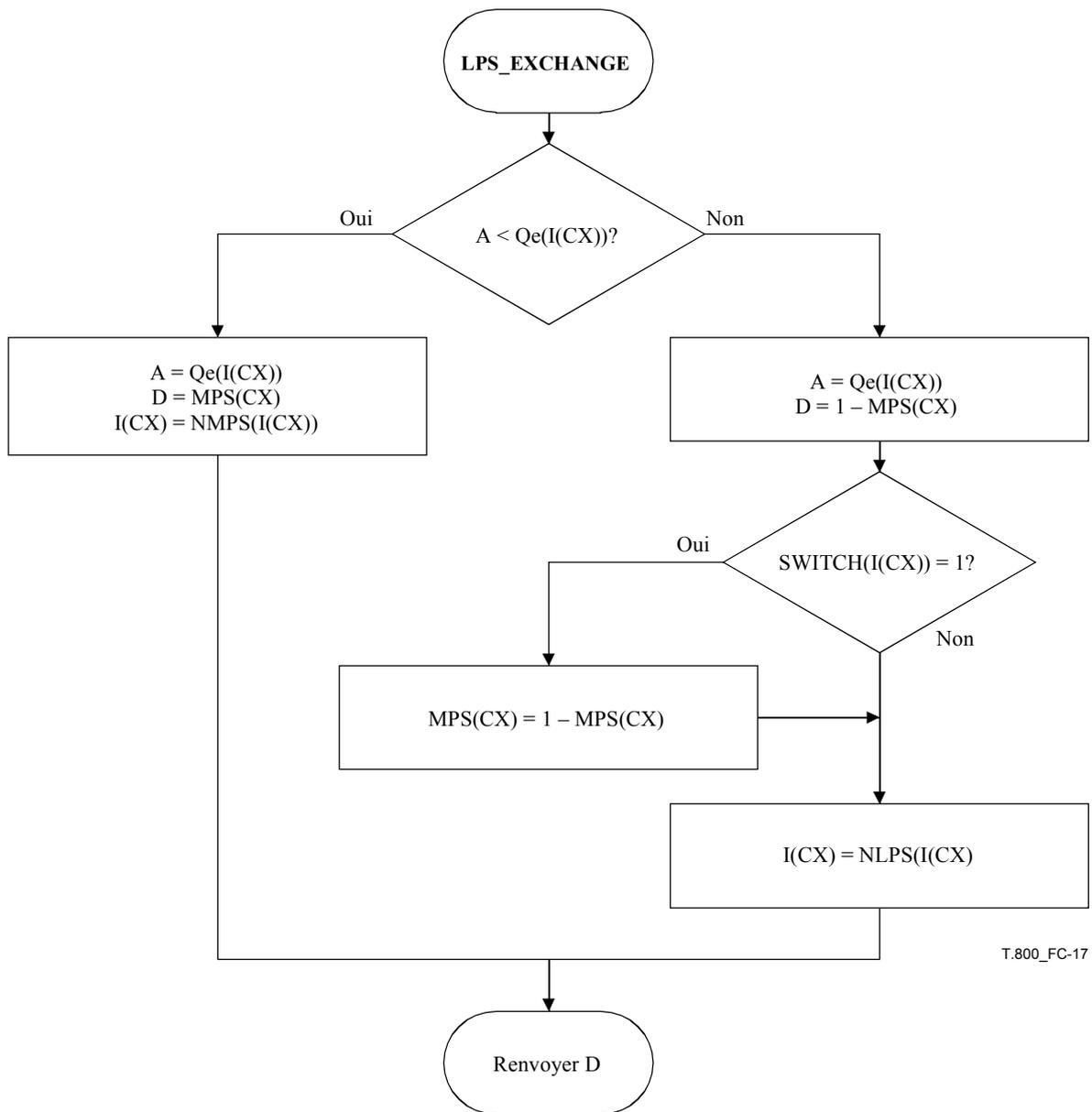


Figure C.17 – Procédure d'échange conditionnel sur le trajet de symbole LPS dans le décodeur

C.3.3 Renormalisation dans le décodeur (RENORMD)

La procédure RENORMD de renormalisation du codeur est illustrée dans la Figure C.18. Un compteur garde trace du nombre de bits comprimés dans l'étage Cbas du registre C. Lorsque CT est zéro, un nouvel octet est inséré dans Cbas par la procédure BYTEIN. Le registre C de cette procédure est la concaténation des registres Chaut et Cbas.

Le registre d'intervalle A et le registre de code C sont tous les deux décalés, un bit à la fois, jusqu'à ce que A ne soit plus inférieur à $0x8000$.

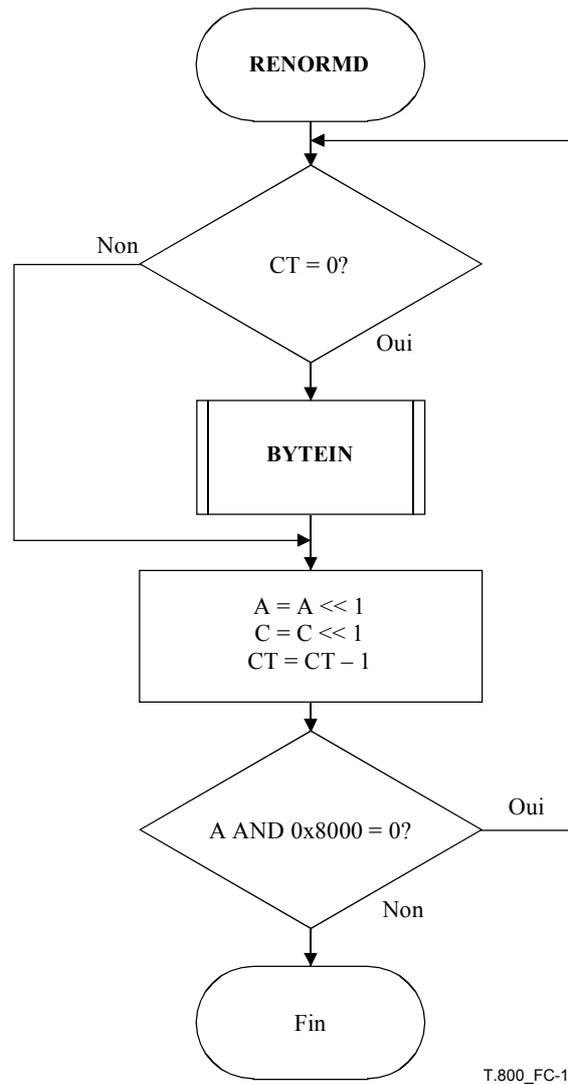


Figure C.18 – Procédure de renormalisation du décodeur

C.3.4 Entrée des données d'image comprimées (BYTEIN)

La procédure BYTEIN appelée par RENORMD est illustrée dans la Figure C.19. Cette procédure lit un seul octet de données, en compensant d'éventuels bits de bourrage suivant l'octet 0xFF dans le processus. Elle détecte également les codes de marqueur qui doivent toujours apparaître à la fin d'une passe de codage. Le registre C dans cette procédure est la concaténation des registres Chaut et Cbas.

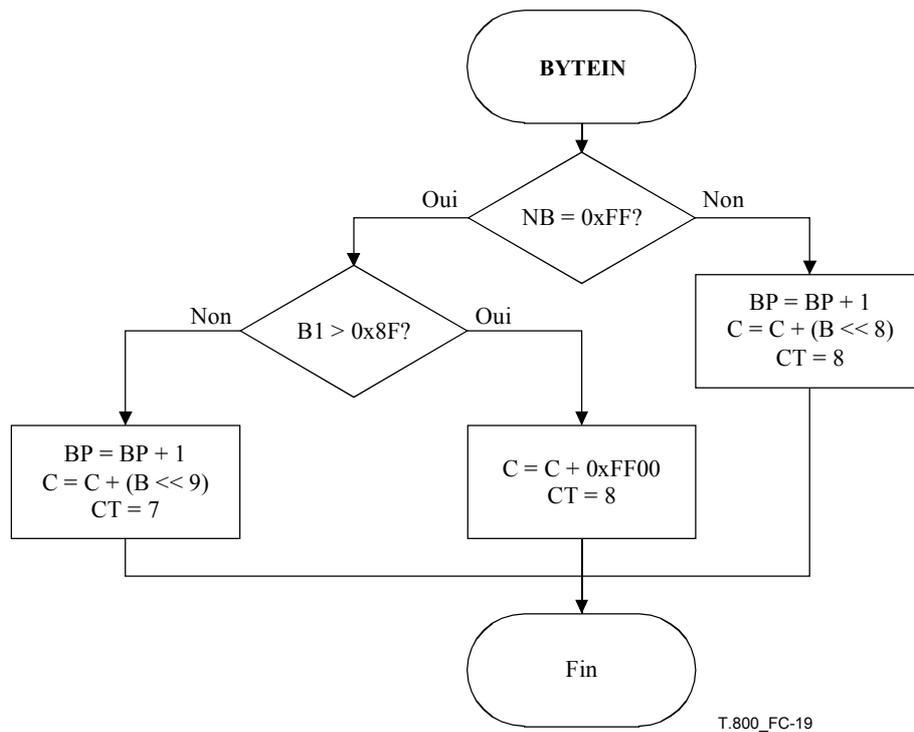


Figure C.19 – Procédure BYTEIN pour décodeur

B est l'octet désigné par le pointeur BP dans le tampon de données d'image comprimées. Si B n'est pas un octet 0xFF, le pointeur BP est incrémenté et la nouvelle valeur de B est insérée dans les 8 bits d'ordre supérieur de Cbas.

Si B est un octet 0xFF, alors B1 (l'octet pointé par BP+1) est essayé. Si B1 est supérieur à 0x8F, alors B1 doit toujours être un des codes de marqueur. Le code de marqueur est interprété comme requis, et le pointeur du tampon continue à désigner le préfixe 0xFF du code de marqueur qui termine les données d'image arithmétiquement comprimées. Des bits 1 sont alors injectés dans le décodeur jusqu'à ce que le décodage soit complet. Cela est représenté par adjonction de la valeur 0xFF00 au registre C et par réglage du compteur de bits CT à 8.

Si B1 n'est pas un code de marqueur, alors BP est incrémenté de façon à pointer sur le prochain octet qui contient un bit inséré. L'octet B est ajouté au registre C avec un alignement tel que le bit de bourrage (qui contient un éventuel report) soit ajouté au bit d'ordre inférieur de Chaut.

C.3.5 Initialisation du décodeur (INITDEC)

La procédure INITDEC sert à lancer le décodeur arithmétique. Une fois que MPS et I sont initialisés, les étapes de base sont représentées dans la Figure C.20.

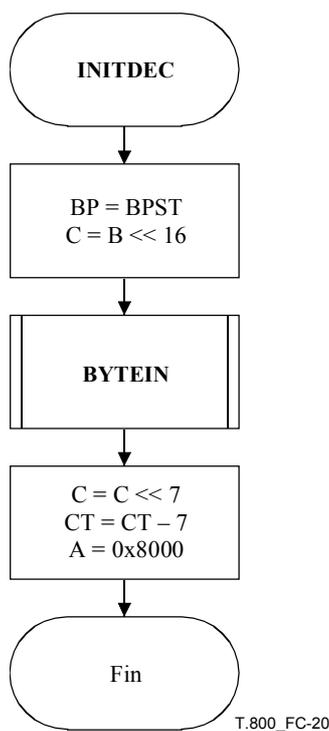


Figure C.20 – Initialisation du décodeur

BP, le pointeur sur les données d'image comprimées, est initialisé à BPST (pointant sur le premier octet comprimé). Le premier octet des données d'image comprimées est décalé dans l'octet d'ordre inférieur de Chaut, et un nouvel octet est alors introduit par lecture. Le registre C est alors décalé de 7 bits et CT est décrémenté de 7, mettant le registre C en alignement avec la valeur de départ de A. Le registre d'intervalle A est mis à la valeur de départ dans le codeur. Les réglages initiaux de MPS et I sont représentés dans le Tableau D.7.

C.3.6 Statistiques de réinitialisation du codage arithmétique

A certains points pendant le décodage, certaines ou la totalité des statistiques de codage arithmétique sont réinitialisées. Ce processus implique le retour de $I(CX)$ et $MPS(CX)$ à leurs valeurs initiales comme défini dans le Tableau D.7 pour certaines ou la totalité des valeurs de CX.

C.3.7 Sauvegarde des statistiques de codage arithmétique

Dans certains cas, le décodeur a besoin de sauvegarder ou de restaurer certaines valeurs de $I(CX)$ et de $MPS(CX)$.

Annexe D

Modélisation des bits de coefficient

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe définit la modélisation et le balayage de bits de coefficient de transformée.

Les blocs codés (voir Annexe B) sont décodés plan binaire par plan binaire à partir du plan binaire de plus fort poids contenant un élément différent de zéro jusqu'au plan binaire de plus faible poids. Pour chaque plan binaire dans un bloc codé, une structure spéciale d'exploration des blocs codés sert à chacune de trois passes de codage. Chaque bit de coefficient dans le plan binaire apparaît dans une seule des trois passes de codage appelées: propagation de signification, raffinement de grandeur et nettoyage. Pour chaque passe, des contextes sont créés puis fournis au codeur arithmétique, CX, en même temps que le flux binaire, CD, (voir § C.3).

D.1 Structure d'exploration des blocs codés dans les blocs codés

Chaque plan binaire d'un bloc codé est balayé dans un ordre particulier. A partir du coin supérieur gauche, les quatre premiers coefficients de la première colonne sont balayés, suivis des quatre premiers coefficients de la seconde colonne et ainsi de suite jusqu'à ce que le côté droit du bloc codé soit atteint. Le balayage revient alors vers la gauche du bloc codé et le second ensemble de quatre coefficients dans chaque colonne est balayé. Le processus est poursuivi jusqu'en bas du bloc codé. Si la hauteur du bloc codé n'est pas divisible par 4, le dernier ensemble de coefficients balayé dans chaque colonne contiendra moins de 4 membres. La Figure D.1 montre un exemple de la structure d'exploration des blocs codés pour un bloc codé.

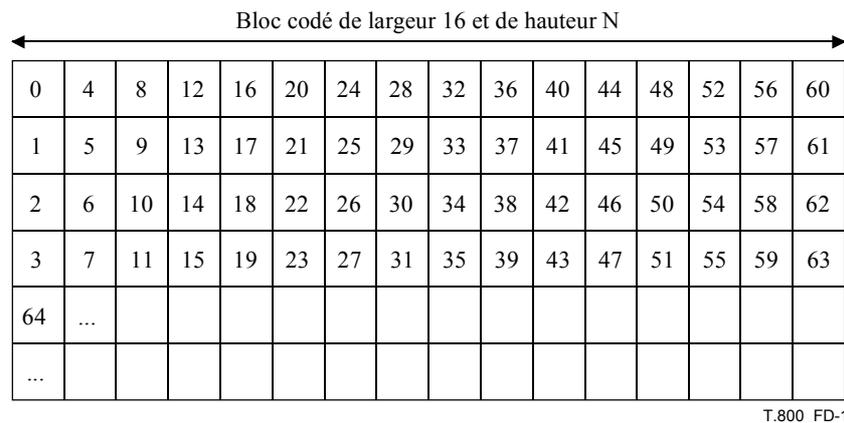


Figure D.1 – Exemple de structure d'exploration du plan binaire d'un bloc codé

D.2 Bits de coefficient et signification

D.2.1 Notations de cas généraux

Les procédures de décodage spécifiées dans la présente annexe produisent pour chaque coefficient de transformée (u, v) de sous-bande b les bits décodés qui seront utilisés pour reconstruire la valeur du coefficient de transformée $q_b(u, v)$. Les bits produits sont les suivants: un bit de signe $s_b(u, v)$ et un certain nombre de bits MSB $N_b(u, v)$ de grandeur décodés, ordonnés de plus fort poids à plus faible poids: $MSB_i(b, u, v)$ est l' i^{e} MSB de coefficient de transformée (u, v) de sous-bande b ($i = 1, \dots, N_b(u, v)$). Comme indiqué dans l'équation (D-1), le bit de signe $s_b(u, v)$ possède une valeur de 1 pour les coefficients négatifs et de zéro pour les coefficients positifs. Le nombre $N_b(u, v)$ de bits MSB décodés comporte le nombre de tous les plans binaires zéro de plus fort poids signalés dans l'en-tête de paquet (voir § B.10.5).

D.2.2 Notation dans le cas d'une région ROI

Dans le cas de la présence du segment marqueur RGN (indiquant la présence d'une région RoI), des modifications doivent être apportées aux bits décodés, ainsi qu'au nombre de bits décodés $N_b(u, v)$. Ces modifications sont spécifiées au § H.1. En l'absence du segment marqueur RGN, aucune modification n'est requise.

D.3 Passes de décodage sur les plans binaires

Chaque coefficient contenu dans un bloc codé possède une variable d'état binaire appelée *état de signification* de ce bloc. Les états de signification sont initialisés à 0 (le coefficient est non significatif) et peuvent devenir 1 (le coefficient est significatif) au cours du décodage du bloc codé. "L'état de signification" modifie de non significatif à significatif (voir le paragraphe ci-dessous) dans le plan binaire où le bit de plus fort poids de grandeur égal à 1 est trouvé. Le vecteur de contexte pour un certain coefficient actuel est le vecteur binaire composé des états de signification de ses 8 coefficients voisins les plus proches, comme représenté dans la Figure D.2. Chaque voisin le plus proche se trouvant hors du bloc codé du coefficient actuel est considéré comme non significatif (c'est-à-dire qu'il est traité comme ayant un état de signification zéro) pour l'objet créant un vecteur de contexte afin de décoder le coefficient actuel.

D ₀	V ₀	D ₁
H ₀	X	H ₁
D ₂	V ₁	D ₃

T.800_FD-2

Figure D.2 – Etats de voisinage utilisés pour former le contexte

En général, un coefficient actuel peut avoir 256 vecteurs de contexte. Ces vecteurs sont regroupés en un plus petit nombre de contextes selon les règles spécifiées ci-dessous pour la formation de contexte. Quatre règles différentes de formation de contexte sont définies, une pour chacune des quatre passes de codage: codage de signification, codage de signe, codage de raffinement de grandeur, et codage de nettoyage. Ces opérations de codage sont effectuées en trois passes de codage sur chaque plan binaire: codage de signification et de signe dans une passe de propagation de signification, codage de raffinement de grandeur dans une passe de raffinement de grandeur, et codage de nettoyage et de signe dans une passe de nettoyage. Pour une opération de codage donnée, l'étiquette de contexte (ou contexte) fournie à l'automate de codage arithmétique est une étiquette attribuée au contexte du coefficient actuel.

NOTE – Bien que (aux fins de la concrétisation) des entiers spécifiques soient utilisés dans les tableaux ci-dessous pour étiqueter des contextes, les jetons utilisés pour les étiquettes de contexte dépendent de l'implémentation et leurs valeurs ne sont pas prescrites par la présente Recommandation | Norme internationale.

Le premier plan binaire dans le bloc actuel contenant un élément différent de zéro possède une seule passe de nettoyage. Les plans binaires restants sont décodés en trois passes de codage. Chaque bit de coefficient est décodé dans exactement une des trois passes de codage. La passe par laquelle un bit de coefficient est décodé dépend des conditions de cette passe. En général, la passe de propagation de signification comporte les coefficients qui sont prévus ou "très susceptibles," de devenir significatifs, avec leur bit de signe selon le cas. La passe de raffinement de grandeur comporte les bits extraits de coefficients déjà significatifs. La passe de nettoyage comporte tous les coefficients restants.

D.3.1 Passe de décodage par propagation de signification

Les huit coefficients voisins qui entourent un coefficient actuel (montrés dans la Figure D.2 où X indique le coefficient actuel) sont utilisés pour créer un vecteur de contexte qui est affecté dans un des 9 contextes représentés dans le Tableau D.1. Si un coefficient est significatif, alors il reçoit une valeur 1 pour la création du contexte, sinon il reçoit une valeur 0. Le mappage aux contextes dépend également de la sous-bande.

Tableau D.1 – Contexte des passes de codage de propagation de signification et de nettoyage

Sous-bandes LL et LH (passe-haut vertical)			Sous-bande HL (passe-haut horizontal)			Sous-bande HH (passe-haut diagonal)		Étiquette de contexte ^{a)}
$\sum H_i$	$\sum V_i$	$\sum D_i$	$\sum H_i$	$\sum V_i$	$\sum D_i$	$\sum(H_i + V_i)$	$\sum D_i$	
2	x ^b	x	x	2	x	x	≥ 3	8
1	≥ 1	x	≥ 1	1	x	≥ 1	2	7
1	0	≥ 1	0	1	≥ 1	0	2	6
1	0	0	0	1	0	≥ 2	1	5
0	2	x	2	0	x	1	1	4
0	1	x	1	0	x	0	1	3
0	0	≥ 2	0	0	≥ 2	≥ 2	0	2
0	0	1	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0

a) Noter que les étiquettes de contexte ne sont indexées qu'aux fins de l'identification dans la présente Recommandation | Norme internationale. Les identificateurs réellement utilisés relèvent de l'implémentation.

b) "x" indique un état indifférent.

La passe de propagation de signification ne comporte que les bits des coefficients qui ont été non significatifs (l'état de signification doit encore être déterminé) et qui ont un contexte différent de zéro. Tous les autres coefficients sont omis. Le contexte est remis au décodeur arithmétique (avec le flux binaire) et le bit de coefficient décodé est renvoyé. Si la valeur de ce bit est 1, alors l'état de signification est mis à 1 et le tout prochain bit à décoder est le bit de signe pour le coefficient. Sinon, l'état de signification reste 0. Lorsque les contextes de coefficients et de passes de codage successifs sont pris en considération, l'état de signification le plus actuel est utilisé pour ce coefficient.

D.3.2 Décodage de bit de signe

L'étiquette de contexte pour le décodage de bit de signe est déterminée au moyen d'un autre vecteur de contexte d'après le voisinage. Le calcul de l'étiquette de contexte peut être considéré comme un processus en deux étapes. La première étape résume la contribution du voisinage vertical et du voisinage horizontal. La seconde étape diminue ces contributions à une étiquette de contexte sur 5.

Pour la première étape, les deux voisinages verticaux (voir Figure D.2) sont considérés ensemble. Chaque voisin peut avoir un des trois états: significatif positif, significatif négatif, ou non significatif. Si les deux voisinages verticaux sont tous les deux significatifs avec le même signe, ou si un seul est significatif, alors la contribution verticale est 1 si le signe est positif ou -1 si le signe est négatif. Si les deux voisinages verticaux sont non significatifs, ou si tous les deux sont significatifs avec des signes différents, alors la contribution verticale est 0. La contribution horizontale est créée de la même façon. De nouveau, si les voisinages s'inscrivent hors du bloc codé, ils sont considérés comme étant non significatifs. Le Tableau D.2 montre ces contributions.

Tableau D.2 – Contributions des voisinages verticaux (et horizontaux) au contexte de signe

V0 (ou H0)	V1 (ou H1)	Contribution V (ou H)
significatif, positif	significatif, positif	1
significatif, négatif	significatif, positif	0
non significatif	significatif, positif	1
significatif, positif	significatif, négatif	0
significatif, négatif	significatif, négatif	-1
non significatif	significatif, négatif	-1
significatif, positif	non significatif	1
significatif, négatif	non significatif	-1
non significatif	non significatif	0

La seconde étape réduit les neuf permutations de contributions verticales et horizontales à 5 étiquettes de contexte. Le Tableau D.3 montre ces étiquettes de contexte. Ce contexte est fourni au décodeur arithmétique avec le flux binaire. Le bit renvoyé, D (voir Annexe C), est alors logiquement combiné par OU exclusif avec le bit $XORbit$ du Tableau D.3 afin de produire le bit de signe. L'équation suivante est utilisée:

$$signbit = D \otimes XORbit \quad (D-1)$$

où $signbit$ est le bit de signe du coefficient actuel (un bit 1 indique un coefficient négatif, un bit zéro un coefficient positif), D est la valeur renvoyée par le décodeur arithmétique en fonction de l'étiquette de contexte et du flux binaire, et le bit $XORbit$ se trouve dans le Tableau D.3 pour l'étiquette de contexte actuelle.

Tableau D.3 – Contextes de signe issus des contributions verticales et horizontales

Contribution horizontale	Contribution verticale	Etiquette de contexte	XORbit
1	1	13	0
1	0	12	0
1	-1	11	0
0	1	10	0
0	0	9	0
0	-1	10	1
-1	1	11	1
-1	0	12	1
-1	-1	13	1

D.3.3 Passe de raffinement de grandeur

La passe de raffinement de grandeur comporte les bits extraits des coefficients qui sont déjà significatifs (sauf ceux qui viennent de devenir significatifs dans la passe de propagation de signification immédiatement antérieure).

Le contexte utilisé est déterminé par la sommation des états de signification des voisinages horizontal, vertical et diagonal. Ce sont les états actuellement connus dans le décodeur, et non les états utilisés avant la passe de décodage de signification. Par ailleurs, le contexte dépend de la question de savoir si c'est le premier bit de raffinement (le bit qui suit immédiatement les bits de signification et de signe) ou non. Le Tableau D.4 montre les trois contextes pour cette passe.

Tableau D.4 – Contexte des passes de codage de raffinement de grandeur

$\sum H_i + \sum V_i + \sum D_i$	Premier raffinement pour ce coefficient	Etiquette de contexte
$x^a)$	Faux	16
≥ 1	Vrai	15
0	Vrai	14

^{a)} "x" indique un état indifférent.

Le contexte est transmis au codeur arithmétique ainsi que le flux binaire. Le bit renvoyé est la valeur du coefficient actuel dans le plan binaire actuel.

D.3.4 Passe de nettoyage

Les coefficients restants étaient déjà non significatifs et non traités par la passe de propagation de signification. La passe de nettoyage non seulement utilise le contexte voisin, comme celui de la passe de propagation de signification d'après le Tableau D.1, mais également un contexte de longueur de plage.

Au cours de cette passe, les contextes voisins pour les coefficients de cette passe sont recréés au moyen du Tableau D.1. L'étiquette de contexte peut maintenant avoir une valeur quelconque parce que les coefficients qui avaient été trouvés significatifs dans la passe de propagation de signification sont considérés comme significatifs dans la passe de nettoyage. Les longueurs de plage sont décodées avec un unique contexte. Si les quatre coefficients contigus dans la colonne balayée sont tous décodés dans la passe de nettoyage et si l'étiquette de contexte pour tous les coefficients est 0 (y compris les coefficients de contexte issus de précédentes passes de grandeur, de signification et de nettoyage), alors

l'unique contexte de longueur de plage est transmis au décodeur arithmétique ainsi que le flux binaire. Si le symbole 0 est renvoyé, alors les quatre coefficients contigus dans la colonne restent non significatifs et sont mis à zéro.

Sinon, si le symbole 1 est renvoyé, alors au moins un des quatre coefficients contigus dans la colonne est significatif. Les deux bits suivants, renvoyés avec le contexte uniforme (index 46 dans le Tableau C.2), indiquent le coefficient qui est le premier à être trouvé significatif du haut en bas de la colonne. Les deux bits décodés avec le contexte uniforme sont décodés dans l'ordre MSB puis LSB. Le bit de signe du coefficient est déterminé comme décrit au § D.3.2. Le décodage d'éventuels coefficients restants continue de la façon décrite au § D.3.1.

Si les quatre coefficients contigus dans une colonne ne sont pas tous décodés dans la passe de nettoyage ou si le segment de contexte pour l'un de ces coefficients est différent de zéro, alors les bits de coefficient sont décodés avec le contexte indiqué dans le Tableau D.1 comme dans la passe de propagation de signification. Les mêmes contextes que dans la propagation de signification sont utilisés ici (l'état est utilisé ainsi que le modèle). Le Tableau D.5 montre la logique pour la passe de nettoyage.

Tableau D.5 – Décodeur de longueur de plage pour passes de nettoyage

Quatre coefficients contigus dans une colonne restant à décoder et ayant chacun le contexte 0	Symboles avec contexte de longueur de plage	Quatre bits contigus à décoder sont à zéro	Symboles décodés avec UNIFORM ^{a)}	Nombre de coefficients à décoder
Vrai	0	Vrai	Néant	Néant
Vrai	1	Faux sauter au premier signe de coefficient sauter au second signe de coefficient sauter au troisième signe de coefficient sauter au quatrième signe de coefficient	MSB LSB 00 01 10 11	3 2 1 0
Faux	Néant	x	Néant	Reste de colonne

^{a)} Voir l'Annexe C.

S'il y a moins de quatre rangées restant dans un bloc codé, alors aucun codage de longueur de plage n'est utilisé. De nouveau, l'état de signification d'un coefficient quelconque est modifié immédiatement après le décodage du premier bit 1 de grandeur.

D.3.5 Exemple de passes de codage et de propagation de signification (pour information)

Le Tableau D.6 montre un exemple de l'ordre de décodage pour les coefficients quantifiés d'une colonne de 4 coefficients dans le balayage. Cet exemple implique que tous les voisinages non inclus dans le tableau sont exactement à zéro, et indique la passe au cours de laquelle chaque bit est décodé. Le bit de signe est décodé après le bit 1 initial et est indiqué dans le tableau par le signe + ou -. La toute première passe dans un nouveau bloc est toujours une passe de nettoyage parce qu'il ne peut y avoir de bits significatifs ou de raffinement prédits. Après la première passe, le bit 1 décodé du premier coefficient provoque le décodage du second coefficient dans la passe de signification pour le prochain plan binaire. Le bit 1 décodé pour le dernier coefficient dans la seconde passe de nettoyage provoque le décodage du troisième coefficient dans la prochaine passe de signification.

Tableau D.6 – Exemple d'ordre de codage de sous-plan binaire et de propagation de signification

Passes de codage	10	1	3	-7	Valeur du coefficient
	+	+	+	-	Signe de coefficient
	1	0	0	0	Poids du coefficient (MSB à LSB)
	0	0	0	1	
1	0	1	1		
0	1	1	1		
Nettoyage	1+	0	0	0	
Signification		0			
Raffinement	0				
Nettoyage			0	1-	
Signification		0	1+		
Raffinement	1			1	
Nettoyage					
Signification		1+			
Raffinement	0		1	1	
Nettoyage					

D.4 Initialisation et terminaison

Lorsque les contextes sont initialisés, ou réinitialisés, ils sont mis aux valeurs indiquées dans le Tableau D.7.

Tableau D.7 – Etats initiaux pour tous les contextes

Contexte	Index initial d'après le Tableau C.3	MPS
UNIFORM	46	0
Longueur de plage	3	0
Tous voisinages zéro (étiquette de contexte 0 dans le Tableau D.1)	4	0
Tous autres contextes	0	0

En fonctionnement normal (sans évitement sélectif du codage arithmétique), le codeur arithmétique doit être terminé soit à la fin de chaque passe de codage ou seulement à la fin de chaque bloc codé (voir § D.4.1). Le Tableau D.8 montre deux exemples de structures de terminaison pour les passes de codage dans un bloc codé. Le marqueur COD ou COC signale la structure de terminaison qui est utilisée (voir § A.6.1 et A.6.2).

Tableau D.8 – Structures de terminaison de codeur arithmétique

#	Passe	Terminaison d'opération de codage seulement à la dernière passe	Terminaison d'opération de codage à chaque passe
1	Nettoyage	Codeur arithmétique (AC)	AC, terminer
2	Propagation de signification	AC	AC, terminer
2	Raffinement de grandeur	AC	AC, terminer
2	Nettoyage	AC	AC, terminer
...
final	Propagation de signification	AC	AC, terminer
final	Raffinement de grandeur	AC	AC, terminer
final	Nettoyage	AC, terminer	AC, terminer

Lorsque de multiples terminaisons du codeur arithmétique sont présentées, la longueur de chaque segment terminé est signalée dans l'en-tête de paquet comme décrit au § B.10.7.

NOTE – Une terminaison ne devrait jamais créer de valeur alignée en octets comprise entre 0xFF90 et 0xFFFF inclus. Ces valeurs sont disponibles en tant que valeurs de marqueur dans le flux binaire.

D.4.1 Terminaison de flux codé attendue

Le décodeur prévoit que le nombre indiqué d'octets de flux codé décodera un certain nombre de passes de codage avant que le codeur arithmétique soit terminé. Pendant le décodage, les octets sont successivement extraits du flux codé jusqu'à ce que tous les octets pour ces passes de codage aient été consommés. Le nombre d'octets correspondant aux passes de codage est spécifié dans l'en-tête de paquet. A ce point, il y a souvent plus de symboles à décodé. Donc, le décodeur doit étendre le flux binaire d'entrée au codeur arithmétique avec des octets 0xFF, autant de fois que nécessaire, jusqu'à ce que tous les symboles aient été décodés.

Il suffit d'adjoindre deux octets 0xFF au plus. Cela aura pour effet que le codeur arithmétique aura au moins une paire d'octets 0xFF consécutifs à son entrée, qui est interprétée comme un marqueur de fin de flux (voir § C.3.4). Le flux binaire ne contient pas en fait de marqueur de terminaison. Cependant, la longueur d'octet est explicitement signalée, ce qui permet de synthétiser le marqueur de terminaison pour le décodeur arithmétique.

NOTE – Deux octets 0xFF adjoints de cette façon est la méthode la plus simple. Cependant, d'autres extensions équivalentes existent. Cela peut être important car certaines implémentations de codeur arithmétique pourraient donner une signification spéciale au marqueur de terminaison spécifique.

D.4.2 Terminaison de codeur arithmétique

La procédure FLUSH exécute cette tâche (voir § C.2.9). Cependant, comme la procédure FLUSH augmente la longueur du flux codé, et qu'une terminaison fréquente peut être désirable, d'autres techniques peuvent être employées. Toute technique qui place tous les octets nécessaires dans le flux codé de telle façon que le décodeur n'ait pas besoin de revenir en arrière pour trouver la position à laquelle le prochain segment du flux codé devrait commencer est acceptable.

Lorsque le fanion de terminaison prévisible est activé (voir COD et COC aux § A.6.1 et A.6.2) la procédure suivante de terminaison doit être utilisée. Au moyen de la notation du § C.2, les étapes suivantes peuvent être suivies:

- 1) identifier le nombre de bits contenus dans le registre de code, C, qui doit toujours être extrait par le tampon d'octets. Cela est donné par $k = (11 - CT) + 1$;
- 2) tant que ($k > 0$):
 - décaler C vers la gauche de CT et mettre $CT = 0$;
 - exécuter la procédure BYTEOUT, qui rend CT égal au nombre de bits effacés du registre C;
 - soustraire CT de k;
- 3) exécuter la procédure BYTEOUT afin d'extraire le contenu du registre du tampon d'octets vers le flux codé. Cette étape doit être omise si l'octet contenu dans le tampon d'octets possède une valeur d'octet 0xFF.

La longueur de troncature applicable dans ce cas est simplement le nombre total d'octets extrait vers le flux codé.

Si le fanion de terminaison prévisible n'est pas activé, le dernier octet émis en sortie par la procédure ci-dessus peut généralement être modifié, dans certaines limites, sans affecter les symboles à décodé. Il sera parfois possible d'augmenter le dernier octet jusqu'à la valeur spéciale, 0xFF, qui ne doit pas être envoyée. L'on peut montrer que cela se produit approximativement 1/8 du temps.

D.4.3 Calcul de longueur (pour information)

Afin d'inclure les données d'image comprimées de la passe de codage dans des paquets, le nombre d'octets à inclure doit toujours être déterminé. Si les données d'image comprimées de la passe de codage sont terminées, l'algorithme indiqué dans le précédent paragraphe peut être utilisé. Sinon, le codeur doit calculer une longueur appropriée telle que les octets correspondants soient suffisants pour que le décodeur puisse reconstruire les passes de codage.

D.5 Symbole de segmentation assurant l'élasticité aux erreurs

Un symbole de segmentation est un symbole spécial. Son utilisation éventuelle est signalée dans les segments marqueurs COD ou COC (voir § A.6.1 et A.6.2). Le symbole est codé avec le contexte uniforme du codeur arithmétique à la fin de chaque plan binaire. Le décodage correct de ce symbole confirme l'exactitude du décodage de ce plan binaire, qui permet la détection d'erreur. Dans le décodeur, un symbole de segmentation "1010" ou "0xA" doit être décodé à la fin de chaque plan binaire (à la fin d'une passe de nettoyage). Si le symbole de segmentation n'est pas décodé correctement, alors des erreurs binaires sont apparues pour ce plan binaire.

NOTE – Ce symbole peut être utilisé avec ou sans la terminaison prévisible.

D.6 Evitement sélectif du codage arithmétique

Ce style de codage permet l'évitement du codeur arithmétique pendant la passe de propagation de signification et pendant les passes de codage de raffinement de grandeur commençant dans le cinquième plan binaire significatif du bloc codé. Le marqueur COD ou COC indique si ce style de codage est ou non utilisé (voir § A.6.1 et A.6.2).

La première passe de nettoyage (qui est le premier plan binaire d'un bloc codé avec un élément différent de zéro) et les trois ensembles suivants des passes de codage de propagation de signification, de raffinement de grandeur et de nettoyage sont décodés avec le codeur arithmétique. La quatrième passe de nettoyage doit inclure une terminaison de codeur arithmétique (voir Tableau D.9).

Tableau D.9 – Evitement sélectif du codage arithmétique

Numéro de plan binaire	Type de passe	Opérations de codage
1	Nettoyage	Codage arithmétique (AC)
2	Propagation de signification	AC
2	Raffinement de grandeur	AC
2	Nettoyage	AC
3	Propagation de signification	AC
3	Raffinement de grandeur	AC
3	Nettoyage	AC
4	Propagation de signification	AC
4	Raffinement de grandeur	AC
4	Nettoyage	AC, terminer
5	Propagation de signification	de base
5	Raffinement de grandeur	de base, terminer
5	Nettoyage	AC, terminer
...
final	Signification	de base
final	Raffinement de grandeur	de base, terminer
final	Nettoyage	AC, terminer

A partir de la quatrième passe de codage de propagation de signification et de raffinement de grandeur, les bits qui auraient été renvoyés par le codeur arithmétique sont au contraire renvoyés directement à partir du flux binaire. (Une routine qui annule les effets de bourrage binaire précède le retour de bits. Spécifiquement, cette routine rejette le premier bit après une valeur d'octet 0xFF.) Après chaque passe de raffinement de grandeur, le flux binaire est "terminé" par bourrage jusqu'à la limite d'octet.

Lorsque le fanion de terminaison prévisible est activé (voir COD et COC aux § A.6.1 et A.6.2) et que tous les bits issus d'une passe de raffinement de grandeur ont été assemblés, les éventuels bits restant dans le dernier octet sont remplis avec une séquence alternante de 0 et de 1. Cette séquence devrait commencer par un 0 quel que soit le nombre de bits à insérer en bourrage.

Lorsque le fanion de terminaison de passe de codage est activé (voir COD et COC aux § A.6.1 et A.6.2), alors les passes de propagation de signification sont terminées de la même façon que les passes de raffinement de grandeur.

Les passes de codage de nettoyage continuent à recevoir des données d'image comprimées directement du codeur arithmétique et sont toujours terminées.

Le bit de signe est calculé avec l'équation (D-2):

$$\text{signbit} = \text{raw_value} \quad (\text{D-2})$$

où $\text{raw_value} = 1$ est a bit de signe négatif et $\text{raw_value} = 0$ est a bit de signe positif. Le Tableau D.9 montre la séquence de codage.

La longueur de chaque segment terminé, plus celle d'éventuelles passes restant non terminées, est signalée dans l'en-tête de paquet comme décrit au § B.10.7. Si la terminaison à chaque passe de codage est choisie (voir § A.6.1 et A.6.2), alors chaque passe est terminée (y compris les deux passes de base).

NOTE 1 – L'utilisation du mode d'évitement sélectif lors du codage d'une image comportant une région ROI peut notablement diminuer le rendement de compression.

Si une valeur 0xFF est rencontrée dans le flux binaire, alors le premier bit du prochain octet est rejeté. La séquence de bits utilisée dans l'évitement sélectif du codage arithmétique a été justifiée en octets au moyen d'une routine de bourrage binaire.

Dans le codeur, les bits sont compactés en octets du bit de plus fort poids au bit de plus faible poids. Une fois qu'un octet complet est assemblé, il est émis vers le flux binaire. Si la valeur de l'octet est une valeur 0xFF, un seul bit zéro est inséré dans le bit de plus fort poids du prochain octet. Une fois que tous les bits de la passe de codage ont été assemblés, le dernier octet est compacté jusqu'à la limite d'octet et émis. Le dernier octet ne doit pas être une valeur 0xFF.

NOTE 2 – Comme le décodeur adjoint des valeurs 0xFF, autant de fois que nécessaire, au flux binaire représentant la passe de codage (voir § D.4.1), la troncature du flux binaire est possible. Lorsque le fanion de terminaison prévisible est activé (voir COD et COC dans les § A.6.1 et A.6.2), une telle troncature n'est pas permise. Le dernier octet ne peut pas être une valeur 0xFF, car la routine de bourrage binaire adjoint un nouvel octet après la valeur FF, dont le bit de plus fort poids a la valeur 0 et dont les bits inutilisés sont remplis par la séquence alternante de bits de valeur 0 ou 1.

D.7 Formation verticale du contexte causal

Ce style de codage contraint la formation de contexte aux explorations du bloc codé actuel et antérieur (quatre rangées de coefficients balayés verticalement). En d'autres termes, tout coefficient issu de l'exploration du prochain bloc codé est considéré comme étant non significatif. Le marqueur COD ou COC indique si ce style de codage est utilisé ou non (voir § A.6.1 et A.6.2).

Aux fins d'illustration, le bit étiqueté 14 dans la Figure D.1 est décodé comme de coutume au moyen des états voisins comme spécifiés dans la Figure D.2, indépendamment de la question de savoir si les contextes sont verticalement en relation causale. Cependant, lorsque la formation verticale du contexte causal est utilisée, le bit étiqueté 15 est décodé en admettant que $D_2 = V_1 = D_3 = 0$ dans la Figure D.2.

D.8 Schéma fonctionnel du codage de bloc codé

Les étapes de modélisation de chaque plan binaire de chaque bloc codé peuvent être considérées graphiquement dans la Figure D.3. Les décisions prises sont indiquées dans le Tableau D.10; les bits et le contexte envoyés au codeur sont indiqués dans le Tableau D.11. Ces tableaux montrent le contexte sans l'évitement sélectif du codage arithmétique ni le modèle en relation causale verticalement.

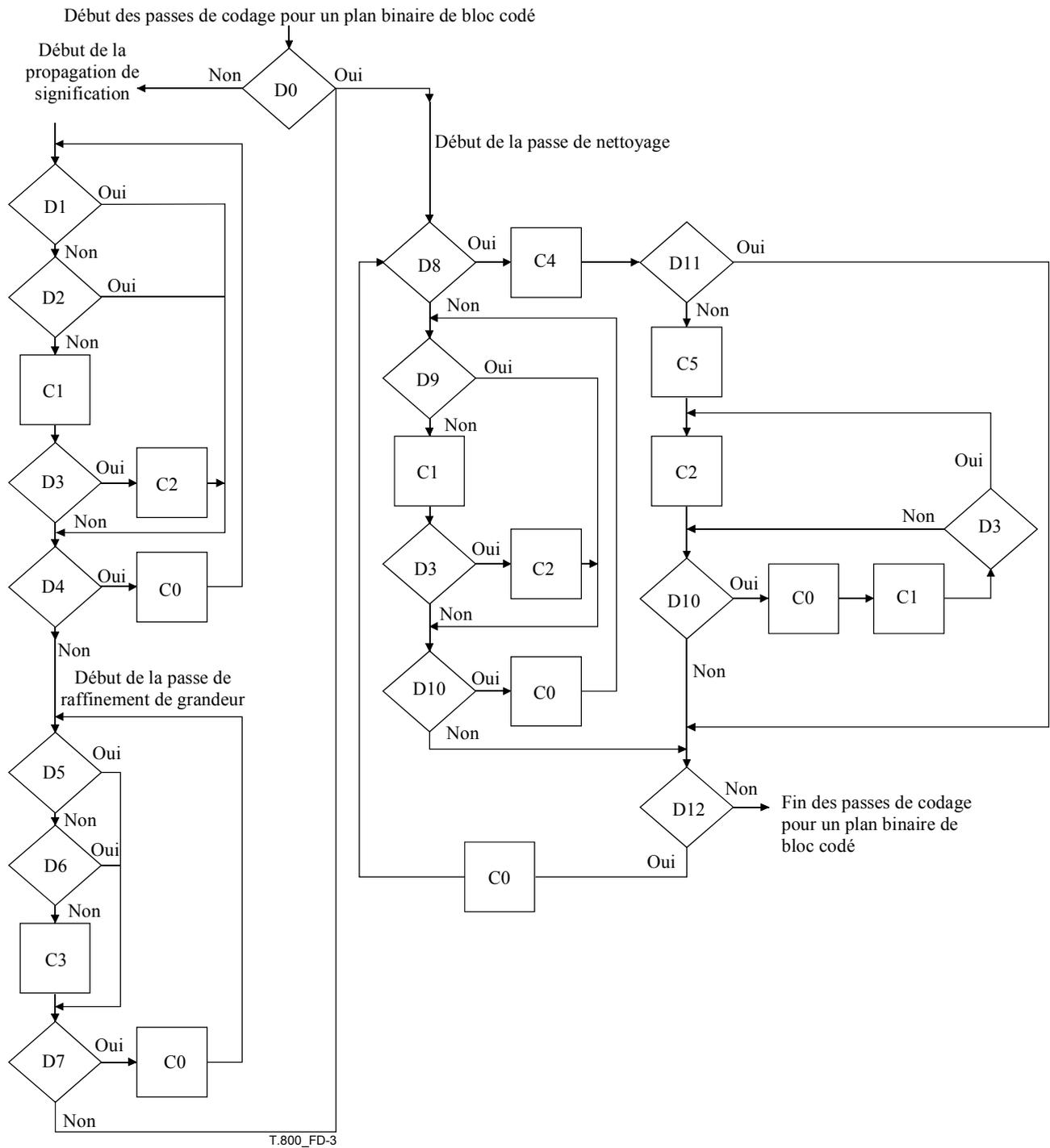


Figure D.3 – Organigramme pour toutes les passes de codage sur le plan binaire d'un bloc codé

Tableau D.10 – Décisions dans l'organigramme de modèle de contexte

Décision	Question	Description
D0	Est-ce le premier plan binaire significatif pour le bloc codé?	Voir § D.3
D1	Le coefficient actuel est-il significatif?	Voir § D.3.1
D2	Le segment de contexte est-il zéro? (voir Tableau D.1)	Voir § D.3.1
D3	Le coefficient actuel vient-il juste de devenir significatif?	Voir § D.3.1
D4	Y a-t-il encore des coefficients dans la propagation de signification?	
D5	Le coefficient est-il non significatif?	Voir § D.3.3
D6	Le coefficient codé était-il dans la dernière propagation de signification?	Voir § D.3.3
D7	Y a-t-il encore des coefficients dans la passe de raffinement de grandeur?	
D8	Y a-t-il quatre coefficients contigus non décodés dans une colonne, chacun avec un contexte 0?	Voir § D.3.4
D9	Le coefficient est-il significatif ou le bit a-t-il déjà été codé pendant la passe de codage de propagation de signification?	Voir § D.3.4
D10	Reste-t-il des coefficients dans les quatre colonnes?	
D11	Les quatre bits contigus sont-ils tous zéro?	Voir § D.3.4
D12	Y a-t-il encore des coefficients dans la passe de nettoyage?	

Tableau D.11 – Décodage dans l'organigramme de modèle de contexte

Code	Symbole décodé	Contexte	Brève explication	Description
C0	–	–	Passer au prochain coefficient ou à la prochaine colonne	
C1	Nouvellement significatif?	Tableau D.1, 9 étiquettes de contexte	Décoder le bit de signification du coefficient actuel (propagation de signification ou nettoyage)	Voir § D.3.1
C2	Bit de signe	Tableau D.3, 5 étiquettes de contexte	Décoder le bit de signe du coefficient actuel	Voir § D.3.2
C3	Bit de grandeur actuelle	Tableau D.4, 3 étiquettes de contexte	Décoder le bit de passe de raffinement de grandeur du coefficient actuel	Voir § D.3.3
C4	0 1	Etiquette de contexte de longueur de plage	Décoder longueur de plage de quatre zéros Décoder longueur de plage non de quatre zéros	Voir § D.3.4
C5	00 01 10 11	UNIFORM	Le premier coefficient est le premier avec bit non zéro Le second coefficient est le premier avec bit non zéro Le troisième coefficient est le premier avec bit non zéro Le quatrième coefficient est le premier avec bit non zéro	Voir § D.3.4 et Tableau C.2

Annexe E

Quantification

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe spécifie les formes de quantification inverse utilisées pour la reconstruction de coefficients de transformée de composante-pavé. Des informations sur la quantification de coefficients de transformée pour codage sont également fournies. La quantification est le processus par lequel les coefficients de transformée sont réduits en précision.

E.1 Procédure de quantification inverse

Pour chaque coefficient de transformée (u, v) d'une sous-bande donnée b , la valeur du coefficient de transformée $q_b(u, v)$ est donnée par l'équation suivante:

$$\overline{q}_b(u, v) = (1 - 2s_b(u, v)) \cdot \left(\sum_{i=1}^{N_b(u, v)} MSB_i(b, u, v) \cdot 2^{M_b - i} \right) \quad (\text{E-1})$$

où $s_b(u, v)$, $N_b(u, v)$ et $MSB_i(b, u, v)$ sont données dans le § D.2, et où M_b est récupéré au moyen de l'équation (E-2), où le nombre de bits de garde G et l'exposant ε_b sont spécifiés dans les segments marqueurs QCD ou QCC (voir § A.6.4 et A.6.5).

$$M_b = G + \varepsilon_b - 1 \quad (\text{E-2})$$

Chaque coefficient de transformée décodé $q_b(u, v)$ de sous-bande b sert à produire un coefficient de transformée reconstruit $Rq_b(u, v)$, comme cela sera décrit dans le § E.1.1.

NOTE – Le fait de décoder seulement (voir § D.2.1) les plans binaires $N_b(u, v)$ est équivalent au décodage de données qui ont été codées au moyen d'un quantificateur scalaire avec échelon $2^{M_b - N_b(u, v)} \cdot \Delta_b$ pour tous les coefficients de ce bloc codé. En raison de la nature des trois passes de codage (voir § D.3), $N_b(u, v)$ peut être différent pour différents coefficients dans le même bloc codé.

E.1.1 Transformation irréversible

E.1.1.1 Détermination de la taille de l'échelon de quantification

Pour la transformation irréversible, la taille d'échelon de quantification Δ_b pour une sous-bande donnée b est calculée d'après l'étendue dynamique R_b de la sous-bande b , d'après l'exposant ε_b et d'après la mantisse μ_b indiquée dans l'équation (E-3).

$$\Delta_b = 2^{R_b - \varepsilon_b} \left(1 + \frac{\mu_b}{2^{11}} \right) \quad (\text{E-3})$$

NOTE – Le dénominateur, 2^{11} , dans l'équation (E-3) est dû à l'affectation de 11 bits dans le flux codé pour μ_b , comme indiqué dans le Tableau A.30.

Dans l'équation (E-3), l'exposant ε_b et la mantisse μ_b sont spécifiés dans les segments marqueurs QCD ou QCC (voir § A.6.4 et A.6.5), et l'étendue dynamique nominale R_b [indiquée par l'équation (E-4)] est la somme de R_l (nombre de bits utilisés pour représenter les échantillons de composante-pavé originaux qui peuvent être extraits du marqueur SIZ – voir Tableau A.11 dans le § A.5.1) et du logarithme en base 2 du gain de sous-bande ($gain_b$) de la sous-bande actuelle b , qui varie selon le type de sous-bande b ($levLL$, $levLH$ ou $levHL$, $levHH$ – Voir § F.3.1) et qui peut être trouvé dans le Tableau E.1.

Tableau E.1 – Gains de sous-bande

Sous-bande b	gain _{b}	log ₂ (gain _{b})
$levLL$	1	0
$levLH$	2	1
$levHL$	2	1
$levHH$	4	2

$$R_b = R_I + \log_2(\text{gain}_b) \quad (\text{E-4})$$

Les paires exposant/mantisse (ϵ_b, μ_b) sont soit signalées dans le flux codé pour chaque sous-bande (quantification expansée) ou sinon signalées seulement pour la sous-bande N_{1LL} et calculées pour toutes les autres sous-bandes (quantification calculée) (voir Tableau A.30). Dans le cas d'une quantification calculée, toutes les paires exposant/mantisse (ϵ_b, μ_b) sont calculées d'après la simple paire exposant/mantisse (ϵ_o, μ_o) correspondant à la sous-bande N_{1LL} , selon l'équation (E-5).

$$(\epsilon_b, \mu_b) = (\epsilon_o - N_L + n_b, \mu_o) \quad (\text{E-5})$$

où n_b indique le nombre de niveaux de décomposition entre la composante-pavé originale et la sous-bande b .

NOTE – Pour une sous-bande donnée b , un coefficient de transformée quantifié peut dépasser l'étendue dynamique R_b .

E.1.1.2 Reconstruction du coefficient de transformée

Pour la transformation irréversible, le coefficient de transformée reconstruit est donné par l'équation (E-6):

$$Rq_b(u, v) = \begin{cases} \left(\overline{q}_b(u, v) + r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b & \text{pour } \overline{q}_b(u, v) > 0 \\ \left(\overline{q}_b(u, v) - r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b & \text{pour } \overline{q}_b(u, v) < 0 \\ 0 & \text{pour } \overline{q}_b(u, v) = 0 \end{cases} \quad (\text{E-6})$$

où r est un paramètre de reconstruction, qui peut être arbitrairement choisi par le décodeur.

NOTE – Le paramètre de reconstruction r peut être choisi par exemple afin de produire la meilleure qualité visuelle ou objective pour la reconstruction. Généralement, les valeurs pour r s'inscrivent dans l'étendue de $0 \leq r < 1$, et une valeur commune est $r = 1/2$. (Cette note s'applique également au § E.1.2.)

E.1.2 Transformation réversible

E.1.2.1 Détermination de la taille de l'échelon de quantification

Pour la transformation réversible, la taille d'échelon de quantification Δ_b est égale à un (sans quantification effectuée).

E.1.2.2 Reconstruction du coefficient de transformée

Pour la transformation réversible, le coefficient de transformée reconstruit $Rq_b(u, v)$ est récupéré différemment selon que tous les bits de coefficient sont décodés, c'est-à-dire si $N_b(u, v) = M_b$ ou $N_b(u, v) < M_b$.

Si $N_b(u, v) = M_b$, alors le coefficient de transformée reconstruit $Rq_b(u, v)$ est donné par l'équation (E-7).

$$Rq_b(u, v) = \overline{q}_b(u, v) \quad (\text{E-7})$$

Si $N_b(u, v) < M_b$, alors le coefficient de transformée reconstruit $Rq_b(u, v)$ est donné par l'équation (E-8).

$$Rq_b(u, v) = \begin{cases} \left\lfloor \left(\overline{q_b}(u, v) + r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b \right\rfloor & \text{pour } \overline{q_b}(u, v) > 0 \\ \left\lceil \left(\overline{q_b}(u, v) - r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b \right\rceil & \text{pour } \overline{q_b}(u, v) < 0 \\ 0 & \text{pour } \overline{q_b}(u, v) = 0 \end{cases} \quad (\text{E-8})$$

E.2 Quantification de coefficient scalaire (pour information)

Lors d'une compression irréversible, chacun des coefficients de transformée $a_b(u, v)$ de la sous-bande b est quantifié en valeur $q_b(u, v)$ selon l'équation (E-9), après la transformation irréversible directe en ondelettes discrètes (voir Annexe F).

$$q_b(u, v) = \text{sign}(a_b(u, v)) \cdot \left\lfloor \frac{|a_b(u, v)|}{\Delta_b} \right\rfloor \quad (\text{E-9})$$

où Δ_b est la taille de l'échelon de quantification. L'exposant ϵ_b et la mantisse μ_b correspondant à la taille Δ_b peuvent être calculés d'après l'équation (E-5), et doivent toujours être enregistrés dans les marqueurs QCD ou QCC du flux codé (voir § A.6.4 et A.6.5).

Pour la compression réversible, la taille d'échelon de quantification est tenue d'être 1. Dans ce cas, un paramètre ϵ_b doit être enregistré dans le flux codé des marqueurs QCD ou QCC (voir § A.6.4 et A.6.5), et est calculé comme suit:

$$\epsilon_b = R_I + \log_2(\text{gain}_b) + \zeta_c \quad (\text{E-10})$$

où R_I et gain_b sont comme décrit dans le § E.1.1, et où ζ_c est zéro si la transformation RCT n'est pas utilisée et où ζ_c est le nombre de bits additionnels ajoutés par la transformation RCT si celle-ci est utilisée comme décrit dans le § G.2.1.

Pour la compression réversible comme pour la compression irréversible, afin d'éviter un éventuel débordement ou excursion au-delà de l'étendue nominale de la représentation par entiers de $|q_b(u, v)|$ due par exemple aux calculs en virgule flottante, le nombre M_b de bits pour la représentation par entiers de $q_b(u, v)$ utilisée du côté codeur est défini par l'équation (E-2). Le nombre G de bits de garde doit être spécifié dans le marqueur QCD ou QCC (voir § A.6.4 et A.6.5). Les valeurs typiques du nombre de bits de garde sont $G = 1$ ou $G = 2$.

Annexe F

Transformation en ondelettes discrètes de composantes-pavés

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe décrit la transformation directe en ondelettes discrètes appliquée à une composante-pavé et spécifie la transformation inverse en ondelettes discrètes utilisée pour reconstruire la composante-pavé.

F.1 Paramètres de composante-pavé

Soit la composante-pavé définie par les coordonnées tcx_0 , tcx_1 , tcy_0 et tcy_1 dans l'équation (B-12), dans le § B.3. Alors les coordonnées (x, y) de la composante-pavé (avec valeurs d'échantillon $I(x, y)$) se trouvent dans l'étendue définie par

$$tcx_0 \leq x < tcx_1 \text{ et } tcy_0 \leq y < tcy_1 \quad (\text{F-1})$$

F.2 Transformations en ondelettes discrètes

F.2.1 Filtrage passe-bas et passe-haut (pour information)

Afin d'effectuer la transformation directe en ondelettes discrètes (FDWT, *forward discrete wavelet transformation*), la présente Recommandation | Norme internationale utilise une décomposition unidimensionnelle en sous-bandes d'une matrice unidimensionnelle d'échantillons en coefficients passe-bas, représentant une version sous-échantillonnée à basse résolution de la matrice originale, et passe-haut coefficients, représentant la version résiduelle sous-échantillonnée de la matrice originale, nécessaire afin de reconstruire parfaitement la matrice originale d'après la matrice passe-bas.

Afin d'effectuer la transformation inverse en ondelettes discrètes (IDWT, *inverse discrete wavelet transformation*), la présente Recommandation | Norme internationale utilise une reconstruction en sous-bande unidimensionnelle d'une matrice unidimensionnelle d'échantillons issus de coefficients passe-bas et passe-haut.

F.2.2 Niveaux de décomposition

Chaque composante-pavé est transformée en un ensemble de signaux bidimensionnels de sous-bande (appelés *sous-bandes*), représentant chacun l'activité du signal dans diverses bandes de fréquences, à diverses résolutions spatiales. N_L indique le nombre de niveaux de décomposition.

F.2.3 Filtrés d'ondelettes discrètes (pour information)

La présente Recommandation | Norme internationale spécifie une seule transformation réversible et une seule transformation irréversible. Compte tenu du fait que les échantillons de composante-pavé sont exprimés en valeurs d'entier, une transformation réversible nécessite la spécification d'une procédure d'arrondissement pour les coefficients de transformée intermédiaires non exprimés en valeurs d'entier.

F.3 Transformation inverse en ondelettes discrètes

F.3.1 La procédure de transformation IDWT

La transformation inverse en ondelettes discrètes (IDWT) transforme un ensemble de sous-bandes, $a_b(u_b, v_b)$ en une composante-pavé $I(x, y)$ décalée au niveau aperiodique (procédure de transformation IDWT). La procédure de transformation IDWT reçoit également en entrée un paramètre N_L qui représente le nombre de niveaux de décomposition (voir Figure F.1). Le nombre de niveaux de décomposition N_L est signalé dans les marqueurs COD ou COC (voir § A.6.1 et A.6.2).

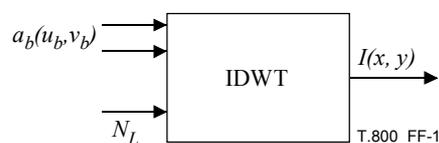


Figure F.1 – Entrées et sorties de la procédure de transformation IDWT

Les sous-bandes sont étiquetées de la façon suivante: un index *lev* correspondant au niveau de décomposition, suivi par deux lettres qui sont soit LL, HL, LH ou HH.

La sous-bande $b = levLL$ correspond à une version sous-échantillonnée de sous-bande $(lev - 1)LL$ qui a été filtrée en passe-bas verticalement et filtrée en passe-bas horizontalement. La sous-bande $b = 0LL$ correspond à la composante-pavé originale. La sous-bande $b = levHL$ correspond à une version sous-échantillonnée de sous-bande $(lev - 1)LL$ qui a été filtrée en passe-bas verticalement et filtrée en passe-haut horizontalement. La sous-bande $b = levLH$ correspond à une version sous-échantillonnée de sous-bande $(lev - 1)LL$ qui a été filtrée en passe-haut verticalement et filtrée en passe-bas horizontalement. La sous-bande $b = levHH$ correspond à une version sous-échantillonnée de sous-bande $(lev - 1)LL$ qui a été filtrée en passe-haut verticalement et filtrée en passe-haut horizontalement.

Pour une valeur donnée de N_L , seules les sous-bandes suivantes sont présentes dans le flux codé, et dans l'ordre suivant (ces sous-bandes sont suffisantes pour reconstruire entièrement la composante-pavé originale):

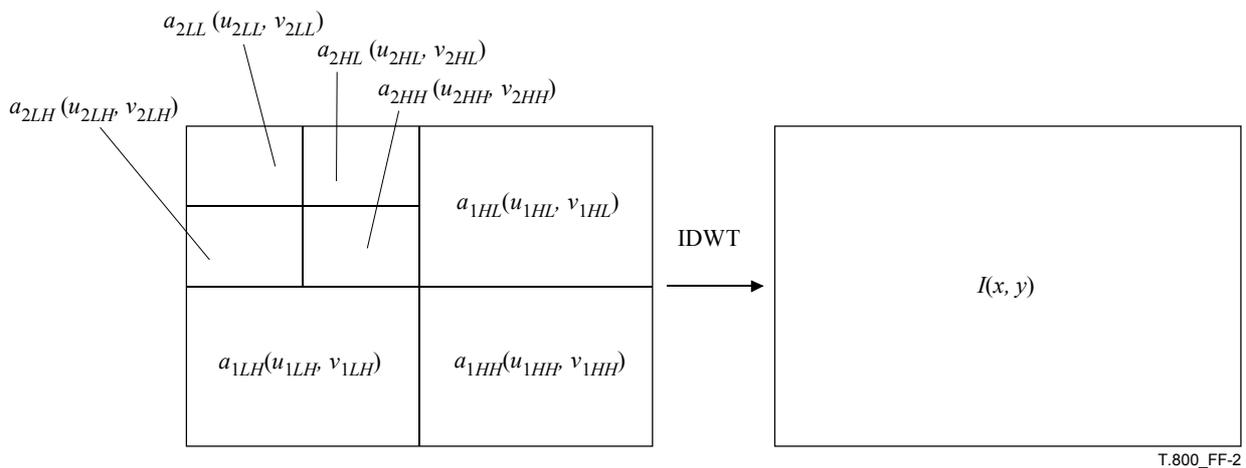
$$N_L LL, N_L HL, N_L LH, N_L HH, (N_L - 1)HL, (N_L - 1)LH, (N_L - 1)HH, \dots, 1HL, 1LH, 1HH.$$

Pour une sous-bande donnée b , le nombre n_b représente le niveau de décomposition auquel il a été produit au moment du codage; ce nombre est indiqué dans le Tableau F.1.

Tableau F.1 – Niveau de décomposition n_b pour sous-bande b

b	$N_L LL$	$N_L HL$	$N_L LH$	$N_L HH$	$(N_L - 1)HL$	$(N_L - 1)LH$	$(N_L - 1)HH$...	1HL	1LH	1HH
n_b	N_L	N_L	N_L	N_L	$N_L - 1$	$N_L - 1$	$N_L - 1$...	1	1	1

Les sous-bandes pour le cas où $N_L = 2$ sont illustrées dans la Figure F.2.



T.800_FF-2

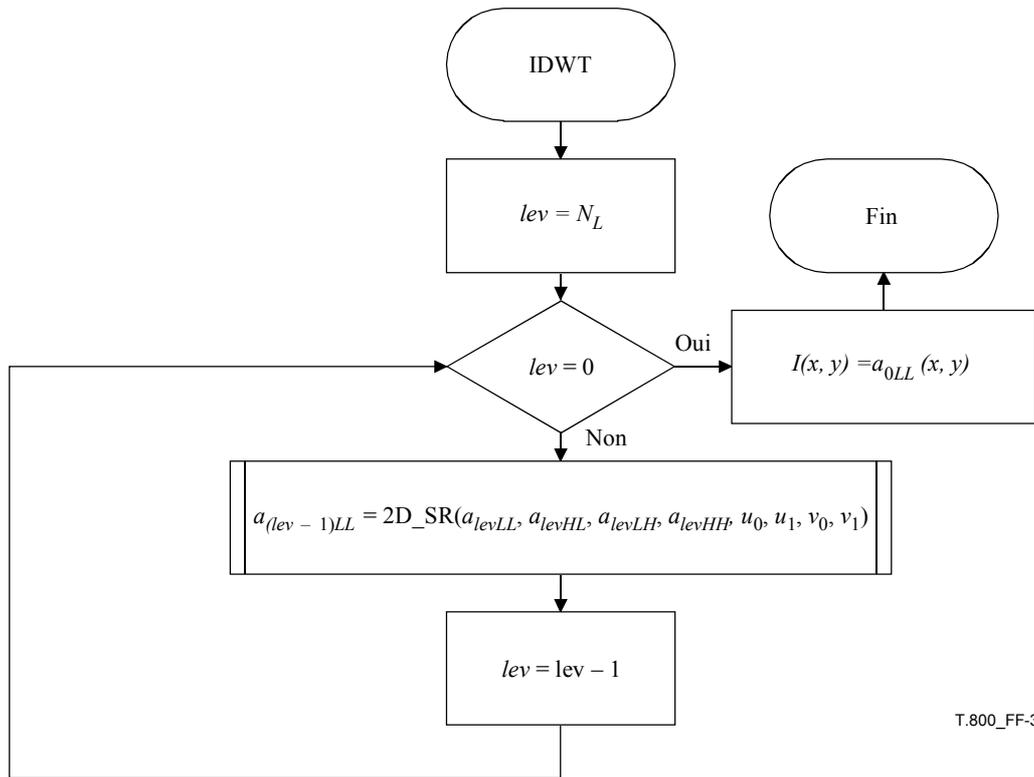
Figure F.2 – La transformation IDWT ($N_L = 2$)

La procédure de transformation IDWT commence avec l'initialisation de la variable *lev* (le niveau de décomposition actuel) à N_L . La procédure 2D_SR (voir § F.3.2) est effectuée à chaque niveau *lev*, où le niveau *lev* diminue à chaque itération, jusqu'à ce que N_L itérations soient effectuées. La procédure 2D_SR est itérée sur la sous-bande $levLL$ produite à chaque itération. Finalement, la sous-bande $a_{0LL}(u_{0LL}, v_{0LL})$ est la matrice de sortie $I(x, y)$.

Comme défini dans l'équation (B-15), les indices (u_b, v_b) des coefficients de sous-bande $a_b(u_b, v_b)$ pour une sous-bande donnée b se trouvent dans l'étendue définie par:

$$tbx_0 \leq u_b < tbx_1 \quad \text{and} \quad tby_0 \leq v_b < tby_1 \tag{F-2}$$

La Figure F.3 décrit la procédure de transformation IDWT.

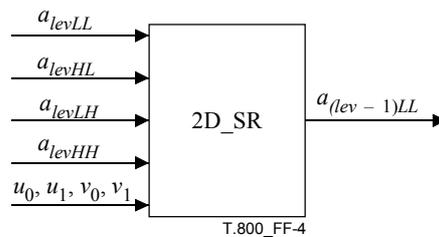


T.800_FF-3

Figure F.3 – La procédure de transformation IDWT

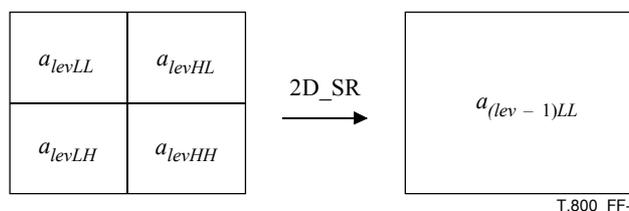
F.3.2 La procédure 2D_SR

La procédure 2D_SR exécute une reconstruction de sous-bande $a_{(lev-1)LL}(u, v)$ d'après les quatre sous-bandes $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$ et $a_{levHH}(u, v)$ (voir Figure F.4). Le nombre total de coefficients de la sous-bande $levLL$ reconstruite est égal à la somme du nombre total de coefficients des quatre sous-bandes d'entrée dans la procédure 2D_SR (voir Figure F.5).



T.800_FF-4

Figure F.4 – Entrées et sorties de la procédure 2D_SR



T.800_FF-5

Figure F.5 – Niveau unique de reconstruction de quatre sous-bandes (procédure 2D_SR) en sous-bandes

Les quatre sous-bandes $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$ et $a_{levHH}(u, v)$ sont entrelacées afin de former une matrice $a(u, v)$ au moyen de la procédure 2D_INTERLEAVE. La procédure 2D_SR applique ensuite la procédure HOR_SR à

toutes les rangées de $a(u, v)$ et finalement applique la procédure VER_SR à toutes les colonnes de $a(u, v)$ afin de produire la sous-bande reconstruite $a_{(lev-1)LL}(u, v)$. La Figure F.6 décrit la procédure 2D_SR.

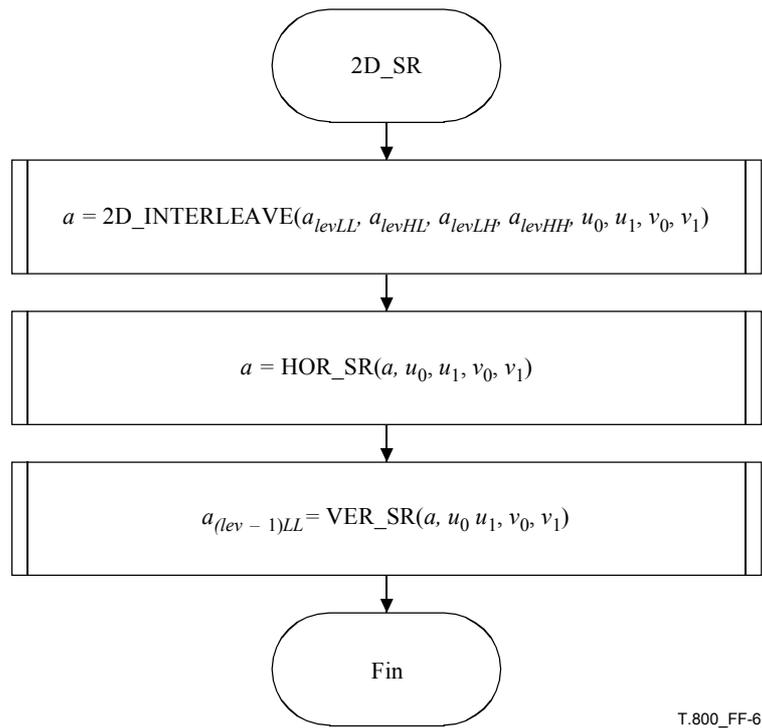


Figure F.6 – La procédure 2D_SR

F.3.3 La procédure 2D_INTERLEAVE

Comme illustré dans la Figure F.7, la procédure 2D_INTERLEAVE entrelace les coefficients des quatre sous-bandes a_{levLL} , a_{levHL} , a_{levLH} , a_{levHH} de former $a(u, v)$. Les valeurs de u_0 , u_1 , v_0 , v_1 utilisées par la procédure 2D_INTERLEAVE sont celles de tbx_0 , tbx_1 , tby_0 , tby_1 qui correspondent à la sous-bande $b = (lev - 1)LL$ (voir définition dans l'équation (B-15)).

La façon dont ces sous-bandes sont entrelacées afin de former la sortie $a(u, v)$ est décrite par la procédure 2D_INTERLEAVE donnée dans la Figure F.8.

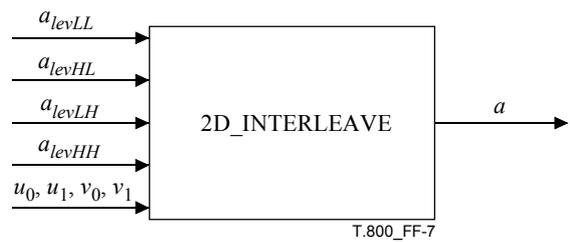
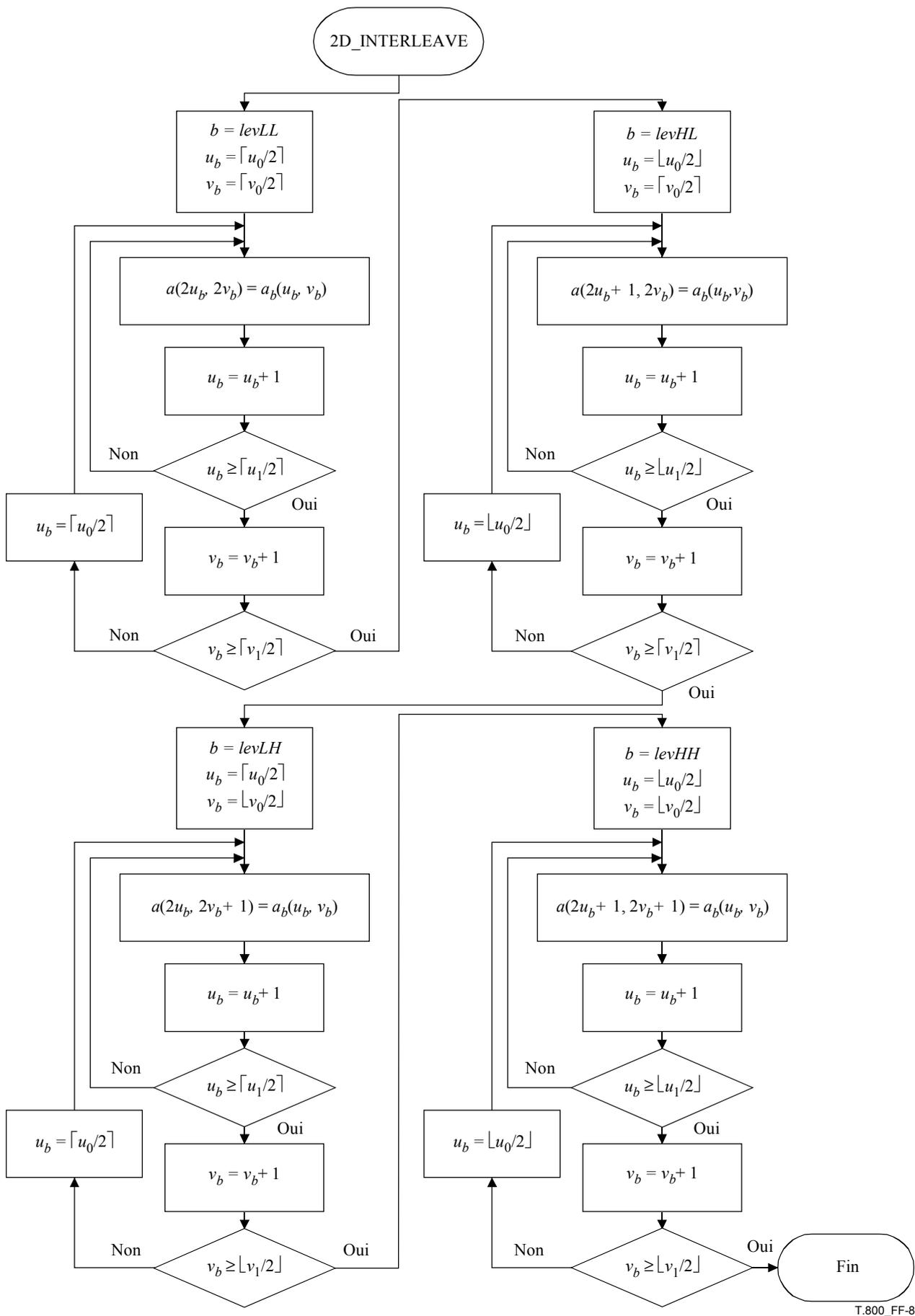


Figure F.7 – Paramètres de la procédure 2D_INTERLEAVE



T.800_FF-8

Figure F.8 – La procédure 2D_INTERLEAVE

F.3.4 La procédure HOR_SR

La procédure HOR_SR exécute une reconstruction en sous-bande horizontale d'une matrice bidimensionnelle de coefficients. Elle reçoit en entrée une matrice bidimensionnelle, $a(u, v)$, les étendues horizontale et verticale de ses coefficients étant indiquées par $u_0 \leq u < u_1$ et $v_0 \leq v < v_1$ (voir Figure F.9). Elle produit en sortie une version filtrée horizontalement de la matrice d'entrée, rangée par rangée.

Comme illustré dans la Figure F.10, la procédure HOR_SR applique la reconstruction en sous-bande unidimensionnelle (procédure 1D_SR) à chaque rangée v de la matrice d'entrée $a(u, v)$, et remémore le résultat dans chaque rangée.

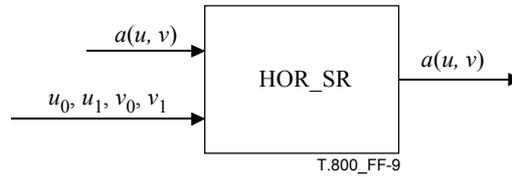


Figure F.9 – Entrées et sorties de la procédure HOR_SR

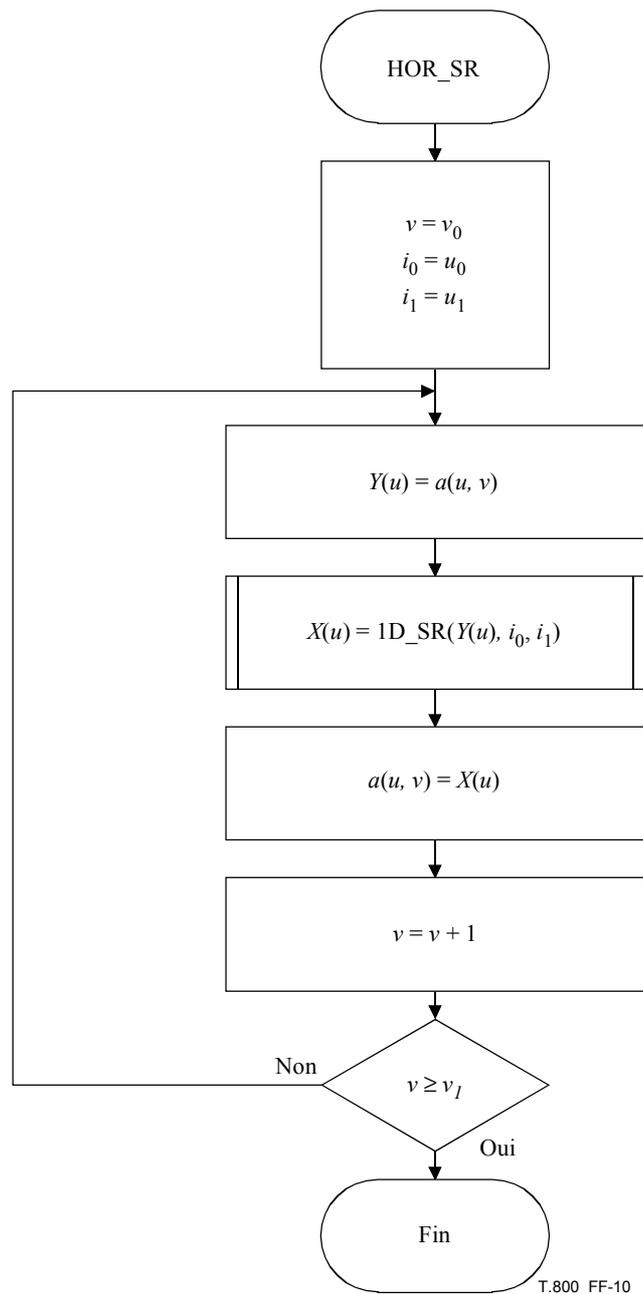


Figure F.10 – La procédure HOR_SR

F.3.5 La procédure VER_SR

La procédure VER_SR exécute la reconstruction en sous-bande verticale d'une matrice bidimensionnelle de coefficients. Elle reçoit en entrée une matrice bidimensionnelle, $a(u, v)$, les étendues horizontale et verticale de ses coefficients étant indiquées par $u_0 \leq u < u_1$ et $v_0 \leq v < v_1$ (voir Figure F.11). Elle produit en sortie une version filtrée verticalement de la matrice d'entrée, colonne par colonne.

Comme illustré dans la Figure F.12, la procédure VER_SR applique la reconstruction en sous-bande unidimensionnelle (procédure 1D_SR) à chaque colonne u de la matrice d'entrée $a(u, v)$ et remémore le résultat dans chaque colonne.

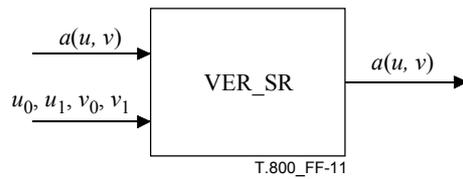


Figure F.11 – Entrées et sorties de la procédure VER_SR

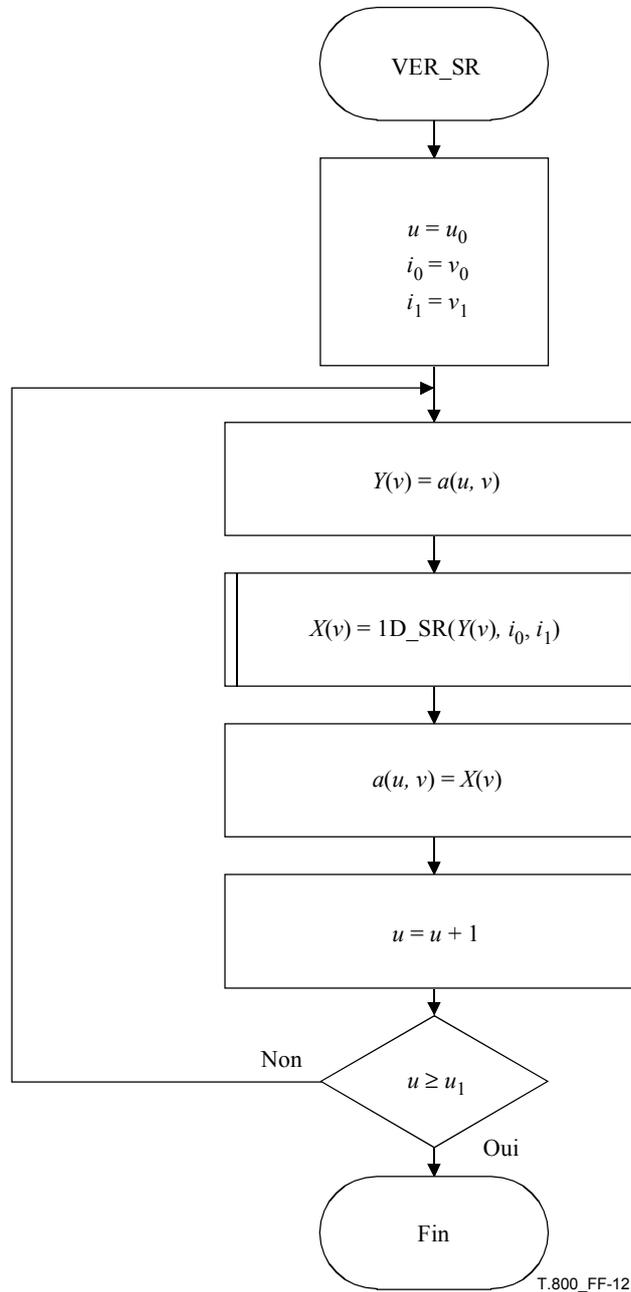


Figure F.12 – La procédure VER_SR

F.3.6 La procédure 1D_SR

Comme illustré dans la Figure F.13, la procédure 1D_SR reçoit en entrée une matrice unidimensionnelle $Y(i)$, l'étendue de ses coefficients étant indiquée par $i_0 \leq i < i_1$. Elle produit en sortie une matrice X , avec les mêmes indices (i_0, i_1) .

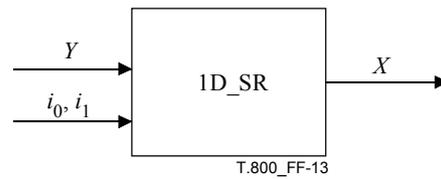


Figure F.13 – Paramètres de la procédure 1D_SR

Pour les signaux de longueur un (c'est-à-dire $i_0 = i_1 - 1$), la procédure 1D_SR met la valeur de $X(i_0)$ à $X(i_0) = Y(i_0)$ si i_0 est un entier pair, et à $X(i_0) = Y(i_0)/2$ si i_0 est un entier impair.

Pour les signaux de longueur supérieure ou égale à deux (c'est-à-dire $i_0 < i_1 - 1$), comme illustré dans la Figure F.14, la procédure 1D_SR utilise d'abord la procédure 1D_EXTR afin d'étendre le signal Y au-delà de ses limites de gauche et de droite ce qui se traduit par le signal étendu Y_{ext} . Elle utilise ensuite la procédure 1D_FILTR afin de filtrer en inverse le signal étendu Y_{ext} et afin de produire le signal filtré souhaité X . Les procédures 1D_EXTR et 1D_FILTR dépendent de la question de savoir si la transformée en ondelettes (irréversible) 9-7 ou la transformée en ondelettes (réversible) 5-3 est choisie: cela est signalé dans les marqueurs COD ou COC (voir § A.6.1 et A.6.2).

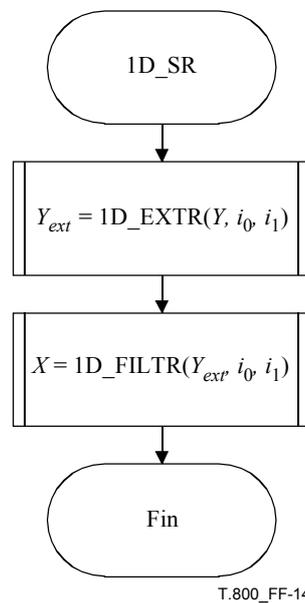


Figure F.14 – La procédure 1D_SR

F.3.7 La procédure 1D_EXTR

Comme illustré dans la Figure F.15, la procédure 1D_EXTR étend le signal Y de i_{left} coefficients vers la gauche et de i_{right} coefficients vers la droite. L'extension du signal est nécessaire afin de permettre le filtrage aux deux limites du signal.

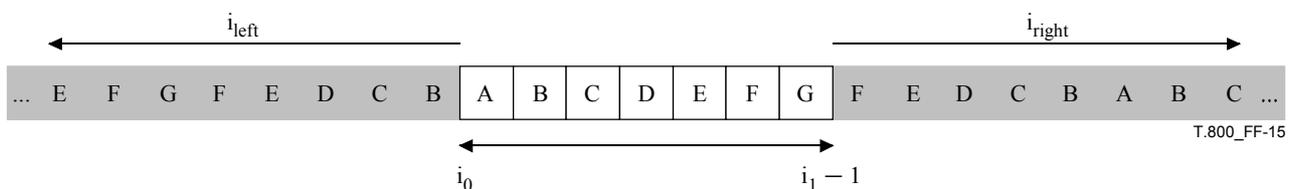


Figure F.15 – Extension périodique et symétrique du signal

Le premier coefficient de Y est le coefficient i_0 , et le dernier coefficient du signal Y est le coefficient $i_1 - 1$. Cette procédure d'extension est dite "extension périodique et symétrique". L'extension symétrique consiste à étendre le signal avec les coefficients de signal obtenus par une réflexion du signal centré sur le premier coefficient (coefficient i_0) pour

l'extension vers la gauche, et à étendre le signal avec les coefficients de signal obtenus par une réflexion du signal centré sur le dernier coefficient (coefficient $i_1 - 1$) pour l'extension vers la droite. L'extension périodique et symétrique est une généralisation de l'extension symétrique dans le cas plus général où le nombre de coefficients permettant d'étendre le signal d'un côté ou d'un autre peut dépasser la longueur de signal $i_1 - i_0$: ce cas peut se produire aux niveaux supérieurs de décomposition.

La procédure 1D_EXTR calcule les valeurs de $Y_{ext}(i)$ pour les valeurs de i au-delà de l'étendue $i_0 \leq i < i_1$, comme indiqué dans l'équation (F-3):

$$Y_{ext}(i) = Y(PSE_O(i, i_0, i_1)) \tag{F-3}$$

où $PSE_O(i, i_0, i_1)$ est donné par l'équation (F-4):

$$PSE_O(i, i_0, i_1) = i_0 + \min(\text{mod}(i - i_0, 2(i_1 - i_0 - 1)), 2(i_1 - i_0 - 1) - \text{mod}(i - i_0, 2(i_1 - i_0 - 1))) \tag{F-4}$$

Deux procédures d'extension sont définies, selon que la transformation en ondelettes 5-3 (procédure 1D_EXTR₅₋₃) ou 9-7 (procédure 1D_EXTR₉₋₇) est appliquée. Ces procédures ne diffèrent que par les valeurs minimales des paramètres d'extension ($i_{left5-3}$ et $i_{right5-3}$ pour la transformation 5-3 en ondelettes, et $i_{left9-7}$ et $i_{right9-7}$ pour la transformation 9-7 en ondelettes) qui sont données dans les Tableaux F.2 et F.3. Ces valeurs ne dépendent que de la parité des indices i_0 et i_1 . Les valeurs égales ou supérieures à celles qui sont indiquées dans les Tableaux F.2 et F.3 produiront la même matrice X à la sortie de la procédure 1D_IFILTR de la Figure F.14.

Tableau F.2 – Extension vers la gauche

i_0	$i_{left5-3}$	$i_{left9-7}$
Pair	1	3
Impair	2	4

Tableau F.3 – Extension vers la droite

i_1	$i_{right5-3}$	$i_{right9-7}$
Impair	1	3
Pair	2	4

F.3.8 La procédure 1D_FILTR

Une procédure de filtrage réversible 1D_FILTR_{5-3R} et une procédure de filtrage irréversible 1D_FILTR_{9-7I} sont spécifiées, selon que la transformation 5-3 réversible ou 9-7 irréversible en ondelettes est utilisée.

Comme illustré dans la Figure F.16, ces deux procédures reçoivent en entrée un signal unidimensionnel étendu Y_{ext} , l'index du premier coefficient i_0 , et l'index du coefficient i_1 suivant immédiatement le dernier coefficient ($i_1 - 1$). Ils produisent tous les deux en sortie le signal X.

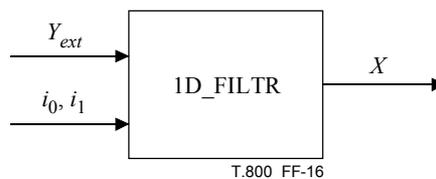


Figure F.16 – Paramètres de la procédure 1D_FILTR

Les deux procédures utilisent le filtrage en mode lissage, qui consiste à appliquer au signal une séquence de très simples opérations de filtrage appelées *étapes de lissage*, qui modifient alternativement les valeurs de coefficient à index impair

du signal avec une somme pondérée de valeurs de coefficient à index pair, et les valeurs de coefficient à index pair avec une somme pondérée de valeurs de coefficient à index impair.

F.3.8.1 La procédure 1D_FILTR_{5-3R}

La procédure 1D_FILTR_{5-3R} utilise le filtrage en mode lissage conjointement avec des opérations d'arrondissement. L'équation (F-5) est effectuée la première pour toutes les valeurs de n indiquées, suivie par l'équation (F-6) qui utilise les valeurs calculées d'après l'équation (F-5):

$$X(2n) = Y_{ext}(2n) - \left\lfloor \frac{Y_{ext}(2n-1) + Y_{ext}(2n+1) + 2}{4} \right\rfloor \text{ pour } \left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1 \quad (\text{F-5})$$

$$X(2n+1) = Y_{ext}(2n+1) + \left\lfloor \frac{X(2n) + X(2n+2)}{2} \right\rfloor \text{ pour } \left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor \quad (\text{F-6})$$

Les valeurs de $X(k)$ sont telles que l'étendue $i_0 \leq k < i_1$ forme la sortie de la procédure 1D_FILTR_{5-3R}.

F.3.8.2 La procédure 1D_FILTR_{9-7I}

La procédure 1D-FILTR_{9-7I} utilise le filtrage en mode lissage (il n'y a aucune opération d'arrondissement). Les paramètres de lissage (α , β , γ , δ) et le paramètre de normalisation K pour toutes les étapes de filtrage sont définis au § F.3.8.2.1.

L'équation (F-7) décrit les deux étapes de normalisation (1 et 2) et les quatre étapes de lissage (3 à 6) du filtrage unidimensionnel effectué sur le signal étendu $Y_{ext}(n)$ afin de produire les coefficients $i_1 - i_0$ du signal X . Ces étapes sont effectuées dans l'ordre suivant:

Premièrement, l'étape 1 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$, et l'étape 2 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor - 2 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$.

Puis l'étape 3 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$, et cette étape utilise les valeurs calculées aux étapes 1 et 2.

Puis l'étape 4 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1$, et cette étape utilise les valeurs calculées aux étapes 2 et 3.

Puis l'étape 5 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1$, et cette étape utilise les valeurs calculées aux étapes 3 et 4.

Finalement, l'étape 6 est effectuée pour toutes les valeurs de n telles que $\left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor$, et cette étape utilise les valeurs calculées aux étapes 4 et 5.

$$\begin{cases} X(2n) = KY_{ext}(2n) & [\text{Etape1}] \\ X(2n+1) = (1/K)Y_{ext}(2n+1) & [\text{Etape2}] \\ X(2n) = X(2n) - \delta(X(2n-1) + X(2n+1)) & [\text{Etape3}] \\ X(2n+1) = X(2n+1) - \gamma(X(2n) + X(2n+2)) & [\text{Etape4}] \\ X(2n) = X(2n) - \beta(X(2n-1) + X(2n+1)) & [\text{Etape5}] \\ X(2n+1) = X(2n+1) - \alpha(X(2n) + X(2n+2)) & [\text{Etape6}] \end{cases} \quad (\text{F-7})$$

où les valeurs des paramètres de lissage (α , β , γ , δ) et de K sont définies dans le Tableau F.4.

Tableau F.4 – Définition des paramètres de lissage pour le filtre irréversible 9-7

Paramètre	Expression exacte	Valeur approchée
α	$-g_4 / g_3$	-1,586 134 342 059 924
β	g_3 / r_1	-0,052 980 118 572 961
γ	r_1 / s_0	0,882 911 075 530 934
δ	s_0 / t_0	0,443 506 852 043 971
K	$1 / t_0$	1,230 174 104 914 001

Les valeurs de $X(k)$ sont telles que l'étendue $i_0 \leq k < i_1$ forme la sortie de la procédure 1D_FILTR₁.

F.3.8.2.1 Paramètres de filtrage pour la procédure 1D_FILTR₉₋₇₁

Les paramètres de filtrage (α , β , γ , δ , K) sont définis dans le Tableau F.4 en termes de paramètres g_n d'après le Tableau F.5, et les paramètres (r_0 , r_1 , s_0 , t_0) sont définis d'après le Tableau F.6. Les paramètres g_n sont définis en termes de paramètres x_1 , $\Re x_2$ et $|x_2|^2$ indiqués dans le Tableau F.7. Tous les tableaux donnent une expression de forme fermée pour tous les paramètres, y compris les approximations jusqu'à 15 décimales.

Tableau F.5 – Définition des coefficients g_n

n	coefficients g_n	valeur approchée de g_n
0	$5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	-0,602 949 018 236 360
1	$-5x_1(8 x_2 ^2 - \Re x_2) / 8$	0,266 864 118 442 875
2	$5x_1(4 x_2 ^2 + 4\Re x_2 - 1) / 16$	0,078 223 266 528 990
3	$-5x_1(\Re x_2) / 8$	-0,016 864 118 442 875
4	$5x_1 / 64$	-0,026 748 757 410 810

Tableau F.6 – Expressions intermédiaires (r_0 , r_1 , s_0 , t_0)

Paramètre	Expression exacte	Valeur approchée
r_0	$-g_0 + 2g_1g_4 / g_3$	1,449 513 704 087 943
r_1	$-g_2 + g_4 + g_1g_4 / g_3$	0,318 310 318 985 991
s_0	$g_1 - g_3 - g_3r_0 / r_1$	0,360 523 644 801 462
t_0	$r_0 - 2r_1$	0,812 893 066 115 961

Tableau F.7 – Expressions intermédiaires

Paramètre	Expression exacte	Valeur approchée
A	$\sqrt[3]{\frac{63 - 14\sqrt{15}}{1080\sqrt{15}}}$	0,128 030 244 703 494
B	$-\sqrt[3]{\frac{63 + 14\sqrt{15}}{1080\sqrt{15}}}$	-0,303 747 672 895 197
x_1	$A + B - 1 / 6$	-0,342 384 094 858 369
$\Re x_2$	$-\frac{(A + B)}{2} - \frac{1}{6}$	-0,078 807 952 570 815
$ x_2 ^2$	$\left[\frac{(A + B)}{2} + \frac{1}{6}\right]^2 + \frac{3(A - B)^2}{4}$	0,146 034 820 982 800

F.4 Transformation directe (pour information)

F.4.1 La procédure FDWT (pour information)

La transformation directe en ondelettes discrètes (FDWT) transforme les échantillons de composante-pavé décalés au niveau apériodique $I(x, y)$ en un ensemble de sous-bandes avec coefficients $a_b(u_b, v_b)$ (procédure FDWT). La procédure FDWT (voir Figure F.17) reçoit également en entrée le nombre de niveaux de décomposition N_L signalé dans les marqueurs COD ou COC (voir § A.6.1 et A.6.2).

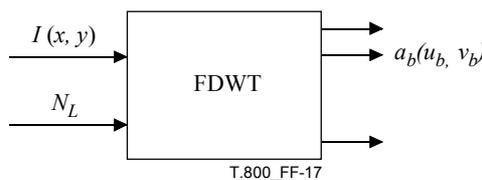


Figure F.17 – Entrées et sorties de la procédure FDWT

Comme illustré dans la Figure F.18, toutes les sous-bandes si $N_L = 2$ peuvent être représentées de la façon suivante.

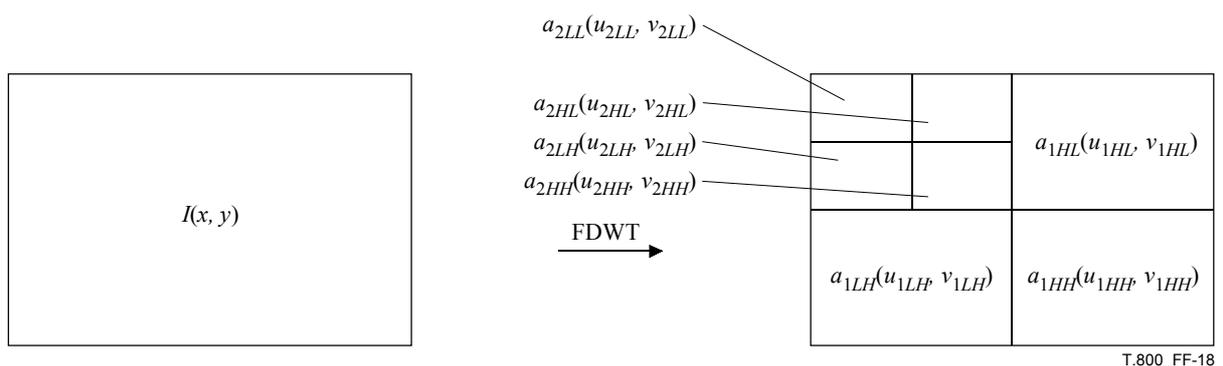


Figure F.18 – La transformation FDWT ($N_L = 2$)

La procédure FDWT commence par l'initialisation de la variable *lev* (le niveau de décomposition actuel) jusqu'à zéro, puis adapte la sous-bande $a_{0LL}(u_{0LL}, v_{0LL})$ à la matrice d'entrée $I(u, v)$. La procédure 2D_SD est effectuée à chaque niveau *lev*, celui-ci augmentant d'un à chaque itération, jusqu'à ce que N_L itérations soient effectuées. La procédure 2D_SD est itérée sur la sous-bande *levLL* produite à chaque itération.

Comme défini dans l'Annexe B (voir équation (B-15)), les coordonnées de la sous-bande $a_{levLL}(u, v)$ se trouvent dans l'étendue définie par:

$$tbx_0 \leq u < tbx_1 \quad \text{et} \quad tby_0 \leq v < tby_1 \tag{F-8}$$

La Figure F.19 décrit la procédure FDWT.

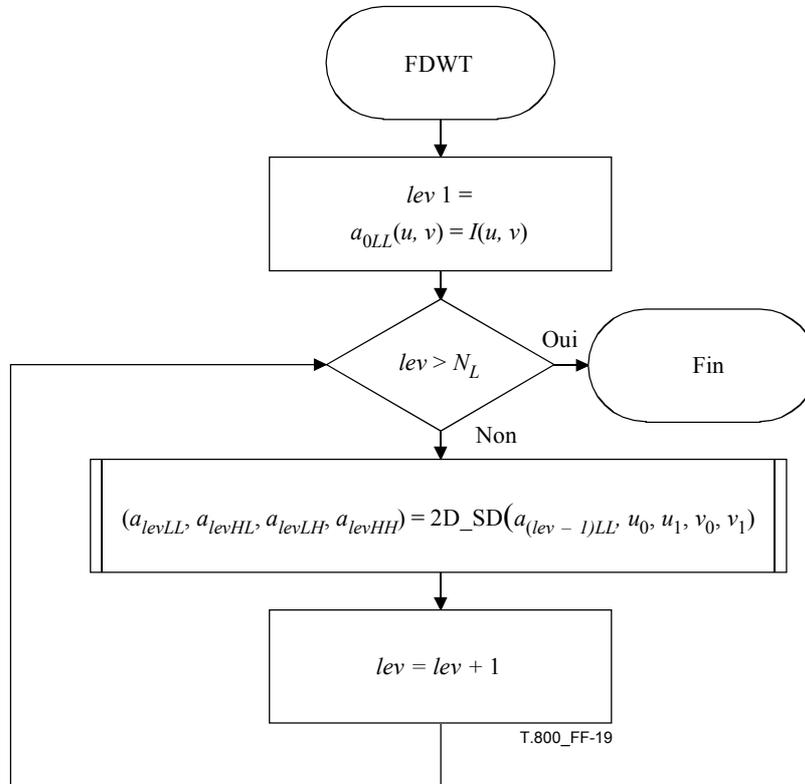


Figure F.19 – La procédure FDWT

F.4.2 La procédure 2D_SD (pour information)

La procédure 2D_SD exécute la décomposition d'une matrice bidimensionnelle de coefficients ou d'échantillons $a_{(lev-1)LL}(u, v)$ en quatre groupes de coefficients de sous-bande $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$, et $a_{levHH}(u, v)$.

Le nombre total de coefficients de la sous-bande lev_{LL} est égal à la somme du nombre total de coefficients des quatre sous-bandes résultant de la procédure 2D_SD.

La Figure F.20 décrit les paramètres d'entrée et de sortie de la procédure 2D_SD.

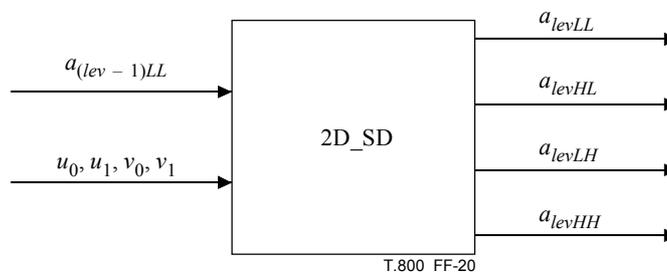


Figure F.20 – Entrées et sorties de la procédure 2D_SD

La Figure F.21 illustre la décomposition en sous-bandes effectuée par la procédure 2D_SD.

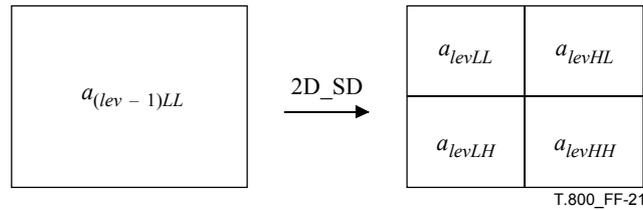


Figure F.21 – Décomposition d'un niveau unique en quatre sous-bandes (procédure 2D_SD)

La procédure 2D_SD applique d'abord la procédure VER_SD à toutes les colonnes de $a(u, v)$. Elle applique ensuite la procédure HOR_SD à toutes les rangées de $a(u, v)$. Les coefficients ainsi obtenus d'après $a(u, v)$ sont désentrelacés dans les quatre sous-bandes au moyen de la procédure 2D_DEINTERLEAVE.

La Figure F.22 décrit la procédure 2D_SD.

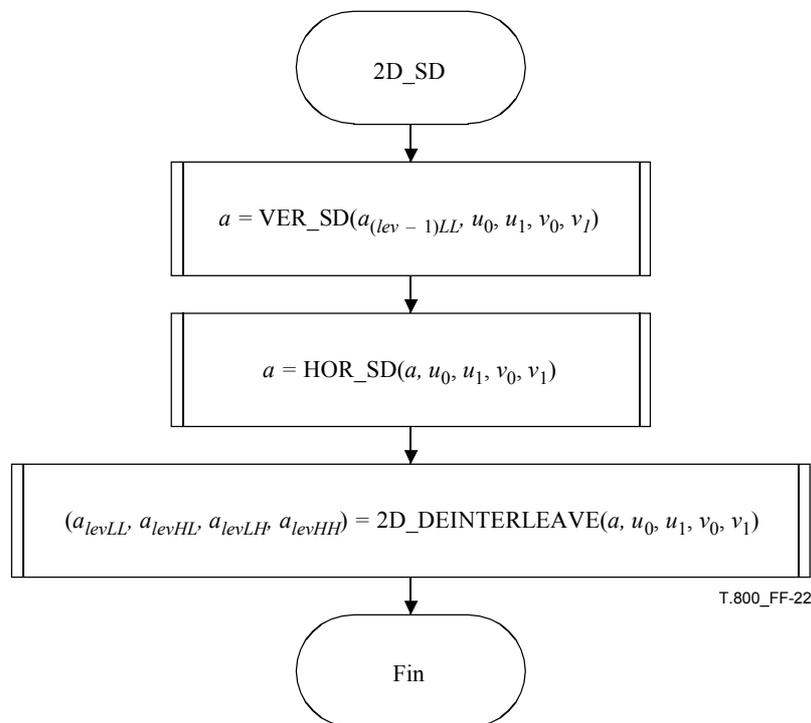


Figure F.22 – La procédure 2D_SD

F.4.3 La procédure VER_SD (pour information)

La procédure VER_SD exécute une décomposition verticale en sous-bandes d'une matrice bidimensionnelle de coefficients. Elle reçoit en entrée la matrice bidimensionnelle $a_{(lev-1)LL}(u, v)$, l'étendue horizontale et verticale de ses coefficients comme indiqué par $u_0 \leq u < u_1$ et $v_0 \leq v < v_1$ (voir Figure F.23) et produit en sortie une version filtrée verticalement $a(u, v)$ de la matrice d'entrée, colonne par colonne. Les valeurs de u_0, u_1, v_0, v_1 utilisées par la procédure VER_SD sont celles de $tbx_0, tbx_1, tby_0, tby_1$ correspondant à la sous-bande $b = (lev-1)LL$ (voir définition dans l'équation (B-15)).

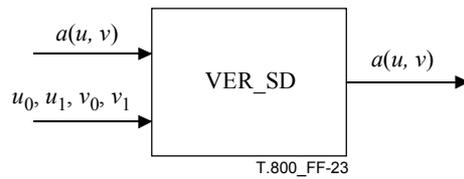


Figure F.23 – Entrées et sorties de la procédure VER_SD

Comme illustré dans la Figure F.24, la procédure VER_SD applique la décomposition unidimensionnelle en sous-bandes (procédure 1D_SD) à chaque colonne de la matrice d'entrée $a(u, v)$, et remémorise le résultat dans chaque colonne.

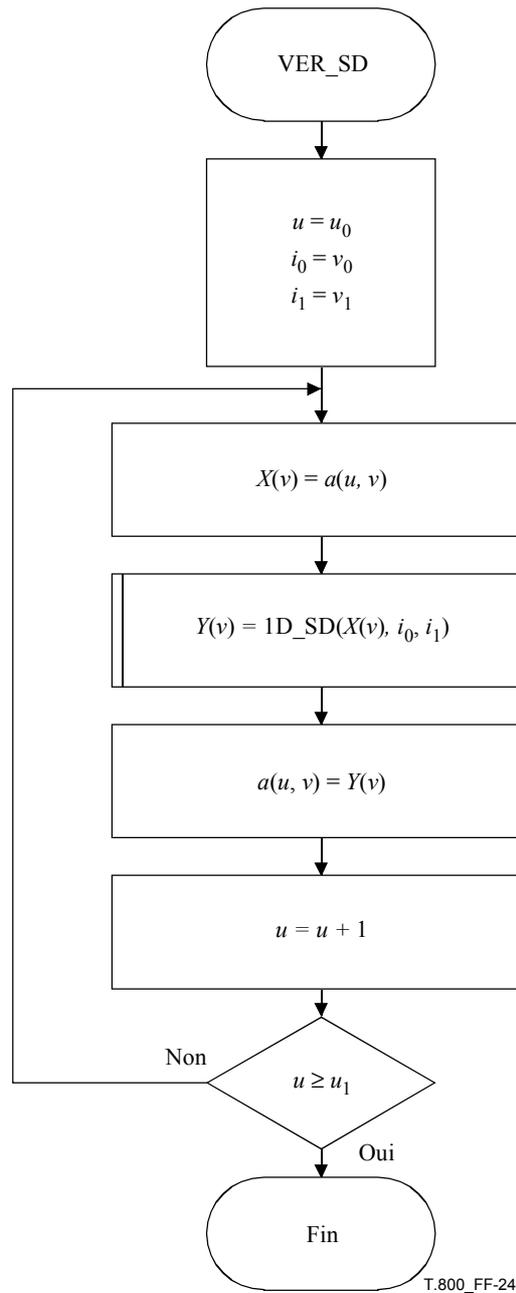


Figure F.24 – La procédure VER_SD

F.4.4 La procédure HOR_SD (pour information)

La procédure HOR_SD exécute une décomposition horizontale en sous-bandes d'une matrice bidimensionnelle de coefficients. Elle reçoit en entrée une matrice bidimensionnelle $a(u, v)$, l'étendue horizontale et verticale de ses coefficients comme indiqué par $u_0 \leq u < u_1$ et $v_0 \leq v < v_1$ (voir Figure F.25) et produit en sortie une version filtrée horizontalement de la matrice d'entrée, rangée par rangée.

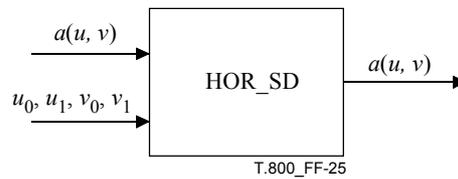


Figure F.25 – Entrées et sorties de la procédure HOR_SD

Comme illustré dans la Figure F.26, la procédure HOR_SD applique la décomposition unidimensionnelle en sous-bandes (procédure 1D_SD) à chaque rangée de la matrice d'entrée $a(u, v)$ et remémore le résultat dans chaque rangée.

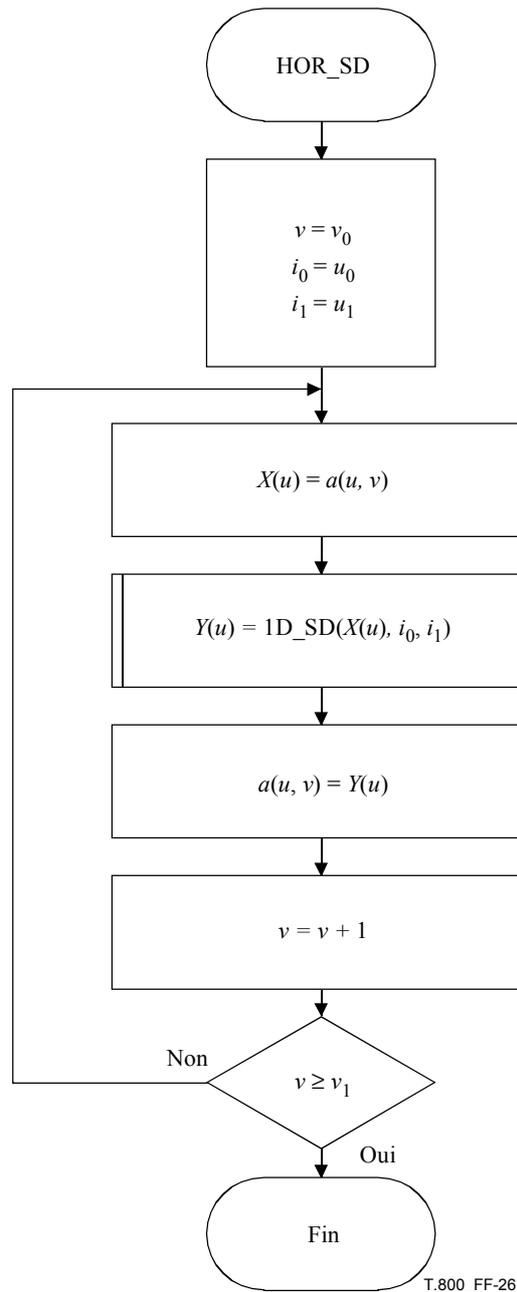


Figure F.26 – La procédure HOR_SD

F.4.5 La procédure 2D_DEINTERLEAVE (pour information)

Comme illustré dans la Figure F.27, la procédure 2D_DEINTERLEAVE désentrelace les coefficients de $a(u, v)$ en quatre sous-bandes. La disposition dépend des coordonnées (u_0, v_0) du premier coefficient de $a(u, v)$.

La façon dont ces sous-bandes sont formées d'après la sortie $a(u, v)$ de la procédure HOR_SD est décrite par la procédure 2D_DEINTERLEAVE illustrée dans la Figure F.28.

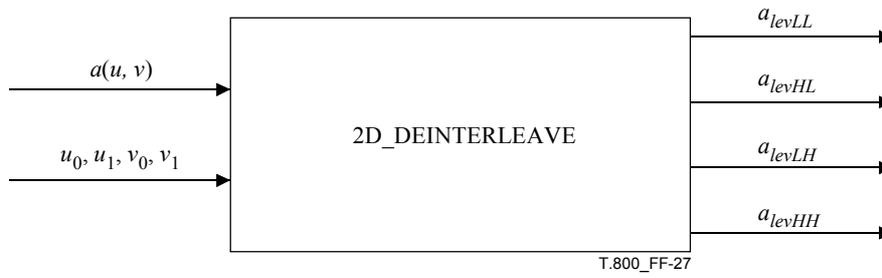
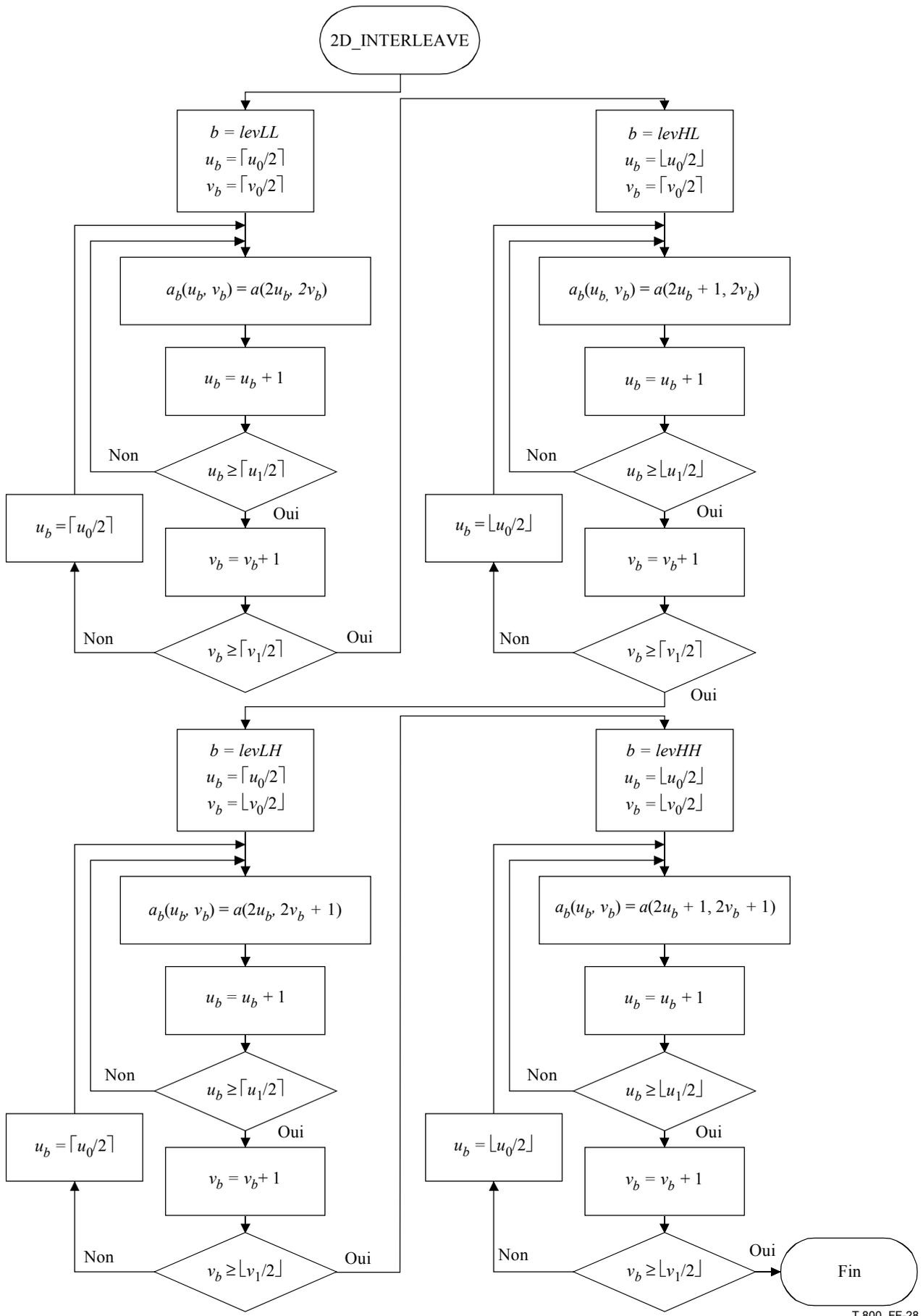


Figure F.27 – Paramètres de la procédure 2D_DEINTERLEAVE



T.800_FF-28

Figure F.28 – La procédure 2D_DEINTERLEAVE

F.4.6 La procédure 1D_SD (pour information)

Comme illustré dans la Figure F.29, la procédure 1D_SD reçoit en entrée une matrice unidimensionnelle $X(i)$, l'étendue de ses coefficients indiquée par $i_0 \leq i < i_1$. Elle produit en sortie une matrice $Y(i)$, avec les mêmes indices (i_0, i_1).

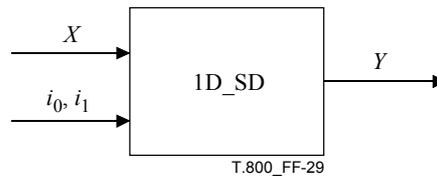


Figure F.29 – Paramètres de la procédure 1D_SD

Pour les signaux de longueur un (c'est-à-dire $i_0 = i_1 - 1$), la procédure 1D_SD met la valeur de $Y(i_0)$ à $Y(i_0) = X(i_0)$ si i_0 est un entier pair, et à $Y(i_0) = 2X(i_0)$ si i_0 est un entier impair.

Pour les signaux de longueur supérieure ou égale à deux (c'est-à-dire $i_0 < i_1 - 1$), comme illustré dans la Figure F.30, la procédure 1D_SD utilise d'abord la procédure 1D_EXTD afin d'étendre le signal X au-delà de ses limites gauche et droite, ce qui se traduit par le signal étendu X_{ext} . Puis elle utilise la procédure 1D_FILTD afin de filtrer le signal étendu X_{ext} et de produire le signal filtré souhaité Y .

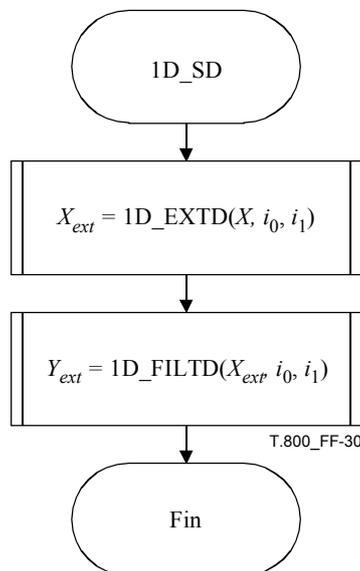


Figure F.30 – La procédure 1D_SD

F.4.7 La procédure 1D_EXTD (pour information)

La procédure 1D_EXTD est identique à la procédure 1D_EXTR, sauf pour les valeurs des paramètres $i_{left_{9,7}}$, $i_{right_{9,7}}$, $i_{left_{5,3}}$ et $i_{right_{5,3}}$ qui sont indiqués dans les Tableaux F.8 et F.9.

Tableau F.8 – Extension vers la gauche

i_0	$i_{left_{5,3}}$	$i_{left_{9,7}}$
Pair	2	4
Impair	1	3

Table F.9 – Extension vers la droite

i_1	$i_{right_{5,3}}$	$i_{right_{9,7}}$
Impair	2	4
Pair	1	3

F.4.8 La procédure 1D_FILTD (pour information)

La présente Recommandation | Norme internationale spécifie une procédure de filtrage irréversible (1D_FILTD_{9-7I}) ou une procédure de filtrage réversible (1D_FILTD_{5-3R}), selon que la transformation en ondelettes irréversible 9-7 ou réversible 5-3 est choisie.

Comme illustré dans la Figure F.31, les deux procédures reçoivent en entrée un signal unidimensionnel étendu X_{ext} , l'index du premier coefficient i_0 , et l'index du coefficient i_1 suivant immédiatement le dernier coefficient ($i_1 - 1$). Elles produisent toutes les deux un signal de sortie, Y . Les coefficients à index pair du signal Y sont une version sous-échantillonnée en passe-bas du signal étendu X_{ext} , alors que les coefficients à index impair du signal Y sont une version sous-échantillonnée en passe-haut du signal étendu X_{ext} .

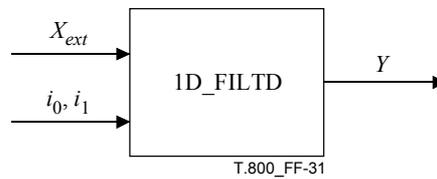


Figure F.31 – Paramètres de la procédure 1D_FILTD

F.4.8.1 La procédure 1D_FILTD_{5-3R} (pour information)

La transformation réversible décrite dans le présent article est l'implémentation réversible en mode lissage du filtrage par le filtre réversible d'ondelette 5-3. La transformation réversible est définie au moyen du filtrage en mode lissage. Les coefficients à index impair du signal de sortie Y sont calculés d'abord pour toutes les valeurs de n telles que

$$\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_1}{2} \right\rceil \text{ comme indiqué dans l'équation (F-9):}$$

$$Y(2n + 1) = X_{ext}(2n + 1) - \left\lfloor \frac{X_{ext}(2n) + X_{ext}(2n + 2)}{4} \right\rfloor \tag{F-9}$$

Puis les coefficients à index pair du signal de sortie Y sont calculés d'après les valeurs à index pair du signal étendu X_{ext} et les coefficients à index impair du signal Y sont calculés pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$ comme indiqué dans l'équation (F-10):

$$Y(2n) = X_{ext}(2n) + \left\lfloor \frac{Y(2n - 1) + Y(2n + 1) + 2}{4} \right\rfloor \tag{F-10}$$

Les valeurs de $Y(k)$ telles que $i_0 \leq k < i_1$ forment la sortie de la procédure 1D_FILTD_R.

F.4.8.2 La procédure 1D_FILTD_I (pour information)

La transformation réversible décrite dans le présent article est l'implémentation de la transformation DWT en mode lissage du filtrage par le filtre irréversible 9-7.

L'équation (F-11) décrit les quatre étapes de lissage (de 1 à 4) et les deux étapes de normalisation (5 et 6) du filtrage unidimensionnel effectué sur le signal étendu $X_{ext}(n)$ afin de produire les coefficients $i_1 - i_0$ du signal Y . Ces étapes sont effectuées dans l'ordre suivant:

Premièrement, l'étape 1 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil - 2 \leq n < \left\lceil \frac{i_1}{2} \right\rceil + 1$.

Puis l'étape 2 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_1}{2} \right\rceil + 1$, et utilise les valeurs calculées à l'étape 1.

Puis l'étape 3 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, et utilise les valeurs calculées à l'étape 1 et 2.

Puis l'étape 4 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, et utilise les valeurs calculées aux étapes 2 et 3.

Finalement, l'étape 5 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$ et utilise les valeurs calculées à l'étape 3. L'étape 6 est effectuée pour toutes les valeurs de n telles que $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$ et utilise les valeurs calculées à l'étape 4.

$$\left\{ \begin{array}{ll} Y(2n+1) = X_{ext}(2n+1) + \alpha(X_{ext}(2n) + X_{ext}(2n+2)) & [Etape1] \\ Y(2n) = X_{ext}(2n) + \beta(Y(2n-1) + Y(2n+1)) & [Etape2] \\ Y(2n+1) = Y(2n+1) + \gamma(Y(2n) + Y(2n+2)) & [Etape3] \\ Y(2n) = Y(2n) + \delta(Y(2n-1) + Y(2n+1)) & [Etape4] \\ Y(2n+1) = KY(2n+1) & [Etape5] \\ Y(2n) = (1/K)Y(2n) & [Etape6] \end{array} \right. \quad (F-11)$$

où les valeurs des paramètres de lissage α , β , γ , δ , et K sont définies dans le Tableau F.4.

Les valeurs de $Y(k)$ telles que l'étendue $i_0 \leq k < i_1$ forme la sortie de la procédure 1D_FILTD₁.

Annexe G

Décalage au niveau aperiodique et transformations multiples de composante

(Cette annexe fait partie integrante de la presente Recommandation | Norme internationale)

Dans la presente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens ou ils definissent une sortie que des implémentations en variante doivent reproduire.

La presente annexe spécifie le décalage au niveau aperiodique qui convertit les valeurs signées résultant du processus de décodage en échantillons convenablement reconstruits.

La presente annexe également décrit deux transformations multiples de composante différentes. Ces transformations multiples de composante sont utilisées afin d'améliorer le rendement de compression. Elles ne sont pas associées aux transformations multiples de composante utilisées afin d'affecter des valeurs chromatiques aux fins d'affichage. Une des transformations multiples de composante est réversible et peut être utilisée pour un codage avec ou sans perte. L'autre transformation est irréversible et ne peut être utilisée pour le codage avec perte.

G.1 Décalage au niveau aperiodique de composantes-pavés

La Figure G.1 montre le flux de décalage au niveau aperiodique dans le système avec transformation multiple de composante.

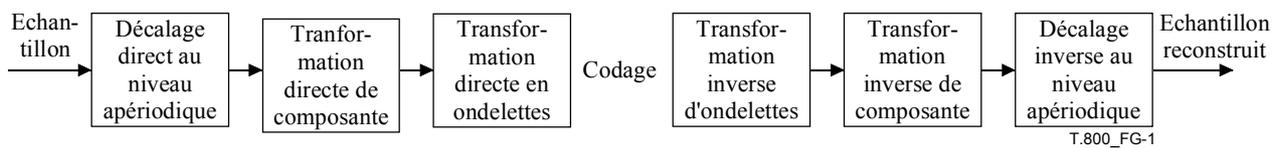


Figure G.1 – Placement du décalage au niveau aperiodique avec transformation de composante

La Figure G.2 montre le flux de décalage au niveau aperiodique dans le système sans transformation multiple de composante.

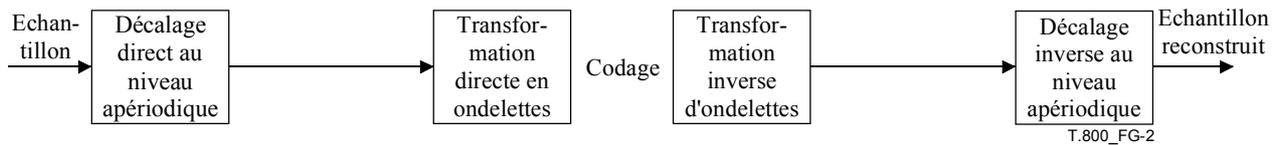


Figure G.2 – Placement du décalage au niveau aperiodique sans transformation de composante

G.1.1 Décalage au niveau aperiodique de composantes-pavés (pour information)

Le décalage au niveau aperiodique n'est effectué que sur des échantillons de composantes qui ne sont pas signés. Il est effectué avant le calcul d'une transformation multiple de composante directe (RCT ou ICT), si une telle transformation est utilisée. Sinon, il est effectué avant la transformation en ondelettes décrite à l'Annexe F. Si le bit MSB de Ss_{iz}^i d'après le segment marqueur SIZ (voir § A.5.1) est zéro, tous les échantillons $I(x, y)$ de la i^e composante sont décalés en niveau par soustraction de la même grandeur issue de chaque échantillon comme suit:

$$I(x, y) \leftarrow I(x, y) - 2^{Ss_{iz}^i} \tag{G-1}$$

G.1.2 Décalage inverse au niveau aperiodique de composantes-pavés

Le décalage inverse au niveau aperiodique n'est effectué que sur des échantillons reconstruits de composantes qui ne sont pas signés. Il est effectué après le calcul de la transformation multiple de composante inverse (RCT ou ICT), si une telle transformation est utilisée. Sinon il est effectué après la transformation inverse d'ondelettes décrite au § F. Si le bit

MSB de $Ssiz^i$ d'après le segment marqueur SIZ (voir § A.6.1) est zéro, tous les échantillons $I(x,y)$ de la i^e composante sont décalés en niveau par adjonction de la même grandeur issue de chaque échantillon comme suit:

$$I(x,y) \leftarrow I(x,y) + 2^{Ssiz^i} \quad (G-2)$$

NOTE – En raison d'effets de quantification, les échantillons reconstruits $I(x,y)$ peuvent dépasser l'étendue dynamique des échantillons originaux. Il n'y a aucune procédure normative pour cette situation de débordement ou de soupassement. Cependant, le fait de ramener la valeur à la plus proche dans l'étendue dynamique originale est une solution courante.

G.2 Transformation multiple de composante réversible (RCT)

L'utilisation de la transformation multiple de composante réversible est signalée dans le segment marqueur COD (voir § A.6.1). La transformation RCT ne doit être utilisée qu'avec le filtre réversible 5-3. La transformation RCT est une transformation réductrice de la redondance qui est appliquée aux trois premières composantes d'une image (indexées par 0, 1 et 2). Les trois composantes introduites dans la transformation RCT doivent avoir la même séparation sur la grille de référence et la même profondeur binaire.

NOTE – La transformation RCT, bien que réversible et donc capable de compression sans perte, peut être utilisée dans des flux codés tronqués afin de fournir une compression avec perte.

G.2.1 Transformation RCT directe (pour information)

Avant d'appliquer la transformation RCT directe, les échantillons de composante d'image sont décalés au niveau aperiodique, pour obtenir des composantes non signées.

La transformation RCT directe est appliquée aux composantes $I_0(x,y)$, $I_1(x,y)$, $I_2(x,y)$ comme suit:

$$Y_0(x,y) = \left\lfloor \frac{I_0(x,y) + 2I_1(x,y) + I_2(x,y)}{4} \right\rfloor \quad (G-3)$$

$$Y_1(x,y) = I_2(x,y) - I_1(x,y) \quad (G-4)$$

$$Y_2(x,y) = I_0(x,y) - I_1(x,y) \quad (G-5)$$

Si I_0 , I_1 , et I_2 sont normalisés à la même précision, alors l'équation (G-4) et l'équation (G-5) produisent une précision numérique de Y_1 et Y_2 qui est supérieure de 1 bit à la précision des composantes originales. Cette augmentation de précision est nécessaire afin d'assurer la réversibilité.

G.2.2 Transformation RCT inverse

Après la transformation inverse d'ondelettes effectuée comme décrit dans l'Annexe F, la transformation RCT inverse est appliquée comme suit:

$$I_1(x,y) = Y_0(x,y) - \left\lfloor \frac{Y_2(x,y) + Y_1(x,y)}{4} \right\rfloor \quad (G-6)$$

$$I_0(x,y) = Y_2(x,y) + I_1(x,y) \quad (G-7)$$

$$I_2(x,y) = Y_1(x,y) + I_1(x,y) \quad (G-8)$$

Après application de la transformation RCT inverse, Les composantes d'image non signées sont décalées en inverse au niveau aperiodique.

G.3 Transformation multiple de composante irréversible (ICT)

Le présent paragraphe spécifie une transformation multiple de composante irréversible. L'utilisation de la transformation de composante irréversible est signalée dans le segment marqueur COD (voir § A.6.1). La

ISO/CEI 15444-1:2004 (F)

transformation ICT ne doit être utilisée qu'avec le filtre irréversible 9-7. La transformation ICT est une transformation réductrice de la redondance qui est appliquée aux trois premières composantes d'une image (indexées par 0, 1 et 2). Les trois composantes introduites dans la transformation ICT doivent avoir la même séparation sur la grille de référence et la même profondeur binaire.

G.3.1 Transformation ICT directe (pour information)

La transformation ICT directe est appliquée à des échantillons de composante d'image $I_0(x, y)$, $I_1(x, y)$, $I_2(x, y)$, comme suit:

$$Y_0(x, y) = -0,299 I_0(x, y) - 0,587 I_1(x, y) + 0,114 I_2(x, y) \quad (\text{G-9})$$

$$Y_1(x, y) = -0,16875 I_0(x, y) - 0,331260 I_1(x, y) + 0,5 I_2(x, y) \quad (\text{G-10})$$

$$Y_2(x, y) = 0,5 I_0(x, y) - 0,41869 I_1(x, y) - 0,08131 I_2(x, y) \quad (\text{G-11})$$

NOTE – Si les trois premières composantes sont les couleurs rouge, vert et bleu, alors la transformation ICT directe est une transformation en composantes YCbCr.

G.3.2 Transformation ICT inverse

Après la transformation inverse d'ondelettes effectuée comme décrit dans l'Annexe F, la transformation ICT inverse est appliquée comme suit:

$$I_0(x, y) = Y_0(x, y) + 1,402 Y_2(x, y) \quad (\text{G-12})$$

$$I_1(x, y) = Y_0(x, y) - 0,34413 Y_1(x, y) - 0,71414 Y_2(x, y) \quad (\text{G-13})$$

$$I_2(x, y) = Y_0(x, y) + 1,772 Y_1(x, y) \quad (\text{G-14})$$

Les équations (G-12), (G-13) et (G-14) n'impliquent pas de précision requise pour les coefficients. Après l'application de la transformation ICT inverse, les échantillons non signés de la composante d'image sont décalés en inverse au niveau aperiodique.

G.4 Sous-échantillonnage de composante de chrominance et la grille de référence

La relation entre les composantes et la grille de référence est signalée dans le marqueur SIZ (voir § A.5.1) et décrite dans le § B.2.

Annexe H

Codage d'images avec régions intéressantes

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

La présente annexe décrit la technique de région intéressante (ROI, *region of interest*). Une région ROI est une partie d'image qui est codée dans le flux codé avant le reste de l'image (l'arrière-plan). Le codage est également effectué de telle façon que les informations associées à la région ROI précèdent les informations associées à l'arrière-plan. La méthode utilisée (et décrite dans la présente annexe) est la méthode du décalage maximal.

H.1 Décodage d'images ROI

La procédure spécifiée dans le présent article n'est appliquée qu'en présence d'un seul segment marqueur RGN (indiquant la présence d'une région ROI).

La procédure réaligne les bits significatifs des coefficients d'image ROI et les coefficients d'arrière-plan. Elle est définie par les étapes suivantes:

- 1) obtenir la valeur de cadrage, s , d'après le paramètre SPRgn du segment marqueur RGN contenu dans le flux codé (voir § A.6.3). Les étapes suivantes (2, 3 et 4) sont appliquées à chaque coefficient (u, v) de sous-bande b ;
- 2) si $N_b(u, v) < M_b$ (voir définition de $N_b(u, v)$ dans le § D.2.1 et de M_b dans l'équation (E-2)), alors aucune modification n'a lieu;
- 3) si $N_b(u, v) \geq M_b$ et si au moins un des M_b (voir définition dans le § E.1) premiers bits MSB ($i = 1, \dots, M_b$) est différent de zéro, alors la valeur de $N_b(u, v)$ est mise à jour en tant que $N_b(u, v) = M_b$;
- 4) si $N_b(u, v) \geq M_b$ et si tous les M_b premiers bits MSB sont égaux à zéro, alors les modifications suivantes sont effectuées:
 - a) ignorer les s premiers bits MSB et décaler les s positions binaires MSB restantes, comme décrit dans l'équation (H-1), pour $i = 1, \dots, M_b$:

$$MSB_i(b, u, v) = \begin{cases} MSB_{i+s}(b, u, v) & \text{si } i + s \leq N_b(u, v) \\ 0 & \text{si } i + s > N_b(u, v) \end{cases} \quad (\text{H-1})$$

- b) mettre à jour la valeur de $N_b(u, v)$ comme indiqué dans l'équation (H-2):

$$N_b(u, v) = \max(0, N_b(u, v) - s) \quad (\text{H-2})$$

H.2 Description de la méthode du décalage maximal

H.2.1 Codage d'images ROI (pour information)

Le codage des coefficients quantifiés de transformée est effectué de façon similaire au codage sans aucune région ROI. Du côté du codeur, un masque de région ROI est créé, décrivant les coefficients quantifiés de transformée qui doivent toujours être codés avec une meilleure qualité (même jusqu'au codage sans perte) afin de coder la région ROI avec une meilleure qualité (jusqu'au codage sans perte). Le masque de région ROI est une matrice binaire décrivant ces coefficients. Voir § H.3 pour les détails sur la façon dont le masque est produit.

Les coefficients quantifiés de transformée à l'extérieur du masque de région ROI, appelés coefficients d'arrière-plan, sont normalisés vers le bas de façon que les bits associés à la région ROI soient placés dans des plans binaires plus élevés que l'arrière-plan. En d'autres termes, lorsque le codeur entropique code les coefficients quantifiés de transformée, les plans binaires associés à la région ROI sont codés avant les informations associées à l'arrière-plan.

La méthode peut être décrite au moyen des étapes suivantes:

- 1) produire un masque ROI, $M(x, y)$ (voir § H.3);
- 2) trouver la valeur de cadrage s (voir § H.2.2);

- 3) ajouter s bits LSB à chaque coefficient $|q_b(u, v)|$. Le nombre M'_b de plans binaires de grandeur sera alors:

$$M'_b = M_b + s \quad (\text{H-3})$$

où le nombre M_b est donné par l'équation (E-2) et où la nouvelle valeur de chaque coefficient est donnée par:

$$|q_b(u, v)| = |q_b(u, v)| \cdot 2^s \quad (\text{H-4})$$

- 4) normaliser vers le bas tous les coefficients d'arrière-plan donnés par $M(x, y)$ au moyen de la valeur de cadrage s (voir § H.3). Donc, si $|q_b(u, v)|$ est un coefficient d'arrière-plan donné par $M(x, y)$, alors

$$|q_b(u, v)| = \frac{|q_b(u, v)|}{2^s} \quad (\text{H-5})$$

- 5) écrire la valeur de cadrage s dans le flux codé au moyen du paramètre SPrgn du segment marqueur RGN.

Après ces étapes les coefficients quantifiés de transformée sont codés entropiquement comme de coutume.

H.2.2 Sélection de valeur de cadrage, s , du côté codeur (pour information)

La valeur de cadrage, s , peut être choisie de façon que l'équation (H-6) soit vérifiée, où $\max(M_b)$ est le plus grand nombre de plans binaires de grandeur, voir équation (E-1), pour tout coefficient d'arrière-plan, $q_{BG}(x, y)$ dans tout bloc codé de la composante actuelle.

$$s \geq \max(M_b) \quad (\text{H-6})$$

Cela garantit que la valeur de cadrage utilisée sera assez grande pour garantir que tous les bits significatifs associés à la région ROI seront dans des plans binaires supérieurs à ceux de tous les bits significatifs associés à l'arrière-plan.

H.3 Remarques sur le codage de région intéressante (pour information)

La fonction de région ROI décrite dans le § H.2 dépend seulement de la valeur de cadrage choisie du côté codeur et donc seulement de l'amplitude des coefficients du côté décodeur. Il appartient au codeur de produire un masque qui correspond aux coefficients qui doivent être codés avec une meilleure qualité afin d'obtenir une région ROI ayant une meilleure qualité que l'arrière-plan. Le § H.3.1 décrit comment produire le masque de région ROI pour une région particulière de l'image. Le § H.3.2 décrit comment produire le masque dans le cas d'images à composantes multiples et le § H.3.3 décrit comment produire le masque de région ROI pour des régions disjointes. Le § H.3.4 décrit une façon possible de traiter l'augmentation de la profondeur binaire des coefficients. Le § H.3.5 décrit comment le masque de région ROI peut être étendu de façon à ne pas correspondre exactement à une région dans le domaine d'image et comment la méthode du décalage maximal peut servir à coder la région ROI et l'arrière-plan avec une qualité différente.

H.3.1 Production de masque de région intéressante (pour information)

Afin d'obtenir une région ROI ayant une meilleure qualité que le reste de l'image tout en conservant un taux de compression satisfaisant, des bits doivent être sauvegardés par envoi de moins d'informations pour l'arrière-plan. A cette fin, un masque de région ROI est calculé. Le masque est un plan binaire indiquant un ensemble de coefficients quantifiés de transformée dont le codage est suffisant pour que le récepteur puisse reconstruire la région recherchée avec une meilleure qualité que l'arrière-plan (jusqu'au codage sans perte).

Afin d'illustrer le concept de production de masque de région ROI, limitons-nous à une seule région ROI et à une seule composante d'image, puis identifions les échantillons qui appartiennent à la région ROI dans le domaine d'image au moyen d'un masque binaire, $M(x, y)$, dans lequel

$$M(x, y) = \begin{cases} 1 & \text{coefficient ondelettes } (x, y) \text{ nécessaire} \\ 0 & \text{la précision } (x, y) \text{ peut être affectée sans affecter la région ROI} \end{cases} \quad (\text{H-7})$$

Le masque est une carte de la région ROI dans le domaine des ondelettes telle qu'il possède une valeur différente de zéro dans la région ROI et 0 en dehors. A chaque étape, la sous-bande LL du masque est mise à jour rangée par rangée puis colonne par colonne. Le masque indiquera alors les coefficients qui sont nécessaires à cette étape de façon que la transformation inverse d'ondelettes reproduise les coefficients du précédent masque.

Par exemple, la dernière étape de la transformation inverse d'ondelettes est une composition de deux sous-bandes en une. Afin de pouvoir récupérer cette étape, l'on recherche les coefficients des deux sous-bandes qui sont nécessaires. L'étape précédente est une composition de quatre sous-bandes en deux. Afin de pouvoir récupérer cette étape, l'on recherche les coefficients des quatre sous-bandes qui sont nécessaires afin de permettre une reconstruction parfaite des coefficients inclus dans le masque pour deux sous-bandes.

Toutes les étapes sont alors récupérées afin d'obtenir le masque. Si les coefficients correspondant au masque sont transmis et reçus, et si la transformation inverse d'ondelettes est calculée d'après ces coefficients, la région ROI recherchée sera reconstruite avec une meilleure qualité que le reste de l'image (jusqu'au codage sans perte si les coefficients de la région ROI ont été codés sans perte).

L'on trouvera ci-dessous une description de la façon dont l'expansion du masque est acquise d'après les divers filtres. De semblables méthodes peuvent être utilisées pour d'autres filtres.

H.3.1.1 Production de masque de région intéressante au moyen du filtre réversible 5-3 (pour information)

Afin d'obtenir l'ensemble optimal de coefficients quantifiés à normaliser, les équations suivantes, décrites dans le présent article, doivent être utilisées.

Afin de voir quels coefficients doivent être dans le masque, la transformation inverse d'ondelettes est étudiée. L'équation (F-5) et l'équation (F-6) donnent les coefficients nécessaires pour reconstruire $X(2n)$ et $X(2n+1)$ sans perte. L'on peut constater immédiatement que ce sont $L(n)$, $L(n+1)$, $H(n-1)$, $H(n)$, $H(n+1)$ (voir Figure H.1). Donc, si $X(2n)$ et $X(2n+1)$ sont dans la région ROI, les coefficients de sous-bande du côté bas et du côté haut sont dans le masque. Noter que $X(2n)$ et $X(2n+1)$ sont respectivement des points à index pair et impair par rapport à l'origine de la grille de référence.

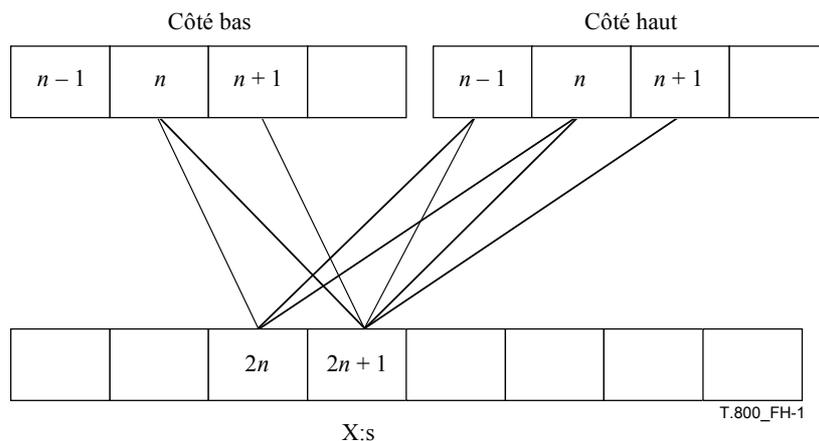


Figure H.1 – La transformation inverse d'ondelettes avec le filtre réversible 5-3

H.3.1.2 Production de masque de région intéressante au moyen du filtre irréversible 9-7 (pour information)

Un décodage correct ne dépend pas de la sélection des échantillons à normaliser. Afin d'obtenir l'ensemble optimal de coefficients quantifiés à normaliser, les équations suivantes, décrites dans le présent article, doivent être utilisées.

Pour savoir quels coefficients doivent être dans le masque, la transformation inverse d'ondelettes est étudiée comme indiqué dans le § H.3.1.1 et comme illustré par la Figure H.2. $X(2n)$ et $X(2n+1)$ sont respectivement des points à index pair et impair, associé à l'origine de la grille de référence.

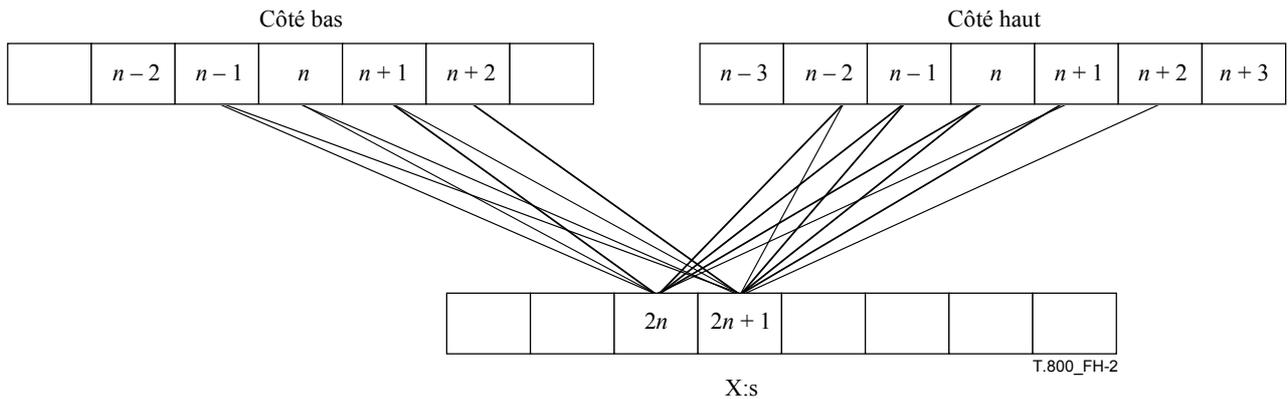


Figure H.2 – La transformation inverse d'ondelettes avec le filtre irréversible 9-7

L'on peut immédiatement constater que les coefficients nécessaires pour reconstruire $X(2n)$ et $X(2n+1)$ sans perte sont $L(n-1)$ à $L(n+2)$ et $H(n-2)$ à $H(n+2)$. Donc, si $X(2n)$ et $X(2n+1)$ sont dans la région ROI, les coefficients de sous-bande du côté bas et du côté haut sont dans le masque.

H.3.2 Remarque sur les composantes multiples (pour information)

Dans le cas d'images en couleur, la méthode s'applique séparément à chaque composante de couleur. Si certaines composantes de couleur sont sous-échantillonnées, le masque pour les composantes sous-échantillonnées est créé de la même façon que le masque pour les composantes non sous-échantillonnées.

H.3.3 Remarque sur les régions disjointes (pour information)

Si la région ROI se compose de parties disjointes alors toutes les parties ont la même valeur de cadrage s .

H.3.4 Implémentation précision remarque (pour information)

Cette méthode de codage de région ROI peut dans certains cas créer des situations où l'étendue dynamique est dépassée. Ce problème est cependant résolu facilement par un simple rejet des plans binaires de plus faible poids qui dépassent la limite en raison de l'opération de sous-échantillonnage. L'effet de ce rejet sera que la région ROI aura une meilleure qualité que l'arrière-plan, même si le flux binaire entier est décodé. Cela peut cependant créer des problèmes lorsque l'image est codée avec des régions ROI en mode sans perte. Le fait de rejeter des plans binaires de plus faible poids pour l'arrière-plan peut avoir pour conséquence que celui-ci ne soit pas codé sans perte et, dans le cas le moins favorable, qu'il ne soit pas reconstruit du tout. Cela dépendra de l'étendue dynamique disponible.

H.3.5 Exemple d'usage de la méthode du décalage maximal (pour information)

La méthode du décalage maximal, décrite ci-dessus, permet à l'utilisateur/l'application de spécifier de multiples régions de forme arbitraire, qui recevront une priorité supérieure par rapport au reste de l'image. La méthode ne nécessite pas de codage ou décodage de la forme de région ROI.

La méthode du décalage maximal permet aux réalisateurs d'un codeur d'exploiter un certain nombre de fonctionnalités qui sont prises en charge par un décodeur conforme. Par exemple, il est possible d'utiliser la méthode du décalage maximal pour coder une image avec une qualité différente pour la région ROI et pour l'arrière-plan. L'image est quantifiée de façon que la région ROI obtienne la qualité recherchée (avec perte ou sans perte), puis la méthode du décalage maximal est appliquée. Si l'image est codée progressivement par couche, toutes les couches des coefficients d'ondelette appartenant à l'arrière-plan n'ont pas besoin d'être codées. Cela correspond à l'utilisation de différents échelons de quantification pour la région ROI et pour l'arrière-plan.

Si la région ROI doit être codée sans perte, le meilleur ensemble de coefficients d'ondelette donnant un résultat sans perte pour la région ROI est décrit par le masque produit au moyen des algorithmes décrits dans le § H.3.1. Cependant, la méthode du décalage maximal prend en charge l'utilisation de tout masque car le décodeur n'a pas besoin de produire le masque. Donc, le codeur peut inclure une sous-bande entière, par. ex. la sous-bande inférieure du côté bas, dans le masque ROI et donc envoyer une version à basse résolution de l'arrière-plan au début de la transmission progressive. Cette opération est effectuée par normalisation de tous les coefficients quantifiés de transformée de la sous-bande entière. En d'autres termes, l'utilisateur peut choisir la sous-bande dans laquelle il va lancer le codage de région ROI: il n'est donc pas nécessaire d'attendre la région ROI entière avant de recevoir des informations pour l'arrière-plan.

Annexe I

Syntaxe de format de fichier JP2

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale.
 Cette annexe est facultative pour le décodeur minimal.)

Dans la présente annexe, les organigrammes et tableaux ne sont normatifs que dans le sens où ils définissent une sortie que des implémentations en variante doivent reproduire.

I.1 Domaine d'application du format de fichier JP2

La présente annexe définit un format de fichier facultatif que les applications peuvent choisir d'utiliser afin d'envelopper des données d'image comprimées JPEG 2000. Bien que toutes les applications ne puissent utiliser ce format, de nombreuses applications constateront que ce format répond à leurs besoins. Cependant, les applications qui implémentent effectivement ce format de fichier doivent le faire comme décrit dans toute la présente annexe.

La présente annexe:

- spécifie un conteneur binaire pour les images comme pour les métadonnées;
- spécifie un mécanisme permettant d'indiquer les propriétés d'image, comme l'échelle de luminances ou l'espace chromatique de l'image;
- spécifie un mécanisme permettant aux lecteurs de reconnaître l'existence d'informations sur les droits de propriété intellectuelle contenues dans le fichier;
- spécifie un mécanisme permettant d'inclure des métadonnées (y compris des informations spécifiques du vendeur) dans des fichiers spécifiés par la présente Recommandation | Norme internationale.

I.2 Introduction au format de fichier JP2

Le format de fichier JPEG 2000 (format de fichier JP2) fournit une base permettant de mémoriser des données propres à l'application (métadonnées) en association avec un flux codé JPEG 2000, comme les informations qui sont requises afin d'afficher l'image. Etant donné que de nombreuses applications nécessitent qu'un tel ensemble d'informations soit associé à des données d'image comprimées, il est utile de définir le format de cet ensemble de données ainsi que la technique de compression et la syntaxe du flux codé.

Théoriquement, le format de fichier JP2 contient le flux codé JPEG 2000 ainsi que d'autres éléments informationnels de base relatifs au flux codé. L'élément de base du format de fichier JP2 est appelé boîte. Toutes les informations contenues dans le fichier JP2 sont encapsulées dans des boîtes. La présente Recommandation | Norme internationale définit plusieurs types de boîtes; la définition de chaque type spécifique de boîte définit les sortes d'informations qui peuvent être trouvées dans une boîte de ce type. Certaines boîtes seront définies comme contenant d'autres boîtes.

I.2.1 Identification de fichier

Les fichiers JP2 peuvent être indiqués au moyen de plusieurs mécanismes. Lorsqu'ils sont mémorisés dans des systèmes de fichiers informatiques traditionnels, il y a lieu que les fichiers JP2 reçoivent l'extension ".jp2" (il convient que les lecteurs admettent le mélange des casses pour les caractères alphabétiques). Dans les systèmes de fichier Macintosh, les fichiers JP2 doivent recevoir le code de type 'jp2\040'.

I.2.2 Organisation des fichiers

Un fichier JP2 représente un ensemble de boîtes. Certaines de ces boîtes sont indépendantes, et certaines de ces boîtes contiennent d'autres boîtes. La structure binaire d'un fichier est une séquence contiguë de boîtes. Le début de la première boîte doit être le premier octet du fichier, et le dernier octet de la dernière boîte doit être le dernier octet du fichier.

La structure binaire d'une boîte est définie dans le § I.4.

Logiquement, la structure d'un fichier JP2 est représentée dans la Figure I.1. Les boîtes avec bordures en trait interrompu sont facultatives dans les fichiers JP2 conformes. Cependant, une boîte facultative peut définir des boîtes obligatoires contenues dans cette boîte facultative. Dans ce cas, si la boîte facultative existe, les boîtes obligatoires doivent exister dans la boîte facultative. Si la boîte facultative n'existe pas, alors des boîtes obligatoires ne doivent pas non plus exister dans ces boîtes.

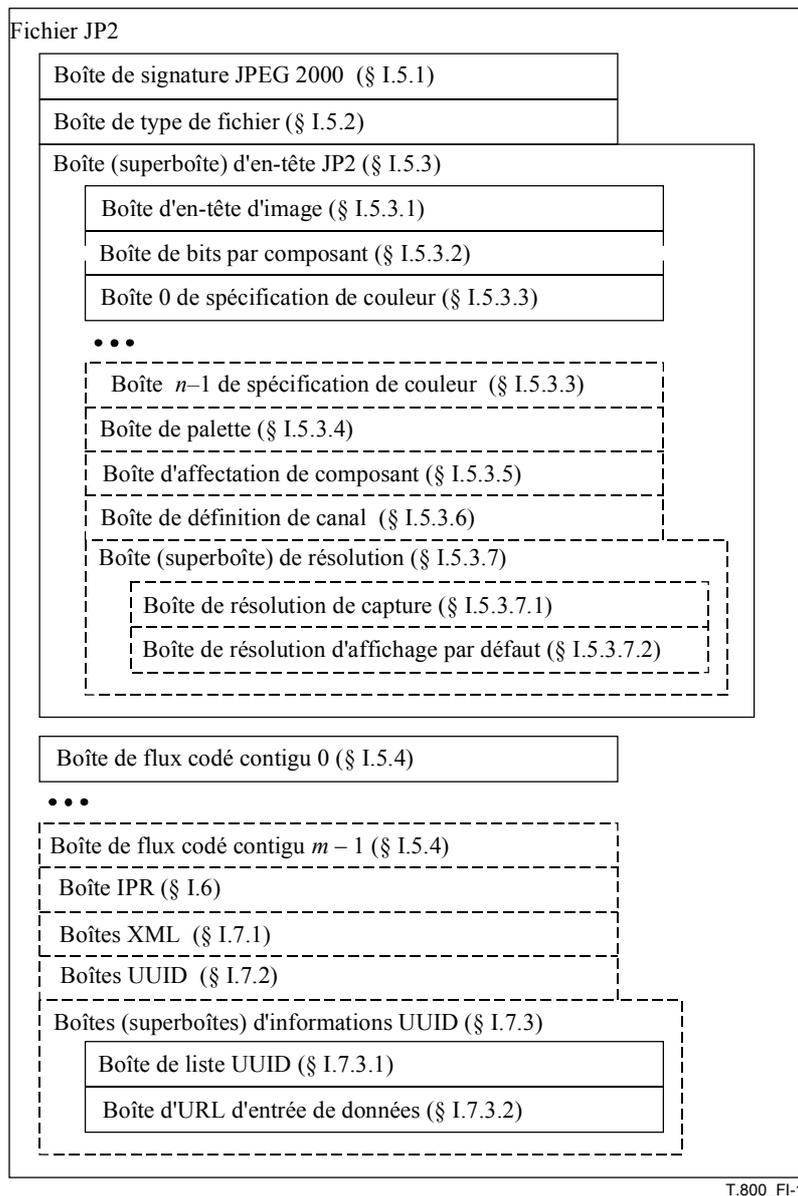


Figure I.1 – Structure théorique d'un fichier JP2

La Figure I.1 spécifie seulement la relation de confinement entre les boîtes du fichier. Un ordre particulier de ces boîtes dans le fichier n'est généralement pas impliqué. Cependant, la boîte de signature JPEG 2000 doit être la première boîte dans un fichier JP2; la boîte de type de fichier doit immédiatement suivre la boîte de signature JPEG 2000 et la boîte d'en-tête de fichier JP2 doit s'inscrire avant la boîte de flux codé contigu.

Le fichier représenté dans la Figure I.1 est une séquence de boîtes. D'autres boîtes peuvent être trouvées entre les boîtes définies dans la présente Recommandation | Norme internationale. Cependant, toutes les informations contenues dans un fichier JP2 doivent être dans le format de boîte; les flux d'octets ne respectant pas le format de boîte ne doivent pas se trouver dans le fichier.

Comme indiqué dans la Figure I.1, un fichier JP2 contient une boîte de signature JPEG 2000, une boîte d'en-tête de fichier JP2, et une ou plusieurs boîtes de flux codé contigu. Un fichier JP2 peut également contenir d'autres boîtes, selon ce qui est déterminé par le rédacteur du fichier. Par exemple, un fichier JP2 peut contenir plusieurs boîtes de langage XML (contenant des métadonnées) entre la boîte d'en-tête de fichier JP2 et la première boîte de flux codé contigu.

I.2.3 Spécification d'échelle de gris, de couleur, de palette et de composantes multiples

Le format de fichier JP2 offre deux méthodes permettant de spécifier l'espace chromatique de l'image. La méthode énumérée spécifie l'espace chromatique d'une image en indiquant une valeur numérique qui spécifie l'espace chromatique. Dans la présente Recommandation | Norme internationale, les images de l'espace chromatique sRGB et les images en échelle de gris peuvent être définies au moyen de la méthode énumérée.

Le format de fichier JP2 permet également de spécifier l'espace chromatique d'une image en imbriquant dans le fichier une forme limitée de profil ICC. Ce profil d'entrée doit être de la classe des profils monochromes ou de celle des profils à trois composantes en matrice définies par la spécification de format de profil ICC, ICC.1:1998-09. Ce format permet de spécifier une large étendue d'espaces chromatiques des classes d'échelle de gris ou de composantes RGB, ainsi que quelques autres espaces qui peuvent être représentés par ces deux classes de profil. Voir dans le § J.9 une description plus détaillée des transformations d'espace chromatique autorisées, des modes de mémorisation de ces transformées dans le fichier, et des façons de traiter une image au moyen de ces transformées sans utiliser de moteur de gestion de couleur ICC. Bien que limités, ces profils ICC sont entièrement conformes et l'image peut donc être traitée par tout moteur ICC conforme qui prend en charge les profils définis dans la norme ICC.1:1998-09.

En plus de spécifier l'espace chromatique de l'image, la présente Recommandation | Norme internationale permet de décoder une image à composantes de palette et de la reconvertir en image à composantes multiples par translation de l'espace indiciel à l'espace de composantes multiples. Toute extraction éventuelle des composantes de palette est appliquée avant que l'espace chromatique soit interprété. Dans le cas d'images à composantes de palette, la spécification de l'espace chromatique de l'image est appliquée aux valeurs de composantes multiples mémorisées dans la palette.

I.2.4 Inclusion de canaux d'opacité

Le format de fichier JP2 permet d'indiquer la présence de canaux auxiliaires (comme les canaux d'opacité), de définir le type de ces canaux, et de spécifier l'ordonnement et la source de ces canaux (qu'ils soient directement extraits du flux codé ou produits par application d'une palette à une composante de flux codé). Lorsqu'un lecteur ouvre le fichier JP2, il détermine l'ordonnement et le type de chaque composante. L'application doit alors adapter la définition de la composante et l'ordonnement d'après le fichier JP2, l'ordonnement de la composante étant défini par la spécification de l'espace chromatique. Une fois que les composantes du fichier ont été affectées aux canaux de couleur, l'image décompressée peut être traitée par d'éventuelles transformations d'espace chromatique qui pourraient être nécessaires.

Dans de nombreuses applications, des composantes autres que les canaux de couleur sont requis. Par exemple, de nombreuses images utilisées sur des pages IP contiennent des informations d'opacité; le navigateur utilise ces informations afin de fondre l'image dans l'arrière-plan. Il est donc désirable d'inclure le canal de couleur et le canal auxiliaire dans un même flux codé.

La façon dont les applications traitent des canaux d'opacité ou d'autres canaux auxiliaires est hors du domaine d'application de la présente Recommandation | Norme internationale.

I.2.5 Métadonnées

Un important aspect du format de fichier JP2 est la capacité d'ajouter des métadonnées à un fichier JP2. Etant donné que toutes les informations sont encapsulées dans des boîtes, et que toutes les boîtes ont des types, le format fournit un simple mécanisme permettant à un lecteur d'extraire les informations utiles tout en négligeant les boîtes contenant des informations non comprises par ce lecteur particulier. De cette façon, de nouvelles boîtes peuvent être créées, soit par la présente Recommandation | Norme internationale, ou par d'autres Recommandations | Normes internationales ou par implémentation privée. De même, aucune nouvelle boîte ajoutée à un fichier JP2 ne doit modifier l'aspect visuel de l'image.

I.2.6 Conformité avec le format de fichier

Tous les fichiers conformes doivent contenir toutes les boîtes requises par la présente Recommandation | Norme internationale, et ces boîtes doivent être comme définies dans la présente Recommandation | Norme internationale. De même, tous les lecteurs conformes doivent interpréter correctement toutes les boîtes requises qui sont définies dans la présente Recommandation | Norme internationale et doivent donc interpréter correctement tous les fichiers conformes.

I.3 Architecture de spécification d'échelle de gris/de couleur/de palette/de composantes multiples

Un des aspects les plus importants d'un format de fichier est qu'il spécifie l'espace chromatique des données d'image qui y sont contenues. Afin d'afficher ou d'interpréter correctement les données d'image, il est essentiel que l'espace chromatique de cette image soit convenablement caractérisé. Le format de fichier JP2 fournit un mécanisme à plusieurs niveaux permettant de caractériser l'espace chromatique d'une image.

I.3.1 Méthode énumérée

La méthode la plus simple pour caractériser l'espace chromatique d'une image est de spécifier un code d'entier représentant l'espace chromatique dans lequel l'image est codée. Cette méthode gère la spécification des images à composantes sRGB, d'échelle de gris et sYCC. Des extensions à cette méthode pourront servir à spécifier d'autres espaces chromatiques, y compris la définition d'images à composantes multiples.

Par exemple, le fichier d'image peut indiquer qu'une image particulière est codée dans l'espace chromatique sRGB. Afin d'interpréter et d'afficher correctement l'image, une application doit toujours comprendre implicitement la définition de l'espace chromatique sRGB. Etant donné qu'une application doit toujours comprendre implicitement chaque espace chromatique spécifié, la complexité de cette méthode dépend des espaces chromatiques exactement spécifiés. De même, la complexité de ce mécanisme est proportionnelle au nombre d'espaces chromatiques qui sont spécifiés et requis afin d'assurer la conformité. Alors que cette méthode fournit un haut niveau d'interopérabilité pour les images codées au moyen d'espaces chromatiques pour lesquels une interprétation correcte est requise afin d'assurer la conformité, cette méthode est très peu flexible. La présente Recommandation | Norme internationale définit un ensemble spécifique d'espaces chromatiques dont l'interprétation est requise afin d'assurer la conformité.

I.3.2 Méthode du profil ICC restreint

Une application peut également spécifier l'espace chromatique d'une image au moyen de deux types de profil ICC limités. Cette méthode gère la spécification des espaces chromatiques les plus communément utilisés: RGB et échelle de gris, au moyen d'une méthode à basse complexité.

Un profil ICC est une représentation normalisée de la transformation requise afin de convertir un espace chromatique en un autre espace chromatique. En ce qui concerne le format de fichier JP2, un profil ICC définit comment des échantillons décompressés et extraits du flux codé sont convertis en espace chromatique normalisé (l'espace de connexion de profil (PCS)). Selon l'espace chromatique original des échantillons, cette transformation peut être soit très simple soit très complexe.

La spécification de format de profil ICC définit deux classes spécifiques de profils ICC qui sont simples à implémenter et qui sont appelées, dans la spécification de profil, entrée monochrome et entrée à trois composantes en matrice. Ces profils limitent la transformation, entre l'espace chromatique d'origine et l'espace PCS_{XYZ}, à l'application d'une courbe de non-linéarité et d'une matrice en 3×3 . Il est pratique de prévoir que toutes les applications, y compris les simples dispositifs, seront en mesure de traiter l'image par cette transformation. Donc, toutes les applications conformes sont appelées à interpréter correctement l'espace chromatique de toute image qui spécifie l'espace chromatique au moyen de ce sous-ensemble de types de profil ICC possibles.

Dans le format de fichier JP2, les profils doivent être conformes à la définition de profil ICC, définition contenue dans la spécification de format de profil ICC, ICC.1:1998-09, ainsi qu'aux restrictions spécifiées ci-dessus. Voir § J.9 une description plus détaillée des transformées d'espace chromatique autorisées, de la façon dont ces transformées sont mémorisées dans le fichier, et de la façon de traiter une image au moyen de ces transformées sans utiliser de moteur de gestion de couleur ICC.

I.3.3 Utilisation de méthodes multiples

Sur le plan de l'architecture, le format permet d'imbriquer de multiples méthodes dans un fichier. Il permet à d'autres normes de définir des méthodes énumérées additionnelles ainsi que des méthodes étendues, ce qui offre aux lecteurs conformes à ces extensions un choix quant au trajet de traitement d'image qui doit servir à interpréter l'espace chromatique de l'image. Cependant, la première méthode trouvée dans le fichier (dans la première boîte de spécification d'espace chromatique contenue dans la boîte d'en-tête de fichier JP2) doit être une des méthodes définies et limitées dans la présente Recommandation | Norme internationale. Un lecteur conforme doit utiliser cette première méthode et ignorer toutes les autres méthodes (contenues dans des boîtes de spécification d'espace chromatique) trouvées dans le fichier.

I.3.4 Images à composantes de palette

En plus de spécifier l'interprétation de l'image en termes d'espace chromatique, la présente Recommandation | Norme internationale permet le décodage d'une même composante lorsque la valeur de cette composante représente un index pointant sur une palette de couleurs. L'entrée d'un échantillon décompressé dans la palette convertit la valeur unique en un nuplet à composantes multiples. La valeur de ce nuplet représente la couleur de cet échantillon; ce nuplet doit alors être interprété selon les autres méthodes de spécification de couleur (énumérée ou profils ICC limités) comme si cet échantillon de composantes multiples avait été directement extrait de composantes multiples contenues dans le flux codé.

I.3.5 Interactions avec la transformée multiple de composante réductrice de la redondance

La spécification de couleur en format de fichier JP2 est indépendante de l'utilisation d'une transformation multiple de composante dans le flux codé (le paramètre CSsiz du segment marqueur SIZ spécifié dans le § A.5.1 et dans l'Annexe G). Les transformations d'espace chromatique spécifiées par la séquence de boîtes de spécification de couleurs doivent être appliquées aux échantillons d'image après que la transformation multiple de composante inverse a été appliquée aux échantillons décompressés. Bien que l'application de ces transformations de composante réductrices de la redondance soit distincte, l'application d'une transformation multiple de composante sur la base d'un codeur améliorera souvent la compression des données d'image en couleur.

I.3.6 Légende des descriptions graphiques (pour information)

Chaque boîte est décrite en termes de sa fonction, de son usage et de sa longueur. La fonction décrit les informations contenues dans la boîte. L'usage décrit l'emplacement logique et la fréquence de cette boîte dans le fichier. La longueur décrit les paramètres qui déterminent la longueur de la boîte.

Ces descriptions sont suivies par une figure qui montre l'ordre et la relation des paramètres dans la boîte. La Figure I.2 montre un exemple de ce type de figure. Un rectangle sert à indiquer les paramètres contenus dans la boîte. La largeur du rectangle est proportionnelle au nombre d'octets dans le paramètre. Un rectangle ombré (zones diagonales) indique que le paramètre est de longueur variable. Deux paramètres avec indices supérieurs et une zone grise intermédiaire indiquent une plage de plusieurs de ces paramètres. Une séquence de deux groupes de paramètres multiples avec indices supérieurs séparés par une zone grise indique une plage de ce groupe des paramètres (un ensemble de chaque paramètre dans le groupe, suivi du prochain ensemble de chaque paramètre dans le groupe). Les paramètres ou boîtes facultatifs seront représentés par un rectangle ombré.

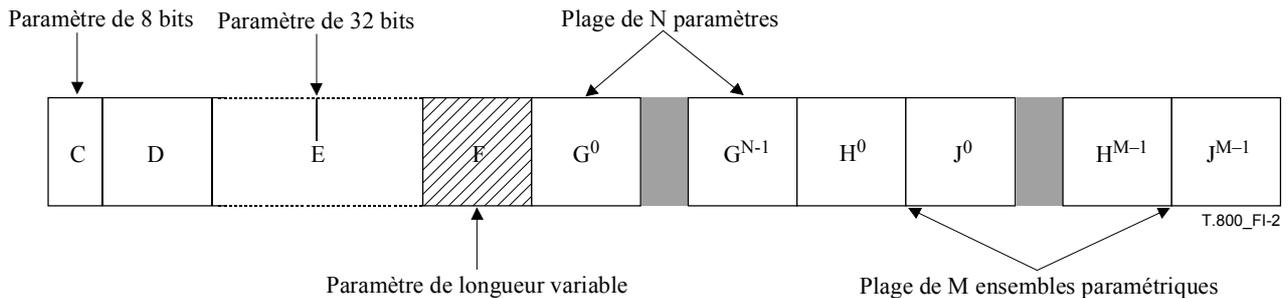


Figure I.2 – Exemple des figures descriptives de boîte

La figure est suivie d'une liste qui décrit la signification de chaque paramètre dans la boîte. Si des paramètres sont répétés, la longueur et nature de la plage de ces paramètres sont définies. Par exemple, dans la Figure I.2, les paramètres C, D, E et F ont respectivement 8, 16, 32 bits et une longueur variable. La notation G^i et G^{N-1} implique qu'il y a N différents paramètres, G^i , dans une rangée. Le groupe des paramètres H^0 et H^{M-1} , et J^0 et J^{M-1} spécifie que la boîte contiendra le paramètre H^0 , suivi par J^0 , suivi par H^1 et J^1 , jusqu'à H^{M-1} et J^{M-1} (M instances de chaque paramètre au total). De même, le champ E est facultatif et ne peut pas être trouvé dans cette boîte.

Après la liste se trouve un tableau qui soit décrit les valeurs paramétriques permises ou fournit des références aux autres tableaux qui décrivent ces valeurs.

En outre, dans une figure décrivant le contenu d'une superboîte, des points de suspension (...) seront utilisés pour indiquer que le contenu du fichier entre deux boîtes n'est pas spécifiquement défini. Toute boîte (ou séquence de boîtes) peut, sauf spécification contraire par la définition de cette boîte, être trouvée au lieu des points de suspension.

Par exemple, la superboîte représentée dans la Figure I.3 doit toujours contenir une boîte AA et une boîte BB, et la boîte BB doit toujours suivre la boîte AA. Cependant, il peut y avoir d'autres boîtes entre les boîtes AA et BB. Le traitement des boîtes inconnues est examiné dans le § I.8.



Figure I.3 – Exemple des figures descriptives de superboîte

I.4 Définition de boîte

Physiquement, chaque objet contenu dans le fichier est encapsulé dans une structure binaire appelée boîte. Cette structure binaire se présente comme dans la Figure I.4:



Figure I.4 – Organisation d'une boîte

- LBox:** longueur de boîte. Ce champ spécifie la longueur de la boîte, mémorisée sous forme d'un entier non signé gros-boutiste de 4 octets. Cette valeur comporte tous les champs de la boîte, y compris la longueur et le type. Si la valeur de ce champ est 1, alors le champ XLBox doit exister et la valeur de ce champ doit être la longueur réelle de la boîte. Si la valeur de ce champ est 0, alors la longueur de la boîte n'était pas connue lorsque le champ LBox a été écrit. Dans ce cas, cette boîte contient tous les octets jusqu'à la fin du fichier. Si une boîte de longueur 0 est contenue dans une autre boîte (sa superboîte), alors la longueur de cette superboîte doit également être 0. En d'autres termes, cette boîte est la dernière dans le fichier. Les valeurs 2 à 7 sont réservées pour utilisation par l'ISO.
- TBox:** type de boîte. Ce champ spécifie le type d'informations trouvé dans le champ DBox. La valeur de ce champ est codé sous forme d'un entier non signé gros-boutiste de 4 octets. Cependant, les boîtes sont généralement désignées par une conversion en chaîne de caractères ISO/CEI 646 de la valeur d'entier. Pour tous les types de boîte définis dans la présente Recommandation | Norme internationale, le type de boîte sera indiqué aussi bien par une chaîne de caractères (normative) que par des entiers hexadécimaux de 4 octets (pour information). De même, un caractère d'espace est représenté dans la conversion en chaîne de caractères du type de boîte par "\040". Toutes les valeurs de TBox non définies dans la présente Recommandation | Norme internationale sont réservées pour utilisation par l'ISO.
- XLBox:** longueur étendue de boîte. Ce champ spécifie la longueur réelle de la boîte si la valeur du champ LBox est 1. Ce champ est mémorisé sous forme d'entier non signé gros-boutiste de 8 octets. La valeur comporte tous les champs de la boîte, y compris les champs LBox, TBox et XLBox.
- DBox:** contenu de boîte. Ce champ contient les informations réelles contenues dans cette boîte. Le format du contenu de boîte dépend du type de boîte et sera défini individuellement pour chaque type.

Tableau I.1 – Structure binaire d'une boîte

Nom du champ	Longueur (bits)	Valeur
LBox	32	0, 1, ou 8 à $(2^{32} - 1)$
TBox	32	Variable
XLBox	64 0	16 à $(264 - 1)$; si LBox = 1 Non applicable; si LBox ≠ 1
DBox	Variable	Variable

Par exemple, la Figure I.5 décrit une séquence de boîtes, y compris une boîte qui contient d'autres boîtes:

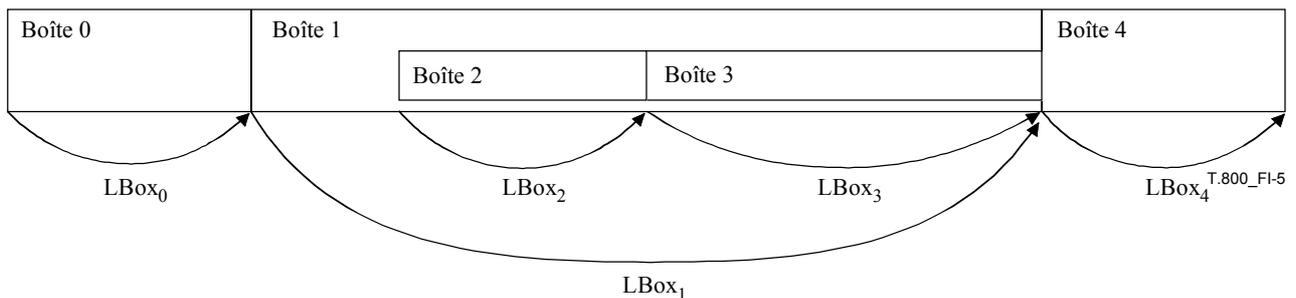


Figure I.5 – Illustration des longueurs de boîte

Comme indiqué dans la Figure I.5, la longueur de chaque boîte comporte d'éventuelles boîtes contenues dans cette boîte. Par exemple, la longueur de la boîte 1 comporte la longueur des boîtes 2 et 3, en plus des champs LBox et TBox pour la boîte 1 proprement dite. Dans ce cas, si le type de boîte 1 n'a pas été compris par un lecteur, celui-ci ne reconnaîtra pas l'existence des boîtes 2 et 3 parce qu'elles seront complètement omises par saut de la longueur de la boîte 1 à partir du début de la boîte 1.

Le Tableau I.2 énumère toutes les boîtes définies par la présente Recommandation | Norme internationale. Les retraits dans ce tableau indiquent la structure de confinement hiérarchique des boîtes dans un fichier JP2:

Tableau I.2 – Boîtes définies

Nom de boîte	Type	Superboîte	Requis?	Commentaires
Boîte de signature JPEG 2000	'jP\040\040' (0x6A50 2020)	Non	Requis	Cette boîte indique uniquement le fichier comme faisant partie de la famille de fichiers JPEG 2000.
Boîte de type de fichier	'ftyp' (0x6674 7970)	Non	Requis	Cette boîte spécifie le type, la version et les informations de compatibilité. Elle spécifie également si ce fichier est un fichier JP2 conforme ou s'il peut être lu par un lecteur JP2 conforme.
Boîte d'en-tête de fichier JP2	'jp2h' (0x6A70 3268)	Oui	Requis	Cette boîte contient une série de boîtes qui contient des informations de type en-tête sur le fichier.
Boîte d'en-tête d'image	'ihdr' (0x6968 6472)	Non	Requis	Cette boîte spécifie la longueur de l'image et d'autres champs associés.
Boîte de bits par composante	'bpc' (0x6270 6363)	Non	Facultatif	Cette boîte spécifie la profondeur binaire des composantes dans le fichier si la profondeur binaire n'est pas constante entre toutes les composantes.
Boîte de spécification de couleur	'colr' (0x636F 6C72)	Non	Requis	Cette boîte spécifie l'espace chromatique de l'image.
Boîte de palette	'pclr' (0x7063 6C72)	Non	Facultatif	Cette boîte spécifie la palette qui affecte une même composante d'espace indiciel à une image à composantes multiples.
Boîte d'affectation des composantes	'cmap' (0x636D 6170)	Non	Facultatif	Cette boîte spécifie l'affectation entre une palette et des composantes de flux codé.
Boîte de définition de canal	'cdef' (0x6364 6566)	Non	Facultatif	Cette boîte spécifie le type et l'ordonnement des composantes dans le flux codé, ainsi que celles qui sont créées par l'application d'une palette.
Boîte de résolution	'res\040' (0x7265 7320)	Oui	Facultatif	Cette boîte contient la résolution de grille.
Boîte de résolution de capture	'resc' (0x7265 7363)	Non	Facultatif	Cette boîte spécifie la résolution de grille à laquelle l'image a été capturée.
Boîte de résolution d'affichage par défaut	'resd' (0x7265 7364)	Non	Facultatif	Cette boîte spécifie la résolution de grille par défaut à laquelle l'image doit être affichée.
Boîte de flux codé contigu	'jp2c' (0x6A70 3263)	Non	Requis	Cette boîte contient le flux codé défini par l'Annexe A.
Boîte de propriété intellectuelle	'jp2i' (0x6A70 3269)	Non	Facultatif	Cette boîte contient des informations de propriété intellectuelle sur l'image.
Boîte de langage XML	'xml\040' (0x786D 6C20)	Non	Facultatif	Cette boîte fournit un outil permettant aux vendeurs d'ajouter des informations formatées en XML à un fichier JP2.
Boîte d'identification UUID	'uuid' (0x7575 6964)	Non	Facultatif	Cette boîte fournit un outil permettant aux vendeurs d'ajouter de nouvelles informations à un fichier sans risquer de conflit avec d'autres vendeurs.
Boîte d'informations d'identification UUID	'uin' (0x7569 6E66)	Oui	Facultatif	Cette boîte fournit un outil permettant à un vendeur de fournir un accès à des informations additionnelles qui sont associées à un UUID.
Boîte de liste d'identificateurs UUID	'ulst' (0x7563 7374)	Non	Facultatif	Cette boîte spécifie une liste des identificateurs UUID.
Boîte d'URL	'url\040' (0x7572 6C20)	Non	Facultatif	Cette boîte spécifie une adresse URL.

I.5 Boîtes définies

Les boîtes suivantes doivent être correctement interprétées par tous les lecteurs conformes. Chacune de ces boîtes est conforme à la structure de boîte normalisée qui est définie dans le § I.4. Les paragraphes suivants définissent la valeur du champ DBox d'après le Tableau I.1 (le contenu de la boîte). Il est supposé que les champs LBox, TBox et XlBox existent pour chaque boîte contenue dans le fichier comme défini dans le § I.4.

I.5.1 Boîte de signature JPEG 2000

La boîte de signature JPEG 2000 indique que le format de ce fichier a été défini par la Recommandation | Norme internationale JPEG 2000. Elle fournit également quelques informations qui peuvent aider à déterminer la validité du reste du fichier. La boîte de signature JPEG 2000 doit être la première boîte dans le fichier, et tous les fichiers doivent contenir une et une seule boîte de signature JPEG 2000.

Le type de la boîte de signature JPEG 2000 doit être 'jp\040\040' (0x6A50 2020). La longueur de cette boîte doit être 12 octets. Le contenu de cette boîte doit être la chaîne de caractères de 4 octets '<CR><LF><0x87><LF>' (0x0D0A 870A). Pour les objets de vérification de fichier, cette boîte peut être considérée comme une chaîne de longueur fixe de 12 octets qui doit avoir la valeur: 0x0000 000C 6A50 2020 0D0A 870A.

La combinaison du type et du contenu particuliers à cette boîte permet à une application de détecter un ensemble commun d'erreurs de transmission de fichier. La séquence CR-LF dans le contenu recueille les mauvais transferts de fichier qui altèrent les séquences de retour à la ligne. Le caractère final d'avancement de ligne règle le problème inverse de conversion de la séquence CR-LF. Le troisième caractère du contenu de boîte a son bit de plus fort poids régler de façon à recueillir les mauvais transferts de fichier qui libèrent le bit 7.

I.5.2 Boîte de type de fichier

La boîte de type de fichier spécifie la Recommandation | Norme internationale qui définit complètement tout le contenu de ce fichier, ainsi qu'une liste séparée des lecteurs, définie par d'autres Recommandations | Norme internationales, avec lesquels ce fichier est compatible. Le fichier peut donc être correctement interprété dans le domaine d'application d'une autre norme. Cette boîte doit suivre immédiatement la boîte de signature JPEG 2000. Cela établit la différence entre la norme qui décrit complètement le fichier et les autres normes qui interprètent un sous-ensemble du fichier.

Tous les fichiers doivent contenir une et une seule boîte de type de fichier.

Le type de la boîte de type de fichier doit être 'ftyp' (0x6674 7970). Le contenu de cette boîte doit être comme décrit dans la Figure I.6:

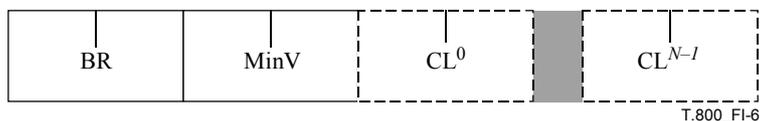


Figure I.6 – Organisation du contenu d'une boîte de type de fichier

BR: marque. Ce champ spécifie la Recommandation | Norme internationale qui définit complètement ce fichier. Ce champ est spécifié par une chaîne de quatre octets de caractères ISO/CEI 646. La valeur de ce champ est définie dans le Tableau I.3:

Tableau I.3 – Valeurs de marque légale

Valeur	Signification
'jp2\040'	IS 15444-1, Annexe I (La présente Recommandation Norme internationale)
Autres valeurs	Réservé pour autres utilisations par l'ISO

En outre, le champ de marque doit être considéré comme fonctionnellement équivalent à un numéro de version majeur. Un changement de version majeur (si tant est qu'il y en ait un), représentant un changement incompatible dans le format de fichier JP2, doit définir une valeur différente pour le champ de marque.

Si la valeur du champ de marque n'est pas 'jp2\040', alors une valeur de 'jp2\040' dans la liste de compatibilité indique qu'un lecteur JP2 peut interpréter le fichier d'une certaine façon comme prévu par le créateur du fichier.

MinV: version mineure. Ce paramètre définit le numéro de version mineure de la spécification JP2 à laquelle le fichier se conforme. Ce paramètre est défini sous forme d'un entier non signé gros-boutiste de 4 octets. La valeur de ce champ doit être zéro. Cependant, les lecteurs doivent continuer à analyser et à interpréter ce fichier même si la valeur de ce champ n'est pas zéro.

CLⁱ: liste de compatibilité. Ce champ spécifie un code représentant la présente Recommandation | Norme internationale, une autre norme, ou un profil d'une autre norme, auquel le fichier est conforme. Ce champ est codé comme une chaîne de quatre octets de caractères ISO/CEI 646. Un fichier qui est conforme à la présente Recommandation | Norme internationale doit avoir au moins un champ CLⁱ dans la boîte de type de fichier, et doit contenir la valeur 'jp2\040' dans un champ CLⁱ dans la boîte de type de fichier, et tous les lecteurs conformes doivent interpréter correctement tous les fichiers contenant 'jp2\040' dans un des champs CLⁱ.

Si un des champs CLⁱ contient la valeur "J2P0" alors le premier flux codé contenu dans ce fichier JP2 est limité comme décrit pour le profil 0 d'après le Tableau A.45.

Si un des champs CLⁱ contient la valeur "J2P1" alors le premier flux codé contenu dans ce fichier JP2 est limité comme décrit pour le profil 1 d'après le Tableau A.45.

Les autres valeurs du champ Liste de compatibilité sont réservées pour utilisation par l'ISO.

Le nombre de champs CLⁱ est déterminé par la longueur de cette boîte.

Tableau I.4 – Format du contenu de la boîte de type de fichier

Nom du champ	Longueur (bits)	Valeur
BR	32	0 à $(2^{32} - 1)$
MinV	32	0
CL ⁱ	32	0 à $(2^{32} - 1)$

I.5.3 Boîte d'en-tête de fichier JP2 (superboîte)

La boîte d'en-tête de fichier JP2 contient des informations génériques sur le fichier, comme le nombre de composantes, l'espace chromatique, et la résolution de grille. Cette boîte est une superboîte. Dans un fichier JP2, il doit y avoir une et une seule boîte d'en-tête de fichier JP2. La boîte d'en-tête de fichier JP2 peut être située n'importe où dans le fichier après la boîte de type de fichier mais avant la boîte de flux codé contigu. Elle doit également être toujours au même niveau que les boîtes de signature JPEG 2000 et de type de fichiers (elle ne doit pas être contenue dans une autre superboîte du fichier).

Le type de la boîte d'en-tête de fichier JP2 doit être 'jp2h' (0x6A70 3268).

Cette boîte contient plusieurs boîtes. Les autres boîtes peuvent être définies dans d'autres normes et peuvent être ignorées par les lecteurs conformes. Ces boîtes, contenues dans la boîte d'en-tête de fichier JP2 et définies dans la présente Recommandation | Norme internationale sont décrites dans la Figure I.7:

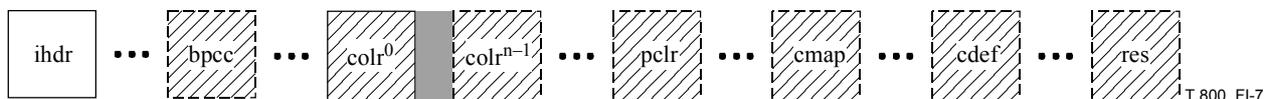


Figure I.7 – Organisation du contenu d'une boîte d'en-tête de fichier JP2

ihdr: boîte d'en-tête d'image. Cette boîte spécifie des informations sur l'image, comme sa hauteur et sa largeur. Sa structure est spécifiée dans le § I.5.3.1. Cette boîte doit être la première boîte dans la boîte d'en-tête de fichier JP2.

bpcc: boîte de bits par composant. Cette boîte spécifie la profondeur binaire de chaque composante contenue dans le flux codé après décompression. Sa structure est spécifiée dans le § I.5.3.2. Cette boîte peut être trouvée n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elle vienne après la boîte d'en-tête d'image.

colrⁱ: boîte de spécification de couleurs. Ces boîtes spécifient l'espace chromatique de l'image décompressée. Leurs structures sont spécifiées dans le § I.5.3.3. Il doit y avoir au moins une boîte de spécification de couleur dans la boîte d'en-tête de fichier JP2. L'utilisation de multiples boîtes de spécification de couleur fournit la capacité de donner à un décodeur de multiples options d'optimisation ou de compatibilité pour le traitement chromatique. Ces boîtes peuvent être trouvées

n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elles viennent après la boîte d'en-tête d'image. Toutes les boîtes de spécification de couleur doivent être contiguës dans la boîte d'en-tête de fichier JP2.

- pclr:** boîte de palette. Cette boîte définit la palette à utiliser afin de créer des composantes multiples à partir d'une même composante. Sa structure est spécifiée dans le § I.5.3.4. Cette boîte peut être trouvée n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elle vienne après la boîte d'en-tête d'image.
- cmap:** boîte de mappage des composantes. Cette boîte définit comment les canaux d'image sont indiqués d'après les composantes réelles contenues dans le flux codé. Sa structure est spécifiée dans le § I.5.3.5. Cette boîte peut être trouvée n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elle vienne après la boîte d'en-tête d'image.
- cdef:** boîte de définition de canal. Cette boîte définit les canaux dans l'image. Sa structure est spécifiée dans le § I.5.3.6. Cette boîte peut être trouvée n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elle vienne après la boîte d'en-tête d'image.
- res:** boîte de résolution. Cette boîte spécifie les résolutions de grille lors de la capture et de l'affichage par défaut de l'image. Sa structure est spécifiée dans le § I.5.3.7. Cette boîte peut être trouvée n'importe où dans la boîte d'en-tête de fichier JP2 à condition qu'elle vienne après la boîte d'en-tête d'image.

I.5.3.1 Boîte d'en-tête d'image

Cette boîte contient des informations génériques de longueur fixe sur l'image, comme les dimensions d'image et le nombre de composantes. Le contenu de la boîte d'en-tête de fichier JP2 doit commencer par une boîte d'en-tête d'image. Les instances de cette boîte à d'autres emplacements dans le fichier doivent être ignorées. La longueur de la boîte d'en-tête d'image doit être de 22 octets, y compris les champs de longueur de boîte et de type. Une grande partie des informations contenues dans la boîte d'en-tête d'image est redondante avec les informations mémorisées dans le flux codé proprement dit.

Toutes les références au "flux codé" dans les descriptions de champs contenues dans cette boîte d'en-tête d'image s'appliquent au flux codé trouvé dans la première boîte de flux codé contiguë dans le fichier. Les fichiers qui contiennent des informations contradictoires entre la boîte d'en-tête d'image et le premier flux codé ne sont pas des fichiers conformes. Cependant, les lecteurs peuvent choisir d'essayer de lire ces fichiers au moyen des valeurs trouvées dans le flux codé.

Le type de la boîte d'en-tête d'image doit être 'ihdr' (0x6968 6472) et le contenu de la boîte doit avoir le format comme décrit dans la Figure I.8:

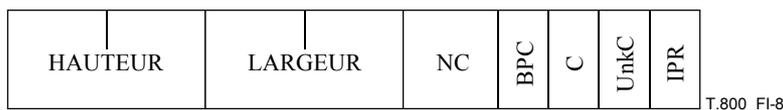


Figure I.8 – Organisation du contenu d'une boîte d'en-tête d'image

HAUTEUR: hauteur de zone d'image. La valeur de ce paramètre indique la hauteur de la zone d'image. Ce champ est mémorisé sous forme d'un entier non signé gros-boutiste de 4 octets. La valeur de ce champ doit être Ysiz – YOsiz, où Ysiz et YOsiz sont les valeurs des champs respectifs dans le marqueur SIZ contenu dans le flux codé. Voir à la Figure B.1 une illustration de la zone d'image. Cependant, les points de grille de référence ne sont pas nécessairement carrés; le rapport hauteur/largeur d'un point de grille de référence est spécifié par la boîte de résolution. Si la boîte de résolution n'est pas présente, alors un lecteur doit partir du principe que les points de grille de référence sont carrés;

LARGEUR: largeur de zone d'image. La valeur de ce paramètre indique la largeur de la zone d'image. Ce champ est mémorisé sous forme d'un entier non signé gros-boutiste de 4 octets. La valeur de ce champ doit être Xsiz – XOsiz, où Xsiz et XOsiz sont les valeurs des champs respectifs dans le marqueur SIZ contenu dans le flux codé. Voir à la Figure B.1 une illustration de la zone d'image. Cependant, les points de grille de référence ne sont pas nécessairement carrés; le rapport hauteur/largeur d'un point de grille de référence est spécifié par la boîte de résolution. Si la boîte de résolution n'est pas présente, alors un lecteur doit partir du principe que les points de grille de référence sont carrés;

NC: nombre de composantes. Ce paramètre spécifie le nombre de composantes contenues dans le flux codé et est mémorisé sous forme d'entier non signé de 2 octets gros-boutiste. La valeur de

ce champ doit être égale à la valeur du Champ Csiz dans le marqueur SIZ contenu dans le flux codé;

- BPC:** bits par composante. Ce paramètre spécifie la profondeur binaire des composantes contenues dans le flux codé, moins 1, et est mémorisé sous forme de champ de 1 octet.
- Si la profondeur binaire et le signe sont les mêmes pour toutes les composantes, alors ce paramètre spécifie cette profondeur binaire et doit être équivalent aux valeurs des champs Ssizⁱ du marqueur SIZ contenu dans le flux codé (qui doivent toutes être égales). Si les composantes varient en profondeur binaire et/ou signe, alors la valeur de ce champ doit être de 255 et la boîte d'en-tête de fichier JP2 doit également contenir une boîte de bits par composante définissant la profondeur binaire de chaque composante (comme défini dans le § I.5.3.2);
- Les 7 bits de plus faible poids de la valeur indiquent la profondeur binaire des composantes. Le bit de plus fort poids indique si les composantes sont signées ou non signées. Si le bit de plus fort poids est 1, alors les composantes contiennent des valeurs signées. Si le bit de plus fort poids est 0, alors les composantes contiennent des valeurs non signées;
- C:** type de compression. Ce paramètre spécifie l'algorithme de compression utilisé pour comprimer les données d'image. La valeur de ce champ doit être 7. Elle est codée sous forme d'entier non signé de 1 octet. Les autres valeurs sont réservées pour utilisation par l'ISO;
- UnkC:** espace chromatique inconnu. Ce champ spécifie si l'espace chromatique réel des données d'image contenues dans le flux codé est connu. Ce champ est codé sous forme d'entier non signé de 1 octet. Les valeurs légales pour ce champ sont 0, si l'espace chromatique de l'image est connu et correctement spécifié dans les boîtes de spécification d'espace chromatique contenues dans le fichier, ou 1, si l'espace chromatique de l'image n'est pas connu. Une valeur de 1 sera utilisée dans des cas comme le transcodage d'images existantes où l'espace chromatique réel des données d'image n'est pas connu. Dans ces cas, alors que les méthodes d'interprétation de l'espace chromatique spécifiées dans le fichier ne peuvent pas reproduire précisément l'image par rapport à l'original, l'image doit être traitée comme si les méthodes reproduisaient précisément l'image. Les valeurs autres que 0 et 1 sont réservées pour utilisation par l'ISO;
- IPR:** propriété intellectuelle. Ce paramètre indique si ce fichier JP2 contient des informations sur les droits de propriété intellectuelle. Si la valeur de ce champ est 0, ce fichier ne contient pas d'informations sur les droits ni donc de boîte IPR. Si la valeur est 1, alors le fichier contient des informations sur les droits et une boîte IPR comme défini dans § I.6. Les autres valeurs sont réservées pour utilisation par l'ISO.

Tableau I.5 – Format du contenu de la boîte d'en-tête d'image

Nom du champ	Longueur (bits)	Valeur
HAUTEUR	32	1 à $(2^{32}-1)$
LARGEUR	32	1 à $(2^{32}-1)$
NC	16	1 à 16 384
BPC	8	Voir Tableau I.6
C	8	7
Unk	8	0 à 1
IPR	8	0 à 1

Tableau I.6 – Valeurs de bits par composante

Valeurs (bits)		Précision d'échantillon de composante
MSB	LSB	
x000	0000 à x010 0101	Profondeur binaire des composantes = valeur + 1. D'une profondeur de 1 bit jusqu'à une profondeur de 38 bits respectivement (en comptant le bit de signe, si approprié)
0xxx	xxxxx	Les composantes sont des valeurs non signées
1xxx	xxxxx	Les composantes sont des valeurs signées
1111	1111	Les composantes varient en profondeur binaire
		Toutes les autres valeurs sont réservées pour utilisation par l'ISO.

I.5.3.2 Boîte de bits par composante

La boîte de bits par composante spécifie la profondeur binaire de chaque composante. Si la profondeur binaire de toutes les composantes contenues dans le flux codé est la même (en signe comme en précision), alors cette boîte ne doit pas être trouvée. Sinon, cette boîte spécifie la profondeur binaire de chaque composante individuelle. L'ordre des valeurs de profondeur binaire dans cette boîte est l'ordre réel dans lequel ces composantes sont énumérées dans le flux codé. L'emplacement exact de cette boîte dans la boîte d'en-tête de fichier JP2 peut varier à condition qu'elle suive la boîte d'en-tête d'image.

Il doit y avoir une et une seule boîte de bits par composante dans une boîte d'en-tête de fichier JP2.

Le type de la boîte de bits par composante doit être 'bpcc' (0x6270 6363). Le contenu de cette boîte doit être comme décrit dans la Figure I.9:

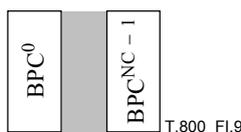


Figure I.9 – Organisation du contenu d'une boîte de bits par composante

BPCⁱ: bits par composante. Ce paramètre spécifie la profondeur binaire de composante i , moins 1, codée sous forme de valeur de 1 octet. L'ordonnement des composantes dans la boîte de bits par composante doit être le même que l'ordonnement des composantes dans le flux codé. Le nombre de champs BPCⁱ doit être le même que la valeur du champ NC d'après la boîte d'en-tête d'image. La valeur de ce champ doit être équivalente au champ Ssizⁱ équivalent dans le marqueur SIZ contenu dans le flux codé.

Les 7 bits de plus faible poids de la valeur indiquent la profondeur binaire de cette composante. Le bit de plus fort poids indique si la composante est signée ou non signée. Si le bit de plus fort poids est 1, alors la composante contient des valeurs signées. Si le bit de plus fort poids est 0, alors la composante contient des valeurs non signées.

Tableau I.7 – Format du contenu de la boîte de bits par composante

Nom du champ	Longueur (bits)	Valeur
BPC ⁱ	8	Voir Tableau I.8

Tableau I.8 – Valeurs BPCⁱ

Valeurs (bits)		Précision d'échantillon de composante
MSB	LSB	
x000	0000 à x010 0101	Profondeur binaire des composantes = valeur + 1. D'une profondeur de 1 bit jusqu'à une profondeur de 38 bits respectivement (en comptant le bit de signe, si approprié)
0xxx	xxxx	Les composantes sont des valeurs non signées
1xxx	xxxx	Les composantes sont des valeurs signées
		Toutes les autres valeurs sont réservées pour utilisation par l'ISO

I.5.3.3 Boîte de spécification de couleur

Chaque boîte de spécification de couleur définit une méthode par laquelle une application peut interpréter l'espace chromatique des données d'image décompressées. La présente Recommandation | Norme internationale de couleur doit être appliquée aux données d'image après qu'elles ont été décompressées et après application aux données d'image d'une éventuelle transformation inverse de composante réductrice de la redondance.

Un fichier JP2 peut contenir de multiples boîtes de spécification de couleur, mais doit toujours en contenir au moins une, spécifiant différentes méthodes pour obtenir des résultats "équivalents". Un lecteur JP2 conforme doit ignorer toutes les boîtes de spécification de couleur venant après la première. Cependant, les lecteurs conformes à d'autres normes peuvent utiliser ces boîtes comme défini dans ces autres normes.

Le type d'une boîte de spécification de couleur doit être 'colr' (0x636F 6C72). Le contenu d'une boîte de spécification de couleur est comme décrit dans la Figure I.10:

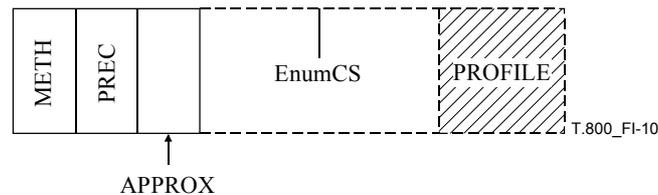


Figure I.10 – Organisation du contenu d'une boîte de spécification de couleur

METH: méthode de spécification. Ce champ spécifie la méthode utilisée par cette boîte de spécification de couleur afin de définir l'espace chromatique de l'image décompressée. Ce champ est codé sous forme d'entier non signé de 1 octet. La valeur de ce champ doit être 1 ou 2, comme défini dans le Tableau I.9.

Tableau I.9 – Valeurs légales du champ METH

Valeur	Signification
1	<p>Espace chromatique énuméré. Cette boîte de spécification d'espace chromatique contient la valeur énumérée de l'espace chromatique de cette image. La valeur énumérée se trouve dans le champ EnumCS de cette boîte. Si la valeur du champ METH est 1, alors le champ EnumCS doit exister dans cette boîte immédiatement après le champ APPROX, et le champ EnumCS doit être le dernier dans cette boîte.</p>
2	<p>Profil ICC restreint. Cette boîte de spécification de couleur contient un profil ICC dans le champ de profil. Ce profil doit spécifier la transformation nécessaire afin de convertir les données d'image décompressées en données PCS_{XYZ}, et doit être conforme soit à la classe de profil d'entrée monochrome soit à la classe de profil d'entrée à trois composantes en matrice. Elle doit contenir toutes les étiquettes qui y sont spécifiées, comme défini dans ICC.1:1998-09. En tant que telle, la valeur du champ d'espace de connexion de profil dans l'en-tête de profil contenu dans le profil imbriqué doit être 'XYZ\040' (0x5859 5A20) indiquant que la sortie d'espace chromatique du profil est dans l'espace chromatique XYZ.</p> <p>D'éventuelles étiquettes privées contenues dans le profil ICC ne doivent pas modifier l'apparence visuelle d'une image traitée au moyen de ce profil ICC.</p> <p>Les composantes extraites du flux codé peuvent avoir une étendue supérieure à l'étendue d'entrée de la courbe de reproduction sonore (TRC, <i>tone reproduction curve</i>) du profil ICC. Toutes les valeurs décodées doivent être écrêtées aux limites de la courbe TRC avant traitement de l'image par le profil ICC. Par exemple, les valeurs négatives d'échantillon de composante signé peuvent être écrêtées jusqu'à zéro avant traitement des données d'image par le profil.</p> <p>Voir le § J.9 pour une description plus détaillée des transformées légales d'espace chromatique, de la façon dont ces transformées sont mémorisées dans le fichier, et de la façon de traiter une image au moyen de ces transformées sans utiliser de moteur de gestion de couleur ICC.</p> <p>Si la valeur du champ METH est 2, alors le champ de profil doit immédiatement suivre le champ APPROX et le champ PROFILE doit être le dernier dans la boîte.</p>
Autres valeurs	<p>Réservées pour autre utilisation par l'ISO. Si la valeur du champ METH n'est pas 1 ou 2, il peut y avoir des champs dans cette boîte après le champ APPROX, et un lecteur JP2 conforme doit ignorer l'ensemble de la boîte de spécification de couleur.</p>

- PREC:** présence. Ce champ est réservé pour utilisation par l'ISO et sa valeur doit être mise à zéro; cependant, les lecteurs conformes doivent ignorer la valeur de ce champ, qui est spécifié sous forme d'entier signé d'un octet.
- APPROX:** approximation de l'espace chromatique. Ce champ spécifie l'étendue dans laquelle cette méthode de spécification de couleur approche la définition "correcte" de l'espace chromatique. La valeur de ce champ doit être mise à zéro; cependant, les lecteurs conformes doivent ignorer la valeur de ce champ. Les autres valeurs sont réservées pour autre utilisation par l'ISO. Ce champ est spécifié sous forme d'entier non signé de 1 octet.
- EnumCS:** espace chromatique énuméré. Ce champ spécifie l'espace chromatique de l'image au moyen de codes d'entier. Afin d'interpréter correctement la couleur d'une image au moyen d'un espace chromatique énuméré, l'application doit toujours connaître en interne la définition de cet espace chromatique. Ce champ contient une valeur d'entier non signé de 4 octets gros-boutiste indiquant l'espace chromatique de l'image. Si la valeur du champ METH est 2, alors le champ EnumCS ne doit pas exister. Les valeurs valides du champ EnumCS pour la première boîte de spécification d'espace chromatique dans les fichiers conformes sont limitées à 16, 17, et 18 comme défini dans le Tableau I.10:

Tableau I.10 – Valeurs légales EnumCS

Valeur	Signification
16	sRGB comme défini par CEI 61966-2-1
17	<p>échelle de gris: espace d'échelle de gris où la luminance de l'image est associée à des valeurs codées au moyen de la relation de non-linéarité sRGB donnée dans les équations (2) à (4) de la Publication 61966-2-1 de la CEI (sRGB):</p> $Y' = Y_{8\text{ bit}} / 255 \quad (\text{I-1})$ <p style="text-align: center;">pour $(Y' \leq 0,04045)$, $Y_{lin} = Y' / 12,92$ (I-2)</p> $\text{pour } (Y' > 0,04045), Y_{lin} = \left(\frac{Y' + 0,055}{1,055} \right)^{2,4}$ <p>où Y_{lin} est la valeur linéaire de luminance d'image dans l'étendue de 0,0 à 1,0. Les valeurs de luminance de l'image doivent être interprétées par rapport aux conditions de référence contenues dans le § 2 de la Publication 61966-2-1 de la CEI.</p>
18	<p>sYCC comme défini par la Publication 61966-2-1 de la CEI Amd. 1</p> <p>NOTE – Il n'est pas recommandé d'utiliser la transformation ICT ou RCT spécifiée dans l'Annexe G avec des données d'image sYCC. Voir § J.15 des directives sur la gestion des flux codés YCC.</p>
Autres valeurs	Réservé pour autres utilisations par l'ISO

PROFILE: profil ICC. Ce champ contient un profil ICC valide, comme indiqué par la spécification de format de profil ICC, qui spécifie la transformation des données d'image décompressées en PCS. Ce champ ne doit pas exister si la valeur du champ METH est 1. Si la valeur du champ METH est 2, alors le profil ICC doit être conforme à la classe de profil d'entrée monochrome ou à la classe de profil d'entrée à trois composantes en matrice comme défini en ICC.1:1998-09.

Tableau I.11 – Format du contenu de la boîte de spécification de couleur

Nom du champ	Longueur (bits)	Valeur
METH	8	1 à 2
PREC	8	0
APPROX	8	0
EnumCS	32 si METH = 1 0 si METH = 2	0 à $(2^{32} - 1)$ Pas de valeur
PROFILE	Variable	Variable; voir la spécification de format de profil ICC, version ICC.1:1998-09.

I.5.3.4 Boîte de palette

Cette boîte spécifie une palette qui peut servir à créer des canaux à partir de composantes. Cependant, la boîte de palette ne spécifie pas la création d'un quelconque canal particulier; la création de canaux sur la base de l'application de la palette à une composante est spécifiée par la boîte de mappage des composantes. L'espace chromatique ou la signification du canal créé est spécifié par la boîte de définition de canal (ou par les valeurs par défaut définies dans la spécification de la boîte de définition de canal si celle-ci n'existe pas). Si la boîte d'en-tête de fichier JP2 contient une boîte de palette, alors elle doit également contenir une boîte de mappage des composantes. Si la boîte d'en-tête de fichier JP2 ne contient pas de boîte de palette, alors elle ne doit pas contenir de boîte de mappage des composantes.

Il doit y avoir au plus une boîte de palette dans une boîte d'en-tête de fichier JP2.

Le type de la boîte de palette doit être 'pclr' (0x7063 6C72). Le contenu de cette boîte doit être comme décrit dans la Figure I.11:

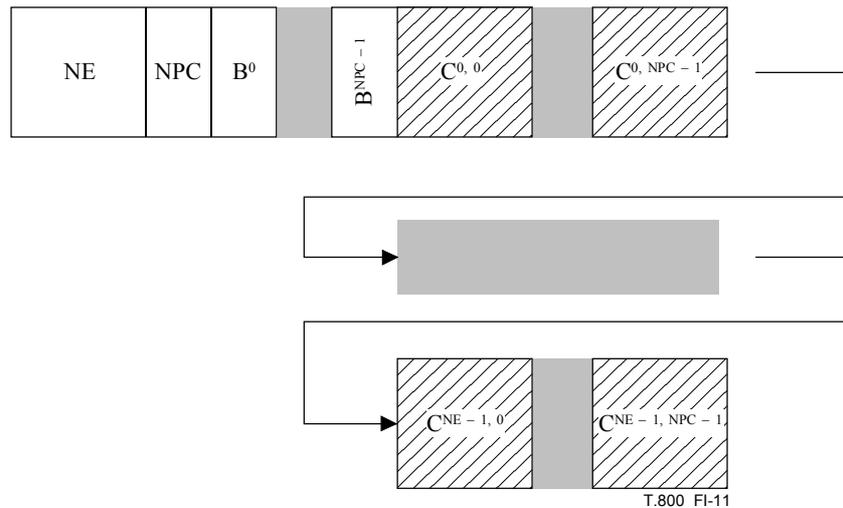


Figure I.11 – Organisation du contenu de la boîte de palette

- NE:** nombre d'entrées dans la table. Cette valeur doit être dans l'étendue de 1 à 1 024 et est codée sous forme d'entier non signé de 2 octets gros-boutiste;
- NPC:** nombre de colonnes de palette spécifiées dans la boîte de palette. Par exemple, si la palette doit servir à affecter une composante à index unique à une image à trois composantes RGB, alors la valeur de ce champ doit être 3. Ce champ est codé sous forme d'entier non signé de 1 octet;
- Bⁱ:** ce paramètre spécifie la profondeur binaire de valeurs créées par la colonne de palette *i*, codées sous forme d'entier gros-boutiste de 1 octet. Les 7 bits de plus faible poids de la valeur indiquent la profondeur binaire de cette colonne de palette. Le bit de plus fort poids indique si la colonne de palette est signée ou non signée. Si le bit de plus fort poids est 1, alors la colonne de palette contient des valeurs signées. Si le bit de plus fort poids est 0, alors la colonne de palette contient des valeurs non signées. Le nombre de valeurs Bⁱ doit être le même que la valeur du champ NPC;
- C^{ji}:** valeur d'entrée *j* dans la colonne de palette *i*. Les valeurs C^{ji} sont organisées dans l'ordre de composante majeure; toutes les valeurs d'entrée *j* sont groupées ensemble, suivies par toutes les valeurs d'entrée *j* + 1. Dans l'exemple ci-dessus, ce tableau serait donc lu comme suit: R₁,G₁,B₁,R₂,G₂,B₂, etc. La longueur de C^{ji} est la valeur spécifiée par le champ Bⁱ. Le nombre de colonnes de palette doit être indiqué dans le champ NPC. Le nombre de valeurs C^{ji} doit être celui des colonnes de palette (le champ NPC) multiplié par le nombre d'entrées dans la palette (NE). Si la valeur de Bⁱ n'est pas un multiple de 8, alors chaque valeur C^{ji} est bourrée de zéros jusqu'à un multiple de 8 bits et la valeur réelle doit être mémorisée dans les bits de plus faible poids de la valeur justifiée. Par exemple, si la valeur de Bⁱ est de 10 bits, alors les valeurs individuelles C^{ji} doivent être mémorisées dans les 10 bits de plus faible poids d'un champ de 16 bits.

Tableau I.12 – Format du contenu de la boîte de palette

Nom du champ	Longueur (bits)	Valeur
NE	16	1 à 1 024
NPC	8	1 à 255
B ⁱ	8	Voir Tableau I.13
C ^{ji}	Variable	Variable

Tableau I.13 – Valeurs Bⁱ

Valeurs (bits)		Précision d'échantillonnage de colonne de palette
MSB	LSB	
x000	0000 à x010 0101	Profondeur binaire de colonne de palette = valeur + 1. D'une profondeur de 1 bit jusqu'à une profondeur de 38 bits respectivement (en comptant le bit de signe, si approprié)
0xxx	xxxxx	Les valeurs de colonne de palette sont des valeurs non signées
1xxx	xxxxx	Les valeurs de colonne de palette sont des valeurs signées
		Toutes les autres valeurs sont réservées pour utilisation par l'ISO

I.5.3.5 Boîte de mappage des composantes

La boîte de mappage des composantes définit comment les canaux d'image sont indiqués d'après les composantes réelles décodées du flux codé. Cette abstraction permet à une seule structure (la boîte de définition de canal) de spécifier la couleur ou le type des deux images: à composantes de palette et sans composantes de palette. Cette boîte contient une matrice de champs CMPⁱ, MTYPⁱ et PCOLⁱ. Chaque groupe de ces champs représente la définition d'un canal dans l'image. Les canaux sont numérotés dans l'ordre à partir de zéro, et le nombre de canaux spécifiés dans la boîte de mappage des composantes est déterminé par la longueur de la boîte.

Il doit y avoir au plus une boîte de mappage des composantes dans une boîte d'en-tête de fichier JP2.

Si la boîte d'en-tête de fichier JP2 contient une boîte de palette, alors la boîte d'en-tête de fichier JP2 doit également contenir une boîte de mappage des composantes. Si la boîte d'en-tête de fichier JP2 ne contient pas de boîte de mappage des composantes, les composantes doivent être affectées directement aux canaux, de façon que la composante *i* soit affectée au canal *i*.

Le type de la boîte de mappage des composantes doit être 'cmap' (0x636D 6170). Le contenu de cette boîte doit être comme décrit dans la Figure I.12:

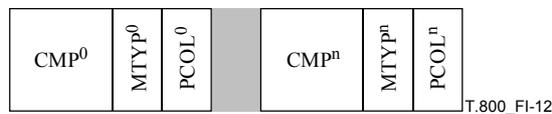


Figure I.12 – Organisation du contenu d'une boîte de mappage des composantes

- CMPⁱ**: ce champ spécifie l'index de composante extraite du flux codé qui est affecté à ce canal (soit directement ou par une palette). Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste.
- MTYPⁱ**: ce champ spécifie comment ce canal est produit d'après les composantes réelles dans le fichier. Ce champ est codé sous forme d'entier non signé de 1 octet. Les valeurs légales du champ MTYPⁱ sont décrites dans le Tableau I.14.

Tableau I.14 – Valeurs de champ MTYPⁱ

Valeur	Signification
0	Usage direct. Ce canal est créé directement d'après une composante réelle contenue dans le flux codé. L'index de la composante mappée à ce canal est spécifié dans le champ CMP ⁱ pour ce canal.
1	Affectation de palette. Ce canal est créé par application de la palette à une composante réelle contenue dans le flux codé. L'index de la composante mappée dans la palette est spécifié dans le champ CMP ⁱ pour ce canal. La colonne de la palette à utiliser est spécifiée dans le champ PCOL ⁱ pour ce canal.
2 à 255	Réservé pour utilisation par l'ISO

- PCOLⁱ**: ce champ spécifie la composante d'index d'après la palette qui sert à affecter la composante réelle extraite du flux codé. Ce champ est codé sous forme d'entier non signé de 1 octet. Si la valeur du champ MTYPⁱ pour ce canal est 0, alors la valeur de ce champ doit être 0.

Tableau I.15 – Format du contenu de la boîte d'affectation des composantes

Nom du champ	Longueur (bits)	Valeur
CMP ⁱ	16	0 à 16 384
MTYP ⁱ	8	0 à 1
PCOL ⁱ	8	0 à 255

I.5.3.6 Boîte de définition de canal

La boîte de définition de canal spécifie la signification des échantillons de chaque canal dans l'image. L'emplacement exact de cette boîte dans la boîte d'en-tête de fichier JP2 peut varier à condition qu'elle suive la boîte d'en-tête d'image. L'affectation de composantes réelles extraites du flux codé à des canaux est spécifiée dans la boîte d'affectation des composantes. Si la boîte d'en-tête de fichier JP2 ne contient pas de boîte d'affectation des composantes, alors un lecteur doit affecter la composante i au canal i , pour toutes les composantes contenues dans le flux codé.

Il doit y avoir au plus une boîte de définition de canal dans une boîte d'en-tête de fichier JP2.

Cette boîte contient une matrice de descriptions de canal. Pour chaque description, trois valeurs sont spécifiées: l'index du canal décrit par cette association, le type de ce canal, et l'association de ce canal avec des couleurs particulières. Cette boîte peut spécifier de multiples descriptions pour un seul canal.

Si une transformée multiple de composante est spécifiée dans le flux codé, l'image doit toujours être dans un espace chromatique RGB et les couleurs rouge, vert et bleu doivent former les canaux 0, 1 et 2 dans le flux codé, respectivement.

Le type de la boîte de définition de canal doit être 'cdef' (0x6364 6566). Le contenu de cette boîte doit être comme décrit dans la Figure I.13:

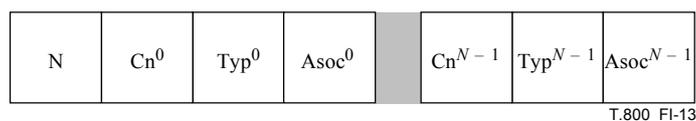


Figure I.13 – Organisation du contenu d'une boîte de définition de canal

- N:** nombre de descriptions de canal. Ce champ spécifie le nombre de descriptions de canal dans cette boîte. Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste.
- Cnⁱ:** index de canal. Ce champ spécifie l'index du canal pour cette description. La valeur de ce champ représente l'index du canal comme défini dans la boîte de mappage des composantes (ou de la composante réelle extraite du flux codé si le fichier ne contient pas de boîte d'affectation des composantes). Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste.
- Typⁱ:** type de canal. Ce champ spécifie le type du canal pour cette description. La valeur de ce champ spécifie la signification des échantillons décomprimés dans ce canal. Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste. Les valeurs légales de ce champ sont représentées dans le Tableau I.16:

Tableau I.16 – Valeur de champ Typⁱ

Valeur	Signification
0	Ce canal est la donnée d'image en couleur pour la couleur associée
1	Opacité. Une valeur d'échantillon de 0 indique que l'échantillon est 100% transparent, et la valeur maximale du canal (associée à la profondeur binaire de la composante de flux codé ou à la composante de palette associée qui est affectée à ce canal) indique un échantillon 100% opaque. Tous les canaux d'opacité doivent être affectés d'après des composantes non signées.
2	Opacité prémultipliée. Canal d'opacité comme spécifié ci-dessus, sauf que la valeur du canal d'opacité a été multipliée dans les canaux de couleur auxquels ce canal est associé. La prémultiplication est définie comme suit: $S_p = S \times \frac{\alpha}{\alpha_{max}} \quad (I-3)$ où S est l'échantillon original, S_p est l'échantillon prémultiplié (l'échantillon mémorisé dans l'image, α est la valeur du canal d'opacité, et α_{max} est la valeur maximale du canal d'opacité comme défini par la profondeur binaire du canal d'opacité.
3 à ($2^{16} - 2$)	Réservé pour utilisation par l'ISO
$2^{16} - 1$	Le type de ce canal n'est pas spécifié

Asocⁱ: association de canal. Ce champ spécifie l'index de la couleur à laquelle ce canal est directement associé (ou une valeur spéciale afin d'indiquer l'image entière ou l'absence d'association). Par exemple, si ce canal est un canal d'opacité pour le canal de rouge dans un espace chromatique RGB, ce champ spécifiera l'index de la couleur rouge. Le Tableau I.17 spécifie les valeurs légales d'association. Le Tableau I.18 spécifie les indices chromatiques légaux. Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste.

Tableau I.17 – Valeurs de champ Asocⁱ

Valeur	Signification
0	Ce canal est associé en tant qu'image dans son ensemble (par exemple un canal d'opacité indépendant qui doit être appliqué à tous les canaux chromatiques).
1 à ($2^{16} - 2$)	Ce canal est associé à une couleur particulière comme indiqué par cette valeur. Cette valeur sert à associer un canal particulier à un aspect particulier de la spécification de l'espace chromatique de cette image. Par exemple, l'indication qu'un canal est associé au canal de rouge d'une image RGB permet au lecteur d'associer ce canal décodé à l'entrée de rouge dans un profil ICC contenu dans une boîte de spécification de couleur. Les indicateurs de couleur sont spécifiés dans le Tableau I.18.
$2^{16} - 1$	Ce canal n'est pas associé à une couleur particulière

Tableau I.18 – Couleurs indiquées par le champ Asocⁱ

Classe d'espace chromatique	Couleur indiquée par la valeur suivante du champ Asoc ⁱ			
	1	2	3	4
RGB	R	G	B	
Echelle de gris	Y			
YCbCr	Y	C _b	C _r	

Les classes suivantes d'espace chromatique sont énumérées pour future référence, ainsi que pour faciliter la compréhension de l'utilisation du champ Asocⁱ.

XYZ	X	Y	Z	
Lab	L	a	b	
Luv	L	u	v	
YCbCr	Y	C _b	C _r	
Yxy	Y	x	y	
HSV	H	S	V	
HLS	H	L	S	
CMYK	C	M	Y	K
CMY	C	M	Y	
Jab	J	a	b	
<i>n</i> espaces chromatiques	1	2	3	4

Les valeurs du Tableau I.18 spécifient des indices qui ont été attribués pour représenter des "couleurs" spécifiques. Elles ne se rapportent pas à des canaux spécifiques (ni à des composantes dans le flux codé ou dans une palette). Les lecteurs doivent toujours utiliser les informations contenues dans la boîte de définition de canal afin de déterminer les canaux qui contiennent les couleurs.

Dans cette boîte, des indices de canal sont affectés d'après des composantes particulières dans le flux codé ou dans la palette. Les indices chromatiques spécifient comment un canal particulier doit être interprété sur la base de la spécification de l'espace chromatique de l'image. Il doit y avoir une définition de canal dans cette boîte pour chaque couleur requise par la spécification de l'espace chromatique de ce fichier, comme spécifié par la boîte de spécification de l'espace chromatique.

Par exemple, la couleur verte dans une image RGB est spécifiée par une valeur {Cn, Typ, Asoc} de {*i*, 0, 2}, où *i* est l'index de ce canal (soit directement ou produit par application de la transformée multiple de composante inverse aux composantes réelles contenues dans le flux codé). Les applications qui ne visent qu'à extraire les canaux de couleur peuvent traiter la paire de champs Typ/Asoc comme une valeur de quatre octets où la valeur combinée affecte directement les indices chromatiques (car le champ Typ d'un canal de couleur doit être 0).

Dans un autre exemple, le flux codé peut contenir un canal *i* qui spécifie des échantillons de fondu d'opacité pour les canaux de rouge et de vert et un canal *j* qui spécifie des échantillons de fondu d'opacité pour le canal de bleu. Dans ce fichier, les nuplets suivants {Cn, Typ, Asoc} seront trouvés dans la boîte de définition de canal pour les deux canaux d'opacité: {*i*, 1, 1}, {*i*, 1, 2} et {*j*, 1, 3}.

Il ne doit pas y avoir plus d'un canal dans un fichier JP2 avec la même paire de valeurs Typⁱ et Asocⁱ, à l'exception des valeurs Typⁱ et Asocⁱ de 2¹⁶ - 1 (non spécifiées). Par exemple un fichier JP2 dans un espace chromatique RGB doit seulement contenir un canal de vert, et une image en échelle de gris doit contenir un seul canal de gris. Il doit y avoir soit exactement un seul canal d'opacité, exactement un seul canal d'opacité prémultipliée, ou aucun canal associé à un seul canal de couleur dans une image.

Si le flux codé contient seulement des canaux de couleur et que ceux-ci soient ordonnés dans le même ordre que les couleurs associées (par exemple, une image RGB avec trois canaux dans l'ordre R, G, puis B), alors cette boîte ne doit pas exister. S'il y a des canaux auxiliaires ou si les canaux ne sont pas dans le même ordre que les indices chromatiques, alors la boîte de définition de canal (voir Tableau I.19) doit se trouver dans la boîte d'en-tête de fichier JP2 avec la liste complète de définitions de canal.

Tableau I.19 – Format de la boîte de définition de canal

Paramètre	Longueur (bits)	Valeur
N	16	1 à $(2^{16}-1)$
Cn ⁱ	16	0 à $(2^{16}-1)$
Typ ⁱ	16	0 à $(2^{16}-1)$
Asoc ⁱ	16	0 à $(2^{16}-1)$

I.5.3.7 Boîte de résolution

Cette boîte spécifie les résolutions de grille de capture et d'affichage par défaut de cette image. Si cette boîte existe, elle doit contenir soit une boîte de résolution de capture, ou une boîte de résolution d'affichage par défaut, ou les deux.

Il doit y avoir au plus une boîte de résolution dans une boîte d'en-tête de fichier JP2.

Le type d'une boîte de résolution doit être 'res\040' (0x7265 7320). Le contenu de la boîte de résolution est le comme décrit dans la Figure I.14:

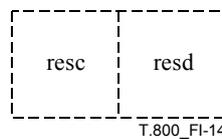


Figure I.14 – Organisation du contenu de la boîte de résolution

- resc:** boîte de résolution de capture. Cette boîte spécifie la résolution de grille à laquelle cette image a été capturée. Le format de cette boîte est spécifié dans le § I.5.3.7.1;
- resd:** boîte de résolution d'affichage par défaut. Cette boîte spécifie la résolution de grille par défaut à laquelle cette image doit être affichée. Le format de cette boîte est spécifié dans le § I.5.3.7.2.

I.5.3.7.1 Boîte de résolution de capture

Cette boîte spécifie la résolution de grille à laquelle la source a été numérisée afin de créer les échantillons d'image spécifiés par le flux codé. Par exemple, cette boîte peut spécifier la résolution du scanneur à plat qui a capturé une page de livre. La résolution de grille de capture pourrait également spécifier la résolution d'une caméra numérique d'extérieur ou de satellite.

Les résolutions verticale et horizontale de grille de capture sont calculées au moyen des six paramètres (Tableau I.20) mémorisés dans cette boîte au moyen des deux équations suivantes, respectivement:

$$VRc = \frac{VRcN}{VRcD} \times 10^{VRcE} \quad (I-4)$$

$$HRc = \frac{HRcN}{HRcD} \times 10^{HRcE} \quad (I-5)$$

Les valeurs *VRc* et *HRc* sont toujours exprimées en points de grille de référence par mètre. Si une application nécessite la résolution de grille dans une autre unité, alors cette application doit toujours appliquer la conversion appropriée.

Le type d'une boîte de résolution de capture doit être 'resc' (0x7265 7363). Le contenu de la boîte de résolution de capture est comme décrit dans la Figure I.15:

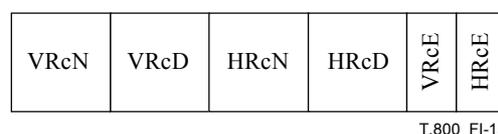


Figure I.15 – Organisation du contenu de la boîte de résolution de capture

- VRcN:** numérateur de résolution verticale de grille de capture. Ce paramètre spécifie la valeur *VRcN* dans l'équation (I-4), qui sert à calculer la résolution verticale de grille de capture. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- VRcD:** dénominateur de résolution verticale de grille de capture. Ce paramètre spécifie la valeur *VRcD* dans l'équation (I-4), qui sert à calculer la résolution verticale de grille de capture. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- HRcN:** numérateur de résolution horizontale de grille de capture. Ce paramètre spécifie la valeur *HRcN* dans l'équation (I-5), qui sert à calculer la résolution horizontale de grille de capture. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- HRcD:** dénominateur de résolution horizontale de grille de capture. Ce paramètre spécifie la valeur *HRcD* dans l'équation (I-5), qui sert à calculer la résolution horizontale de grille de capture. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- VRcE:** exposant de résolution verticale de grille de capture. Ce paramètre spécifie la valeur *VRcE* dans l'équation (I-4), qui sert à calculer la résolution verticale de grille de capture. Ce paramètre est codé sous forme d'entier signé de 1 octet en complément à deux;
- HRcE:** exposant de résolution horizontale de grille de capture. Ce paramètre spécifie la valeur *HRcE* dans l'équation (I-5), qui sert à calculer la résolution horizontale de grille de capture. Ce paramètre est codé sous forme d'entier signé de 1 octet en complément à deux.

Tableau I.20 – Format du contenu de la boîte de résolution de capture

Nom du champ	Longueur (bits)	Valeur
VRcN	16	1 à (2 ¹⁶ - 1)
VRcD	16	1 à (2 ¹⁶ - 1)
HRcN	16	1 à (2 ¹⁶ - 1)
HRcD	16	1 à (2 ¹⁶ - 1)
VRcE	8	-128 à 127
HRcE	8	-128 à 127

I.5.3.7.2 Boîte de résolution d'affichage par défaut

Cette boîte spécifie une résolution de grille d'affichage recherchée. Par exemple, cette boîte peut être utilisée afin de déterminer la longueur de l'image sur une page lorsque l'image est placée dans un programme de mise en page. Cependant, cette valeur est seulement par défaut. Chaque application doit toujours déterminer une taille d'affichage appropriée pour cette application.

Les résolutions verticale et horizontale de grille d'affichage sont calculées au moyen des six paramètres (Tableau I.21) mémorisés dans cette boîte par les deux équations suivantes, respectivement:

$$VRd = \frac{VRdN}{VRdD} \times 10^{VRdE} \tag{I-6}$$

$$HRd = \frac{HRdN}{HRdD} \times 10^{HRdE} \tag{I-7}$$

Les valeurs *VRd* et *HRd* sont toujours exprimées en points de grille de référence par mètre. Si une application nécessite la résolution de grille dans une autre unité, alors cette application doit toujours appliquer la conversion appropriée.

Le type d'une boîte de résolution d'affichage par défaut doit être 'resd' (0x7265 7364). Le contenu de la boîte de résolution d'affichage par défaut est comme décrit dans la Figure I.16:

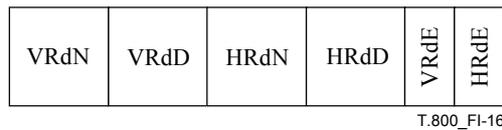


Figure I.16 – Organisation du contenu de la boîte de résolution d'affichage par défaut

- VRdN:** numérateur de résolution verticale de grille d'affichage. Ce paramètre spécifie la valeur *VRdN* dans l'équation (I-6), qui sert à calculer la résolution verticale de grille d'affichage. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- VRdD:** dénominateur de résolution verticale de grille d'affichage. Ce paramètre spécifie la valeur *VRdD* valeur dans l'équation (I-6), qui sert à calculer la résolution verticale de grille d'affichage. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- HRdN:** numérateur de résolution horizontale de grille d'affichage. Ce paramètre spécifie la valeur *HRdN* dans l'équation (I-7), qui sert à calculer la résolution horizontale de grille d'affichage. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- HRdD:** dénominateur de résolution horizontale de grille d'affichage. Ce paramètre spécifie la valeur *HRdD* dans l'équation (I-7), qui sert à calculer la résolution horizontale de grille d'affichage. Ce paramètre est codé sous forme d'entier non signé de 2 octets gros-boutiste;
- VRdE:** exposant de résolution verticale de grille d'affichage exposant. Ce paramètre spécifie la valeur *VRdE* dans l'équation (I-6), qui sert à calculer la résolution verticale de grille d'affichage. Ce paramètre est codé sous forme d'entier signé de 1 octet en complément à deux;
- HRdE:** exposant de résolution horizontale de grille d'affichage. Ce paramètre spécifie la valeur *HRdE* dans l'équation (I-7), qui sert à calculer la résolution horizontale de grille d'affichage. Ce paramètre est codé sous forme d'entier signé de 1 octet en complément à deux.

Tableau I.21 – Format du contenu de la boîte de résolution d'affichage par défaut

Nom du champ	Longueur (bits)	Valeur
VRdN	16	1 à $(2^{16} - 1)$
VRdD	16	1 à $(2^{16} - 1)$
HRdN	16	1 à $(2^{16} - 1)$
HRdD	16	1 à $(2^{16} - 1)$
VRdE	8	-128 à 127
HRdE	8	-128 à 127

I.5.4 Boîte de flux codé contigu

La boîte de flux codé contigu contient un flux codé JPEG 2000 valide et complet, comme défini dans l'Annexe A. Lors de l'affichage de l'image, un lecteur conforme doit ignorer tous les flux codés postérieurs au premier flux codé trouvé dans le fichier. La boîte de flux codé contigu peut être trouvée n'importe où dans le fichier sauf avant la boîte d'en-tête de fichier JP2.

Le type d'une boîte de flux codé contigu doit être 'jp2c' (0x6A70 3263). Le contenu de la boîte doit être comme décrit dans la Figure I.17:

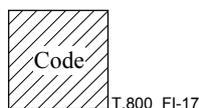


Figure I.17 – Organisation du contenu de la boîte de flux codé contigu

Code: ce champ contient un flux codé JPEG 2000 valide et complet, comme spécifié par l'Annexe A.

Tableau I.22 – Format du contenu de la boîte de flux codé contigu

Nom du champ	Longueur (bits)	Valeur
Code	Variable	Variable

I.6 Adjonction d'informations sur les droits de propriété intellectuelle dans JP2

La présente Recommandation | Norme internationale spécifie un type pour une boîte qui est spécialisée dans le transport des informations relatives aux droits de propriété intellectuelle dans un fichier JP2. L'inclusion de ces informations dans un fichier JP2 est facultative si celui-ci est conforme. La définition du format du contenu de cette boîte est réservée pour l'ISO. Cependant, le type de cette boîte est défini dans la présente Recommandation | Norme internationale comme devant permettre aux applications de reconnaître l'existence d'informations IPR. L'utilisation et l'interprétation de ces informations sont hors du domaine d'application de la présente Recommandation | Norme internationale.

En général, une boîte d'informations IPR trouvée au niveau supérieur du fichier spécifie les informations IPR pour le fichier dans son ensemble. Les boîtes IPR peuvent être trouvées à d'autres emplacements, y compris dans des superboîtes définies par d'autres Recommandations | Normes internationales. Pour ces boîtes IPR, les droits spécifiés se rapportent à l'entité définie par la superboîte contenante.

Le type de la boîte de propriété intellectuelle doit être 'jp2i' (0x6A70 3269).

I.7 Adjonction d'informations spécifiques du vendeur au format de fichier JP2

Les boîtes suivantes fournissent un ensemble d'outils permettant aux applications d'ajouter des informations spécifiques du vendeur au format de fichier JP2. Toutes les boîtes suivantes sont facultatives dans les fichiers conformes et peuvent être ignorées par les lecteurs conformes.

I.7.1 Boîtes de langage XML

Une boîte de langage XML contient des informations spécifiques du vendeur (en format XML) autres que les informations contenues dans les boîtes définies par la présente Recommandation | Norme internationale. Il peut y avoir de multiples boîtes de langage XML dans le fichier, et ces boîtes peuvent être trouvées n'importe où dans le fichier sauf avant la boîte de type de fichier.

Le type d'une boîte de langage XML est 'xml\040' (0x786D 6C20). Le contenu de la boîte doit être comme décrit dans la Figure I.18:



Figure I.18 – Organisation du contenu de la boîte de langage XML

DATA: ce champ doit contenir un document XML bien formé comme défini par REC-xml-19980210.

L'existence éventuelle d'une boîte de langage XML est facultative pour les fichiers conformes. De même, une boîte de langage XML ne doit pas contenir d'informations nécessaires pour décoder l'image dans la mesure qui est définie dans la présente Recommandation | Norme internationale. Et l'interprétation correcte du contenu d'une boîte de langage XML ne doit pas modifier l'apparence visuelle de l'image. Tous les lecteurs peuvent ignorer une boîte de langage XML dans le fichier.

I.7.2 Boîte d'identification UUID

Une boîte d'identification UUID contient des informations spécifiques du vendeur autres que les informations contenues dans les boîtes définies dans la présente Recommandation | Norme internationale. Il peut y avoir de multiples boîtes d'identification UUID dans le fichier, et ces boîtes peuvent être trouvées n'importe où dans le fichier sauf avant la boîte de type de fichier.

Le type d'une boîte d'identification UUID doit être 'uuid' (0x7575 6964). Le contenu de la boîte doit être comme décrit dans la Figure I.19:

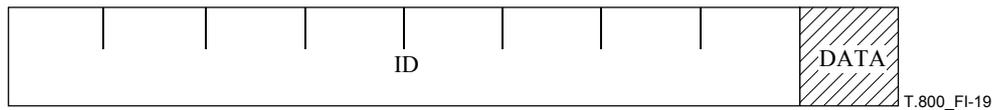


Figure I.19 – Organisation du contenu de la boîte d'identification UUID

- ID:** ce champ contient un identificateur UUID de 16 octets comme spécifié par l'ISO/CEI 11578. La valeur de cet identificateur UUID spécifie le format des informations spécifiques du vendeur mémorisées dans le champ DATA et l'interprétation de ces informations.
- DATA:** ce champ contient les informations spécifiques du vendeur. Le format de ces informations est défini hors du domaine d'application de la présente Recommandation | Norme internationale, mais est indiqué par la valeur du champ UUID.

Tableau I.23 – Format du contenu d'une boîte d'identification UUID

Nom du champ	Longueur (bits)	Valeur
UUID	128	Variable
DATA	Variable	Variable

L'existence de boîtes d'identification UUID est facultative pour les fichiers conformes. De même, une boîte d'identification UUID ne doit pas contenir d'informations nécessaires pour décoder l'image dans la mesure qui est définie dans la présente Recommandation | Norme internationale. L'interprétation des informations contenues dans une boîte d'identification UUID ne doit pas modifier l'apparence visuelle de l'image. Tous les lecteurs peuvent ignorer les boîtes d'identification UUID.

I.7.3 Boîtes d'informations d'identification UUID (superboîte)

Bien qu'il soit utile de permettre aux vendeurs d'étendre les fichiers JP2 par adjonction d'informations au moyen de boîtes d'identification UUID, il est également utile de fournir des informations sous une forme normalisée qui peut être utilisée par des applications non étendues afin d'obtenir de plus amples informations sur les extensions contenues dans le fichier. Ces informations sont contenues dans les boîtes d'informations d'identification UUID. Un fichier JP2 peut contenir zéro ou plusieurs boîtes d'informations d'identification UUID. Ces boîtes peuvent être trouvées n'importe où dans le niveau supérieur du fichier (la superboîte d'une boîte d'informations d'identification UUID doit être le fichier JP2 proprement dit) sauf avant la boîte de type de fichier.

Ces boîtes, si présentes, ne peuvent pas fournir un index complet pour les identificateurs UUID dans le fichier, peuvent faire référence à des identificateurs UUID non utilisés dans le fichier, et peuvent éventuellement fournir de multiples références pour le même UUID.

Le type d'une boîte d'informations d'identification UUID doit être 'uin' (0x7569 6E66). Le contenu d'une boîte d'informations d'identification UUID est comme décrit dans la Figure I.20:

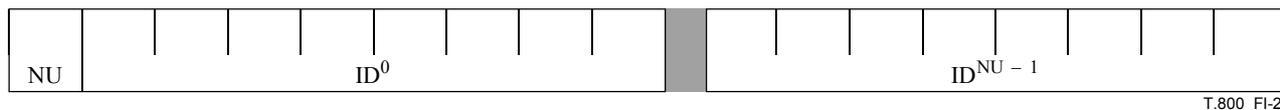


Figure I.20 – Organisation du contenu d'une boîte d'informations d'identification UUID

- UList:** boîte de liste d'identificateurs UUID. Cette boîte contient une liste d'identificateurs UUID pour lesquels cette boîte d'informations d'identification UUID spécifie un lien vers de plus amples informations. Le format de la boîte de liste d'identificateurs UUID est spécifié dans le § I.7.3.1.
- DE:** boîte d'URL d'entrée de données. Cette boîte contient une adresse URL. Une application peut obtenir de plus amples informations sur les identificateurs UUID contenus dans la boîte de liste d'identificateurs UUID. Le format d'une boîte d'URL d'entrée de données est spécifié dans le § I.7.3.2.

I.7.3.1 Boîte de liste d'identificateurs UUID

Cette boîte contient une liste des identificateurs UUID. Le type d'une boîte de liste d'identificateurs UUID doit être 'ulst' (0x756C 7374). Le contenu d'une boîte de liste d'identificateurs UUID doit être comme décrit dans la Figure I.21:



T.800_FI-21

Figure I.21 – Organisation du contenu d'une boîte de liste d'identificateurs UUID

- NU:** nombre d'identificateurs UUID. Ce champ spécifie le nombre d'identificateurs UUID trouvés dans cette boîte de liste d'identificateurs UUID. Ce champ est codé sous forme d'entier non signé de 2 octets gros-boutiste.
- IDⁱ:** ID. Ce champ spécifie un identificateur UUID, comme spécifié dans l'ISO/CEI 11578, qui doit être associé aux adresses URL contenues dans la boîte d'URL de la même boîte d'informations d'identification UUID. Le nombre de champs UUIDⁱ doit être égal à la valeur du champ NU. La valeur de ce champ doit être un identificateur de 16 octets.

Tableau I.24 – Valeurs de structure de données de contenu de boîte de liste d'identificateurs UUID

Paramètre	Longueur (bits)	Valeur
NU	16	0 à $(2^{16} - 1)$
UUID ⁱ	128	0 à $(2^{128} - 1)$

I.7.3.2 Boîte d'URL d'entrée de données

Cette boîte contient une adresse URL qui peut être utilisée par une application afin d'obtenir de plus amples informations sur les extensions spécifiques du vendeur associé. Le format des informations acquises par l'utilisation de cette adresse URL n'est pas défini dans la présente Recommandation | Norme internationale. Le type d'adresse URL doit être celui d'un service qui fournit un fichier (par ex. URL de type fichier, http, ftp, etc.), permettant également et théoriquement un accès aléatoire. Des adresses URL relatives sont admissibles. Elles se rapporteront au fichier contenant cette boîte d'URL d'entrée de données.

Le type d'une boîte d'URL d'entrée de données doit être 'url\040' (0x7572 6C20). Le contenu d'une boîte d'URL d'entrée de données doit être comme décrit dans la Figure I.22:



T.800_FI-22

Figure I.22 – Organisation du contenu d'une boîte d'URL d'entrée de données

- VERS:** numéro de version. Ce champ spécifie le numéro de version du format de cette boîte et est codé sous forme d'entier non signé de 1 octet. La valeur de ce champ doit être 0.
- FLAG:** fanions. Ce champ est réservé pour d'autres utilisations afin de signaler des attributs particuliers de cette boîte. Il est codé sous forme d'entier non signé de 3 octets. La valeur de ce champ doit être 0.
- LOC:** emplacement. Ce champ spécifie l'adresse URL des informations additionnelles qui sont associées aux identificateurs UUID contenus dans la boîte de liste d'identificateurs UUID de la même superboîte d'informations d'identification UUID. L'adresse URL est codée par une chaîne de caractères UTF-8 terminée par un caractère néant.

**Tableau I.25 – Valeurs de structure de données de contenu de boîte d'URL
d'entrée de données**

Paramètre	Longueur (bits)	Valeur
VERS	8	0
FLAG	24	0
LOC	Variable	Variable

I.8 Traitement des boîtes inconnues

Un fichier conforme JP2 peut contenir des boîtes inconnues des applications fondées seulement sur la présente Recommandation | Norme internationale. Si un lecteur conforme trouve une boîte qu'il ne comprend pas, il doit omettre et négliger cette boîte.

Annexe J

Exemples et directives

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe comporte un certain nombre d'exemples destinés à indiquer comment fonctionne le processus de codage, et comment le flux codé résultant doit être émis en sortie. La présente annexe est entièrement pour information.

J.1 Décodeur entropique adaptatif de conventions logicielles

La présente annexe fournit certains organigrammes en variante pour une version du décodeur entropique adaptatif. Cette version en variante peut être plus efficace lorsqu'elle est implémentée dans le logiciel, car elle possède un moins grand nombre d'opérations dans le raccourci.

La version en variante est obtenue par les substitutions suivantes. Remplacer l'organigramme de la Figure C.20 par l'organigramme de la Figure J.1. Remplacer l'organigramme de la Figure C.15 par l'organigramme de la Figure J.2. Remplacer l'organigramme de la Figure C.19 par l'organigramme de la Figure J.3.

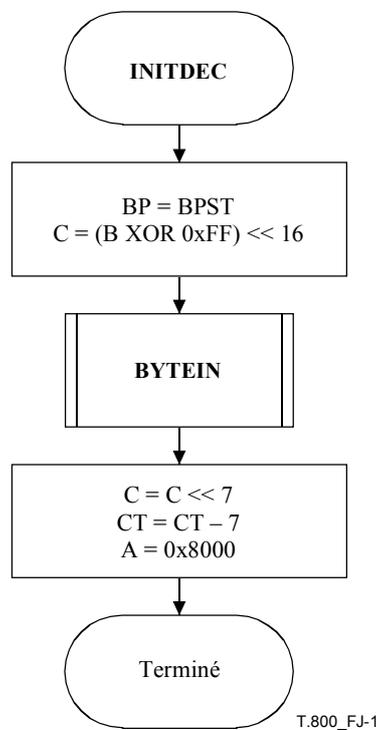


Figure J.1 – Initialisation du décodeur de conventions logicielles

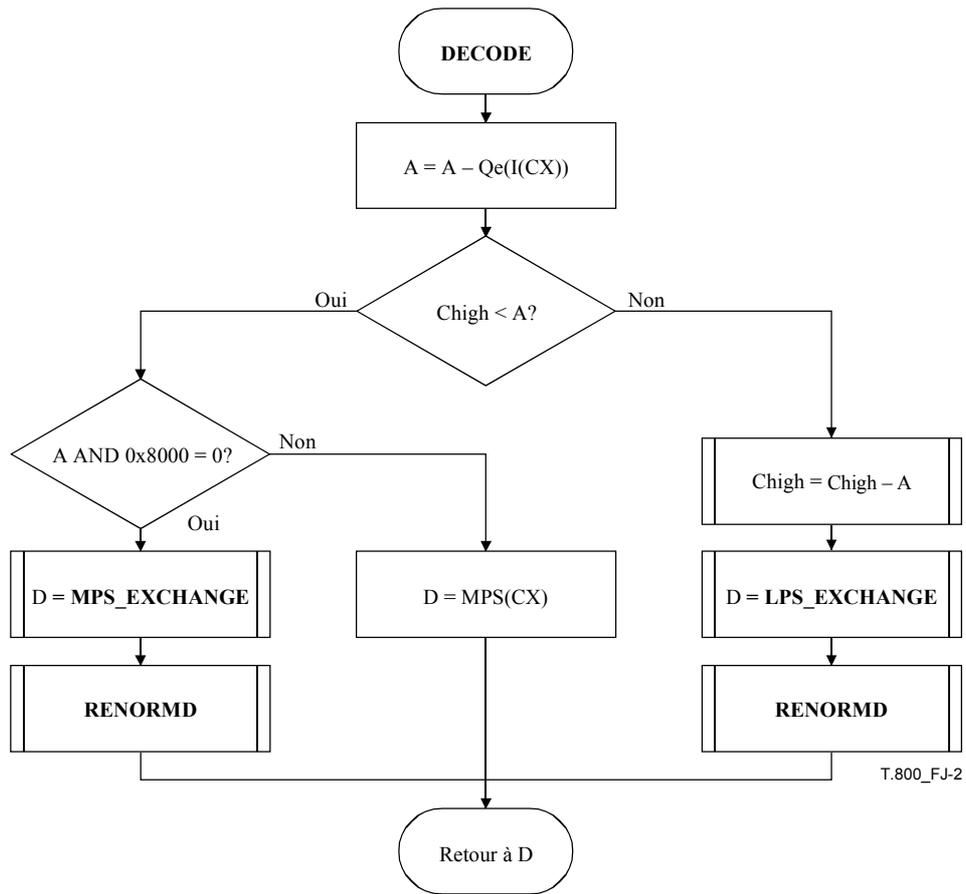


Figure J.2 – Décodage d'un symbole MPS ou d'un symbole LPS dans le décodeur de conventions logicielles

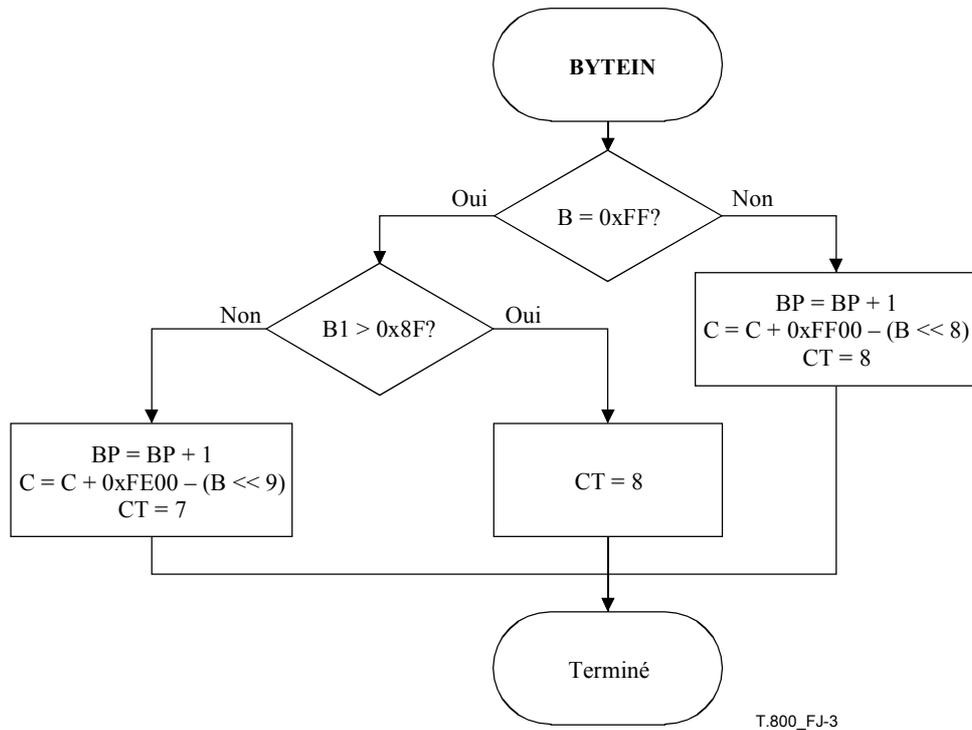


Figure J.3 – Insertion d'un nouvel octet dans le registre C du décodeur de conventions logicielles

J.2 Sélection de grandeurs d'échelon de quantification pour transformations irréversibles

Pour la compression irréversible, aucune sélection particulière de la taille d'échelon de quantification n'est requise dans la présente Recommandation | Norme internationale. Différentes applications peuvent spécifier les grandeurs d'échelon de quantification selon des caractéristiques spécifiques de composante-pavé. Une façon efficace de choisir l'échelon de quantification pour chaque sous-bande b consiste à normaliser un échelon par défaut Δ_d par prise en compte des procédures de filtrage horizontal et vertical qui produisent ces coefficients de sous-bande. Une méthode consiste à normaliser Δ_d avec un paramètre de pondération d'énergie γ_b (quantité d'erreurs quadratiques introduites par une erreur unitaire dans un coefficient transformé de sous-bande b) de la façon suivante [12]:

$$\Delta_b = \frac{\Delta_d}{\sqrt{\gamma_b}} \quad (\text{J-1})$$

J.3 Réponses impulsionnelles de filtre correspondant à des procédures de filtrage irréversibles en mode lissage

Les procédures de filtrage irréversible décrites dans l'Annexe F implémentent le banc de filtres convolutionnel de Cohen-Daubechies-Feauveau à 9 prises/7 prises [20], [21]. Les réponses impulsionnelles équivalentes des filtres d'analyse et de synthèse sont données dans les Tableaux J.1 et J.2.

Tableau J.1 – Définition des réponses impulsionnelles pour le banc 9-7 de filtres irréversibles d'analyse

n	Filtre passe-bas	Valeur approchée
0	$-5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	0,602 949 018 236 360
± 1	$-5x_1(8 x_2 ^2 - \Re x_2) / 8$	0,266 864 118 442 875
± 2	$-5x_1(4 x_2 ^2 - 4\Re x_2 - 1) / 16$	-0,078 223 266 528 990
± 3	$-5x_1(\Re x_2) / 8$	-0,016 864 118 442 875
± 4	$-5x_1 / 64$	0,026 748 757 410 810
n	Filtre passe-haut	Valeur approchée
-1	$(6x_1 - 1) / 8x_1$	1,115 087 052 457 000
-2, 0	$-(16x_1 - 1) / 32x_1$	-0,591 271 763 114 250
-3, 1	$(2x_1 + 1) / 16x_1$	-0,057 543 526 228 500
-4, 2	$-1 / 32x_1$	0,091 271 763 114 250

Tableau J.2 – Définition des réponses impulsionnelles pour le banc 9-7 de filtres irréversibles de synthèse

n	Filtre passe-bas	Valeur approchée
0	$(6x_1 - 1) / 8x_1$	1,115 087 052 457 000
± 1	$(16x_1 - 1) / 32x_1$	0,591 271 763 114 250
± 2	$(2x_1 + 1) / 16x_1$	-0,057 543 526 228 500
± 3	$1 / 32x_1$	-0,091 271 763 114 250
n	Filtre passe-haut	Valeur approchée
1	$-5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	0,602 949 018 236 360
0, 2	$5x_1(8 x_2 ^2 - \Re x_2) / 8$	-0,266 864 118 442 875
-1, 3	$-5x_1(4 x_2 ^2 - 4\Re x_2 - 1) / 16$	-0,078 223 266 528 990
-2, 4	$5x_1(\Re x_2) / 8$	0,016 864 118 442 875
-3, 5	$-5x_1 / 64$	0,026 748 757 410 811

J.4 Exemple de transformation en ondelettes discrètes

Le Tableau J.3 contient les échantillons exprimés en valeurs d'entier $I(x,y)$ d'une composante-pavé qui a une largeur de 13 échantillons et une hauteur de 17 échantillons.

Tableau J.3 – Echantillons de composante-pavé de source

$I(x, y)$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	2	2	3	4	5	6	7	8	9	10	11	12
3	3	3	3	4	5	5	6	7	8	9	10	11	12
4	4	4	4	5	5	6	7	8	8	9	10	11	12
5	5	5	5	5	6	7	7	8	9	10	11	12	13
6	6	6	6	6	7	7	8	9	10	10	11	12	13
7	7	7	7	7	8	8	9	9	10	11	12	13	13
8	8	8	8	8	8	9	10	10	11	12	12	13	14
9	9	9	9	9	9	10	10	11	12	12	13	14	15
10	10	10	10	10	10	11	11	12	12	13	14	14	15
11	11	11	11	11	11	12	12	13	13	14	14	15	16
12	12	12	12	12	12	13	13	13	14	15	15	16	16
13	13	13	13	13	13	13	14	14	15	15	16	17	17
14	14	14	14	14	14	14	15	15	16	16	17	17	18
15	15	15	15	15	15	15	16	16	17	17	18	18	19
16	16	16	16	16	16	16	17	17	17	18	18	19	20

J.4.1 Exemple de transformation irréversible en ondelettes 9-7

Les Tableaux J.4, J.5, J.6, J.7, J.8, J.9 et J.10 contiennent les coefficients des sous-bandes 2LL, 2HL, 2LH, 2HH, 1HL, 1LH, 1HH résultant de la décomposition à deux niveaux avec la transformation 9-7 irréversible des échantillons de composante-pavé de source donnée dans le Tableau J.3 (voir Figure F.18). Les valeurs de coefficient affichées dans ces tableaux ont été arrondies à l'entier le plus proche.

Tableau J.4 – Coefficients de sous-bande 2LL (transformation irréversible en ondelettes 9-7)

$a_{2LL}(u, v)$	0	1	2	3
0	1	4	8	11
1	4	5	8	11
2	8	9	11	13
3	12	12	14	16
4	15	15	17	18

Tableau J.5 – Coefficients de sous-bande 2HL (transformation irréversible en ondelettes 9-7)

$a_{2HL}(u, v)$	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

Tableau J.6 – Coefficient de sous-bande 2LH (transformation irréversible en ondelettes 9-7)

$a_{2LH}(u, v)$	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

Tableau J.7 – Coefficients de sous-bande 2HH (transformation irréversible en ondelettes 9-7)

$a_{2HH}(u, v)$	0	1	2
0	-1	0	0
1	0	1	0
2	0	0	0
3	0	0	0

Tableau J.8 – Coefficients de sous-bande 1HL (transformation irréversible en ondelettes 9-7)

$a_{1HL}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	-1	0	0	0	0
4	0	0	0	-1	0	0
5	0	0	1	1	0	-1
6	0	0	0	0	0	0
7	0	0	-1	-1	-1	0
8	0	0	0	0	0	0

Tableau J.9 – Coefficients de sous-bande 1LH (transformation irréversible en ondelettes 9-7)

$a_{1LH}(u, v)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	-1	0	0	0
2	0	0	0	0	0	1	1
3	0	0	0	0	-1	1	0
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

Tableau J.10 – Coefficients de sous-bande 1HH (transformation irréversible en ondelettes 9-7)

$a_{1HH}(u, v)$	0	1	2	3	4	5
0	-1	0	0	0	0	0
1	0	0	-1	0	0	0
2	0	-1	1	0	0	0
3	0	0	0	0	0	1
4	0	0	0	0	-1	0
5	0	0	0	0	0	0
6	0	0	-1	0	-1	1
7	0	0	0	0	-1	0

J.4.2 Exemple de transformation réversible en ondelettes 5-3

Les Tableaux J.11, J.12, J.13, J.14, J.15, J.16 et J.17 contiennent les coefficients des sous-bandes 2LL, 2HL, 2LH, 2HH, 1HL, 1LH, 1HH résultant de la décomposition à deux niveaux avec la transformation 5-3 irréversible des échantillons de composante-pavé de source donnée dans le Tableau J.3 (voir Figure F.18). Les valeurs de coefficient affichées dans ces tableaux ont été arrondies à l'entier le plus proche.

Tableau J.11 – Coefficients de sous-bande 2LL (transformation réversible en ondelettes 5-3)

$a_{2LL}(u, v)$	0	1	2	3
0	0	4	8	12
1	4	5	8	12
2	8	8	11	15
3	12	12	14	18
4	16	16	18	20

Tableau J.12 – Coefficients de sous-bande 2HL (transformation réversible en ondelettes 5-3)

$a_{2HL}(u, v)$	0	1	2
0	0	0	0
1	0	1	0
2	0	1	0
3	0	0	1
4	0	0	0

Tableau J.13 – Coefficients de sous-bande 2LH (transformation réversible en ondelettes 5-3)

$a_{2LH}(u, v)$	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	0	0	0	0
3	0	0	0	0

Tableau J.14 – Coefficients de sous-bande 2HH (transformation réversible en ondelettes 5-3)

$a_{2HH}(u, v)$	0	1	2
0	-1	0	0
1	0	-1	0
2	0	1	0
3	0	0	0

Tableau J.15 – Coefficients de sous-bande 1HL (transformation réversible en ondelettes 5-3)

$a_{1HL}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	1	0	1	0	0
3	0	0	0	0	-1	1
4	0	0	0	0	1	1
5	0	0	1	1	0	-1
6	0	0	1	0	1	1
7	0	0	0	0	0	0
8	0	0	0	0	0	0

Tableau J.16 – Coefficients de sous-bande 1LH (transformation réversible en ondelettes 5-3)

$a_{1LH}(u, v)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	0	0	0	1	1
3	0	0	1	0	0	1	1
4	0	0	0	0	1	0	2
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	1	1	0

Tableau J.17 – Coefficients de sous-bande 1HH (transformation réversible en ondelettes 5-3)

$a_{1HH}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	0	0	1
4	0	0	0	0	0	1
5	0	0	0	1	0	0
6	0	0	0	0	0	1
7	0	0	0	0	-1	0

J.5 Transformée en ondelettes par rangée

L'on décrit ici un exemple de transformation en ondelettes par rangée pour le filtre irréversible 9-7 bien adaptée aux dispositifs de compression qui reçoivent et transfèrent des données d'image en mode série. Les implémentations traditionnelles de transformation en ondelettes nécessitent que l'image entière soit mise en mémoire tampon, puis que le filtrage soit effectué en sens vertical et en sens horizontal. Alors que le filtrage dans le sens horizontal est très simple, le

filtrage dans le sens vertical est plus élaboré. Le filtrage d'une rangée nécessite la lecture de celle-ci; le filtrage d'une colonne nécessite la lecture de l'image entière. Cela explique les énormes exigences en largeur de bande de l'implémentation d'une transformation en ondelettes. La transformation en ondelettes par rangée résout la précédente limitation tout en fournissant exactement les mêmes coefficients transformés qu'une implémentation traditionnelle de transformation en ondelettes. Cependant, la transformation en ondelettes par rangée seule ne fournit pas un paradigme complet de codage par rangée. Un codeur par rangée complet doit également prendre en compte toutes les étapes suivantes de codage jusqu'à celles du codage entropique et de l'attribution des débits.

J.5.1 La procédure de transformation FDWT_ROW

La procédure de transformation FDWT_ROW pour le filtre irréversible 9-7 utilise un tampon $buf(i, j)$ de cinq rangées $0 \leq i \leq 4$ afin d'effectuer l'équivalent de la procédure 2D_SD décrite dans le § F.4.2, sauf pour la procédure 2D_DEINTERLEAVE. L'étendue des échantillons de composante-pavé d'entrée $I(x, y)$ est censée être définie par l'équation (F-1). Chaque rangée du tampon $buf(i, j)$ est de taille $tx_1 - tx_0 + 1$. La description générale de la transformation FDWT_ROW appliquée à une composante-pavé d'image est illustrée dans la Figure J.4 pour le premier niveau de décomposition. La transformation FDWT_ROW reçoit en entrée une rangée d'échantillons de composante-pavé d'image décalé en niveau et produit en sortie une rangée de coefficients de transformée. Il est supposé dans tout ce paragraphe que la composante-pavé d'image possède au moins cinq rangées.

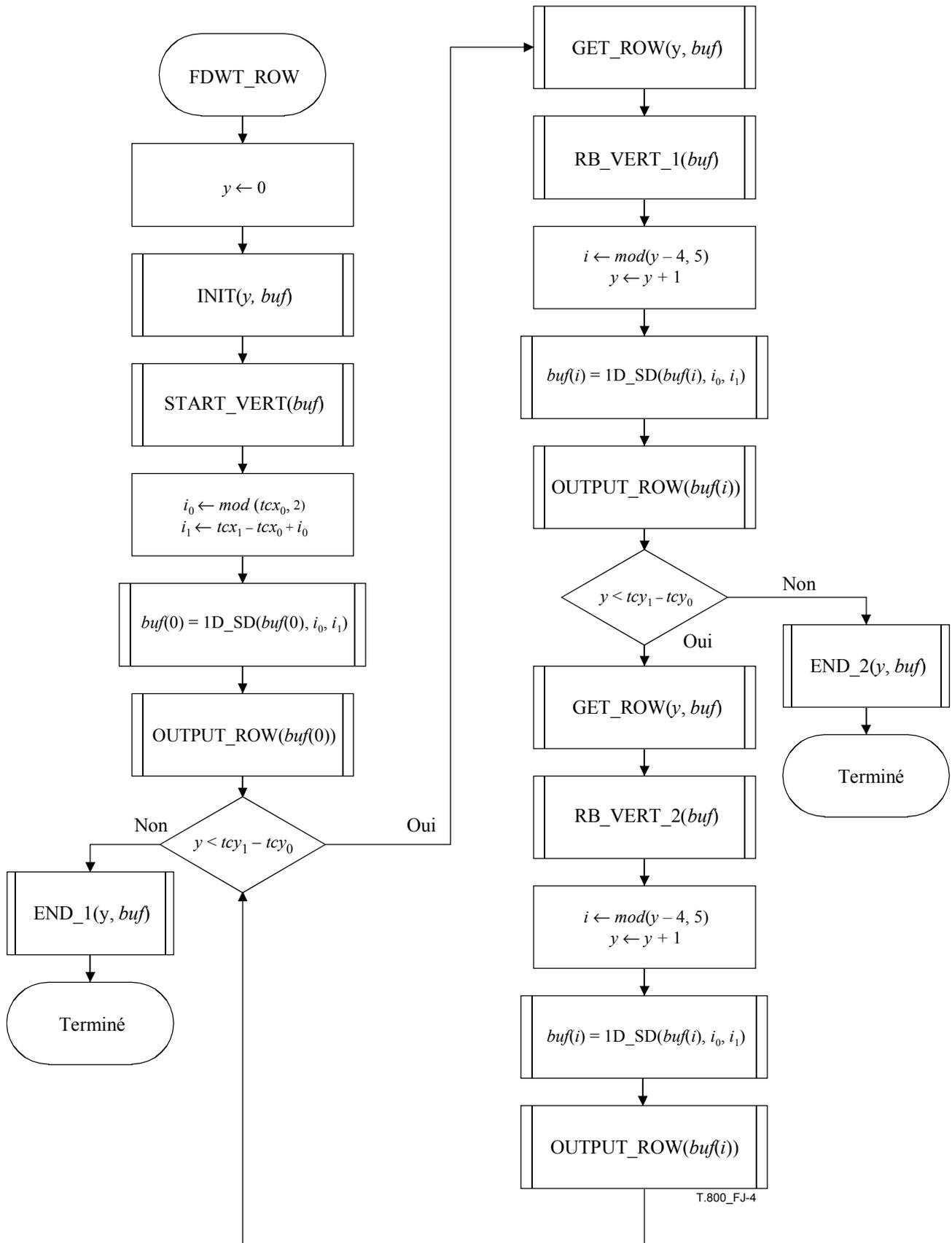


Figure J.4 – La procédure de transformation FDWT_ROW

J.5.1.1 La procédure GET_ROW

Dans cette description, la composante-pavé d'image décalée en niveau est censée être mémorisé dans une mémoire externe $I(x,y)$. Comme illustré dans la Figure J.5, la procédure GET_ROW lit une rangée d'échantillons de la composante-pavé d'image décalée en niveau et transfère cette rangée d'échantillons dans le tampon, buf .

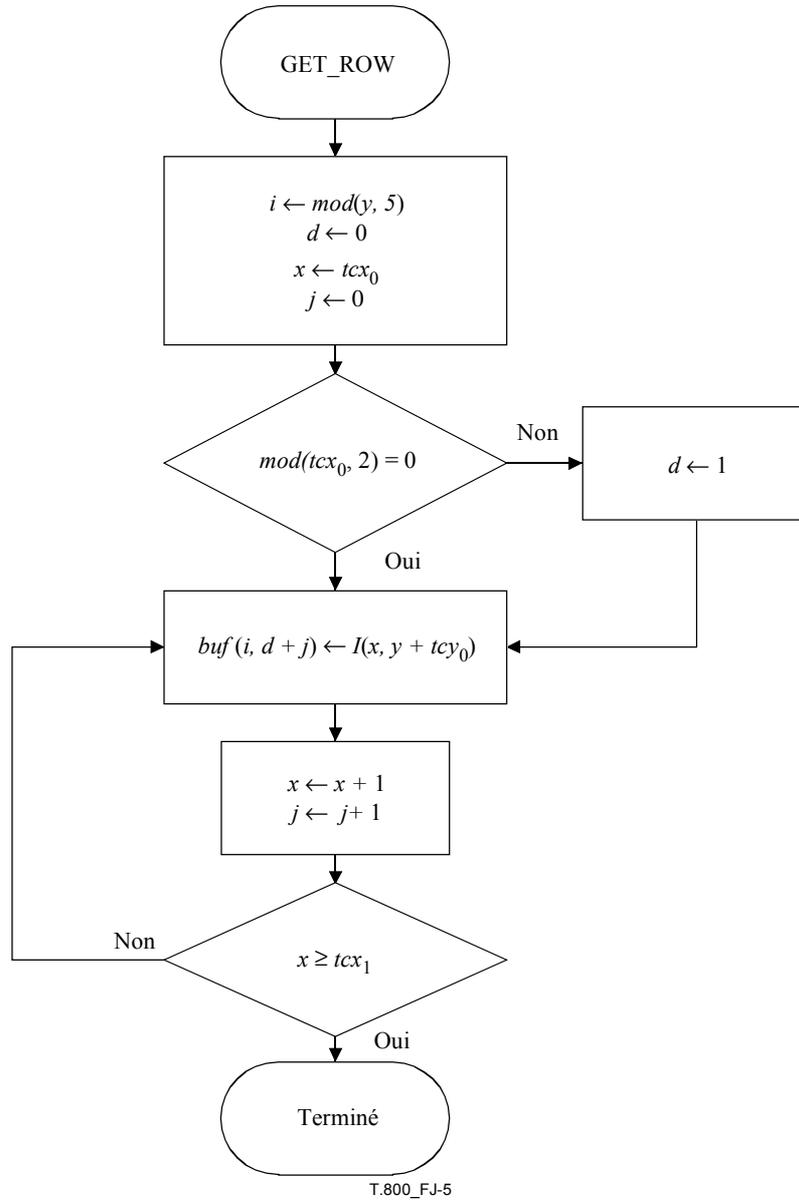


Figure J.5 – La procédure GET_ROW

J.5.2 La procédure INIT

Comme illustré dans la Figure J.6, la procédure INIT lit cinq rangées d'échantillons de la composante-pavé d'image décalée en niveau et transfère ces rangées d'échantillons dans le tampon, buf .

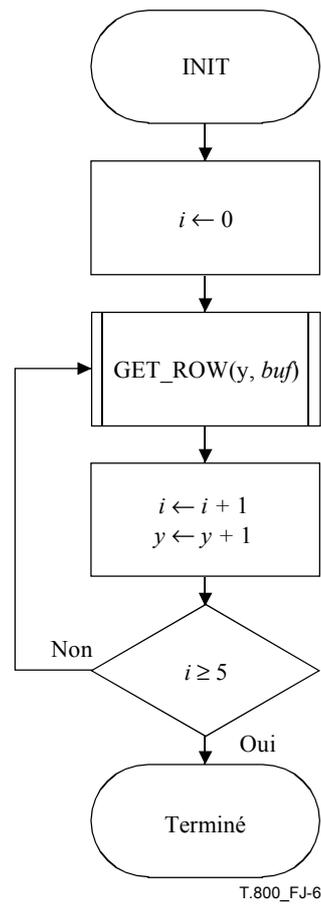


Figure J.6 – La procédure INIT

J.5.3 La procédure START_VERT

Comme illustré dans la Figure J.7, la procédure START_VERT modifie les coefficients dans le tampon $buf(i, j)$. Dans cette figure ainsi que dans toutes les figures suivantes de ce paragraphe, l'expression $buf(i) \leftarrow buf(i) + \alpha \cdot buf(i_2)$ est équivalente à $buf(i, j) \leftarrow buf(i, j) + \alpha \cdot buf(i_2, j)$ for $d \leq j < tcx_0 - tcx_1 + d$.

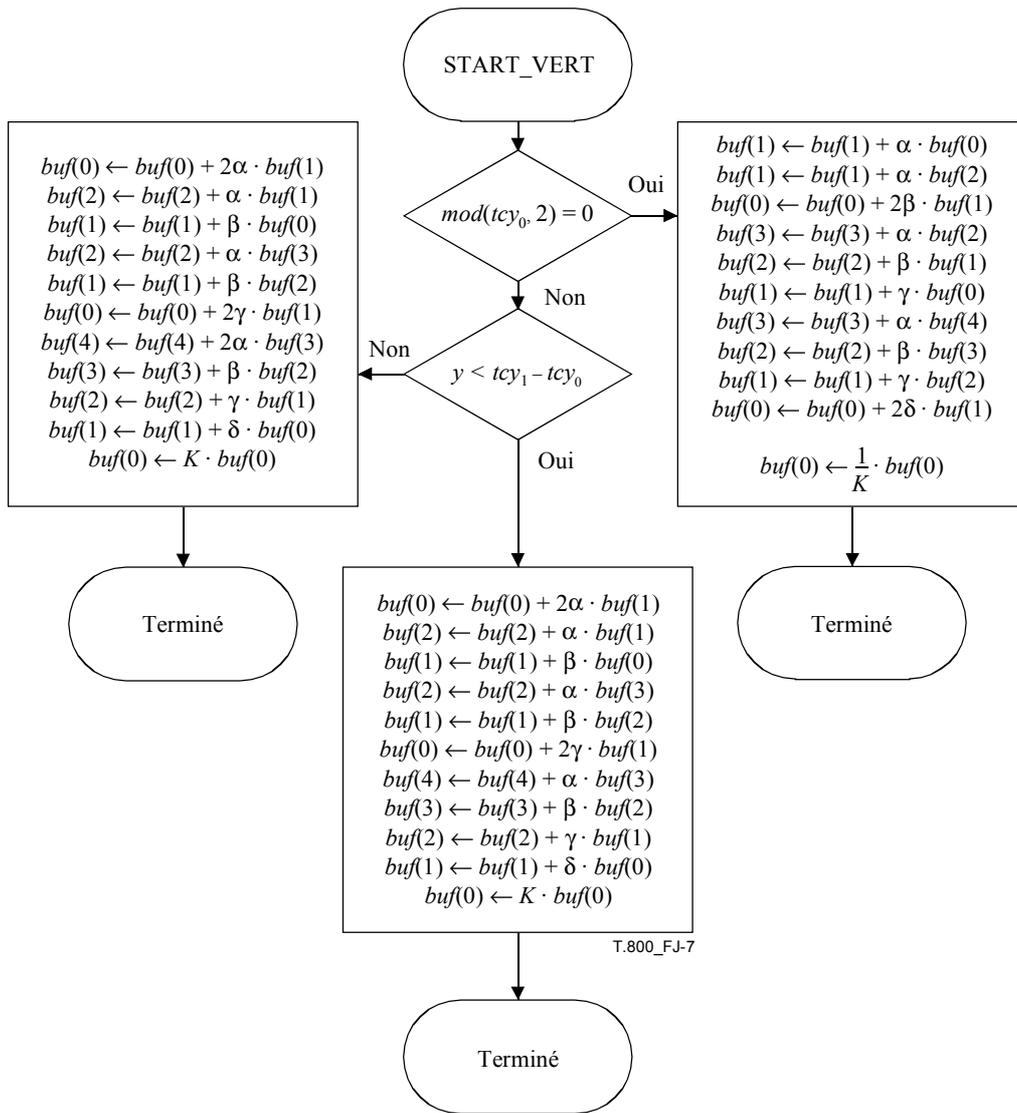


Figure J.7 – La procédure START_VERT

J.5.3.1 La procédure RB_VERT_1

Comme illustré dans la Figure J.8, la procédure RB_VERT_1 modifie les coefficients dans le tampon $buf(i, j)$.

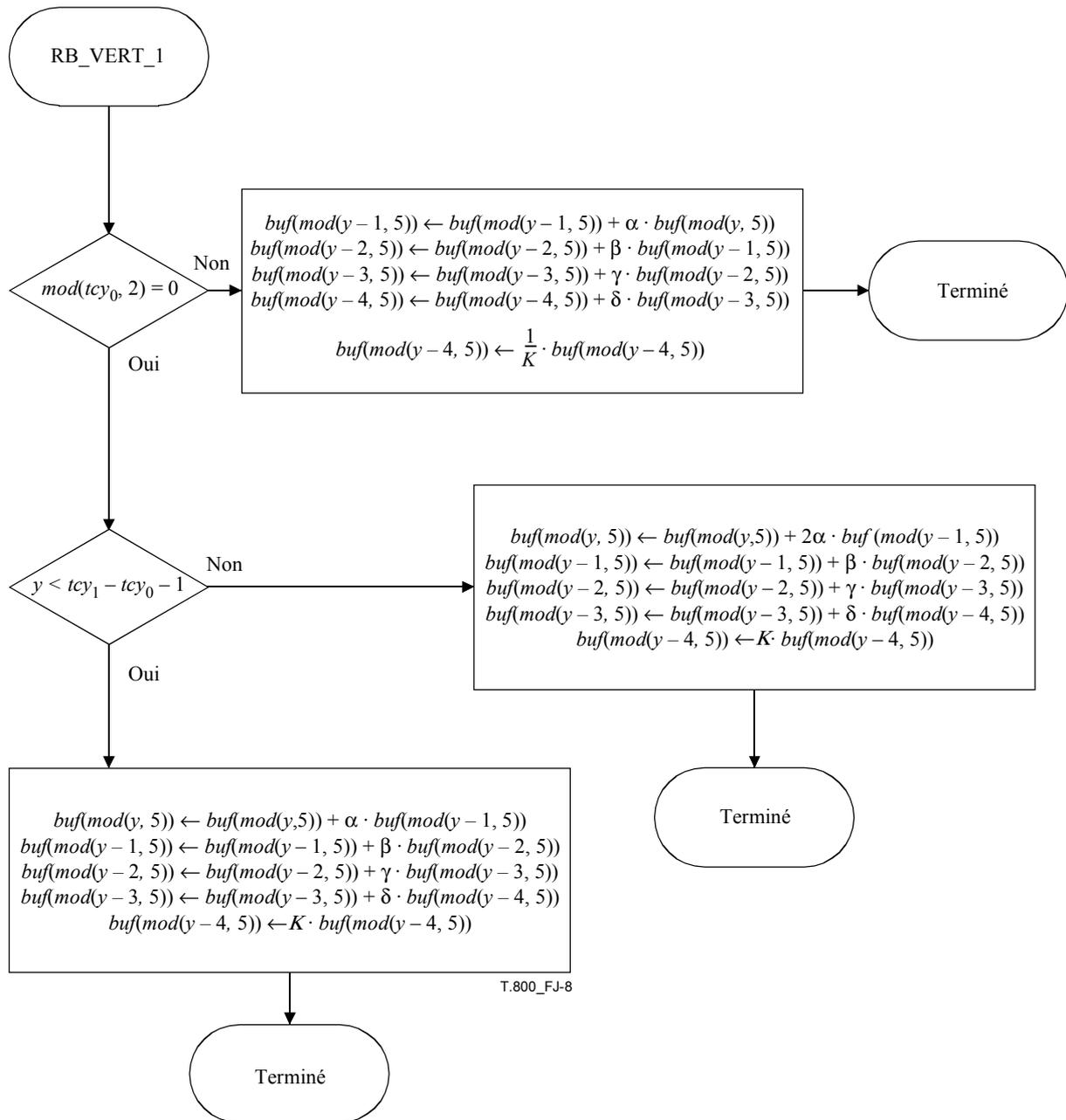


Figure J.8 – La procédure RB_VERT_1

J.5.3.2 La procédure RB_VERT_2

Comme illustré dans la Figure J.9, la procédure RB_VERT_2 modifie les coefficients dans le tampon $\text{buf}(i, j)$.

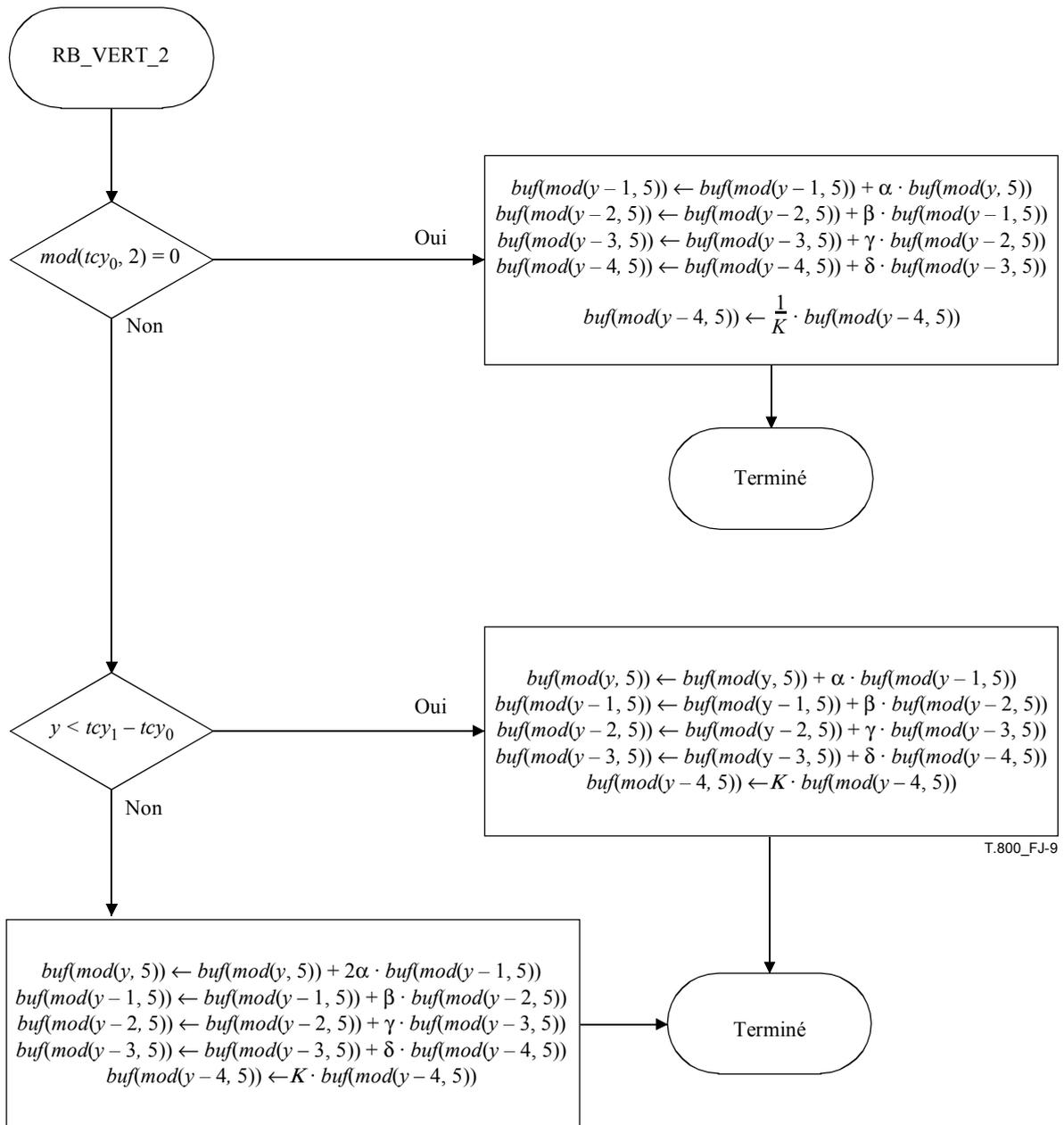
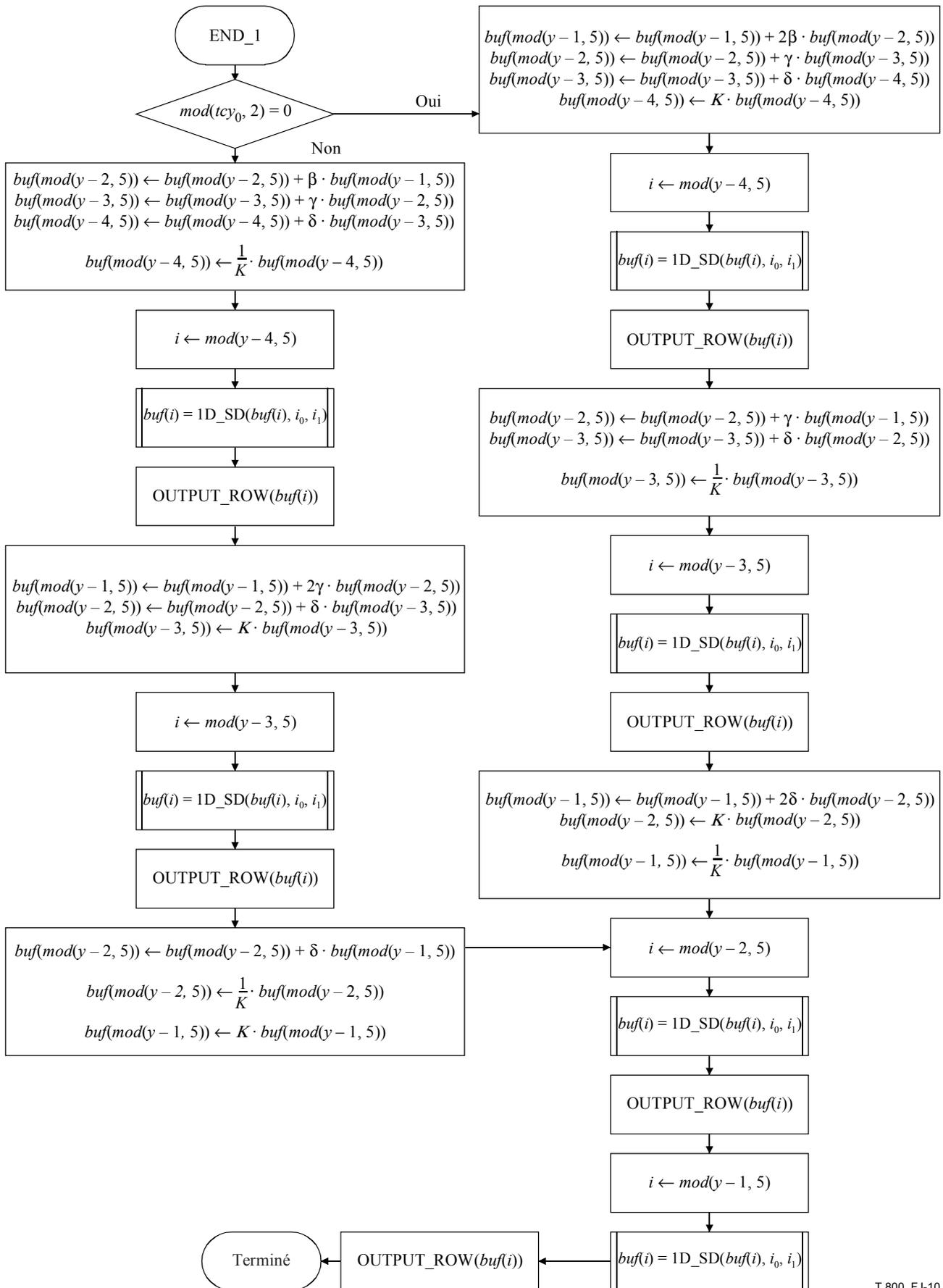


Figure J.9 – La procédure RB_VERT_2

J.5.3.3 La procédure END_1

La procédure END_1 est détaillée dans la Figure J.10.



T.800_FJ-10

Figure J.10 – La procédure END_1

J.5.3.4 La procédure END_2

La procédure END_2 est détaillée dans la Figure J.11.

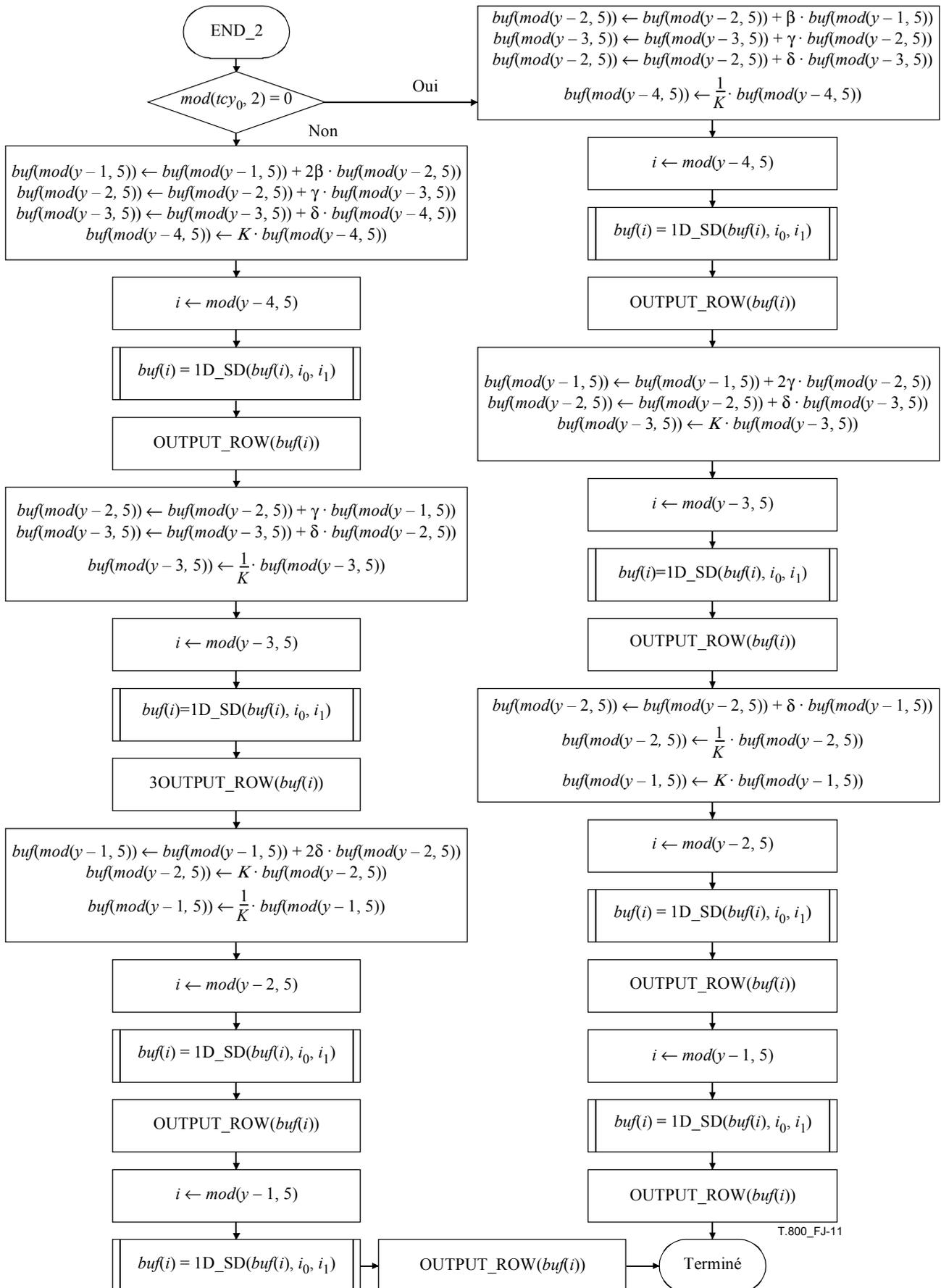


Figure J.11 – La procédure END_2

J.5.4 La procédure OUTPUT_ROW

Cette procédure renvoie une rangée $buf(i)$ de coefficients transformés, qui correspond soit aux sous-bandes 1LL et 1HL soit aux sous-bandes 1LH et 1HH. Cette rangée de coefficients de transformée peut être soit mémorisée dans une mémoire externe ou traitée immédiatement.

J.6 Codage par exploration

Certaines applications utilisent des capteurs de balayage qui créent des images (éventuellement non contraintes en longueur) rangée par rangée et qui ont des capacités limitées de mémoire de traitement. Ces applications nécessitent un codage par exploration complet, dans lequel seul le nombre minimal requis d'octets est conservé en mémoire à un instant donné sans perte significative de performance. Des implémentations de tels systèmes de codage par exploration ont été montrées à titre d'exemple [39], [40]. La procédure recommandée est décrite ci-dessous.

La commande de débit décrite dans le § J.14 nécessite la mise en mémoire tampon de tout le flux codé, comprimé à un débit supérieur au débit entrelacé final. En variante, une approche par exploration peut être utilisée, dans laquelle la transformation en ondelettes par rangée (voir § J.5) est suivie d'une procédure d'attribution de débit et de codage par exploration afin de faire en sorte que les coefficients d'ondelette comprimés soient transmis aussitôt après avoir été produits. A cette fin, une mémoire tampon limitée (le tampon de balayage) est introduite après la transformée en ondelettes. Les segments discrets de données comprimées qui y sont contenus sont appelés "éléments de balayage". Un élément de balayage se compose d'un ensemble localisé de coefficients d'ondelette. Il peut être un pavé ou un district et correspond à un petit nombre de rangées dans l'espace d'image. Le tampon de balayage peut contenir un ou plusieurs éléments de balayage.

L'algorithme de commande de débit est appliqué aux données comprimées dans le tampon de balayage et le premier élément de balayage est injecté dans le flux binaire. S'il y a plus d'un seul élément de balayage dans le tampon de balayage, un mécanisme de commande de débit à fenêtre glissante est implémenté. Cette méthode peut donner de meilleurs résultats de compression au prix d'une légère augmentation de la complexité et de la mémoire nécessaire.

Cette approche par exploration n'a pas d'incidence sur le processus de décodage JPEG 2000.

J.7 Elasticité aux erreurs

Le présent paragraphe décrit une méthode de décodage d'images qui ont été codées au moyen d'une syntaxe élastique aux erreurs.

De nombreuses applications nécessitent l'acheminement de données d'image par différents types de canaux de communication. Les canaux de communications sans fil traditionnels donnent naissance à des erreurs binaires aléatoires et sporadiques. Les communications IP sont exposées aux pertes en raison des encombrements du trafic. Afin d'améliorer la performance d'émission d'images comprimées sur ces canaux exposés aux erreurs, une syntaxe et des outils de flux binaire élastique aux erreurs sont inclus dans la présente Recommandation | Norme internationale.

Les outils d'élasticité aux erreurs (voir Tableau J.18) contenus dans la présente Recommandation | Norme internationale traitent les erreurs de canal au moyen des approches suivantes: partitionnement et resynchronisation des données comprimées, détection et masquage d'erreur, et transmission de qualité de service (QS) transmission sur la base de priorités. Les outils d'élasticité aux erreurs sont décrits dans chaque catégorie.

Tableau J.18 – Outils d'élasticité aux erreurs

Type d'outil	Nom	Référence
Niveau du codage entropique	Blocs codés Terminaison du codeur arithmétique pour chaque passe Réinitialisation de contextes après chaque passe de codage Evitement sélectif du codage arithmétique Symboles de segmentation	Annexe D
Niveau des paquets	Format court de paquet Paquet avec marqueur de resynchronisation (SOP)	Annexe B

Le codage entropique des coefficients quantifiés est effectué dans les blocs codés. Comme codage et décodage des blocs codés sont indépendants, les erreurs binaires contenues dans le flux binaire d'un bloc codé seront contenues dans ce bloc codé (voir Annexe D).

La terminaison du codeur arithmétique est permise après chaque passe de codage. De même, les contextes peuvent être réinitialisés après chaque passe de codage. Cela permet au codeur arithmétique de continuer à décoder des passes de codage après des erreurs (voir § D.4).

Le style facultatif d'évitement du codage arithmétique place des bits de base dans le flux binaire sans codage arithmétique. Cela prévient les types de propagation d'erreur auxquels le codage de longueur variable est exposé (voir § D.6).

Les paquets courts sont obtenus par déplacement des en-têtes de paquet vers les segments marqueurs PPM ou PPT (voir § A.7.4 et A.7.5). S'il y a des erreurs, les en-têtes de paquet contenus dans les segments marqueurs PPM ou PPT peuvent encore être associés au paquet correct au moyen du numéro séquentiel contenu dans le marqueur SOP.

Un symbole de segmentation est un symbole spécial. Le décodage correct de ce symbole confirme l'exactitude du décodage de ce plan binaire, ce qui permet la détection d'erreur. Voir § D.5.

Un paquet contenant un marqueur de resynchronisation SOP (voir § A.8.1) permet un partitionnement spatial et une resynchronisation. Ce marqueur est placé au début de chaque paquet dans un pavé avec un numéro séquentiel commençant à zéro. Il est incrémenté avec chaque paquet. L'ordonnement des paquets est décrit au § B.10.

J.8 Exigence de compatibilité avec fichiers JFIF/SPIFF

Le présent paragraphe ne s'applique qu'aux extensions de spécification de la Rec. UIT-T T.81 | ISO/CEI 10918-1. Il n'a aucune incidence sur la présente Recommandation | Norme internationale et n'en aura pas non plus dans ses extensions. Il n'y a aucune exigence de prise en charge de ce profil dans un décodeur JPEG 2000.

J.8.1 Méthode de compatibilité

Afin d'éviter tout type de modification dans le format de fichier décrit dans la partie normative de la présente Recommandation | Norme internationale, un nouveau profil est défini dans la Rec. UIT-T T.81 | ISO/CEI 10918-1 selon les méthodes d'extension décrites dans la Rec. UIT-T T.86 | ISO/CEI 10918-4. Ce profil facilitera la transition entre l'utilisation de la Rec. UIT-T T.81 | ISO/CEI 10918-1 et la présente Recommandation | Norme internationale.

Ce nouveau profil est disponible par l'intermédiaire de l'organisme d'enregistrement établi aux fins des extensions et est spécifié dans les Recommandations UIT-T T.84 | ISO/CEI 10918-3 et UIT-T T.86 | ISO/CEI 10918-4. Il apparaîtra donc sur le site IP de JURA (*JPEG Utilities Registration Authority*), à l'adresse <http://jura.jpeg.org>. Le statut de ce nouveau profil est le même que tous les autres paramètres enregistrés; il est une extension de la Rec. UIT-T T.81 | ISO/CEI 10918-1.

J.8.2 Paramètres d'étude de compatibilité

- 1) Transparent à un décodeur JFIF (conforme à la Rec. UIT-T T.81 | ISO/CEI 10918-1 et à la Rec. UIT-T T.83 | ISO/CEI 10918-2).
- 2) Transparent à un décodeur SPIFF (conforme à la Rec. UIT-T T.84 | ISO/CEI 10918-3).
- 3) Structure fondamentale de boîte JP2/JPX.
- 4) Transcodage de DCT en flux codé d'ondelettes flux inutile.
- 5) Préserve l'accès à d'éventuelles informations sur les droits IPR et leur intégrité.
- 6) Utilise l'extension ".jpg" ou ".spf" selon la provenance de son fichier d'origine.
- 7) Tire parti de tout paramètre enregistré comme décrit dans la Rec. UIT-T T.86 | ISO/CEI 10918-4.

J.9 Implémentation de la méthode ICC limitée à l'extérieur d'un moteur complet de gestion de couleur ICC

La présente annexe décrit la méthode ICC restreinte pour spécifier l'espace chromatique d'un fichier JP2 au moyen de profils ICC sur la base de la version ICC.1:1998-09 de la spécification de format de profil ICC [41]. La présente annexe est spécifiquement orientée vers les développeurs qui ne font pas appel à un moteur complet de gestion de couleur ICC et qui donc doivent toujours extraire les paramètres de transformation du profil ICC et traiter l'image au moyen d'un code propre à l'application.

J.9.1 Extraction de la transformation chromatique d'un profil ICC

J.9.1.1 Format de profil ICC

Un profil ICC utilise un format de données balisées afin d'organiser ses informations. Il est décrit dans le § 6 de la spécification de format de profil ICC. Ce format se compose d'un en-tête de 128 octets, d'une table de balises, et de données balisées. Chaque balise est indiquée par une signature de 32 bits qui correspond habituellement à quatre

caractères ASCII. Les données de chaque balise sont mémorisées dans un format qui spécifie les divers éléments de données. Chaque format est indiqué par une signature de type de données, constituée des premiers 32 bits des données balisées. Afin d'obtenir les données pour une balise, localiser d'abord la signature de cette balise dans la table de balises, qui spécifie la position et la taille des données pour cette balise, puis extraire les données sur la base de leur position et taille dans les données balisées. Une fois récupérée, la signature de type de données balisées spécifie comment interpréter les données balisées.

Les principales balises utilisées dans le traitement d'une image par un profil ICC restreint sont résumées dans le Tableau J.19.

Tableau J.19 – Balises de traitement utilisées par un profil ICC restreint

Nom de balise	Signature de balise	Type de données balisées	Signature de type de données balisées
redTRCTag	'rTRC'	curveType	'curv'
greenTRCTag	'gTRC'	curveType	'curv'
blueTRCTag	'bTRC'	curveType	'curv'
redColorantTag	'rXYZ'	XYZType	'XYZ\040'
greenColorantTag	'gXYZ'	XYZType	'XYZ\040'
blueColorantTag	'bXYZ'	XYZType	'XYZ\040'
grayTRCTag	'kTRC'	curveType	'curv'

Noter qu'un profil ICC, et donc un profil ICC restreint, peut contenir d'autres balises, comme *mediaWhitePoint*. Alors que ces balises ne sont pas utilisées dans le trajet de traitement par défaut d'un profil ICC restreint comme décrit dans le § J.9.2, des scénarios plus complexes de restitution peuvent tirer parti de ces informations pour fournir une restitution plus précise ou optimisée de l'image.

Le format de profil ICC, et donc un profil ICC restreint, spécifie un modèle de traitement qui assure la conversion entre des valeurs codées de dispositif et l'espace de connexion de profil (PCS, *profile connection space*). Ce modèle se compose de deux parties: un ensemble de trois tableaux d'interpolation unidimensionnels et une matrice de trois par trois. Les tableaux d'interpolation sont formés à partir des balises redTRCTag, greenTRCTag, et blueTRCTag. La matrice est formée à partir des balises redColorantTag, greenColorantTag, et blueColorantTag. Le modèle de traitement de base utilisant ces éléments est spécifié par l'équation (4), § 6.3.1.2 de la spécification de format de profil ICC et est décrit dans le § J.9.2 ci-dessous. Noter que le profil décrit la conversion entre valeurs de dispositif et espace PCS. La matrice et les tableaux d'interpolation doivent être inversés afin d'assurer la conversion dans le sens PCS valeurs de dispositif.

La spécification complète du format d'un profil ICC (et donc d'un profil ICC restreint) est contenue dans la spécification de format de profil ICC, version ICC.1:1998-09.

J.9.1.2 Tableaux d'interpolation

Les tableaux d'interpolation utilisent le format de balise curveType. Une balise de type curveType se compose d'un décompte suivi du nombre d'entrées tabulaires non signées de 16 bits.

Si le décompte est 1, l'entrée tabulaire unique est une valeur gamma codée. Dans ce cas, le tableau unidimensionnel est formé au moyen de la formule suivante:

$$linéaire = \left(\frac{dVal}{dMax} \right)^{\frac{tVal}{256}} \times 65\,535 \tag{J-2}$$

où *dVal* est la valeur de la composante de dispositif, *dMax* est la valeur maximale du dispositif et *tVal* est la valeur d'entrée tabulaire.

Si le décompte est supérieur à un, les entrées sont les valeurs d'une table d'interpolation. La première entrée correspond à une valeur de dispositif de 0 et la dernière entrée correspond à la valeur maximale du dispositif (par exemple 255 pour données de 8 bits, 65 535 pour données de 16 bits). Les entrées restantes sont uniformément espacées entre ces deux entrées. Par exemple, si l'on a des données de 8 bits et 6 entrées, alors la 4^e entrée correspond à une valeur de dispositif de $255 * ((4 - 1)/(6 - 1)) = 153$. Noter que 1 est soustrait de la position de l'entrée dans le tableau afin de la convertir en

index de base zéro, et que 1 est soustrait du nombre des entrées afin de le convertir en nombre d'intervalles entre entrées.

Afin de convertir la table d'interpolation en table d'exploration, le nombre des entrées dans l'interpolation doit toujours être réglé de façon à correspondre au nombre de valeurs de dispositif possibles. Dans le cas de données de 8 bits, il y a 256 valeurs possibles. Si la table d'interpolation possède 6 entrées, elle devra être étendue à 256 entrées. Il y a plusieurs méthodes à cette fin, et l'interpolation linéaire est souvent utilisée. Au moyen de cette méthode, les entrées situées entre celles de la table d'interpolation fournie sont calculées par interpolation linéaire entre les entrées adjacentes de la table d'interpolation au moyen de la valeur de dispositif comme interpolant. Par exemple, soit une table d'interpolation de 6 entrées avec les valeurs T3 dans la 3^e entrée et T4 dans la 4^e entrée. L'index de table d'exploration 128 aura une position dans la table d'interpolation de $(128/255) * 5 = 2,5098$. La valeur de table d'exploration à l'index 128 est calculée par interpolation entre la valeur à l'entrée 3 et la valeur à l'entrée 4 au moyen d'un interpolant de 0,5098: la valeur à l'index 128 = $T3 + ((T4 - T3) * 0,5098)$.

Chaque entrée dans le tableau peut être convertie en virgule flottante par division par 65 535.

Lors du traitement d'un pixel, chaque composante de celui-ci est appliqué à sa table d'exploration correspondante. Les valeurs en virgule flottante sont obtenues à partir du vecteur de colonne à 3 éléments apparaissant dans l'équation (4) du § 6.3.1.2 de la spécification de format de profil ICC.

J.9.1.3 Matrice

La matrice est formée d'après les valeurs des balises redTRCTag, greenTRCTag et blueTRCTag. Ces balises utilisent le type XYZType, qui contient trois sous-types XYZNumberType. Le premier est la composante X, le second est la composante Y et le troisième est la composante Z. Chaque type XYZNumberType contient un entier signé de 32 bits qui peut être converti en virgule flottante par division par 65 536. Les valeurs XYZ de chaque balise correspondent à une rangée de la matrice, comme représenté dans l'équation (4) du § 6.3.1.2 de la spécification de format de profil ICC. Cette matrice est multipliée par le vecteur de colonne produit par la table d'interpolation afin de produire les valeurs XYZ de l'espace PCS.

J.9.1.4 Combinaison des profils de source et de destination

Le profil imbriqué dans le fichier JP2 décrit comment convertir les données d'image en espace PCS. Il est désigné par le terme de "profil de source". Normalement, les données d'image doivent être converties en données pour un autre dispositif, comme l'affichage. Ce dispositif est désigné par le terme de "dispositif de destination" et son profil est désigné par le terme de "profil de destination". La conversion est effectuée par combinaison des modèles de traitement des profils de source et de destination. En admettant que la transformation du dispositif de destination est limitée à une matrice 3 × 3 suivie par un tableau unidimensionnel, la procédure sera:

- 1) obtenir les tableaux d'interpolation et la matrice d'après le profil de source;
- 2) obtenir les tableaux d'interpolation et la matrice d'après le profil de destination;
- 3) inverser les tableaux d'interpolation et la matrice du profil de destination;
- 4) combiner les deux matrices au moyen de la multiplication de matrice.

Cette procédure produit un modèle de traitement global qui est un ensemble constitué de tableaux unidimensionnels, d'une matrice, et d'un second ensemble de tableaux unidimensionnels. Cette procédure peut servir à convertir les pixels d'image de source en images de pixels de destination.

J.9.2 Equations de traitement chromatique pour images RGB à trois composantes

L'objectif de la méthode du profil ICC restreint est de limiter l'ensemble de tous les profils ICC à un ensemble qui peut être décrit au moyen d'un simple système d'équations de traitement chromatique. La spécification ICC définit cette classe de profil sous la forme de profils d'entrée matricielle trichromatique (définis au § 6.3.1.2 de la spécification de format de profil ICC) et de profils d'entrée monochrome (définis au § 6.3.1.1 de la spécification de format de profil ICC). Les profils de la classe des profils d'entrée matricielle trichromatique peuvent être décrits au moyen des équations suivantes:

$$\begin{aligned} linear_r &= redTRC [decompressed_r] \\ linear_g &= greenTRC [decompressed_g] \\ linear_b &= blueTRC [decompressed_b] \end{aligned} \quad (J-3)$$

$$\begin{bmatrix} connection_x \\ connection_y \\ connection_z \end{bmatrix} = \begin{bmatrix} redColorant_x & greenColorant_x & blueColorant_x \\ redColorant_y & greenColorant_y & blueColorant_y \\ redColorant_z & greenColorant_z & blueColorant_z \end{bmatrix} \begin{bmatrix} linear_r \\ linear_y \\ linear_z \end{bmatrix} \quad (J-4)$$

où le terme "*decompressed_{rgb}*" est le pixel décomprimé d'origine et où le terme "*connection_{xyz}*" est le pixel converti en la forme XYZ de l'espace de connexion de profil (XYZ_{PCS}). Dans l'équation (J-3), les trois tables d'exploration sont chargées d'après le profil ICC restreint issu des balises redTRCTag, greenTRCTag et blueTRCTag respectivement, comme défini aux § 6.4.38, 6.4.18 et 6.4.4, respectivement, de la spécification de format de profil ICC. Le format de données commun de ces balises est défini au § 6.5.25 de la spécification de profil. Dans l'équation (J-4), les rangées de la matrice sont chargées d'après les balises redColorantTag, greenColorantTag et blueColorantTag respectivement, comme défini aux § 6.4.39, 6.4.19 et 6.4.5, respectivement, de la spécification de format de profil ICC. Le format de données commun de ces balises est défini au § 6.5.2 de la spécification de profil.

La classe de profil d'entrée monochrome peut être décrite par l'équation suivante:

$$connection = grayTRC[device] \quad (J-5)$$

où le terme "device" est le pixel décomprimé d'origine et où le terme "connection" est le canal achromatique de l'espace de connexion de profil. Dans l'équation (J-5), la table d'exploration est chargée d'après le profil ICC restreint à partir de la balise grayTRCTag, comme spécifié au § 6.3.17. Le format de données de cette balise est défini au § 6.5.2 de la spécification de profil.

J.9.3 Conversion d'images en format sRGB

Un des scénarios applicatifs les plus communs sera la situation dans laquelle une image spécifiée au moyen de la méthode du profil ICC restreint doit toujours être convertie en l'espace chromatique sRGB pour l'affichage d'une version électronique (par exemple édition assistée par ordinateur et navigateurs IP) [42].

Cette transformation est utilisée conjointement avec la méthode ICC restreinte afin de créer des valeurs sRGB provenant de valeurs chromatiques de source originales [47]. Si applicable, des transformées analogues (tables d'exploration ou matrices unidimensionnelles) peuvent être combinées afin d'améliorer le rendement du traitement. Dans cet exemple, seule la transformation d'après l'espace de connexion de profil (XYZ_{PCS}) sera représentée. Elle pourra ultérieurement être combinée avec les transformées des équations (J-3) et (J-4).

Lors du transfert de couleurs codées dans l'espace XYZ_{PCS} vers des couleurs codées dans l'espace chromatique sRGB, trois éléments sont nécessaires afin d'effectuer cette transformation. Ces éléments sont imbriqués dans deux matrices 3 × 3. Puis une équation de conversion de linéaire en non linéaire peut être appliquée canal par canal en pratique au moyen de trois tables d'exploration unidimensionnelles.

La première matrice de la transformation est requise afin d'effectuer une transformation d'adaptation chromatique entre le point du blanc adaptatif défini dans l'espace de connexion du profil ICC (chromaticités CIE D50) et le point du blanc adaptatif défini dans l'espace sRGB (chromaticités CIE D65). Plusieurs choix de transformation différents peuvent être utilisés. Pour cet exemple de transformation, c'est la transformation d'adaptation chromatique de Bradford (BFD) qui sera utilisée [43]. L'on a montré que la transformation de Bradford produisait des résultats précis. Elle a été adoptée comme partie intégrante du modèle d'aspect chromatique recommandé par la CIE (CIECAM97s) [44], [45]. La transformation BFD normalement comporte une portion linéaire et une portion non linéaire. Dans le cas de cet exemple de transformation, la portion non linéaire de la transformation de Bradford a été laissée de côté afin de permettre un traitement par simple matrice 3 × 3. L'on a montré que la performance de la transformation de Bradford restait encore très bonne avec cette omission [46].

La seconde matrice de la transformation est une matrice de transformation de couleurs primaires requise afin de transférer des couleurs primaires de l'espace XYZ_{PCS} à l'ensemble primaire UIT-R BT.709-2 défini dans la norme sRGB, CEI/TC100/PT61966-2.1.

Par ailleurs, la transformation se présente comme suit, la transformation primaire étant désignée par "PT" et la matrice d'adaptation chromatique de Bradford étant désignée par "BFD":

$$\begin{bmatrix} slinear_r \\ slinear_g \\ slinear_b \end{bmatrix} = \begin{bmatrix} 3,2406_{PT} & -1,5372_{PT} & -0,4986_{PT} \\ -0,9689_{PT} & 1,8758_{PT} & 0,0415_{PT} \\ 0,0557_{PT} & -0,2040_{PT} & 1,0570_{PT} \end{bmatrix} \begin{bmatrix} 0,9554_{BDF} & -0,0231_{BDF} & 0,0633_{BDF} \\ -0,0284_{BDF} & 1,0100_{BDF} & 0,0211_{BDF} \\ 0,0123_{BDF} & -0,0205_{BDF} & 1,3305_{BDF} \end{bmatrix} \begin{bmatrix} connection_x \\ connection_y \\ connection_z \end{bmatrix} \quad (J-6)$$

Cependant, ces matrices peuvent être combinées afin de former une seule matrice comme représenté dans les équations suivantes:

$$\begin{bmatrix} \text{slinear}_r \\ \text{slinear}_g \\ \text{slinear}_b \end{bmatrix} = \begin{bmatrix} 3,1337 & -1,6173 & -0,4907 \\ -0,9785 & 1,9162 & 0,0334 \\ 0,0720 & -0,2290 & 1,4056 \end{bmatrix} \begin{bmatrix} \text{connection}_x \\ \text{connection}_y \\ \text{connection}_z \end{bmatrix} \quad (\text{J-7})$$

Il est ensuite nécessaire de convertir les valeurs slinear_{rgb} en valeurs sRGB non linéaires. Pour chaque canal de rouge, vert et bleu de la valeur slinear_{rgb} (désignée par slinear_X), le canal respectif de la valeur non linéaire sRGB (désignée par sRGB_X) est calculé au moyen de l'équation (J-8).

$$\text{sRGB}_X = \begin{cases} 12,92 \cdot \text{slinear}_X & \text{slinear}_X \leq 0,0031308 \\ 1,055 \cdot \text{slinear}_X^{(1,0/2,4)} - 0,055 & \text{slinear}_X > 0,0031308 \end{cases} \quad (\text{J-8})$$

Noter que la conversion d'un pixel décomprimé en valeur sRGB peut être optimisée par combinaison de la matrice de colorants décrite dans l'équation (J-4) avec la matrice de conversion de XYZ à sRGB décrite dans l'équation (J-7), comme suit:

$$\begin{bmatrix} \text{slinear}_r \\ \text{slinear}_g \\ \text{slinear}_b \end{bmatrix} = \begin{bmatrix} 3,1337 & -1,6173 & -0,4907 \\ -0,9785 & 1,9162 & 0,0334 \\ 0,0720 & -0,2290 & 1,4056 \end{bmatrix} \begin{bmatrix} \text{redColorant}_x & \text{greenColorant}_x & \text{blueColorant}_x \\ \text{redColorant}_y & \text{greenColorant}_y & \text{blueColorant}_y \\ \text{redColorant}_z & \text{greenColorant}_z & \text{blueColorant}_z \end{bmatrix} \begin{bmatrix} \text{linear}_r \\ \text{linear}_g \\ \text{linear}_z \end{bmatrix} \quad (\text{J-9})$$

Cette optimisation réduit le traitement de l'espace chromatique d'un pixel décomprimé vers l'espace sRGB à l'application d'une table d'exploration unidimensionnelle, d'une seule matrice 3×3 et d'une autre table d'exploration unidimensionnelle.

Les transformées représentées ci-dessus pour sRGB peuvent être généralisées de façon à servir lors de la conversion en de nombreux autres espaces chromatiques cibles que l'espace sRGB. Souvent, les étapes suivies correspondront exactement à celles qui sont nécessaires pour la conversion en sRGB. Cependant, dans d'autres cas, de moins nombreuses étapes peuvent être requises, par exemple lorsque le point du blanc adaptatif de l'espace chromatique cible correspond à celui de l'espace PCS XYZ, supprimant ainsi la nécessité d'une transformée par adaptation chromatique. Il est également possible que certains cas puissent nécessiter des étapes supplémentaires afin de compenser différents facteurs comme des différences de conditions de visualisation. Les transformées de condition de visualisation sont hors du domaine d'application de la présente annexe, mais ont été traitées dans d'autres publications [41], [42], [46], [48].

Il convient de noter que, selon l'espace chromatique de mémorisation, il peut y avoir une perte d'informations due à la conversion en sRGB, ou en tout autre espace chromatique de sortie à palette limitée. Soit par exemple le cas où l'espace chromatique de mémorisation est à gamme étendue. La conversion en sRGB se traduira par l'écrtage des couleurs qui sont hors de la gamme chromatique sRGB. Bien qu'elles soient une étape nécessaire lors de l'affichage de l'image, des transformations chromatiques (comme un décalage chromatique) peuvent avoir été en mesure d'utiliser ces données écrtées (par décalage en gamme sRGB). En tant que telles, il est souvent préférable d'effectuer la plupart des transformations chromatiques sur les données mémorisées originales avant de les convertir en l'espace chromatique sRGB. De même, si l'image doit ensuite être imprimée sur un dispositif de sortie capable d'imprimer les couleurs qui ont été écrtées, alors il sera préférable de revenir à l'image dans l'espace chromatique de mémorisation, plutôt que d'imprimer l'image sRGB. Comme autre exemple, soit le cas où l'image de l'espace chromatique de mémorisation est un codage chromatique étendu d'une scène à étendue dynamique. La conversion en sRGB comportera nécessairement une étape de restitution dans laquelle les informations de rehaussement sont écrtées en fonction de l'étendue dynamique de l'affichage de sortie. (L'étape de restitution est couramment implémentée par application d'une fonction d'échelle de luminances dans le cadre des courbes TRC représentées dans l'équation (J-3).) Les informations qui sont perdues dans l'étape de restitution ne pourront pas servir à modifier l'image ultérieurement. De même, la conversion en espace sRGB peut introduire des erreurs de quantification dans l'image, ce qui en limiterait la qualité. Dans ces deux exemples, il peut être désirable de conserver l'image dans l'espace chromatique de mémorisation, et d'appliquer à l'image présente d'éventuelles manipulations. En variante, l'image peut être convertie en un codage chromatique intermédiaire à gamme étendue. La conversion en espace sRGB peut être effectuée aux seules fins de la prévisualisation, ou en tant qu'étape finale dans la chaîne de mise en images.

J.9.4 Conversion d'images en autres espaces chromatiques

En variante, il peut être désirable, dans certaines applications, de convertir des images en d'autres espaces chromatiques que l'espace sRGB aux fins de l'affichage sur des dispositifs de sortie spécifiques, ou aux fins d'une manipulation dans un espace chromatique propre à une application. Par exemple, si l'on souhaite afficher l'image sur un écran cathodique dont les caractéristiques connues sont différentes de l'affichage sRGB de référence, alors la matrice et la non-linéarité spécifiées dans l'équation (J-6) jusqu'à l'équation (J-9) peuvent être remplacées par la matrice et la non-linéarité correspondant à cet écran cathodique particulier. Généralement, la matrice associée aux chromaticités et au point du blanc des luminophores peut servir à convertir les valeurs trichromatiques de l'espace PCS en valeurs RGB linéaires particulières, tandis que la non-linéarité peut servir à associer les valeurs RGB linéaires aux valeurs codées correspondantes. De même, d'autres espaces chromatiques à couleurs RGB additives, comme l'espace RGB à codage de référence ROMM, peuvent également être calculés par substitution des appropriées matrice et non-linéarité appropriées [49].

Dans certains cas, il peut être souhaitable de convertir des images en d'autres espaces chromatiques qui ne peuvent pas être décrits par une simple transformation à matrice/non-linéarité. A cette fin, l'on peut remplacer la transformation décrite dans le § J.9.3 par la transformation appropriée dans l'espace chromatique recherché. L'on pourra souvent réaliser cela de façon commode au moyen d'un profil ICC pour l'espace chromatique recherché.

J.9.5 Etendues d'entrée/sortie et quantification

Les valeurs codées d'entrée dans les tables d'exploration indiquées dans l'équation (J-3) (redTRC, greenTRC et blueTRC) doivent être des entiers de même précision que les valeurs codées décompressées, et doivent être indexées de façon que TRC[i] produise la valeur correcte d'intensité linéaire pour une valeur codée d'entrée de i . Les valeurs codées d'entrée qui sont plus grandes que le nombre d'éléments de la table d'exploration - 1 doivent être écrêtées jusqu'au nombre d'éléments de la table d'exploration - 1.

Le pixel de sortie d'après l'équation (J-3) doit être une valeur réelle d'intensité linéaire dans l'étendue nominale (0,0, 1,0).

L'entrée dans la matrice colorante de l'équation (J-4) doit également être une valeur réelle d'intensité linéaire dans l'étendue (0,0, 1,0). La sortie de cette équation (les valeurs XYZ_{PCS}) est normalisée de façon que la valeur Y soit dans l'étendue (0,0, 1,0). Les valeurs neutres dans l'image devraient correspondre aux valeurs XYZ ayant la chromaticité du point blanc de l'espace PCS (ce qui implique que $X/Y = 0,9642$, et $Z/Y = 0,8250$). Si l'application convertit les valeurs codées d'entrée en l'espace chromatique sRGB, cette étendue de sortie permet une concaténation directe des matrices comme dans l'équation (J-8).

Les étendues et la quantification de la transformation de XYZ_{PCS} en sRGB sont semblables. Les entrées et sorties de l'équation (J-6), et donc les entrées dans l'équation (J-8), sont également des valeurs réelles dans l'étendue (0,0, 1,0).

Les sorties de l'équation (J-8) sont des valeurs dans l'étendue (0,0, 1,0). Cependant, ces valeurs seront généralement normalisées par 255 afin de produire des valeurs sRGB de 8 bits. Cela dépend beaucoup de l'application et du type de traitement additionnel qui sera éventuellement effectué. Cependant, il est fortement suggéré que tout traitement chromatique soit effectué sur les données d'image de source (décompressées_r, décompressées_g, décompressées_b) avant qu'elles soient converties en sRGB, car la possibilité de quantification notablement réduite existe.

J.9.6 Avantage de multiples spécifications d'espace chromatique

Le format JP2 permet qu'un fichier spécifie de multiples méthodes d'interprétation de l'espace chromatique d'une image. Par exemple, une application peut écrire des images dans lesquelles les valeurs de pixel ont déjà été converties en signaux nécessaires pour piloter un dispositif de sortie particulier. Dans cette situation, il est utile que l'application fournisse un mécanisme simple permettant au dispositif de déterminer qu'un traitement chromatique additionnel n'est pas requis. A cette fin, on peut spécifier le nom de l'espace chromatique du dispositif au moyen de la méthode d'espace chromatique énuméré, dans une seule boîte de spécification de couleur du fichier.

Cependant, d'autres applications, comme les navigateurs IP, doivent toujours convertir l'image en signaux appropriés à l'affichage sur d'autres dispositifs; il est très probable que ces applications ne connaîtront pas la définition de l'espace chromatique propre à ce vendeur. Il est donc très utile que le rédacteur du fichier original rédige une seconde boîte de spécification de couleur dans le fichier qui utilise la méthode du profil ICC restreint ou celle du profil ICC générique. La fourniture d'un mécanisme secondaire permet d'augmenter notablement le nombre d'applications qui ont la capacité d'interpréter correctement l'espace chromatique de l'image.

Noter que la façon de choisir entre de multiples méthodes de spécification de couleur définies dans un même fichier n'est pas spécifiée par la présente Recommandation | Norme internationale. C'est à chaque application qu'il appartient de choisir la méthode qui satisfait le mieux les exigences de cette application particulière.

J.10 Exemple d'interprétation de composantes multiples

Un exemple d'une interprétation non traditionnelle est le codage de régions intéressantes (ROI, *regions of interest*) dans un ensemble complexe de données SAR. Chaque région ROI peut être considérée comme un ensemble de deux fragments d'image représentant les parties réelle (I) et imaginaire (Q) des données. L'ensemble des fragments I et Q peut être assemblé en un ensemble de "composantes multiples", même si les fragments individuels sont disjoints et peuvent avoir des dimensions spatiales différentes. Après évitement de la transformée d'espace chromatique, l'ensemble de fragments peut alors être soumis à une compression avec ou sans perte. Cette procédure présente deux avantages: toutes les régions ROI contenues dans un certain ensemble de données peuvent être comprimées par une seule passe; et l'attribution des bits peut être optimisée dans tout l'ensemble d'images ROI plutôt que fragment par fragment.

J.11 Exemple de décodage montrant les étapes intermédiaires

Soit les flux binaires comprimés suivants où le décalage à partir du début du fichier est donné en valeurs octales à gauche, et où les valeurs du fichier sont données en valeurs hexadécimales.

```

00000 FF4F FF51 0029 0000 0000 0001 0000 0009
00020 0000 0000 0000 0000 0000 0001 0000 0009
00040 0000 0000 0000 0000 0001 0701 01FF 5C00
00060 0740 4048 4850 FF52 000C 0000 0001 0001
00100 0404 0001 FF90 000A 0000 0000 001E 0001
00120 FF93 C7d4 0C01 8F0D C875 5DC0 7C21 800F
00140 B176 FFD9

```

Les diverses parties de ce flux binaire peuvent être décodées comme suit.

J.11.1 En-tête principal

L'en-tête principal commence à l'octet 0 comme indiqué par le marqueur SOC et se termine avant l'octet 0104 (octal), qui est connu en raison du marqueur SOT.

```

00000 FF4F           Marqueur SOC
00002 FF51           Marqueur SIZ
00004 0029           Lsiz longueur du marqueur SIZ
00006 0000           Rsiz
00010 0000 0001     Xsiz
00014 0000 0009     Ysiz
00020 0000 0000     XOsiz
00024 0000 0000     YOsiz
00030 0000 0001     XTsiz
00034 0000 0009     YTsiz
00040 0000 0000     XTOsiz
00044 0000 0000     YTOsiz
00050 0001           Csiz
00052 07            Ssiz
00053 01            XRsiz
00054 01            YRsiz

```

Donc "l'image" est une composante ayant 8 bits/échantillons non signés, 1 échantillon horizontalement, et 9 échantillons verticalement. Tous les échantillons sont contenus dans un même pavé.

```

00055 FF5C           Marqueur QCD
00057 0007           Lqcd longueur du marqueur QCD
00061 40            Sqcd
00062 4048 4850     SPqcd

```

Il y a 2 bits de garde, aucune quantification n'est effectuée (autre qu'une éventuelle troncature), et les exposants d'échelon de quantification ϵ_n sont {8,9,9, 10}.

```

00066 FF52           Marqueur COD

```

ISO/CEI 15444-1:2004 (F)

00070	000C	Lcod longueur du marqueur COD
00072	00	Scod (PPx = PPy = 15, aucun SOP, aucun EPH)
00073	00	Ordre de progression
00074	0001	Nombre de couches
00076	00	Transformée multiple de composante
00077	01	Nombre de niveaux de décomposition
00100	04	Valeur du décalage de l'exposant de largeur de bloc codé
00101	04	Valeur du décalage de l'exposant de hauteur de bloc codé
00102	00	Style des passes de codage de bloc codé
00103	01	Transformée

Aucun district n'est utilisé. Il y a un seul niveau de transformée en ondelettes. L'ordre de progression est: couche-niveau de résolution-composante-position, mais il n'y a qu'une seule couche. Les blocs codés ont 64×64 échantillons (noter que la longueur est 2^6 alors que la valeur contenue dans le flux binaire est 4). Il n'y a aucun évitement sélectif du codage arithmétique, aucune réinitialisation de probabilités contextuelles ou terminaison à chaque passe de codage, pas de contextes générateurs de bande verticale, pas de terminaison prévisible, et pas de symboles de segmentation. Le filtre réversible 5-3 est utilisé.

J.11.2 En-tête d'élément de pavé

Le premier et seul en-tête d'élément de pavé commence à la valeur octale de l'octet 0104 avec le marqueur SOT et se termine à la valeur octale de l'octet 0120 avec le marqueur SOD.

00104	FF90	Marqueur SOT
00106	000A	Lsot longueur du marqueur SOT
00110	0000	Isot
00112	0000 001E	Psot
00116	00	TPsot
00117	01	TNsot

Il s'agit du pavé numéro 0. La longueur de l'élément de pavé est de 30 octets. Donc, le prochain élément de pavé ou la fin du flux codé est à $0104 + 036 = 0142$. C'est l'élément de pavé 0. Ce pavé a un seul élément.

00120	FF93	SOD marker
-------	------	------------

Il n'y avait pas de marqueur COD, QCD ni de marqueur de commentaires dans cet en-tête d'élément de pavé. Donc, tous les paramètres de codage sont déterminés d'après l'en-tête principal. Les 16 prochains octets sont des données comprimées (30 octets de longueur – 14 octets de segments marqueurs).

Données comprimées (en-tête de paquet et corps de paquet)

00122	C7D4 0C01 8F0D C875 5DC0 7C21 800F
00140	B176

Fin d'image

00142	FFD9	Marqueur de fin EOC
-------	------	---------------------

J.11.3 En-tête de paquet

Comme l'image est de format 1×9 et qu'il n'y a qu'un seul niveau de transformation (et que les blocs codés, districts, et pavés sont trop grands pour avoir un effet), il y aura 5 coefficients d'ondelette passe-bas, et 4 coefficients passe-bas horizontalement et passe-haut verticalement. Les données comprimées commencent par un en-tête de paquet qui est décodé comme représenté dans le Tableau J.20.

Tableau J.20 – Décodage du premier en-tête de paquet

Octets de flux codé	Bits utilisés	Commentaires
0xC7	1	Paquet de longueur non nulle
	1	Seul le bloc codé est inclus
	0001	3 plans binaires zéro ^{a)}
0xD4	11 1101010	16 passes de codage pour ce bloc codé ^{b)}
	0	<i>LBlock</i> reste 3
0x0C	0000110	6 octets de données comprimées ^{c)}
	0	Bit de bourrage inutilisé
<p>a) Le nombre maximal de plans binaires d'après l'équation (E-2) est 9, dont 3 sont à zéro, de sorte que le premier bit sera décodé dans le plan binaire 6.</p> <p>b) Le nombre de bits à lire dépend des données comprimées; voir Tableau B.2.</p> <p>c) Le nombre de bits à lire est <i>LBlock</i> plus la fonction plancher de base 2 log du nombre de passes: $3 + \lfloor \log 16 \rfloor = 7$.</p>		

Le décodage du premier en-tête de paquet nécessite 3 octets et indique que 6 octets de données comprimées en codage arithmétique sont utilisés pour le seul bloc codé contenu dans ce paquet. Donc, le prochain en-tête de paquet commencera au décalage 0134. Cet en-tête de paquet est décodé dans le Tableau J.21.

Tableau J.21 – Décodage du second en-tête de paquet

Octets de flux codé	Bits utilisés	Commentaire
0xC0	1	Paquet de longueur non nulle
	1	Seul le bloc codé est inclus
0x7C	000000 01	7 plans binaires zéro ^{a)}
0x21	111100 001	7 passes de codage pour ce bloc codé
	0	<i>LBlock</i> reste 3
0x80	0001 1	3 octets de données comprimées ^{b)}
0xC0	1	Bit de bourrage inutilisé
<p>a) Le nombre maximal de plans binaires est 10, dont 7 sont zéro, de sorte que le premier bit sera décodé dans le plan binaire 3.</p> <p>b) Le nombre de bits à lire est $3 + \lfloor \log 7 \rfloor = 5$.</p>		

J.11.4 Données comprimées à codage arithmétique

Les six octets de données comprimées du premier bloc codé (issu du premier paquet) peuvent être décodés comme représenté dans le Tableau J.22. Le premier élément est l'étiquette de contexte d'après l'Annexe D (qui peut être complètement différent pour chaque implémentation). Le second élément est le type de contexte. Finalement le bit renvoyé par le codeur arithmétique est indiqué. Ces bits sont utilisés afin de déterminer les coefficients passe-bas horizontalement et passe-haut verticalement. Les octets fournis au codeur arithmétique sont ceux qui commencent au décalage 0125.

0000125 01 8F0D C875 5D

Tableau J.22 – Décodage arithmétique du premier bloc codé

CTX	Type de contexte	Bit	Commentaire
17	C4(ZERO_RUN)	1	Plage différente de zéro
18	C5(UNIFORM)	1	Le premier coefficient différent de zéro est le quatrième (numéroté à partir de 1)
18	C5(UNIFORM)	1	
9	C2(SIGN)	1	Négatif
3	C1 (NEW_SIGNIFICANT)	0	Le cinquième coefficient n'est pas significatif
3	C1 (NEW_SIGNIFICANT)	1	Le troisième coefficient est significatif (premier coefficient qui est dans la passe de signification)
10	C2 (SIGN)	0	Négatif (le bit XOR est 1)
3	C1 (NEW_SIGNIFICANT)	1	Cinquième coefficient significatif dans cette passe de codage
10	C2 (SIGN)	0	Négatif (le bit XOR est 1)
15	C3 (REFINE)	0	Le prochain bit du quatrième coefficient est 0
0	C1 (NEW_SIGNIFICANT)	1	Le premier coefficient est significatif
9	C2 (SIGN)	1	Négatif
4	C1 (NEW_SIGNIFICANT)	1	Le second coefficient est significatif
10	C2 (SIGN)	0	Négatif
15	C3 (REFINE)	1	Tous les coefficients sont dans la passe de raffinement. Le bit décodé est le prochain bit du coefficient en séquence du premier au cinquième
15	C3 (REFINE)	0	
15	C3 (REFINE)	1	
16	C3 (REFINE)	0	
15	C3 (REFINE)	0	
16	C3 (REFINE)	0	Prochain plan binaire
16	C3 (REFINE)	1	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	Prochain plan binaire
16	C3 (REFINE)	1	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	Dernier plan binaire.
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	

Donc les coefficients décodés sont les suivants:

-26, -22, -30, -32, -19

Les données comprimées pour le seul bloc codé dans le second paquet, représentant la sous-bande passe-haut verticalement et passe-bas horizontalement, commencent au décalage 0137 en valeur octale.

0000137 0F B176

Le processus de décodage est décrit dans le Tableau J.23.

Tableau J.23 – Décodage arithmétique du second bloc codé

CTX	Type de contexte	Bit	Commentaire
17	C4 (ZERO_RUN)	1	Plage différente de zéro
18	C5 (UNIFORM)	0	Le premier coefficient différent de zéro est le second
18	C5 (UNIFORM)	1	
9	C2 (SIGN)	0	Positif
3	C1 (NEW_SIGNIFICANT)	0	Troisième et quatrième coefficients en passe de nettoyage
0	C1 (NEW_SIGNIFICANT)	0	
3	C1 (NEW_SIGNIFICANT)	0	1 ^{er} et 3 ^e coefficients en passe de signification
3	C1 (NEW_SIGNIFICANT)	0	
14	C3 (REFINE)	0	Deuxième coefficient
0	C1 (NEW_SIGNIFICANT)	0	Quatrième coefficient en passe de nettoyage
3	C1 (NEW_SIGNIFICANT)	1	Premier coefficient en passe de signification
10	C2 (SIGN)	0	Positif
3	C1 (NEW_SIGNIFICANT)	1	Troisième coefficient en passe de signification
10	C2 (SIGN)	0	Positif
3	C1 (NEW_SIGNIFICANT)	0	Quatrième coefficient en passe de signification
16	C3 (REFINE)	1	Deuxième coefficient en passe de raffinement

Les coefficients passe-haut verticalement et passe-bas horizontalement sont les suivants après décodage:

1, 5, 1, 0

J.11.5 Ondelette et décalage de niveau

Après le filtre inverse réversible 5-3 et le décalage de niveau, les échantillons de composante en décimal sont les suivants:

101, 103, 104, 105, 96, 97, 96, 102, 109

J.12 Pondération de fréquence visuelle

Le système visuel humain joue un rôle important dans la qualité perçue d'images comprimées. Il est donc souhaitable de permettre aux concepteurs et utilisateurs de système de tirer parti des connaissances actuelles en matière de perception visuelle, par exemple, d'utiliser des modèles de la sensibilité variable aux fréquences spatiales du système visuel, telle que mesurée dans la fonction de sensibilité au contraste (CSF, *contrast sensitivity function*). Comme la pondération de la fonction CSF est déterminée par la fréquence visuelle du coefficient de transformation, il y aura une seule pondération par fonction CSF par sous-bande dans la transformée en ondelettes. La conception de la pondération par fonction CSF relève du codeur et dépend de la condition spécifique de visualisation dans laquelle l'image décodée doit être vue. Prière de se rapporter à [34] et [35] pour de plus amples détails sur la conception des pondérations par la fonction CSF.

Souvent, un seul ensemble de pondérations par fonction CSF est choisi et appliqué selon la condition de visualisation. Cette application de pondération de fréquence visuelle est désignée par le terme de *pondération visuelle fixe*. Dans le cas de codeurs imbriqués, étant donné que le flux binaire de codage peut être tronqué ultérieurement, les conditions de visualisation à différentes étapes de l'imbrication peuvent être très différentes. A bas débit, la qualité de l'image comprimée est médiocre et les caractéristiques détaillées de l'image ne sont pas disponibles. L'image est habituellement visualisée à une distance relativement grande et les observateurs sont plus intéressés par les caractéristiques globales. Au fur et à mesure que plus de bits sont reçus, la qualité d'image améliore, et les détails de l'image sont révélés. L'image est habituellement examinée à une plus faible distance, ou est même agrandie pour un examen de près, ce qui équivaut à diminuer la distance de visualisation. Donc, différents ensembles de pondérations par fonction CSF sont appelés à différentes étapes de l'imbrication. Cette application réglable de la pondération de fréquence visuelle est désignée par le terme de *codage visuel progressif*. Il est clair que la pondération visuelle fixe peut être considérée comme un cas particulier de codage visuel progressif.

En pondération visuelle fixe, un ensemble de pondérations par fonction CSF, $\{w_i\}$, est choisi selon la condition de visualisation finale, où w_i est la pondération pour la i ème sous-bande. L'ensemble de pondérations par fonction CSF peut être incorporé d'une des deux façons suivantes.

J.12.1 Modification de la taille de l'échelon de quantification

Dans le codeur, la taille d'échelon de quantification q_i des coefficients de transformation de la i ème sous-bande est réglée de façon à être inversement proportionnelle à la pondération par fonction CSF w_i . Plus la pondération par fonction CSF diminue, plus la taille de l'échelon de quantification augmente. Les indices de quantification normalisés par la fonction CSF sont alors traités uniformément dans le processus d'optimisation R-D, qui n'est pas modifié afin de prendre en compte d'éventuels changements dans la taille de l'échelon de quantification. Les pondérations par fonction CSF ne doivent pas être transmises au décodeur. Les informations sont incluses dans les grandeurs d'échelon de quantification, qui sont explicitement transmises pour chaque sous-bande. Cette méthode doit spécifier explicitement le quantificateur. Elle peut donc ne pas être très appropriée au codage imbriqué, spécialement au codage imbriqué avec perte dans tous les modes jusqu'au codage sans perte.

J.12.2 Modification de l'ordre de codage imbriqué

Les grandeurs d'échelon de quantification ne sont pas modifiées mais les pondérations de distorsion introduites dans l'optimisation R-D sont au contraire altérées ce qui permet de contrôler effectivement la signification relative de différents numéros de plans binaires d'après le flux binaire imbriqué de chaque bloc codé. La table de pondération en fréquence peut être transmise explicitement. Cette méthode est recommandée car elle produit des résultats semblables dans le § J.12.1 et est compatible avec la compression sans perte. Cette méthode affecte seulement le compresseur et est compatible avec toutes les stratégies de quantification, y compris la quantification implicite.

J.12.3 Codage visuel progressif (VIP)

Si les pondérations de fréquence visuelle doivent être modifiées pendant le processus de codage imbriqué, il est très peu pratique de modifier les valeurs de coefficient ou les grandeurs d'échelon de quantification. Par ailleurs, la performance du codeur entropique subséquent peut se dégrader en raison des variations statistiques de la représentation binaire. Une façon élégante d'implémenter le codage visuel progressif (VIP, *visual progressive coding*) consiste à changer, à la volée, l'ordre dans lequel les sous-plans binaires de blocs codés doivent apparaître dans l'ensemble du flux binaire imbriqué sur la base des pondérations visuelles, au lieu de modifier les valeurs de coefficient ou les grandeurs d'échelon de quantification. En d'autres termes, c'est l'ordre plutôt que le contenu du codage qui est affecté par les pondérations visuelles.

Une série d'ensembles de pondérations visuelles pour différentes étendues de débit est indiquée comme suit:

$$\begin{aligned} \text{Ensemble de pondérations } 0 : r(0), \text{ avec } W(0) &= \{w_0(0), w_1(0), \dots, w_n(0)\}; \\ \text{Ensemble de pondérations } 1 : r(1), \text{ avec } W(1) &= \{w_0(1), w_1(1), \dots, w_n(1)\}; \\ &\dots \\ \text{Ensemble de pondérations } m : r(m), \text{ avec } W(m) &= \{w_0(m), w_1(m), \dots, w_n(m)\}, \end{aligned} \tag{J-10}$$

où $r(j)$ représente le débit binaire auquel les facteurs de pondération sont modifiés, $r(0) < r(1) < \dots < r(m)$, et $w_i(j)$ sont les pondérations appliquées à la sous-bande i dans l'étendue de débit allant de $r(j)$ à $r(j+1)$. Chaque ensemble de pondérations visuelles prendra effet dans une certaine étendue de débit. Si $m = 0$, c'est-à-dire s'il y a un seul ensemble de pondérations visuelles, l'on revient au cas élémentaire de la pondération visuelle fixe. L'ensemble de pondérations visuelles, de $W(0)$ à $W(m)$, sera utilisé afin de déterminer l'ordre d'imbrication dans leurs étendues de débit correspondantes. Pour l'imbrication à haut débit, en particulier pour le codage imbriqué dans toutes ses formes avec perte jusqu'à sans perte, les pondérations visuelles finales $W(m)$ doivent être codées par des séries de 1 (comme pour l'absence de pondération en codage sans perte). Le codage visuel progressif peut ajuster les pondérations visuelles afin d'obtenir une bonne qualité visuelle à tous les débits binaires.

La pondération VIP affecte seulement le codeur et aucune signalisation n'est requise dans le décodeur.

Le codeur est appelé à calculer l'ordre dans lequel les sous-plans binaires de blocs codés doivent apparaître dans la hiérarchie stratifiée de l'ensemble du flux binaire, sur la base de critères de distorsion de débit. Une simple implémentation de pondération visuelle progressive modifie progressivement la mesure de distorsion en fonction des pondérations visuelles pendant la formation du flux binaire. Comme la formation du flux binaire est régie par l'optimisation R-D après la compression, la modification progressive des pondérations visuelles commande effectivement l'ordre d'imbrication des sous-plans binaires de blocs codés à la volée.

J.12.4 Tableaux de pondération de fréquence recommandée

Le Tableau J.24 spécifie trois ensembles de pondérations par fonction CSF qui ont été conçus pour la composante de luminance sur la base de la valeur de la fonction CSF à mi-fréquence de chaque sous-bande. La distance de visualisation est censée être de 1 000, 2 000, et 4 000 échantillons (par exemple, correspondant à 10 pouces pour

100 dpi, 200 dpi, et 400 dpi d'impression ou d'affichage), respectivement. Noter que les tableaux visent une décomposition à 5 niveaux en ondelettes.

Le Tableau J.24 ne comporte pas de pondération pour la sous-bande de fréquence la plus basse, nLL, qui est toujours 1. Les niveaux 1, 2, ..., 5 indiquent les niveaux de sous-bande dans l'ordre croissant des fréquences. L'expression (HL, LH, HH) indique les trois orientations de fréquence dans chaque sous-bande.

Tableau J.24 – Pondération de fréquence recommandée

Niveau	Distance de visualisation 1000			Distance de visualisation 2000			Distance de visualisation 4000		
	HL	LH	HH	HL	LH	HH	HL	LH	HH
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	0,731 668
3	1	1	1	1	1	0,727 203	0,564 344	0,564 344	0,285 968
4	1	1	0,727 172	0,560 841	0,560 841	0,284 193	0,179 609	0,179 609	0,043 903
5	0,560 805	0,560 805	0,284 173	0,178 494	0,178 494	0,043 631	0,014 774	0,014 774	0,000 573

Pour les images en couleur, les tableaux de pondération de fréquence des composantes Y, Cr, et Cb doivent différer afin de tirer parti des capacités du système visuel humain. Par exemple, il est habituellement désirable de mettre en valeur la composante de luminance plutôt que les composantes de chrominance. Le Tableau J.25 spécifie trois ensembles de pondérations par fonction CSF pour les composantes de luminance et de chrominance.

Tableau J.25 – Pondération de fréquence recommandée pour images (en couleurs) à composantes multiples

Composante	Niveau	Distance de visualisation 1000			Distance de visualisation 1700			Distance de visualisation 3000		
		HL	LH	HH	HL	LH	HH	HL	LH	HH
Y (Y0)	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	0,921 045	0,921 045	0,848 324
	4	0,998 276	0,998 276	0,996 555	0,861 593	0,861 593	0,742 342	0,410 628	0,410 628	0,182 760
	5	0,756 353	0,756 353	0,573 057	0,307 191	0,307 191	0,108 920	0,038 487	0,038 487	0,003 075
Cb (Y1)	1	0,883 196	0,883 196	0,833 582	0,818 766	0,818 766	0,745 875	0,717 086	0,717 086	0,613 777
	2	0,793 487	0,793 487	0,712 295	0,689 404	0,689 404	0,579 220	0,539 437	0,539 437	0,403 353
	3	0,650 482	0,650 482	0,531 700	0,501 652	0,501 652	0,362 279	0,319 773	0,319 773	0,185 609
	4	0,450 739	0,450 739	0,309 177	0,280 068	0,280 068	0,152 290	0,124 021	0,124 021	0,044 711
	5	0,230 503	0,230 503	0,113 786	0,097 816	0,097 816	0,031 179	0,023 308	0,023 308	0,003 413
Cr (Y2)	1	0,910 877	0,910 877	0,872 378	0,860 885	0,860 885	0,803 172	0,780 091	0,780 091	0,695 128
	2	0,841 032	0,841 032	0,776 180	0,757 626	0,757 626	0,665 951	0,631 632	0,631 632	0,509 729
	3	0,725 657	0,725 657	0,625 103	0,598 537	0,598 537	0,470 893	0,428 659	0,428 659	0,287 593
	4	0,552 901	0,552 901	0,418 938	0,388 492	0,388 492	0,248 566	0,211 871	0,211 871	0,100 658
	5	0,336 166	0,336 166	0,200 507	0,177 435	0,177 435	0,077 130	0,060 277	0,060 277	0,014 977

J.13 Sous-échantillonnage de composantes par le codeur

Il est devenu de pratique courante, dans certaines applications de compression, d'utiliser un sous-échantillonnage par composantes conjointement avec certaines transformations réductrices de la redondance. Un exemple caractéristique est l'utilisation d'une transformation réductrice de la redondance de RGB en YCrCb suivie par un sous-échantillonnage des composantes de chrominance (Cr, Cb). Bien que ce soit un moyen efficace de réduire la quantité de données d'image à coder par les algorithmes de compression de type transformation DCT (Rec. UIT-T T.81 | ISO/CEI 10918-1), il n'est pas recommandé de l'utiliser dans la présente Recommandation | Norme internationale.

La nature multirésolutive de la transformation en ondelettes décrite dans la présente Recommandation | Norme internationale peut servir à obtenir le même effet que par sous-échantillonnage des composantes. Par exemple, si les sous-bandes 1HL, 1LH, 1HH d'une décomposition de composante en ondelettes sont ignorées et que toutes les autres sous-bandes soient conservées, un sous-échantillonnage 2:1 a été réalisé dans les dimensions horizontale et verticale de la composante. Cette technique fournit les mêmes avantages qu'un sous-échantillonnage explicite de la composante avant toute transformée en ondelettes.

Par ailleurs, il se révèle souvent avantageux, en termes de qualité d'image, de conserver quelques coefficients d'ondelette dans les sous-bandes 1HL, 1LH, 1HH, tout en continuant à en rejeter la grande majorité. Dans de tels cas, le nombre des coefficients est encore approximativement réduit de 2:1, mais l'imagerie décodée résultante offrira une meilleure qualité avec moins d'artifices de compression. L'utilisation d'une technique de sous-échantillonnage empêche les codeurs d'effectuer de tels choix et peut dégrader la qualité d'image décodée.

J.14 Commande de débit

La commande de débit est utile pour répondre à une exigence particulière de débit binaire ou de temps de transmission. La commande de débit garantit que le nombre d'octets souhaité est utilisé par le flux codé tout en assurant la plus haute qualité d'image possible.

J.14.1 Introduction aux concepts de clé pour commande de débit

L'on subdivise chaque sous-bande en blocs codés d'échantillons qui sont codés indépendamment. Comme chaque bloc est codé tout à fait indépendamment avec exactement le même algorithme dans chaque sous-bande, l'association entre sous-bandes et blocs codés peut être ignorée provisoirement. Soit $\{B_i\}_{i=1,2,\dots}$ l'ensemble de tous les blocs codés qui représentent l'image. Pour chaque bloc codé, B_i , un flux binaire séparé est produit sans utiliser d'informations en provenance d'autres blocs codés. Par ailleurs, le flux binaire possède la caractéristique qu'il peut être tronqué selon diverses longueurs discrètes $R_i^1, R_i^2, R_i^3, \dots$ et la distorsion subie lors de la reconstruction de chacun de ces sous-ensembles tronqués est estimée et indiquée par $D_i^1, D_i^2, D_i^3, \dots$. La mesure de distorsion par erreur quadratique moyenne est souvent utilisée, mais n'est pas nécessaire. Pendant le processus de codage, les longueurs, R_i^n , et les distorsions, D_i^n , sont calculées et temporairement mémorisées sous une forme compacte avec le flux binaire comprimé proprement dit.

Une fois que l'image entière a été comprimée, une opération de post-traitement passe sur tous les blocs codés comprimés et détermine la mesure dans laquelle le flux binaire imbriqué de chaque bloc codé doit être tronqué afin d'obtenir une mesure particulière de débit binaire cible, une mesure liée à la distorsion ou un autre paramètre de qualité. Plus généralement, le flux binaire final est composé d'un ensemble de "couches", dans lequel chaque couche possède une interprétation en termes de qualité d'image globale. La première couche, de qualité la plus basse, est formée à partir des flux binaires de bloc codé tronqués de façon optimale comme décrit ci-dessus. Chaque couche subséquente est formée par troncature optimale des flux binaires de bloc codé afin d'obtenir successivement une mesure particulière de débit binaire cible, une mesure liée à la distorsion ou un autre paramètre de qualité, selon ce qui convient, y compris les mots de code additionnels qui sont requis afin d'augmenter les informations représentées dans les couches précédentes en fonction des nouveaux points de troncature. Ces concepts de flux binaire stratifié sont examinés de façon plus approfondie dans le § J.14.2.

J.14.2 Abstraction du flux binaire stratifié

Un aspect important est la façon dont le codeur forme un flux binaire final à partir des flux binaires imbriqués produits indépendamment pour chaque bloc codé. Le problème de la formation du flux binaire est très simplifié lorsque le codeur fonctionne sur plusieurs sous-bandes entières à la fois, car l'organisation spatiale imposée par blocs codés n'existe pas.

Fondamentalement, le flux binaire est organisé sous la forme d'une succession de couches, où chaque couche contient les contributions additionnelles issues de chaque bloc codé (certaines contributions peuvent être vides), comme illustré dans la Figure J.12. Les points de troncature de bloc codé associés à chaque couche sont optimaux en terme de distorsion du débit, ce qui signifie que le flux binaire obtenu par rejet d'un nombre entier de couches moins importantes sera toujours optimal en terme de distorsion de débit. Si le flux binaire est tronqué partiellement dans une couche, alors il ne sera pas strictement optimal, mais l'écart par rapport au flux optimal peut être petit si le nombre de couches est grand. Au fur et à mesure que le nombre de couches est augmenté de façon que le nombre d'octets de code dans chaque couche soit réduit, les pentes de distorsion de débit associées à tous les points de troncature de bloc codé dans la couche convergeront; cependant, le nombre de blocs codés qui ne contribuent pas à la couche augmentera également, de façon que le surdébit associé à l'identification des blocs codés qui contribuent effectivement à la couche va augmenter. En pratique, il se trouve que la performance optimale de compression pour des applications à rapport SNR progressif est réalisée lorsque le nombre de couches est approximativement deux fois plus grand que le nombre de passes de sous-

plan binaire effectuées par le codeur entropique. Les limites des passes du sous-plan binaire sont également les points de troncature du flux binaire imbriqué de chaque bloc codé. Par conséquent, chaque couche contient en moyenne des contributions issues d'approximativement la moitié des blocs codés de façon que le surdébit dû à l'indication qu'un bloc codé contribue ou non à une couche donnée (environ 2 bits par bloc codé) est bien inférieur au surdébit dû à l'indication d'un ordre strict des contributions de bloc codé. Par ailleurs, la contribution relative de ce surdébit au débit binaire est indépendante de la longueur de l'image.

La Figure J.12 est une illustration des contributions de bloc codé aux couches de flux binaire. Seules cinq couches sont représentées avec sept blocs codés, par souci de simplicité. Noter que tous les blocs codés n'ont pas besoin de contribuer à chaque couche et que le nombre d'octets apportés par les blocs codés à une couche donnée est généralement très variable. Noter également que l'opération de codage de bloc codé procède verticalement dans chaque bloc codé indépendamment, tandis que l'organisation en flux binaire stratifié est horizontale, répartissant les passes de codage du bloc codé dans les diverses couches.

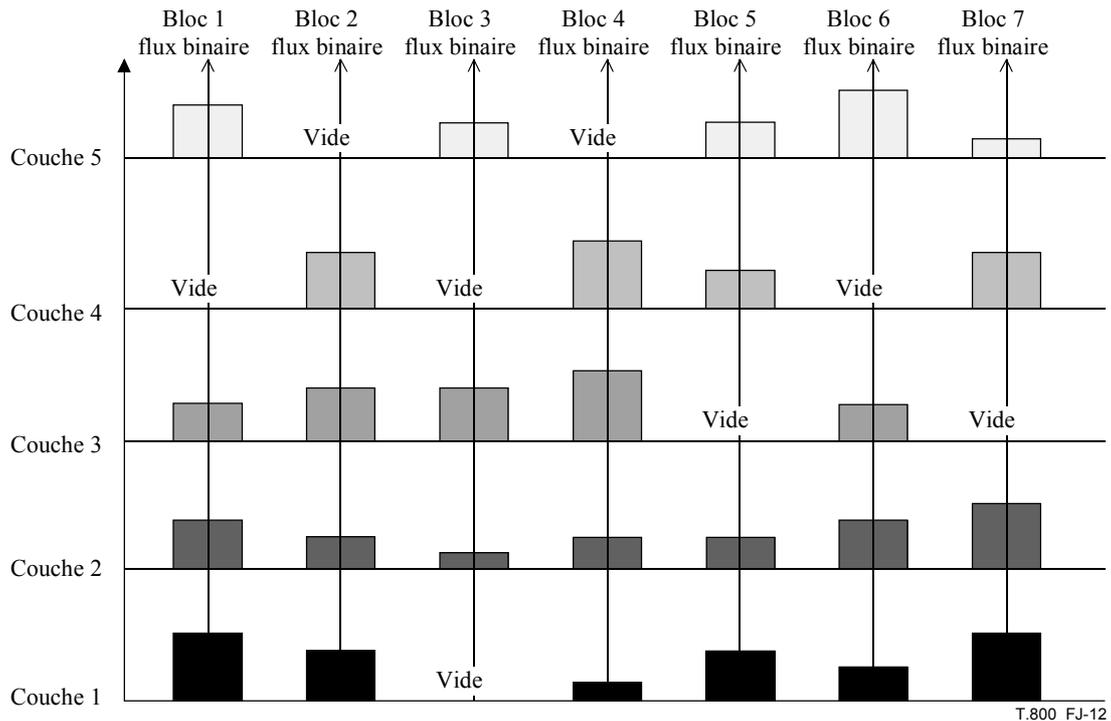


Figure J.12 – Illustration des contributions de bloc codé aux couches de flux binaire

J.14.3 Optimisation de la distorsion de débit

L'algorithme de distorsion de débit décrit ici n'est justifié que si la mesure de distorsion adoptée pour les blocs codés est additive. En d'autres termes, la distorsion, D , doit satisfaire l'équation suivante dans l'image finale reconstruite:

$$D = \sum_i D_i^{n_i} \quad (\text{J-11})$$

où n_i est le point de troncature du bloc codé B_i . Sous réserve de normalisation appropriée, cette caractéristique additive est satisfaite par l'erreur quadratique moyenne (MSE, *mean squared error*) et par l'erreur MSE pondérée (par exemple erreur MSE pondérée visuellement), à condition que la transformation en ondelettes soit orthogonale. L'additivité est également vérifiée si les erreurs de quantification de valeurs individuelles d'échantillon ne sont pas corrélées, que la transformation soit ou non orthogonale. En pratique, la transformation n'est habituellement qu'approximativement orthogonale et les erreurs de quantification ne sont pas complètement décorrélées, de sorte que même les mesures d'erreur quadratique ne sont qu'approximativement additives; mais cette approximation est habituellement suffisante. Soit R le nombre d'octets de code associés à une certaine couche dans le flux binaire (et toutes les couches précédentes). Alors, pour un certain ensemble de points de troncature, n_i :

$$R = \sum_i R_i^{n_i} \quad (\text{J-12})$$

L'objectif consiste à trouver l'ensemble des n_i valeurs qui minimise D sous réserve de la contrainte $R \leq R_{max}$. Le problème de l'optimisation d'une contrainte par la méthode des multiplicateurs de Lagrange est une solution bien connue de ce problème. Spécifiquement, le problème revient à minimiser:

$$\sum (R_i^{n_i} - \lambda D_i^{n_i}) \quad (J-13)$$

où la valeur de λ doit toujours être réglée jusqu'à ce que le débit fourni par les points de troncature qui minimisent l'équation (J-13) satisfasse $R = R_{max}$. Il n'y a aucun algorithme simple qui puisse fournir un ensemble globalement optimal de points de troncature en général. Cependant, tout ensemble de points de troncature, n_i , qui minimise l'équation (J-13) pour certaines valeurs λ est garanti d'être optimal dans le sens que la distorsion minimale est réalisée au débit binaire correspondant. Si la plus grande valeur de λ , est trouvée telle que l'ensemble de points de troncature, n_i , obtenu après avoir minimisé l'équation (J-13), fournit un débit $R \leq R_{max}$, alors il n'est pas possible de trouver un ensemble de points de troncature qui puisse fournir une plus petite distorsion globale et un débit inférieur ou égal à R . En pratique, il se trouve qu'il est habituellement possible de trouver des valeurs de λ telles que R soit très proche de R_{max} (presque toujours à 100 octets près), de façon que l'on puisse négliger une éventuelle valeur sous-optimale.

Si l'on revient maintenant au problème de minimiser l'expression de l'équation (J-13), il s'agit d'un problème d'optimisation pour chaque bloc codé individuel. Spécifiquement, pour chaque bloc codé, B_i , il faut trouver le point de troncature, n_i , qui minimise $(R_i^{n_i} + \lambda D_i^{n_i})$. Un algorithme simple est à cette fin le suivant:

Soit $n_i = 0$ (c'est-à-dire qu'aucune information n'est incluse pour le bloc codé)

Pour $k = 1, 2, 3, \dots$

Soit $\Delta R_i^k = R_i^k - R_i^{n_i}$ et $\Delta D_i^k = D_i^k - D_i^{n_i}$

Si $(\Delta D_i^k / \Delta R_i^k) > \lambda^{-1}$, on pose alors $n_i = k$

Comme il est possible que cet algorithme doive être exécuté pour de nombreuses valeurs différentes de λ , il est logique de commencer par identifier le sous-ensemble, N_i , de seuils tels que les valeurs de pente de distorsion de débit, $S_i^k = \Delta D_i^k / \Delta R_i^k$, soient en décroissance monotone avec k , pour tous les k du sous-ensemble N_i . Spécifiquement, un algorithme approprié afin de déterminer N_i se présente comme suit:

- 1) soit $N_i = \{n\}$, c'est-à-dire l'ensemble de tous les points de troncature;
- 2) soit $p = 0$;
- 3) pour $k = 1, 2, 3, 4, \dots$

Si k appartient à N_i

Soit $\Delta R_i^k = R_i^k - R_i^p$ et $\Delta D_i^k = D_i^k - D_i^p$

Soit $S_i^k = \Delta D_i^k / \Delta R_i^k$

Si $p \neq 0$ et $S_i^k > S_i^p$, alors supprimer p de N_i et passer à l'étape (2)

Sinon, poser $p = k$

Une fois que ces informations ont été précalculées, la tâche d'optimisation pour toute donnée λ consiste simplement à poser P égal au plus grand k contenu dans N_i tel que $S_i^k > \lambda^{-1}$. Il est clair que λ peut être interprété comme un paramètre de qualité, car les très grandes valeurs de λ correspondent à une troncature moins sévère des flux binaires du bloc codé; son inverse peut être indiqué sous la forme d'un seuil de pente de distorsion de débit.

L'ensemble N_i et les pentes S_i^k sont calculés immédiatement après le codage du bloc codé B_i , et après que l'on a mémorisé assez d'informations pour déterminer ultérieurement les points de troncature qui appartiennent à N_i et les valeurs correspondantes R_i^k et S_i^k pendant la phase d'optimisation de la distorsion de débit. Ces informations sont généralement plus petites que le flux binaire proprement dit, qui est mémorisé pour le bloc codé.

J.14.4 Estimation de distorsion efficace pour troncature R-D optimale

Les points de troncature éligibles afin de représenter chaque bloc codé dans le flux binaire imbriqué correspondent à la conclusion de chaque passe de codage. Pendant la compression, le nombre d'octets, R_i^n , requis pour représenter tous les symboles codés jusqu'à chaque point de troncature, n , ainsi que la distorsion, D_i^k , subie par troncature du flux binaire à chaque point, n , doivent toujours être évalués. En fait, l'estimation de distorsion n'est pas strictement nécessaire afin de produire un flux binaire décompressible correct, mais elle est importante pour le bon fonctionnement de l'algorithme d'optimisation de la distorsion de débit décrit dans le § J.14.3.

J.14.4.1 Considérations pour transformations irréversibles

L'algorithme d'optimisation de la distorsion de débit décrit dans le § J.14.3 dépend seulement de la mesure dans laquelle chaque passe de codage diminue la distorsion. Spécifiquement, si D_i^0 indique la distorsion subie par omission du bloc codé dans son ensemble (c'est-à-dire en mettant tous les échantillons à zéro), alors seules les différences, $D_i^n - D_i^{n-1}$, doivent être calculées pour $n = 1, 2, 3, \dots$. Il s'avère que ce calcul peut être effectué à l'aide de deux petites tables d'exploration qui ne dépendent pas de la passe de codage, du plan binaire ou de sous-bande en jeu. A titre de démonstration, soit $\omega_i \Delta_i^2$ indiquant la contribution à la distorsion dans l'image reconstruite qui résulterait d'une erreur d'un seul échelon exactement dans un même échantillon de bloc codé B_i . Ici ω_i est une pondération positive qui est calculée d'après la norme L2 de la forme d'onde de synthèse d'ondelette pour la sous-bande correspondante. Cette pondération peut en outre être modifiée de façon à refléter la pondération visuelle ou d'autres critères. L'on définit maintenant:

$$v_i^p[m, n] = 2^{-p} v_i[m, n] - 2 \left[\frac{2^{-p} v_i[m, n]}{2} \right] \quad (\text{J-14})$$

Donc $v_i^p[m, n]$ vérifie la différence normalisée entre la grandeur de l'échantillon $s_i[m, n]$ et le plus grand seuil de quantification dans le plan binaire précédent qui n'était pas plus grand que cette grandeur. Il est aisé de vérifier que $0 \leq v_i^p[m, n] \leq 2$. Bien que $s_i[m, n]$ soit en fait une grandeur d'entier quantifiée, l'on tiendra compte du fait que le quantificateur peut fournir des bits fractionnaires pour $s_i[m, n]$ et donc $v_i[m, n]$, qui peut être utilisé dans l'équation (J-14) afin de produire des estimations précises de la distorsion associée aux passes de codage dans les plans binaires les moins significatifs. Or, lorsqu'un même échantillon devient d'abord significatif dans un certain plan binaire, p , l'on doit toujours avoir $v_i[m, n] \geq 2^p$ et donc $v_i^p[m, n] \geq 1$, de sorte que la réduction en distorsion peut être exprimée par:

$$2^{2p} \omega_i \Delta_i^2 \left[v_i^p[m, n]^2 - \left(v_i^p[m, n] - 1,5 \right)^2 \right] = 2^{2p} \omega_i \Delta_i^2 f_s \left(v_i^p[m, n] \right) \quad (\text{J-15})$$

à condition que les niveaux de représentation utilisés pendant la quantification inverse soient à mi-chemin entre les seuils de quantification. De même, la réduction en distorsion qui peut être attribuée au raffinement de grandeur d'un échantillon dans un plan binaire p peut être exprimée par:

$$2^{2p} \omega_i \Delta_i^2 \left[\left(v_i^p[m, n] - 1 \right)^2 - \left(v_i^p[m, n] - 0,5 - v \right)^2 \right] = 2^{2p} \omega_i \Delta_i^2 f_m \left(v_i^p[m, n] \right) \quad (\text{J-16})$$

Donc, la réduction de la distorsion subie pendant une seule passe de codage peut être calculée par sommation des signaux de sortie d'une de deux fonctions différentes, $f_s(\cdot)$ ou $f_m(\cdot)$ selon le cas, chaque fois qu'un échantillon devient significatif ou que sa grandeur est raffinée. Puis par normalisation du résultat à la fin de la passe de codage par une valeur constante qui est facilement calculée d'après l'index de plan binaire et la valeur de $\omega_i \Delta_i^2$. L'argument appliqué à ces fonctions, $v_i^p[m, n]$, possède une représentation binaire de la forme v.xxxxx, où v, seul bit avant la virgule binaire, est simplement la valeur du bit de grandeur p , c'est-à-dire $v_i^p[m, n]$. Exactement 6 bits supplémentaires au-delà de la

virgule binaire sont utilisés pour pointer sur une table d'exploration de 7 bits pour $f_m(\cdot)$ et sur une table d'exploration de 6 bits pour $f_s(\cdot)$ (ne pas perdre de vue que l'on doit toujours avoir $1 \leq v_i^p[m, n] < 2$ lorsqu'un échantillon devient d'abord significatif). Chaque entrée de ces tables d'exploration contient une représentation en virgule fixe de $2^{13} f_s(v_i^p[m, n])$ ou $2^{13} f_m(v_i^p[m, n])$, selon le cas, ce qui signifie que la réduction totale de distorsion associée à toute passe de codage peut être calculée par cumul de ces valeurs d'entier dans un cumulateur de 32 bits, sans aucun risque de débordement.

J.14.4.2 Considérations pour transformations réversibles

Généralement, le processus d'estimation de la distorsion pendant le codage des coefficients produits par une transformation réversible n'est pas différent du processus d'une transformation irréversible. Il y a, cependant, deux différences subtiles qui doivent être précisées ici. L'équation (J-15) et l'équation (J-16) sont fondées sur l'hypothèse que le déquantificateur représentera chaque coefficient au milieu de l'intervalle de quantification correspondant. C'est le comportement le plus probable pour le quantificateur la plupart du temps, sauf pour le plan binaire le moins significatif dans le mode réversible. Dans ce cas $\Delta_i = 1$, sans erreur de quantification. La reconstruction au point milieu de l'intervalle n'est pas applicable ici et le déquantificateur représente les coefficients de transformée au moyen du seuil inférieur (en grandeur) de l'intervalle de quantification applicable. En conséquence, l'équation (J-15) et l'équation (J-16) doivent être modifiées comme suit:

$$2^{2p} \omega_i \Delta_i^2 v_i^p[m, n]^2 = 2^{2p} \omega_i \Delta_i^2 f'_m(v_i^p[m, n]) \quad (\text{J-17})$$

et:

$$2^{2p} \omega_i \Delta_i^2 (v_i^p[m, n] - 1)^2 = 2^{2p} \omega_i \Delta_i^2 f'_m(v_i^p[m, n]) \quad (\text{J-18})$$

respectivement.

J.15 Directives sur la gestion des flux codés YCC

En imagerie aussi bien fixe qu'animée pour les consommateurs, il existe un grand nombre d'applications et de dispositifs qui ne peuvent pas être envisagés sans prise en charge des composantes YCC ni sans production directe de données de chrominance sous-échantillonnées. Dans de tels cas, la signalisation de transformation multiple de composante dans le flux codé peut ne pas être nécessaire. Le présent paragraphe fournit des directives sur la façon de traiter les données YCC.

Cependant, le présent paragraphe ne vise pas à impliquer que les données YCC doivent être sous-échantillonnées.

J.15.1 Utilisation de la transformation multiple de composante

Il n'est pas nécessaire d'utiliser la transformation multiple de composante afin de prendre en charge les données YCC car les composantes sont déjà décorrélées. Donc, le signal de transformation multiple de composante du paramètre SGcod défini dans le Tableau A.17 doit toujours être "0000 0000".

J.15.2 Utilisation du format JP2

Il existe des dispositifs qui produisent automatiquement en sortie des données transformées en composantes YCC dans l'espace chromatique sYCC. Le format JP2 prend en charge ces cas en spécifiant la valeur EnumCS = "18" comme défini dans le Tableau I.10.

J.15.3 Décalage de chrominance

Le décalage de chrominance (terme courant désignant le sous-échantillonnage de composantes de chrominance et leurs décalages relatifs) est spécifié dans un flux codé JPEG 2000 au moyen du marqueur CRG (voir § A.9.1). Les Figures J.13, J.14, J.15 et J.16 montrent des exemples de structures de décalage de chrominance bien connues. Le Tableau J.26 montre un exemple de valeurs paramétriques de segments marqueurs SIZ (voir § A.5.1) et CRG pour chaque structure.

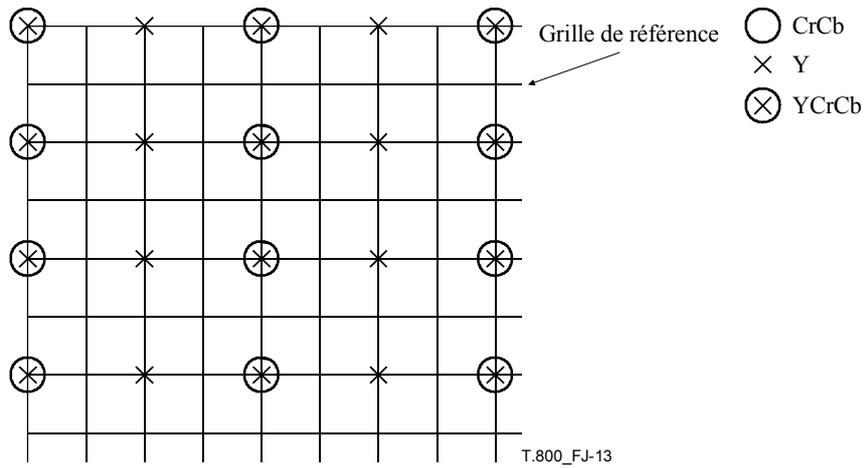


Figure J.13 – Format 4:2:2 (cositionné)

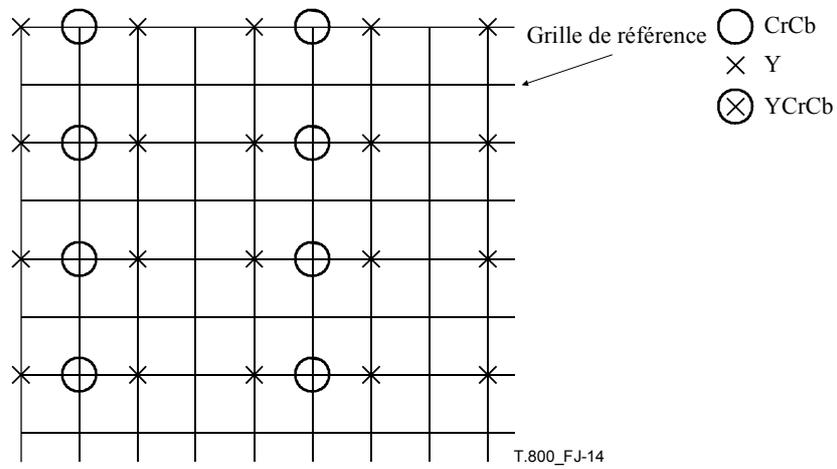


Figure J.14 – Format 4:2:2 (centré)

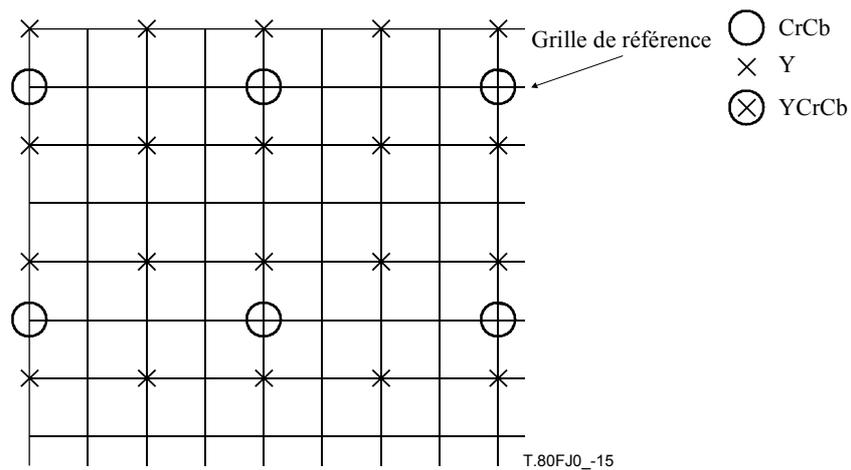


Figure J.15 – Format 4:2:0 (cositionné)

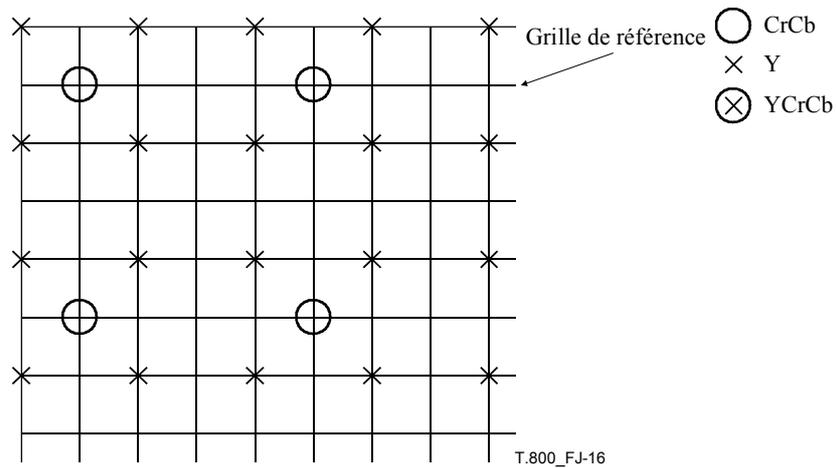


Figure J.16 – Format 4:2:0 (centré)

Tableau J.26 – Valeurs d'enregistrement de composante (CRG)

		Figure J.13	Figure J.14	Figure J.15	Figure J.16
Y	(XRsiz, YRsiz)	(2, 2)	(2, 2)	(2, 2)	(2, 2)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Cb	(XRsiz, YRsiz)	(4, 2)	(4, 2)	(4, 4)	4, 4)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(16 384, 0)	(0, 16 384)	(16 384, 16 384)
Cr	(XRsiz, YRsiz)	(4, 2)	(4, 2)	4, 4)	4, 4)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(16 384, 0)	(0, 16 384)	(16 384, 16 384)

NOTE – La valeur CRG est définie de façon que tous les échantillons de composante soient situés dans les points de la grille de référence. Donc, le fait que (XRsiz, YRsiz) = (2, 2) pour Y ne signifie pas que les échantillons de Y sont également sous-échantillonnés.

Annexe K

Bibliographie

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

K.1 Généralités

- [1] BOLIEK (M.), GORMISH (M.J.), SCHWARTZ (E.L.), KEITH (A.): Decoding compression with reversible embedded wavelets (CREW) codestreams, *Electronic Imaging*, Vol. 7, N° 3, juillet 1998.
- [2] EBRAHIMI (T.), CHRISTOPOULOS (C.): JPEG 2000 standard: Still image compression scheme of 21st century, Tampere, Finlande, 5-8 septembre 2000.
- [3] MARCELLIN (M.W.), GORMISH (M.), BILGIN (A.), BOLIEK (M.): An Overview of JPEG-2000, *Proc. of IEEE Data Compression Conference*, Snowbird, Utah, mars 2000.
- [4] SKODRAS (A.), CHRISTOPOULOS (C.), EBRAHIMI (T.): JPEG2000: The upcoming still image compression standard, (*Invited paper*) *Proceedings of the 11th Portuguese Conference on Pattern Recognition (RECPAD 2000)*, p. 359-366, Porto, Portugal, 11-12 mai 2000.
- [5] ZANDI (A.), ALLEN (J.D.), SCWHARTZ (E.L.), BOLIEK (M.): CREW: Compression with reversible embedded wavelets, *Proceedings of Data Compression Conference*, Snowbird, Utah, p. 212-221, mars 1995.

K.2 Quantification et codage entropique

- [6] ORDENTLICH (E.), WEINBERGER (M.J.), SEROUSSI (G.): A low-complexity modelling approach for embedded coding of wavelet coefficients, *Proceedings of Data Compression Conference*, p. 408-417, Snowbird, Utah, 29 mars-1^{er} avril 1998.
- [7] ORDENTLICH (E.), TAUBMAN (D.), WEINBERGER (M.J.), SEROUSSI (G.), MARCELLIN (M.): Memory Efficient Scalable Line-based Image Coding, *Proceedings of Data Compression Conference*, p. 218-227, Snowbird, Utah, 29-31 mars 1999.
- [8] PENNEBAKER (W.), MITCHELL (J.): JPEG Still Image Data Compression Standard, *Van Nostrand Reinhold*, New York, 1993.
- [9] SEMENTILLI (P.J.), BILGIN (A.), KASNER (J.H.), MARCELLIN (M.W.): Wavelet TCQ: Submission to JPEG 2000, (invited paper), *Proceedings, Applications of Digital Image Processing, SPIE*, San Diego, Californie, juillet 1998.
- [10] TAUBMAN (D.), ZAKHOR (A.): Multirate 3-D subband coding of video, *IEEE Trans. Image Processing*, Vol. 3, p. 572-88, septembre 1994.
- [11] TAUBMAN (D.): High performance scalable image compression with EBCOT, *IEEE Trans. on Image Processing*, juillet 2000.
- [12] WOODS (J.W.), NAVEEN (J.): A filter based bit allocation scheme for subband compression of HDTV, *IEEE Trans. on Image Processing*, juillet 1992.

K.3 Transformation en ondelettes

- [13] ADAMS (M.D.), KOSENTINI (F.): Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis, *IEEE Trans. on Image Processing*, 2000.
- [14] ANTONINI (M.), BARLAUD (M.), MATHIEU (P.), DAUBECHIES (I.): Image coding using wavelet transform, *IEEE Trans. on Image Processing* Vol. 1, p. 205-220, avril 1992.
- [15] BRISLAWN (C.M.): Classification of nonexpansive symmetric extension transforms for multirate filter banks, *Applied and Computational Harmonic Analysis*, Vol. 3, p. 337-57, 1996.
- [16] CALDERBANK (A.R.), DAUBECHIES (I.), SWELDENS (W.), YEO (B.-L.): Wavelet transforms that map integers to integers, *Applied and Computational Harmonic Analysis*, Vol. 5, N° 3, p. 332-369, juillet 1998.

ISO/CEI 15444-1:2004 (F)

- [17] CHRYSAFIS (C.), ORTEGA (A.): Line-Based, Reduced Memory, Wavelet Image Compression, *IEEE Trans. on Image Processing*, Vol. 9, N° 3, p. 378-389, mars 2000.
- [18] CHUI (C.K.): An Introduction to Wavelets, *Academic Press*, Boston, 1992.
- [19] CHUI (C.K.): Wavelets: A Mathematical Tool for Signal Analysis, *SIAM Publ.*, Philadelphie, 1997.
- [20] DAUBECHIES (I.): Ten Lectures on Wavelets, *SIAM Publ.*, Philadelphie, 1992.
- [21] DAUBECHIES (I.), SWELDENS (W.): Factoring wavelet transforms into lifting steps, *Journal of Fourier Analysis and Applications*, Vol. 4, N° 3, p. 247-269, 1999.
- [22] LEGALL (D.), TABATABAI (A.): Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques, *IEEE International Conference on Acoustics, Speech and Signal Processing*, New York, NY, p. 761-765, 1988.
- [23] MALLAT (S.): A Wavelet Tour of Signal Processing, *Academic Press*, 2nd Edition, San Diego, 1999.
- [24] SHENG (F.), BILGIN (A.), SEMENTILLI (P.J.), MARCELLIN (M.W.): Lossy and lossless image compression using reversible integer wavelet transforms, *Proceedings, IEEE International Conference on Image Processing*, Chicago, Illinois, octobre 1998.
- [25] SWELDENS (W.): The lifting scheme: a custom-design construction of biorthogonal wavelets, *Applied and Computational Harmonic Analysis*, Vol. 3, N° 2, p. 186-200, 1996.
- [26] SWELDENS (W.): The lifting scheme: A construction of second generation wavelets, *SIAM Journal on Mathematical Analysis*, Vol. 29, N° 2, p. 511-546, 1997.

K.4 Codage de région intéressante

- [27] CHRISTOPOULOS (C.), ASKELOF (J.), LARSSON (M.): Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard, *IEEE Signal Processing letters*, septembre 2000.
- [28] ATSUMI (E.), FARVARDIN (N.): Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees, *Proc. IEEE International Conference on Image Processing (ICIP-98)*, p. 87-91, 4-7 octobre 1998, Chicago, Illinois.
- [29] NISTER (D.), CHRISTOPOULOS (C.): Lossless Region of Interest with a naturally progressive still image coding algorithm, *Proceedings of IEEE International Conference on Image Processing (ICIP 98)*, p. 856-860, 4-7 octobre 1998, Chicago, Illinois,
- [30] NISTER (D.), CHRISTOPOULOS (C.): Lossless region of interest with embedded wavelet image coding, *Signal Processing*, Vol. 78, N° 1, p. 1-17, octobre 1999.
- [31] SANTA CRUZ (D.), LARSSON (M.), ASKELOF (J.), EBRAHIMI (T.), CHRISTOPOULOS (C.): Region of Interest coding in JPEG2000 for interactive client/server applications, *IEEE International Workshop on Multimedia Signal Processing*, Copenhague, Danemark, 13-15 septembre 1999.

K.5 Pondération de fréquence visuelle

- [32] ALBANESI (M.), BERTOLUZZA (S.): Human vision model and wavelets for high-quality image compression, *Proc. of 5th Int. Conference in Image Processing and its Applications*, Edimbourg, Royaume-Uni, N° 410, p. 311-315, 4-6 juillet 1995.
- [33] ECKERT (M.): Lossy compression using wavelets, block DCT, and lapped orthogonal transforms optimised with a perceptual model, *SPIE*, Vol. 3031, p. 339-350, 1997.
- [34] O'ROURKE (T.), STEVENSON (R.): Human visual system based wavelet decomposition for image compression, *J. VCIP V. 6*, p. 109-121, 1995.
- [35] WATSON (A.B.), YANG (G.), SOLOMON (J.), VILLASENOR (J.): Visibility of wavelet quantization noise, *IEEE Trans. on Image Processing*, Vol. 6, p. 1164-1175, 1997.

K.6 Elasticité aux erreurs

- [36] MAN (H.), KOSENTINI (F.), SMITH (M.): A Family of Efficient and Channel Error Resilient Wavelet/Subband Image Coders, *Special issue of the IEEE Transactions on Circuits and Systems for Video Technology on Interactive Multimedia*, 9(2), février 1999.

- [37] LIANG (J.), TALLURI (R.): Tools for Robust Image and Video Coding in JPEG2000 and MPEG-4 Standards, *Proceedings of SPIE Visual Communications and Image Processing Conference (VCIP)*, janvier 1999, San Jose, CA.
- [38] MOCCAGATA (I.), SODAGAR (S.), LIANG (J.), CHEN (H.): Error Resilient Coding in JPEG-2000 and MPEG-4, *IEEE Journal of Selected Areas in Communications (JSAC)*, juin 2000.

K.7 Codage par exploration

- [39] LAMBERT-NEBOUT (C.), LATRY (C.), MOURY (G.), ANTONINI (M.), BARLAUD (M.), PARISOT (C.): On-Board Optical image compression for future high resolution space remote sensing systems, *Proc. SPIE*, Vol. 4115, 31 juillet-3 août 2000.
- [40] FLOHR (T.J.), MARCELLIN (M.W.), ROUNTREE (J.C.): Scan-Based Processing with JPEG 2000, *Proc. SPIE*, Vol. 4115, 31 juillet-3 août 2000.

K.8 Couleur

- [41] International Color Consortium (ICC), *ICC Profile Format Specification 1:1998-09*, (1998).
- [42] CEI 61966-2-1 (1999), *Mesure et gestion de la couleur dans les systèmes et appareils multimédia – Partie 2-1: gestion de la couleur – Espace chromatique RVB par défaut – sRVB*.
- [43] LAM (K.M.): Metamerism and Colour Constancy, *University of Bradford*, 1985.
- [44] LUO (M.R.), LO (M.C.), KUO (W.G.): The LLab(l:c) Colour Model, *Color Research and Application*, Vol. 21, p. 412-429 (1996).
- [45] LUO (M.R.), HUNT (R.W.G.): A Chromatic Adaptation Transform and a Colour Constancy Index, *Color Research and Application*, Vol. 23, p. 154-158, 1997.
- [46] LUO (M.R.), HUNT (R.W.G.): The Structure of the CIE 1997 Colour Appearance Model (CIECAM97s), *Color Research and Application*, Vol. 23, 1997.
- [47] NIELSEN (M.), STOKES (M.): The Creation of the sRGB ICC Profile, *Proceedings of the IS&T/SID 6th Color Imaging Conference*, p. 253-257, 1998.
- [48] GIORGIANNI (E.), MADDEN (T.): Digital Color Management: Encoding Solution, *Addison-Wesley*, 1998.
- [49] SPAULDING (K.E.), WOOLFE (G.J.), GIORGIANNI (E.J.): Reference Input/Output Medium Metric RGB color encodings (RIMM/ ROMM RGB), *Proc. PICS 2000 Conference*, 26-29 mars, 2000, Portland, OR.

Annexe L**Déclaration relative aux brevets**

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

L'Organisation Internationale de Normalisation (ISO, *International Organization for Standardization*) et la Commission électrotechnique internationale (CEI) attirent l'attention sur la revendication que la conformité à la présente Recommandation | Norme internationale peut impliquer l'utilisation de brevets, comme indiqué dans le Tableau L.1.

Tableau L.1 – Revendications de droits de propriété intellectuelle reçues

Numéro	Compagnie
1	Algo Vision plc
2	Canon Incorporated
3	Digital Accelerator Corporation
4	Telefonaktiebolaget LM Ericsson
5	Hewlett Packard Company
6	International Business Machines, Inc.
7	LizardTech, Incorporated
8	LuraTech
9	Mitsubishi Electric Corporation
10	PrimaComp Incorporated
11	Ricoh Company, Limited
12	Sarnoff Corporation
13	Sharp Corporation
14	TeraLogic Incorporated
15	University of Arizona
16	Washington State University
17	Sony Corporation

L'ISO et la CEI ne prennent aucune position concernant l'évidence, la validité et le domaine d'application de ces droits de brevet.

Les détenteurs de ces droits de brevet ont assuré l'ISO et la CEI qu'ils étaient disposés à négocier des concessions à des termes et conditions raisonnables et non discriminatoires avec des requérants dans le monde entier. A ce propos, les revendications des détenteurs de ces droits de brevets sont enregistrées auprès de l'ISO et de la CEI.

L'attention est attirée sur la possibilité que certains éléments de la présente Recommandation | Norme internationale fassent l'objet de droits de brevet autres que ceux qui sont indiqués ci-dessus. L'ISO et la CEI ne doivent pas être tenus responsables de l'indication de tout ou partie de ces droits de brevet.

SÉRIES 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	Réseaux câblés et 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, circuits téléphoniques, télégraphie, télécopie 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 de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication