



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.82

(03/93)

**ÉQUIPEMENTS TERMINAUX ET PROTOCOLES
POUR LES SERVICES DE TÉLÉMATIQUE**

**TECHNOLOGIE DE L'INFORMATION –
REPRÉSENTATION CODÉE DES IMAGES
ET DU SON – COMPRESSION PROGRESSIVE
DES IMAGES EN DEUX TONS**

Recommandation UIT-T T.82

(Antérieurement «Recommandation du CCITT»)

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. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT) (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie 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. Le texte de la Recommandation T.82 de l'UIT-T a été approuvé par la CMNT (Helsinki, 1-12 mars 1993). Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 11544.

NOTES

1 A la suite de la restructuration de l'Union internationale des télécommunications (UIT), le CCITT a cessé d'exister le 28 février 1993. Le Secteur de la normalisation des télécommunications de l'UIT fut créé en son lieu et place le 1^{er} mars 1993. Au cours de cette restructuration, le CCIR et l'IFRB ont été également remplacés par le Secteur des radiocommunications de l'UIT.

Afin de ne pas retarder la publication de la présente Recommandation, aucune correction n'a été apportée dans le texte aux références contenant les expressions CCITT, CCIR ou IFRB ou le nom de leurs entités connexes telles que Assemblée plénière, Secrétariat spécialisé, etc. Les prochaines versions de la présente Recommandation utiliseront la terminologie appropriée relative à la nouvelle structure de l'UIT.

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

© UIT 1994

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

TABLE DES MATIÈRES

		<i>Page</i>
Intro. 1	Caractéristiques générales	i
Intro. 2	Ordre des bandes et des données	ii
Intro. 3	Blocs fonctionnels du codeur	iv
Intro. 3.1	Réduction de résolution et codage de couche différentielle.....	v
Intro. 3.2	Codeur de la couche de plus basse résolution	vii
Intro. 4	Blocs fonctionnels du décodeur.....	viii
1	Domaine d'application.....	1
2	Références normatives	1
3	Définitions.....	1
4	Symboles et abréviations.....	3
4.1	Sigles.....	3
4.2	Constantes symboliques.....	3
4.3	Symboles mathématiques, opérateurs et indicateurs.....	3
4.4	Variables de noms mnémoniques.....	3
5	Conventions.....	4
5.1	Conventions et symboles relatifs aux organigrammes.....	4
5.2	Représentation graphique des gabarits.....	4
5.3	Phases spatiales.....	5
5.4	Graphiques de structure des données	6
6	Spécifications	7
6.1	Règles générales.....	7
6.2	Organisation des données	8
6.3	Réduction de résolution	13
6.4	Prédiction typique de couche différentielle.....	14
6.5	Prédiction typique dans la couche de résolution la plus basse.....	16
6.6	Prédiction déterministe (DP).....	20
6.7	Gabarits modèles et gabarits adaptatifs.....	24
6.8	Codage arithmétique	27
7	Exemples de méthodes et de suites de données de test	44
7.1	Codage arithmétique	45
7.2	Algorithme paramétré	52
7.3	Exemples de suites de données	56
Annexe A	– Plages minimales suggérées pour les paramètres libres	57
Annexe B	– Conception de la table de réduction de résolution.....	58
B.1	Filtrage.....	58
B.2	Exceptions.....	59
Annexe C	– Changements de gabarit adaptatif.....	61
C.1	Considérations générales.....	61
C.2	Couches différentielles.....	61
C.3	Couche de plus basse résolution	61

Annexe D – Conception de la table d'estimation de probabilité.....	64
D.1 Estimation bayésienne.....	64
D.2 Environnements multicontextes	64
D.3 Paramétrage MPS/LPS.....	64
D.4 Réponse rapide.....	65
D.5 Réduction de la charge de calcul.....	65
Annexe E – Brevets	69
E.1 Remarques d'introduction	69
E.2 Liste des brevets.....	69
E.3 Adresses à contacter pour des renseignements sur les brevets.....	70
Annexe F – Bibliographie	72

Introduction et vue d'ensemble

(Cette introduction ne fait pas partie de la présente Recommandation | Norme internationale)

La présente Recommandation | Norme internationale a été élaborée par le Groupe mixte d'experts sur les images en deux tons JBIG (*joint bi-level image expert group*) du groupe de travail ISO/CEI JTC1/SC29/WG9 et de la Commission d'études SG VIII du CCITT. Le groupe d'experts JBIG a été constitué en 1988 pour élaborer une norme sur le codage progressif des images en deux tons.

Un système de codage progressif transmet une image comprimée en envoyant d'abord les données comprimées d'une version de l'image à résolution réduite: cette image comprimée est ensuite améliorée en fonction des besoins, en transmettant des données comprimées additionnelles, qui complètent celles qui ont déjà été transmises. La présente Recommandation | Norme internationale définit une méthode de codage comportant des modes progressif, séquentiel compatible progressif et séquentiel simple. Elle suggère une méthode permettant d'obtenir n'importe quelle restitution à basse résolution. Il a été constaté qu'il est possible d'utiliser de façon efficace les algorithmes de codage et de réduction de résolution ici définis, pour le codage sans perte d'images en niveaux de gris et en couleurs, aussi bien que d'images en deux tons.

L'article «Introduction et vue d'ensemble», ainsi que les Annexes A à F sont informatifs et ne font donc pas partie intégrante de la présente Recommandation | Norme internationale.

Intro. 1 Caractéristiques générales

La présente Spécification définit une méthode de codage à compression sans perte d'une image en deux tons (c'est-à-dire d'une image comportant seulement deux couleurs, comme une image noir et blanc). La méthode définie peut également être utilisée pour le codage d'images en niveaux de gris et en couleurs. Cette méthode peut s'adapter aux caractéristiques de l'image: elle peut donc être utilisée pour traiter divers types d'images. Sur les images obtenues par balayage de caractères imprimés, les taux de compression observés sont 1,1 à 1,5 fois supérieurs à ceux réalisés par l'algorithme de codage MMR (*modified modified Read*), moins complexe, décrit dans les Recommandations T.4 (G3) et T.6 (G4). Sur des images de caractères imprimés générées par ordinateur, les taux de compression observés peuvent être au moins 5 fois supérieurs. Sur des images à niveaux de gris, restituées en demi-teintes ou en tramés, les taux de compression observés sont de 2 à 30 fois supérieurs.

Comme celle des Recommandations T.4 et T.6, cette méthode est sans perte: elle n'introduit aucune distorsion, l'image décodée finale est identique à l'original.

La méthode présente également une capacité «progressive»: lors du décodage d'une image à codage progressif, une restitution à basse résolution de l'image d'origine est d'abord disponible, des doublements de résolution étant ensuite opérés au fur et à mesure du décodage des données. A noter que la réduction de résolution est opérée des couches à haute résolution vers les couches à basse résolution, alors que le décodage est opéré des couches à basse résolution vers les couches à résolution plus élevée. L'image de résolution la plus basse, transmise sous la forme d'une séquence progressive, est une image à codage séquentiel. Dans une application à codage séquentiel simple, c'est la seule image transmise.

Les codages progressifs présentent deux avantages distincts. Le premier est qu'il est possible de concevoir une application qui peut desservir de façon efficace, à partir d'une base de données commune, des dispositifs de sortie de résolutions très différentes. Seule est transmise et décodée la partie du fichier de l'image comprimée nécessaire à la reconstruction à la résolution du dispositif de sortie utilisé. En outre, si une amélioration de résolution est désirée, par exemple pour une sortie papier d'une image déjà affichée sur écran, il suffit de transmettre les informations nécessaires à améliorer la résolution.

L'autre avantage des codages progressifs est qu'ils peuvent procurer, lors de la consultation d'images par défilement sur un écran, une qualité subjective plus élevée, lorsque l'application utilise des voies de communication à faible ou moyen débit. La transmission et l'affichage rapide d'une image peuvent être suivis d'autant d'améliorations de résolution que désiré. Chaque stade d'amélioration de résolution construit une image immédiatement disponible. Le codage progressif peut faciliter la reconnaissance rapide d'une image, lors de sa construction, par un utilisateur, qui peut interrompre la transmission, dès qu'il a reconnu l'image.

Soit D le nombre de doublements de résolution (appelés couches différentielles) fournis par le codage progressif, soit I_D , la résolution d'image la plus élevée et X_D et Y_D , les dimensions horizontale et verticale en pixels. Soit R_D la résolution d'échantillonnage de l'image I_D .

La présente Spécification n'impose presque aucune restriction aux paramètres R_D , X_D , Y_D et D . Les choix tels que 400 ou 200 points par pouce (dpi) (*dots-per-inch*) pour la résolution R_D de la couche de plus haute résolution permettent de disposer d'une hiérarchie de résolutions correspondant aux normes actuelles de télécopie. Le choix de 600 ou 300 dpi pour R_D donne une hiérarchie progressive plus compatible avec les résolutions d'imprimantes à laser disponibles lors de la rédaction de la présente Spécification.

Il est prévu que D sera généralement choisi en sorte que la résolution la plus basse soit de l'ordre de 10 à 25 dpi. Ramenées à cette résolution, des images typiques en deux tons ne sont pas lisibles. Des restitutions à une telle résolution sont néanmoins très utiles comme icônes à génération automatique. L'organisation des pages est en général apparente et il est souvent possible de reconnaître les pages qui ont été visualisées auparavant dans une résolution plus élevée.

Comme mentionné plus haut, la présente Spécification ne restreint pas le nombre D de doublements de résolution. D peut être mis à 0, si le codage progressif n'est pas nécessaire, comme dans le cas, par exemple, d'une télécopie sur papier. On conserve ainsi le gain de compression offert par JBIG par rapport à celui de la méthode MMR (en fait, on l'augmente un peu), en évitant la nécessité d'une mémoire tampon et en simplifiant l'algorithme. Le codage JBIG séquentiel simple a des applications potentielles identiques à celles du codage MMR. Les images comprimées par un codeur séquentiel simple seront lisibles par des décodeurs capables d'effectuer un décodage progressif, alors que seule la version à résolution la plus basse d'une image à codage progressif est décodable par un décodeur séquentiel simple.

Il est possible d'utiliser la présente Spécification pour le codage sans perte d'images en niveaux de gris et en couleurs, en codant indépendamment les plans binaires, comme si chacun de ces plans était lui-même une image en deux tons. Cette méthode de codage des images en niveaux de gris et en couleurs, peut être utilisée pour remplacer le mode sans perte du codage photographique spécifié dans la Rec. T.81 du CCITT | ISO/CEI 10918-1 (JPEG). Jusqu'à 6 bits par pixel, des résultats expérimentaux préliminaires montrent que JBIG présente, pour des images en niveaux de gris, un gain de compression par rapport au mode sans perte de JPEG. De 6 à 8 bits par pixel, les taux de compression sont similaires pour JBIG et JPEG. La présente Spécification peut s'appliquer à des images comportant plusieurs plans binaires, mais ne formule aucune prescription quant à la mise en correspondance des intensités en niveaux de gris ou en couleurs, avec les plans binaires. On a pu constater expérimentalement, pour des images en niveaux de gris, qu'une mise en correspondance basée sur un codage de Gray de l'intensité est meilleure qu'une mise en correspondance basée sur un codage binaire à pondération simple.

Intro. 2 Ordre des bandes et des données

Lorsqu'il est nécessaire de distinguer le codage progressif de la forme de codage d'images plus classique, dans laquelle l'image est codée à pleine résolution de gauche à droite et de haut en bas, cette forme de codage plus ancienne est appelée codage «séquentiel». L'avantage du codage séquentiel sur le codage progressif est qu'aucune mémoire tampon de page (de trame) n'est nécessaire. A une résolution juste inférieure à la résolution maximale, le codage progressif nécessite une mémoire de page, car le codage d'une image utilise alors des images de résolution plus basse.

Il est possible de créer un train de données JBIG à partir de la seule couche de plus basse résolution: ce mode est appelé codage séquentiel simple. Dans ce mode de codage, une image en pleine résolution est codée sans référence à aucune couche différentielle de résolution. Le paramètre D (mentionné en Intro. 1) est mis à zéro. A noter que dans le codage progressif d'une image, la couche de plus basse résolution est en fait codée en mode séquentiel simple. Si une image a été codée en pleine résolution en mode séquentiel simple, il n'est pas possible de la décoder progressivement.

Le codage en mode séquentiel compatible progressif est dit «compatible» avec le codage en mode progressif, parce que les trains de données créés (codeurs) ou lus (décodeurs) dans les deux modes, véhiculent exactement les mêmes informations. Lorsque l'on passe du codage progressif au codage séquentiel compatible progressif, seul change l'ordre de création des parties des données comprimées par le codeur. Lorsqu'on passe du décodage progressif au décodage séquentiel compatible progressif, seul change l'ordre d'utilisation de ces parties par le décodeur.

Cette compatibilité est obtenue en décomposant l'image, avant compression, en segments plus petits. Ces segments sont créés en divisant chacune des couches de résolution de l'image en «bandes» horizontales. Le codage séquentiel compatible progressif nécessite l'utilisation d'une mémoire tampon de «bande» (beaucoup plus petite qu'une mémoire de page) et une certaine capacité de mémoire «d'état» individuelle, pour le codage entropique adaptatif de chaque couche de résolution et de chaque plan binaire.

La Figure Intro. 1 montre une telle décomposition, avec trois couches de résolution, trois bandes par couche, et un seul plan binaire. Le Tableau Intro. 1 montre divers enchaînements possibles des neuf bandes.

Le bit SEQ précise si le codage est progressif ou séquentiel; le bit HITOLO indique l'ordre de résolution. Les codeurs fonctionnant par résolution décroissante, le plus naturel est de coder les bandes dans l'ordre de HITOLO. Il est naturel, pour les décodeurs qui doivent construire l'image par résolution croissante, de traiter les bandes dans l'ordre opposé. Lorsqu'une application utilise un codeur qui envoie progressivement des données codées directement à un décodeur, le codeur ou le décodeur doit utiliser une mémoire tampon pour inverser l'ordre des données. Lorsqu'une application comprend une base de données, cette dernière (complétée d'une installation appropriée) peut être utilisée comme mémoire tampon pour inverser l'ordre (en mettant HITOLO à la valeur correcte): il n'est alors plus nécessaire de prévoir une mémoire tampon au niveau du codeur ou du décodeur.

La taille verticale d'une bande est généralement plus petite que celle de l'image complète. Le nombre L_0 de lignes par bande dans la couche la plus basse est un autre paramètre libre. Par exemple, L_0 doit être choisi pour avoir une bande d'environ 8 mm. Dans ce cas, le nombre S de bandes d'une image d'une feuille de papier format A4 serait d'environ 35.

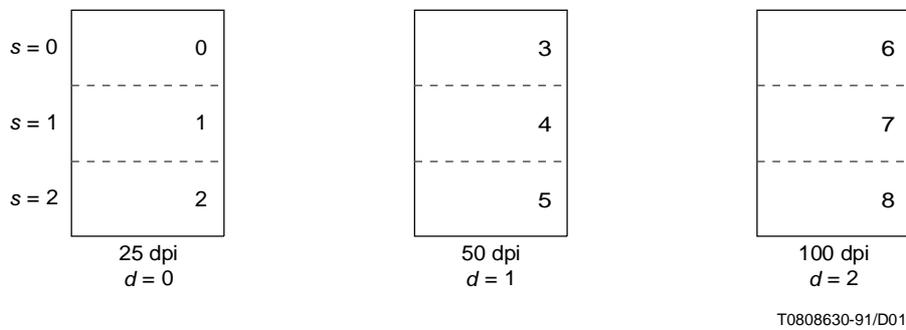


Figure Intro. 1 – Décomposition, dans le cas particulier de trois couches, trois bandes et un plan binaire

Tableau Intro. 1 – Ordres possibles des données en deux tons

HITOLO	SEQ	Exemples d'ordre
0	0	0, 1, 2 3, 4, 5 6, 7, 8
0	1	0, 3, 6 1, 4, 7 2, 5, 8
1	0	6, 7, 8 3, 4, 5 0, 1, 2
1	1	6, 3, 0 7, 4, 1 8, 5, 2

Avec plusieurs plans binaires (Figure Intro. 2), douze ordres de bande, donnés dans le Tableau Intro. 2, sont définis. Comme plus haut, le bit HITOLO indique l'ordre de résolution et le bit SEQ précise si le codage est progressif ou séquentiel. Le bit ILEAVE mis à 1 indique l'entrelacement de plusieurs plans binaires. Le bit SMID mis à 1 indique que l'indice s , relatif à la bande, est l'indice central, comme cela apparaît plus clairement dans le Tableau 11, en 6.2.4.

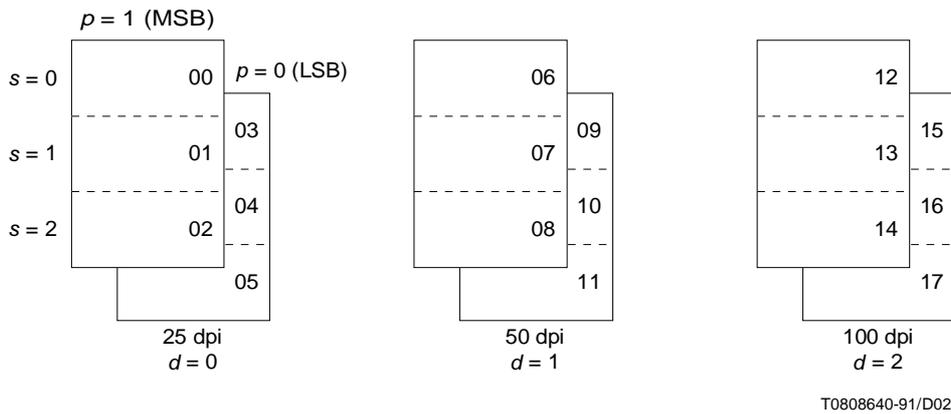


Figure Intro. 2 – Décomposition, dans le cas particulier de trois couches, trois bandes et deux plans binaires

Tableau Intro. 2 – Ordres possibles des données dans le cas de plusieurs plans

HITOLO	SEQ	ILEAVE	SMID	Exemples d'ordre
0	0	0	0	(00,01,02 06,07,08 12,13,14) (03,04,05 09,10,11 15,16,17)
0	0	1	0	(00,01,02 03,04,05) (06,07,08 09,10,11) (12,13,14 15,16,17)
0	0	1	1	(00,03 01,04 02,05) (06,09 07,10 08,11) (12,15 13,16 14,17)
0	1	0	0	(00,06,12 03,09,15) (01,07,13 04,10,16) (02,08,14 05,11,17)
0	1	0	1	(00,06,12 01,07,13 02,08,14) (03,09,15 04,10,16 05,11,17)
0	1	1	0	(00,03 06,09 12,15) (01,04 07,10 13,16) (02,05 08,11 14,17)
1	0	0	0	(12,13,14 06,07,08 00,01,02) (15,16,17 09,10,11 03,04,05)
1	0	1	0	(12,13,14 15,16,17) (06,07,08 09,10,11) (00,01,02 03,04,05)
1	0	1	1	(12,15 13,16 14,17) (06,09 07,10 08,11) (00,03 01,04 02,05)
1	1	0	0	(12,06,00 15,09,03) (13,07,01 16,10,04) (14,08,02 17,11,05)
1	1	0	1	(12,06,00 13,07,01 14,08,02) (15,09,03 16,10,04 17,11,05)
1	1	1	0	(12,15 06,09 00,03) (13,16 07,10 01,04) (14,17 08,11 02,05)

Les deux nouvelles variables ILEAVE et SMID ainsi que les deux variables déjà définies HITOLO et SEQ permettent d'indexer les douze ordres. Aucun ordre de bande n'est attaché aux quatre autres combinaisons (sur les seize possibles) de ces quatre variables binaires. S'il n'y a qu'un seul plan, l'ordre des bandes ne dépend pas des variables ILEAVE et SMID, dont les valeurs sont ignorées.

Les données comprimées $C_{s,d,p}$ de la bande s de la couche de résolution d du plan binaire p sont indépendantes de l'ordre des bandes. Tous les changements de valeur d'HITOLO, de SEQ, d'ILEAVE et de SMID évoluent dans l'ordre de concaténation des données au train de données. Ainsi se manifeste la caractéristique de compatibilité signalée plus haut.

Pour simplifier, la suite de cet article d'introduction suppose un seul plan binaire, l'indice p , qui désigne le plan binaire, étant supprimé de $C_{s,d,p}$.

Intro. 3 Blocs fonctionnels du codeur

Un codeur peut être décomposé comme montré sur la Figure Intro. 3. (En codage séquentiel simple, seul le codeur de la couche de résolution la plus basse serait utilisé.)

Conceptuellement, le codeur se décompose en D codeurs de couche différentielle, d'algorithme identique, comme montré Figure Intro. 3: en pratique, on peut choisir pour certaines réalisations de n'utiliser, de façon récursive, qu'un seul codeur de couche différentielle physique.

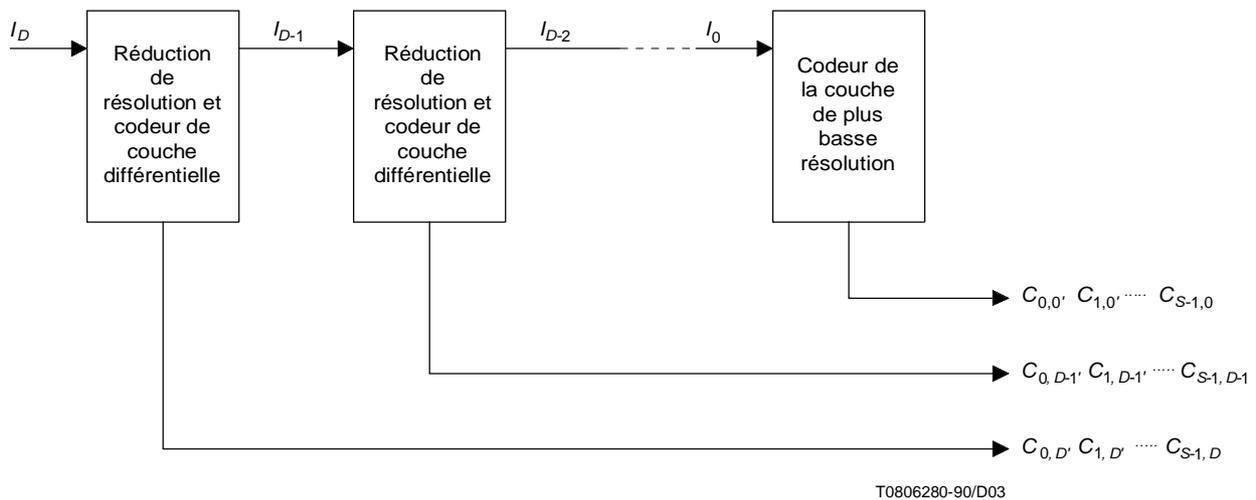


Figure Intro. 3 – Décomposition du codeur

Intro. 3.1 Réduction de résolution et codeur de couche différentielle

Tous les blocs «réduction de résolution et codeur de couche différentielle» de la Figure Intro. 3 ayant une fonction identique, il suffit de décrire cette fonction pour une couche. La description fait intervenir seulement deux couches de résolution. Pour simplifier, l'image entrante sera appelée, dans la suite de ce paragraphe, image à «haute résolution», et l'image sortante, image à «basse résolution». A noter que les images à «haute» et «basse» résolution d'un bloc «réduction de résolution et codage de couche différentielle» déterminé de la Figure Intro. 3 ne sont en général pas les images de résolution respectivement la plus haute et la plus basse du système complet.

Un bloc «réduction de résolution et codage de couche différentielle» de la Figure Intro. 3 peut lui-même être décomposé en sous-blocs, comme montré sur la Figure Intro. 4. Tous ces sous-blocs ne sont pas forcément utilisés dans tous les systèmes. (Se reporter aux tableaux en 4 pour une définition des noms des signaux.)

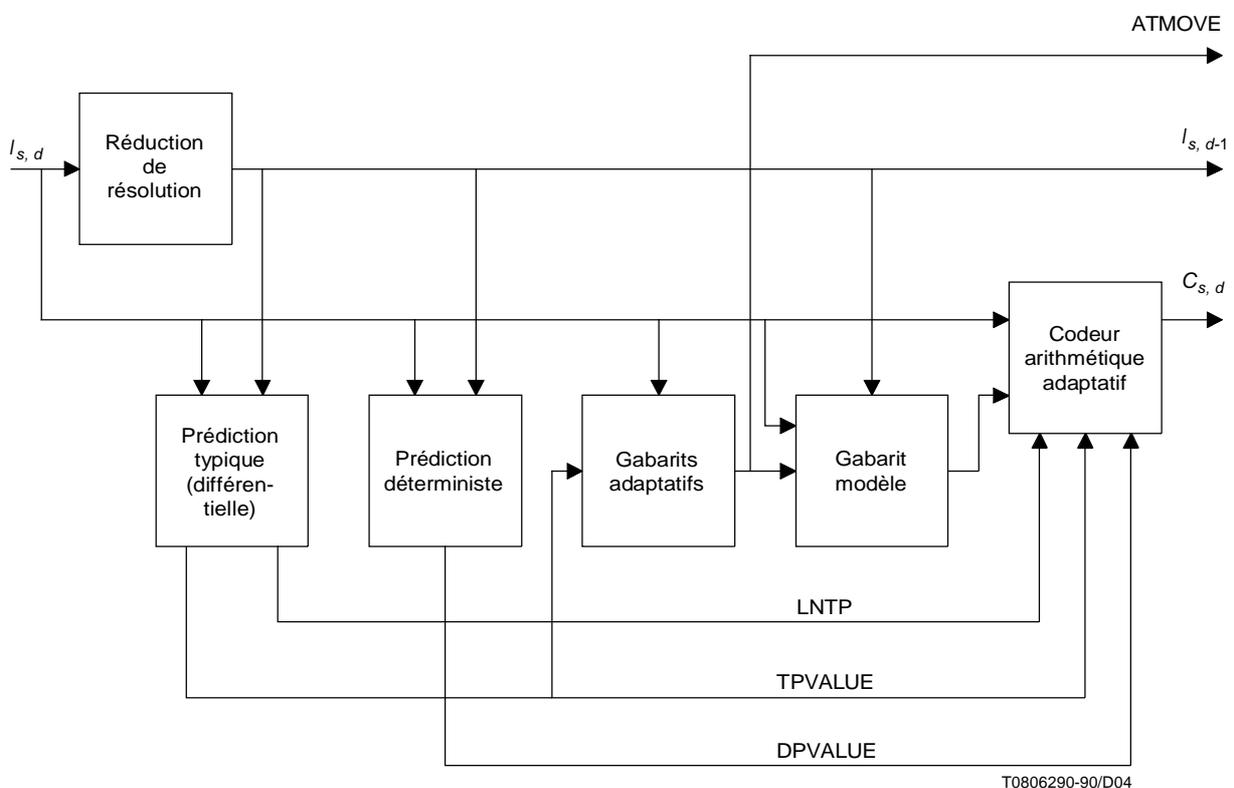


Figure Intro. 4 – Réduction de résolution et codeur de couche différentielle

Le Tableau Intro. 3 donne la signification des sigles des blocs de traitement de cette figure, ainsi que de certains autres traités dans cet article.

Tableau Intro. 3 – Sigles des blocs de traitement

Sigle	Signification
AAD	Décodeur arithmétique adaptatif (<i>adaptive arithmetic decoder</i>)
AAE	Codeur arithmétique adaptatif (<i>adaptive arithmetic encoder</i>)
AT	Gabarit(s) adaptatif(s) (<i>adaptive templates</i>)
DP	Prédiction déterministe (<i>deterministic prediction</i>)
MT	Gabarit modèle (<i>model templates</i>)
RR	Réduction de résolution (<i>resolution reduction</i>)
TPB	Prédiction typique (basse) [<i>typical prediction (bottom)</i>]
TPD	Prédiction typique (différentielle) [<i>typical prediction (differential)</i>]

Intro. 3.1.1 Réduction de résolution

Le bloc RR effectue une réduction de résolution. Il accepte une image à haute résolution, pour créer une image à basse résolution comportant un nombre de rangées et un nombre de colonnes aussi voisins que possible de la moitié de ceux des rangées et colonnes d'origine.

Une façon évidente de réduire la résolution d'une image donnée d'un facteur deux, dans les deux dimensions, est de la sous-échantillonner, en prenant une rangée sur deux et une colonne sur deux. Ce sous-échantillonnage est simple, mais produit des images d'une qualité subjective médiocre, notamment lorsque l'image d'entrée est en deux tons.

Pour les images en deux tons contenant du texte et des dessins au trait, le sous-échantillonnage donne un résultat médiocre, car il supprime souvent des traits fins. Le résultat est également médiocre pour les images en deux tons qui contiennent des demi-teintes ou des tramés ordonnés pour restituer les niveaux de gris, car les gris ne sont pas bien conservés, spécialement si la période de trame est une puissance de deux, ce qui est fréquemment le cas.

La présente Spécification suggère une méthode de réduction de résolution. Cette méthode a été conçue avec soin. Elle a fait l'objet de tests approfondis et s'est avérée donner d'excellents résultats pour les textes, les dessins au trait, les niveaux de gris tramés, les niveaux de gris en demi-teintes, ainsi que les niveaux de gris à diffusion d'erreur.

Intro. 3.1.2 Prédiction typique de couche différentielle

Le bloc «prédiction typique de couche différentielle» (TP) procure un certain gain de codage, mais son objet est essentiellement d'accélérer les mises en œuvre. L'algorithme de prédiction typique de couche différentielle recherche les régions de couleur uniforme: si un pixel de haute résolution en cours de codage appartient à une telle région, aucun des traitements normalement effectués dans les blocs DP, AT, MT et AAE n'est alors nécessaire. Sur les images de textes ou de dessins au trait, la TP de couche différentielle permet en général d'éviter le codage de plus de 95% des pixels. Sur les images en deux tons restituant des niveaux de gris, les économies de traitement sont nettement moindres.

Intro. 3.1.3 Prédiction déterministe

L'objet du bloc «prédiction déterministe» (DP) est de procurer un gain de codage. Sur un des jeux d'images de test utilisés pour l'élaboration de la présente Spécification, un gain de 7% a été constaté: cette valeur est considérée comme typique.

Lorsque la résolution d'image est réduite au moyen d'un algorithme déterminé, il arrive souvent que la valeur du pixel de haute résolution à coder puisse être déduite de pixels déjà connus du codeur et du décodeur, c'est-à-dire de pixels de l'image basse résolution et de l'image haute résolution qui ont une relation de causalité avec le pixel courant (pour un sens donné de balayage). On dit alors que le pixel courant peut faire l'objet d'une prédiction déterministe. Le bloc DP marque les pixels de ce type, pour empêcher leur traitement par le codeur arithmétique.

La prédiction déterministe est un algorithme piloté par table. Les valeurs de pixels particuliers entourant le pixel à coder et appartenant à l'image basse résolution et à l'image «causale» haute résolution, sont utilisées pour pointer vers une table, afin de vérifier le caractère déterministe de ces valeurs et, dans l'affirmative, d'obtenir la prédiction déterministe. Les tables DP dépendent fortement de la méthode de réduction de résolution utilisée. Lorsqu'un codeur utilise un algorithme de réduction de résolution privé, il est prévu qu'il télécharge des tables DP dans le décodeur. Si une application utilise l'algorithme DP par défaut, les décodeurs doivent toujours disposer des tables DP par défaut, et aucune table DP n'est transmise, ce qui est le cas pour l'algorithme de réduction de résolution suggéré dans la présente Spécification.

Intro. 3.1.4 Gabarit modèle

Pour chaque pixel de haute résolution à coder, le bloc MT «gabarit modèle» fournit au codeur arithmétique un entier appelé «contexte». Cet entier est déterminé par les couleurs (niveaux binaires) de pixels particuliers de l'image de prédiction à haute résolution, par des pixels particuliers de l'image à basse résolution déjà disponible et par la phase spatiale du pixel en cours de codage. La «phase spatiale» détermine l'orientation du pixel de haute résolution par rapport au pixel de basse résolution correspondant.

Le codeur arithmétique conserve pour chaque contexte une estimation de la probabilité conditionnelle du symbole attaché à ce contexte. Le gain de codage le plus élevé est réalisé lorsque cette estimation de probabilité est à la fois précise et voisine de 0 ou 1. Ainsi, les bons gabarits offrent une bonne valeur de prédiction, de telle sorte que lorsque les valeurs de leurs pixels sont connues, la valeur du pixel à coder est hautement prévisible.

Intro. 3.1.5 Gabarits adaptatifs

Le bloc AT «gabarits adaptatifs» procure un gain substantiel de codage (atteignant parfois 80%) sur des images restituant les niveaux de gris en demi-teintes. Le bloc AT recherche la périodicité présente dans l'image: lorsqu'il trouve une périodicité, il modifie le gabarit en y incorporant le pixel précédant le pixel courant, modulo la périodicité trouvée. Un tel pixel a une excellente valeur prédictive.

De telles modifications sont peu fréquentes: lorsque l'une d'entre elles a lieu, une séquence de contrôle (indiquée symboliquement par `ATMOVE` dans la Figure Intro. 4) est concaténée au train de données de sortie. Ainsi, les décodeurs n'ont à effectuer aucun traitement de recherche pour positionner correctement le bloc fonctionnel AT.

Intro. 3.1.6 Codeur arithmétique adaptatif

Le bloc «codeur arithmétique adaptatif» AAE est un codeur entropique. Il enregistre les sorties des blocs TP et DP et détermine s'il est nécessaire de coder un pixel donné. Dans l'affirmative, il enregistre le contexte et utilise un estimateur de probabilité interne pour évaluer la probabilité conditionnelle pour que le pixel courant ait une couleur donnée. Le pixel est souvent hautement prévisible d'après le contexte, en sorte que la probabilité conditionnelle est très voisine de 0 ou 1, et qu'un important gain de codage entropique peut être obtenu.

La conservation des estimations de probabilité relatives à chaque contexte est un problème statistique non trivial. Il faut trouver un compromis entre deux impératifs contradictoires: précision des estimations et adaptation rapide à l'évolution de la structure statistique de l'entrée.

Intro. 3.2 Codeur de la couche de plus basse résolution

La Figure Intro. 5 représente le codeur de la couche de résolution la plus basse. Il est conceptuellement plus simple que le codeur de couche différentielle car il n'utilise pas les blocs RR et DP et met en œuvre des blocs TP, AT et MT différents, car aucune couche de résolution inférieure n'est utilisée en entrée. Se reporter aux tableaux en 4, pour une définition des noms des signaux. (Tous les sous-blocs ne sont pas nécessairement utilisés dans tous les systèmes.)

Comme la TP de couche différentielle, la TP de la couche de plus basse résolution est essentiellement destinée à accélérer le traitement. Les algorithmes utilisés pour les deux versions de TP sont toutefois tout à fait différents: il n'est pas possible, avec la TP de la couche de plus basse résolution, de «sauter» un pourcentage aussi élevé de pixels qu'avec la TP de couche différentielle. Avec des images comportant des textes et des dessins au trait, la TP de la couche de plus basse résolution permet de sauter environ 40% des pixels.

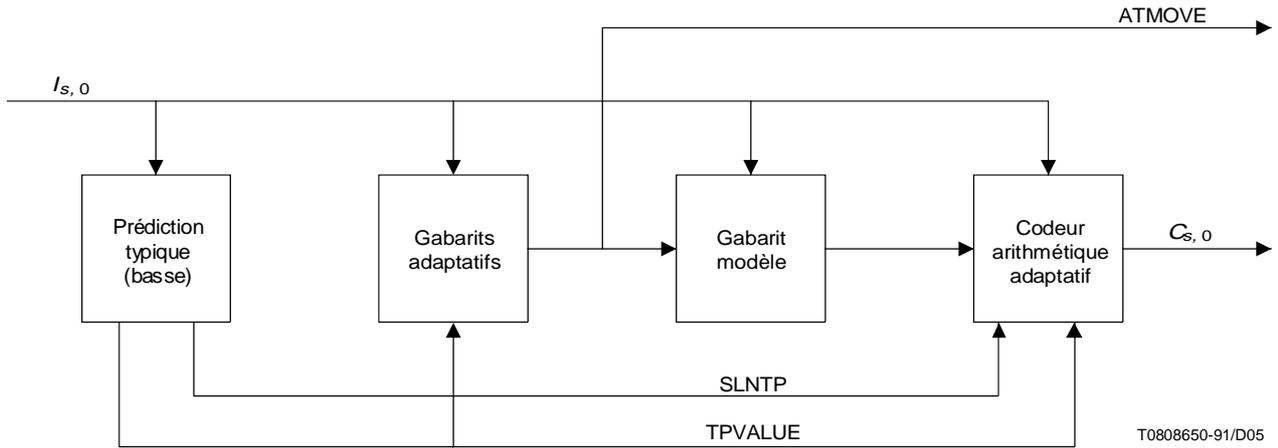


Figure Intro. 5 – Codeur de la couche de plus basse résolution

Intro. 4 Blocs fonctionnels du décodeur

Les Figures Intro. 6, Intro. 7 et Intro. 8 sont analogues aux Figures Intro. 3, Intro. 4 et Intro. 5 mais représentent le décodage. A noter que les blocs RR et AT ne figurent pas dans le décodeur. Se reporter aux tableaux en 4 pour la définition des noms des signaux. En codage séquentiel simple, seul le bloc «décodeur de la couche de plus basse résolution» de la Figure Intro. 6 est utilisé. Tous les sous-blocs des Figures Intro. 7 et Intro. 8 ne sont pas nécessairement utilisés dans tous les systèmes.

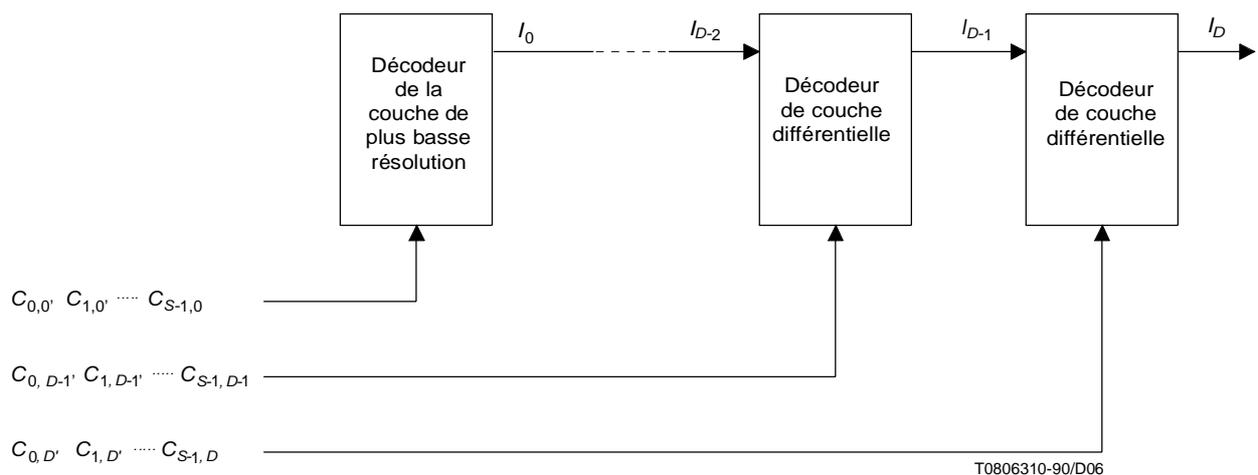


Figure Intro. 6 – Décomposition du décodeur

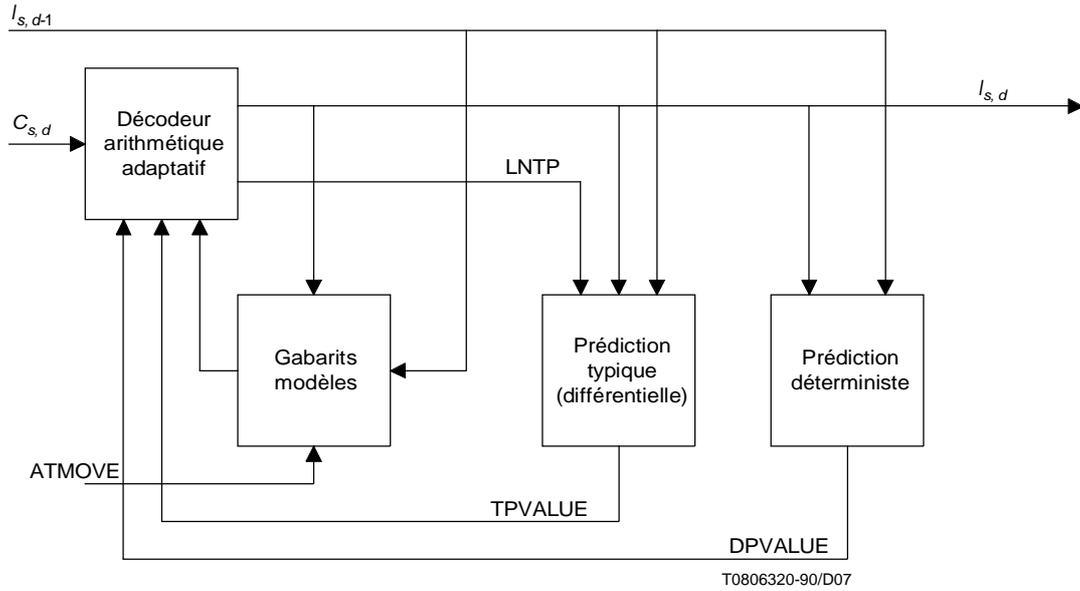


Figure Intro. 7 – Décodeur de couche différentielle

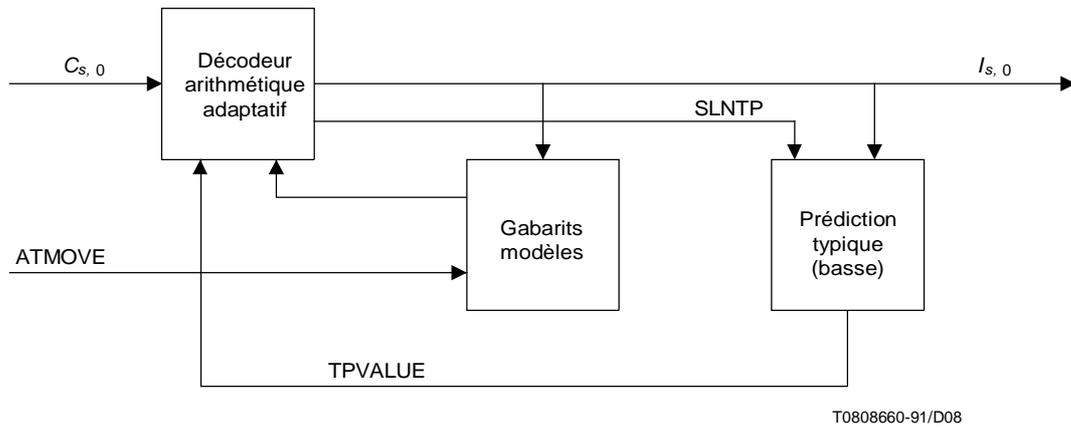


Figure Intro. 8 – Décodeur de la couche de plus basse résolution

NORME INTERNATIONALE

RECOMMANDATION DU CCITT

**TECHNOLOGIE DE L'INFORMATION –
REPRÉSENTATION CODÉE DES IMAGES ET DU SON –
COMPRESSION PROGRESSIVE DES IMAGES EN DEUX TONS**

1 **Domaine d'application**

La présente Recommandation | Norme internationale définit une méthode de compression sans perte, pour le codage des plans binaires d'images, particulièrement appropriée aux images en deux tons (qui incluent les images en noir et blanc).

2 **Références normatives**

Aucune référence normative. Les références informatives à des normes et à des publications techniques sont données dans l'Annexe F.

3 **Définitions**

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

- 3.1** **codeur arithmétique adaptatif**: Mécanisme de compression ou de décompression adaptative des données, utilisant des caractéristiques observées des données pour prédire et coder des symboles de données futurs.
- 3.2** **gabarits adaptatifs (AT)** (*adaptive template*): Gabarits modèles pouvant être modifiés par déplacement d'un pixel AT durant le traitement d'une image, pour tirer parti des motifs observés dans cette image.
- 3.3** **décalage AT**: Distance, en pixels, entre le pixel en cours de codage et le pixel AT.
- 3.4** **pixel AT**: Pixel spécifique du gabarit modèle dont la position peut changer de façon adaptative, durant le traitement d'une image.
- 3.5** **plan binaire**: Matrice (ou «plan») de symboles en deux tons, construite en choisissant un bit de même position dans tous les pixels d'une image.
- 3.6** **entrelacement de plans binaires**: Méthode de multiplexage des données de plusieurs plans binaires, dans un même train de données.
- 3.7** **octet**: Huit bits de données.
- 3.8** **bourrage d'octets**: Mécanisme permettant de distinguer sans ambiguïté des octets d'échappement prédéfinis indiquant le début d'un segment marqueur, d'octets identiques se présentant naturellement dans un train de données comprimées.
- 3.9** **contexte**: Entier correspondant au motif spécifique du gabarit et (si nécessaire) à la phase spatiale, utilisé pour identifier l'indice de l'état du codeur arithmétique adaptatif à utiliser pour le codage du pixel courant.
- 3.10** **prédiction déterministe (DP)** (*deterministic prediction*): Méthode de prédiction exacte (donc sans codage) des pixels d'une image, à partir d'une version à plus basse résolution de la même image, et basée sur une connaissance très précise de la méthode de réduction de résolution utilisée.
- 3.11** **codeur de couche différentielle**: Mécanisme de codage ou de décodage d'images de couche différentielle.
- 3.12** **image de couche différentielle**: Image de résolution donnée, décrite par référence aux pixels d'une image à plus basse résolution.
- 3.13** **codeur entropique**: Toute méthode de compression ou de décompression de données sans perte.

- 3.14 octet d'échappement:** Octet d'un train de données signifiant que les informations qui suivent ont la signification spéciale d'un marqueur.
- 3.15 pixel de haute résolution:** Pixel de l'image de plus haute résolution, dans les deux couches de résolution concernées.
- 3.16 ligne atypique (LNTF) (*line not typical*):** Condition se présentant lors d'une prédiction typique, lorsqu'un ou plusieurs des pixels associés à une ligne donnée à basse résolution donneraient lieu à une prédiction incorrecte.
- 3.17 codeur de la couche de plus basse résolution:** Mécanisme de codage ou de décodage des images de plus basse résolution.
- 3.18 image de la couche de plus basse résolution:** Image de résolution donnée, décrite sans aucune référence à une image de résolution plus basse.
- 3.19 pixel de basse résolution:** Pixel de l'image de plus basse résolution, dans les deux couches de résolution concernées.
- 3.20 marqueur:** Combinaison d'un octet d'échappement et d'un octet marqueur, annonçant des informations de contrôle.
- 3.21 octet marqueur:** Octet suivant immédiatement un octet d'échappement et définissant le type d'informations de contrôle annoncées.
- 3.22 segment marqueur:** Marqueur suivi de n'importe quel nombre d'octets d'informations de contrôle associées.
- 3.23 gabarit modèle (MT) (*model template*):** Motif géométrique précisant la position de pixels par rapport à un pixel à coder. Est utilisé pour modéliser des caractéristiques locales de l'image.
- 3.24 pixel:** Élément d'une image décrite par une matrice rectangulaire de tels éléments.
- 3.25 comportement progressif:** Une technique de codage présente un comportement progressif si la première image codée est celle de la couche de résolution la plus basse, la résolution étant ensuite augmentée par les apports successifs d'images de couches différentielles.
- 3.26 codage progressif:** Méthode de codage dans laquelle l'image peut être segmentée en bandes – l'image complète étant d'abord codée sous la forme d'une image de la couche de plus basse résolution, dont la résolution est ensuite augmentée par les apports successifs d'images de couches différentielles. Ce codage est compatible, par une remise en ordre des données de bande et de couche, avec le codage séquentiel compatible progressif.
- 3.27 codage séquentiel compatible progressif:** Méthode de codage d'une image dans laquelle l'image peut être segmentée en bandes, codées successivement, l'image de chaque bande étant codée progressivement jusqu'à la pleine résolution. Ce codage est compatible, par une remise en ordre des données de bande et de couche, avec le codage progressif.
- 3.28 données codées de bande protégées (PSCD) (*protected stripe coded data*):** Train de données d'image comprimées modifiées par bourrage d'octets, pour permettre de distinguer les octets d'échappement prédéfinis qui signalent des segments marqueurs spéciaux (ne faisant pas partie du train de données comprimées), d'octets identiques à l'octet d'échappement, qui se présentent naturellement dans le train de données comprimées.
- 3.29 méthode de réduction de résolution (RR) (*resolution reduction method*):** Méthode de transformation d'une image de résolution donnée, en une image décrivant le même sujet, mais de plus basse résolution.
- 3.30 comportement séquentiel:** Une technique de codage présente un comportement séquentiel si des parties voisines du haut de l'image sont complètement décrites avant de commencer la description de parties inférieures à celles-ci.
- 3.31 codage séquentiel simple:** Méthode de codage d'une image, telle que l'image est entièrement codée en une seule couche de résolution, ligne par ligne, de gauche à droite et de haut en bas, sans référence à aucune image de plus basse résolution. Elle est compatible avec le codage progressif et avec le codage séquentiel compatible progressif, si le nombre de couches différentielles est nul.
- 3.32 phase spatiale:** Attribut d'un pixel d'une image de couche différentielle, qui précise son orientation par rapport au pixel de basse résolution qui lui est associé.
- 3.33 résolution spatiale:** Nombre de pixels utilisés pour décrire une région d'une image de dimensions spatiales déterminées.

- 3.34 bande:** Zone d'une image, de taille verticale fixe, couvrant toute la largeur horizontale de cette image.
- 3.35 pixel cible:** Pixel à traiter (appelé également pixel courant ou pixel actif).
- 3.36 prédiction typique (TP) (*typical prediction*):** Méthode de prédiction exacte (et donc sans codage) de blocs de pixels d'une image, exploitant la présence de grandes zones de couleur uniforme.

4 Symboles et abréviations

4.1 Sigles

Voir le Tableau 1.

Tableau 1 – Sigles

Sigles	Signification
AT	Gabarits adaptatifs (<i>adaptive templates</i>)
DP	Prédiction déterministe (<i>deterministic prediction</i>)
LPS	Symbole le moins probable (<i>less probable symbol</i>)
LSB	Bit de plus faible poids (<i>least significant bit</i>)
MPS	Symbole le plus probable (<i>more probable symbol</i>)
MSB	Bit de plus fort poids (<i>most significant bit</i>)
MT	Gabarit modèle (<i>model templates</i>)
RR	Réduction de résolution (<i>resolution reduction</i>)
TP	Prédiction typique (<i>typical prediction</i>)

4.2 Constantes symboliques

Voir le Tableau 2.

Tableau 2 – Constantes symboliques

Constante	Signification	Valeur	
		ISO	Hexadécimal
ABORT	Abandon	00/04	0x04
ATMOVE	Changement d'AT	00/06	0x06
COMMENT	Commentaire privé	00/07	0x07
ESC	Echappement	15/15	0xff
NEWLEN	Nouvelle longueur	00/05	0x05
RESERVE	Marqueur réservé	00/01	0x01
SDNORM	Fin normale des données de la bande	00/02	0x02
SDRST	Réinitialisation à la fin des données de la bande	00/03	0x03
STUFF	Bourrage	00/00	0x00

4.3 Symboles mathématiques, opérateurs et indicateurs

Voir le Tableau 3.

4.4 Variables de noms mnémoniques

Voir le Tableau 4.

5 Conventions

5.1 Conventions et symboles relatifs aux organigrammes

Tous les organigrammes ont leur entrée en haut et leur sortie en bas. Le symbole «<<<» désigne un décalage binaire à gauche avec remplissage de zéros des positions binaires de plus faible poids; le symbole «>>>» indique un décalage binaire à droite, avec remplissage de zéros des positions binaires de plus fort poids. Pour «<<<» et «>>>», la quantité indiquée à gauche est la quantité décalée, et la quantité indiquée à droite est la valeur du décalage. Le ET logique binaire de deux nombres est indiqué par «&».

5.2 Représentation graphique des gabarits

Il sera fréquemment nécessaire de représenter graphiquement les relations des pixels d'une image de haute résolution avec ceux d'une image de basse résolution. La Figure 1 est un graphique tridimensionnel, représentant une telle correspondance. Dans la présente Spécification, on préfère utiliser des représentations bidimensionnelles, plus compactes (comme celle de la Figure 2). A noter que dans la représentation graphique bidimensionnelle, les pixels basse résolution sont représentés par des cercles, et les pixels haute résolution correspondants par des carrés, partiellement cachés par les cercles basse résolution.

Tableau 3 – Symboles mathématiques, opérateurs et indicateurs

Notation	Signification
$C_{s,d}$	Données codées de la bande s de la couche d
D	Nombre pour la couche de résolution finale dans la BIE
D_L	Nombre pour la couche de résolution initiale dans la BIE
h	Pixel de haute résolution
I_d	Image de la couche d
l	Pixel de basse résolution
L_c	Longueur du commentaire privé, en octets
L_d	Lignes par bande dans la couche d
M_X	Décalage horizontal maximal permis pour un pixel AT
M_Y	Décalage vertical maximal permis pour un pixel AT
p	Probabilité
P	Nombre de plans binaires
R_d	Résolution à la couche d
S	Nombre de bandes
X_d	Dimension horizontale de l'image à la couche d
y_{AT}	Ligne à laquelle doit être opéré un changement d'AT
Y_d	Dimension verticale de l'image à la couche d
τ_X	Décalage horizontal du pixel AT
τ_Y	Décalage vertical du pixel AT
$>>$	Décalage binaire à droite
$<<$	Décalage binaire à gauche
$\&$	ET logique
$!$	Non logique
\oplus	OU exclusif
$\lceil \cdot \rceil$	Fonction plafond (plus petit entier \geq argument)
0x	Indicateur hexadécimal

5.3 Phases spatiales

Une «phase spatiale» peut être associée aux pixels de n'importe quelle couche de résolution autre que la plus basse. Cette «phase» précise l'orientation du pixel par rapport au pixel de plus basse résolution qui lui est associé. Ainsi, le pixel supérieur de gauche, parmi les quatre pixels associés à un pixel de basse résolution unique, est dit avoir une «phase 0». De même, le pixel supérieur droit a la phase 1, le pixel inférieur gauche a la phase 2 et le pixel inférieur droit a la phase 3 (voir Figure 3).

Tableau 4 – Variables de noms mnémoniques

Variable	Signification
A	Registre de longueur d'intervalle
BID	Données d'image en deux tons
BIE	Entité d'image en deux tons
BIH	En-tête d'image en deux tons
BUFFER	Mémoire tampon
C	Registre de code
CE	Echange conditionnel
CHIGH	Les deux octets supérieurs du registre de code
CLOW	Les deux octets inférieurs du registre de code
CT	Compteur de bits
CX	Contexte
DPLAST	Dernière DP
DPON	DP activée
DPPRIV	DP privée
DPTABLE	Table de DP
DPVALUE	Valeur de DP
HITOLO	A résolution décroissante
I LEAVE	Plusieurs plans binaires entrelacés
LNTP	Ligne atypique
LPIX	Pixel de basse résolution
LRLTWO	Gabarit de deux lignes de la couche de plus basse résolution
LSZ	Longueur de l'intervalle de codage du LPS
MPS	Symbole le plus probable
NLPS	Prochain si LPS
NMPS	Prochain si MPS
PIX	Pixel
PSCD	Données codées de bande protégées
SC	Compteur de pile (d'octets 0xff)
SCD	Données codées de bande
SDE	Entité de données de bande
SEQ	Séquentiel
SLNTP	Même LNTP
SMID	L'indice relatif à la bande est en position centrale
SWTCH	Permutation
ST	Etat
TPBON	TP de la couche de plus basse résolution activée
TPDON	TP de couche différentielle activée
TPVALUE	Valeur de TP
VLENGTH	Longueur de variable

5.4 Graphiques de structure des données

Les Tableaux 5 à 16 précisent la décomposition des champs en sous-champs. Dans ces tableaux, les sous-champs de gauche doivent être transmis en premier. La dernière rangée de ces tableaux donne la longueur des champs en octets. Une entrée «1/8» indique un bit unique. Une entrée «varie» signale que la longueur du champ est variable et dépend des options ou paramètres choisis, ou de l'image codée.

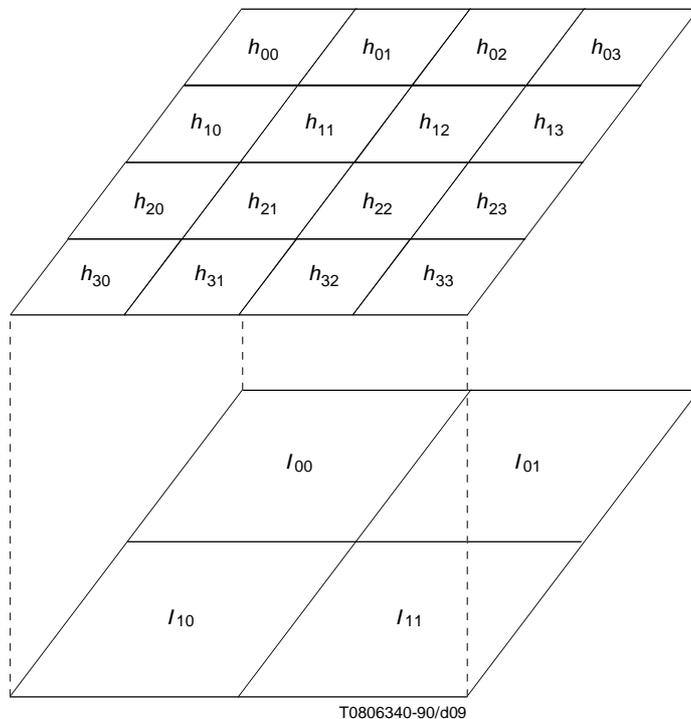


Figure 1 – Représentation graphique tridimensionnelle de pixels de haute et de basse résolution

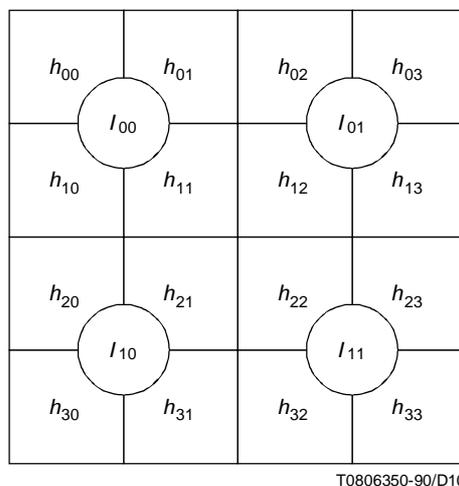


Figure 2 – Représentation graphique bidimensionnelle de pixels de haute et de basse résolution

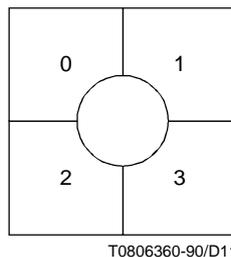
6 Spécifications

6.1 Règles générales

6.1.1 Affectation des couleurs

Chacun des bits de chaque plan de pixels vaut 0 ou 1. Lorsque l'image est en deux tons, un bit 1 indique la couleur d'avant-plan et un bit 0 indique la couleur d'arrière-plan. Si l'image comporte plusieurs plans binaires, la présente Spécification ne définit pas la correspondance des intensités et couleurs avec les plans binaires.

NOTE – La couleur d'avant-plan peut être représentée par la valeur 1 ou 0, ce choix n'ayant d'incidence sur aucun aspect de la présente Spécification, sauf sur la méthode de réduction de résolution décrite. Cette méthode présente en effet une légère asymétrie entre les couleurs d'arrière-plan et d'avant-plan.



T0806360-90/D11

Figure 3 – Quatre phases possibles pour des pixels de haute résolution

6.1.2 Conventions relatives aux bords

Les algorithmes de réduction de résolution, de prédiction typique, de prédiction déterministe et de codage effectuent tous des itérations sur l'image, dans l'ordre usuel de balayage, c'est-à-dire de gauche à droite et de haut en bas. Le traitement d'un pixel cible courant doit se référer aux couleurs de certains pixels se trouvant dans une relation spatiale fixe avec ce pixel cible. Aux bords de l'image, les pixels voisins, auxquels il est ainsi fait référence, peuvent ne pas se trouver dans l'image réelle. Pour les images à haute et basse résolution, les règles de traitement des références hors image sont les suivantes:

- un bord de couleur d'arrière-plan (0) est supposé se trouver au-dessus, à gauche et à droite de l'image réelle;
- le bas de l'image doit être étendu vers le bas, aussi loin que nécessaire, en répétant les pixels de la dernière ligne réelle de l'image.

En outre, les règles suivantes doivent être appliquées lors de références à des pixels situés au-delà des limites d'une des bandes:

- une référence à un pixel, dans une bande se trouvant au-dessus de la bande courante, doit renvoyer la valeur réelle du pixel, sauf si le pixel est au-dessus de l'image, auquel cas la règle relative au bord d'arrière-plan du haut de l'image doit être appliquée;
- une référence à un pixel, dans une bande située en dessous de la bande courante, doit être satisfaite par répétition des pixels de la dernière ligne de cette bande courante. En particulier, il ne faut pas utiliser de valeurs réelles, même si le pixel de référence se trouve à l'intérieur de l'image.

NOTE – Cette dernière règle n'a de conséquence que pour l'image à basse résolution, car pour que le décodage soit possible, il ne doit jamais exister de référence à des pixels à haute résolution appartenant à la ligne du dessous. Il se trouve en outre que l'algorithme de réduction de résolution ici décrit ne fait jamais référence à des pixels de la ligne du dessous, même à basse résolution.

6.1.3 Calage des octets

NOTE – Du fait des conventions relatives à l'en-tête et aux marqueurs, décrites en 6.2, les segments marqueurs sont toujours calés sur les octets du train de données.

6.2 Organisation des données

6.2.1 Décomposition de l'image

La structure de données du plus haut niveau décrite dans la présente Spécification est appelée entité d'image en deux tons (BIE) (*bi-level image entity*). Une BIE donnée contient les données d'une ou plusieurs couches de résolution et d'un ou plusieurs plans binaires. Les données décrivant une image donnée, dans toutes ses résolutions disponibles et pour tous ses plans binaires, peut, mais pas nécessairement, être contenues dans plusieurs BIE.

NOTE – La description d'une image par plusieurs BIE est nécessaire lorsque les images sont d'abord disponibles à une résolution basse ou intermédiaire ou à la précision du plan binaire, et qu'une amélioration de la résolution ou de la précision peut éventuellement être demandée.

6.2.2 Décomposition d'une entité et d'un en-tête d'image en deux tons (BIE et BIH)

Comme montré dans le Tableau 5, une entité d'image de deux tons (BIE) comprend un en-tête d'image en deux tons (BIH) (*bi-level image header*) et des données d'images en deux tons (BID) (*bi-level image data*).

Tableau 5 – Décomposition d'une BIE

BIE	
BIH	BID
Varie	Varie

L'en-tête d'image en deux tons comprend les champs indiqués dans les Tableaux 6, 7 et 8.

Tableau 6 – Décomposition d'un BIH

BIH											
D_L	D	P	–	X_D	Y_D	L_0	M_X	M_Y	Ordre	Options	DPTABLE
1	1	1	1	4	4	4	1	1	1	1	0 ou 1728

Tableau 7 – Décomposition de l'octet d'ordre

Ordre							
MSB	...						LSB
–	–	–	–	HITOLO	SEQ	ILEAVE	SMID
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Tableau 8 – Décomposition de l'octet d'options

Options							
MSB	...						LSB
–	LRLTWO	VLENGTH	TPDON	TPBON	DPON	DPPRIV	DPLAST
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Le premier octet du BIH désigne D_L , la couche de résolution initiale à spécifier dans la BIE. Le plus souvent, ce nombre sera nul, auquel cas les données transmises doivent permettre la construction de l'image sans en avoir aucune connaissance préalable. D_L n'est pas nul si une BIE antérieure a déjà défini l'image d'une couche intermédiaire: il suffit alors de spécifier les informations incrémentales. Le deuxième octet spécifie D , la couche de résolution finale décrite dans cette BIE. Noter qu'avec de telles BIE multiples, lorsque D_L est zéro, D est égal au nombre de couches différentielles, non au nombre total de couches.

Le troisième octet spécifie le nombre de plans binaires P . Si l'image est en deux tons, P a la valeur 1.

Le quatrième octet est un octet de remplissage. Il a toujours la valeur 0.

Les trois champs suivants, de quatre octets, spécifient X_D , Y_D et L_0 qui sont respectivement la dimension horizontale à la plus haute résolution, la dimension verticale à la plus haute résolution et le nombre de lignes par bande à la plus basse résolution. Ces trois entiers sont codés en commençant par l'octet de plus fort poids. En d'autres termes, X_D est la somme de 256^3 fois le cinquième octet de BIH, 256^2 fois le sixième octet, 256 fois le septième octet et du huitième octet.

Les dix-septième et dix-huitième octets spécifient M_X et M_Y , les décalages maximaux horizontaux et verticaux permis pour le pixel AT. Ces paramètres sont plus amplement traités en 6.7.3.

Le dix-neuvième octet du BIH véhicule les paramètres binaires HITOLO, SEQ, ILEAVE et SMID, qui spécifient ensemble l'ordre dans lequel les données de la bande sont concaténées pour former les BID. De plus amples détails sont donnés en 6.2.4. Les quatre bits de plus fort poids de cet octet ont une fonction de remplissage et doivent toujours être à zéro.

Le vingtième octet du BIH spécifie les options. Son bit de plus fort poids est un bit de remplissage toujours à 0. Le gabarit utilisé pour le codage de la couche de plus basse résolution doit avoir respectivement deux ou trois lignes, lorsque LRLTWO vaut 1 ou 0 (voir 6.7.1). Si le bit VLENGTH vaut 0, il n'y a pas de segment marqueur NEWLEN (voir 6.2.6.2). Si VLENGTH vaut 1, des segments marqueurs NEWLEN peuvent éventuellement figurer. Les bits TPDON, TPBON et DPON sont mis à 1 pour activer respectivement, le cas échéant, la TP de couche différentielle, la TP de couche de plus basse résolution et la DP. Les bits DPPRIV et DPLAST ne sont significatifs que si DPON est égal à 1. Si DPON et DPPRIV sont à 1, une table de DP privée doit être utilisée. Si DPLAST est à 0, la table de DP privée (1728 octets) doit être chargée. Autrement, la dernière table de DP utilisée est réutilisée.

Le champ DPTABLE de BIH ne doit figurer que si DPON est à 1, DPPRIV à 1 et DPLAST à 0. Sa taille et son interprétation sont définies en 6.6.

Les variables D_L , D , P , X_D , Y_D , L_0 , M_X , M_Y , HITOLO, SEQ, ILEAVE, SMID, LRLTWO, VLENGTH, TPDON, TPBON, DPON, DPPRIV et DPLAST sont des paramètres libres. Certaines normes d'application peuvent restreindre les choix, pour certains ou pour l'ensemble de ces paramètres libres. Le Tableau 9 montre les limites de ces paramètres, déterminées par leur nature implicite ou par les longueurs de champs permises pour eux dans un BIH.

Un train de données JBIG est constitué par tout train de données créé comme décrit dans les parties normatives de cette Spécification, avec des paramètres libres appartenant aux plages du Tableau 9. Afin de faciliter la création d'un ensemble d'applications aussi important que possible, pouvant utiliser des matériels communs et échanger des données décodables, des plages minimales pour ces paramètres libres sont suggérées dans l'Annexe A. Chaque fois que possible, il est conseillé de choisir, pour des applications différentes, des valeurs de paramètres dans les plages minimales suggérées. Il peut être souhaitable, pour des mises en œuvre que l'on désire compatibles avec une large gamme d'applications, de prendre en charge tous les choix de paramètres libres, dans les plages suggérées.

6.2.3 Itération de paramètres dépendants de la couche de résolution

Les dimensions de l'image aux couches inférieures (indexées par d) sont définies récursivement pour $D \geq d \geq 1$, par les itérations:

$$X_{d-1} = \lceil X_d / 2 \rceil \quad (1)$$

$$Y_{d-1} = \lceil Y_d / 2 \rceil \quad (2)$$

où $\lceil \cdot \rceil$ désigne la fonction plafond (le plus petit entier supérieur ou égal à l'argument).

Tableau 9 – Limites absolues des paramètres libres

Paramètre	Minimum	Maximum
D_L	0	D
D	D_L	255
P	1	255
X_D	1	4 294 967 295
Y_D	1	4 294 967 295
L_0	1	4 294 967 295
M_X	0	127
M_Y	0	255
HITOLO	0	1
SEQ	0	1
ILEAVE	0	1
SMID	0	1
LRLTWO	0	1
VLENGTH	0	1
TPDON	0	1
TPBON	0	1
DPON	0	1
DPPRIV	0	1
DPLAST	0	1

Pour $1 \leq d \leq D$, le nombre de lignes par bande dans la couche d est défini par:

$$L_d = 2 \times L_{d-1} \tag{3}$$

Toutes les couches comportent nécessairement:

$$S = \lceil Y_0 / L_0 \rceil \tag{4}$$

bandes. Dans la plupart des cas, la dernière bande d'une couche d (quelconque) aura moins de L_d lignes.

6.2.4 Décomposition des données d'une image en deux tons (BID)

Les données codées $C_{s, d, p}$, définissant une bande donnée s de résolution d , au plan binaire p sont contenues dans une entité de données de bande SDE. La BID consiste en la concaténation des segments d'une SDE et d'un marqueur flottant, comme montré dans le Tableau 10.

Tableau 10 – Décomposition des BID

BID						
Segment(s) marqueur(s) flottant(s)	SDE _{s, d, p}	Segment(s) marqueur(s) flottant(s)	SDE _{s, d, p}	...	Segment(s) marqueur(s) flottant(s)	SDE _{s, d, p}
Varie	Varie	Varie	Varie	...	Varie	Varie

L'ordre des SDE dépend de HITOLO, SEQ, ILEAVE et SMID. L'imbrication des indices est définie par le Tableau 11. Seules les six combinaisons des trois variables SEQ, ILEAVE et SMID indiquées dans le Tableau 11 sont permises. Les deux combinaisons restantes ne doivent jamais se présenter. Les boucles relatives aux variables fictives s et p vont respectivement de 0 à $S - 1$ (de haut en bas) et $P - 1$ à 0 (de MSB à LSB). Si HITOLO est à 0, la variable fictive d doit évoluer de D_L à D . Autrement, elle doit évoluer de D à D_L .

Voir Tableau Intro. 2, pour un exemple pédagogique.

Tableau 11 – Ordre de codage des bandes des BID

			Boucles		
SEQ	ILEAVE	SMID	Ext.	Centrale	Int.
0	0	0	p	d	s
0	1	0	d	p	s
0	1	1	d	s	p
1	0	0	s	p	d
1	0	1	p	s	d
1	1	0	s	d	p

6.2.5 Décomposition d'une entité de données d'une bande (SDE)

Comme montré dans le Tableau 12, chaque SDE doit être terminée par un octet ESC, et un octet SDNORM ou SDRST.

Tableau 12 – Structure de la SDE

SDE		
PSCD	ESC	SDNORM ou SDRST
Varie	1	1

Normalement, l'octet de terminaison est un SDNORM, qui implique que toutes les informations «d'état» sont sauvegardées. Si l'octet de terminaison est un SDRST, «l'état» du plan binaire et de la couche de résolution traités doit être réinitialisé avant le codage ou le décodage de la bande suivante de ce plan et de cette couche. La réinitialisation de l'état indiqué par SDRST, nécessite l'initialisation des estimateurs de probabilité adaptatifs en haut de l'image, le repositionnement (le cas échéant) du pixel AT à sa position par défaut, et l'initialisation de LNT_{y-1} à 1, dans la couche de plus basse résolution. Cette réinitialisation nécessite également que toutes les fonctions concernant la réduction de résolution, la prédiction déterministe, la prédiction typique et les gabarits modèles, commencent à traiter la bande suivante, comme elles le feraient en haut de l'image, c'est-à-dire comme défini en 6.1.2.

NOTE – La réinitialisation de l'état indiqué par SDRST ne doit pas être effectuée inutilement, car cela réduirait l'efficacité de compression et risquerait d'introduire des artefacts au bord des bandes, dans des images à basse résolution.

Les données codées de bande protégées (PSCD) sont définies comme les octets de la SDE qui restent après la suppression des deux octets de terminaison. Un décodeur crée des données codées de bande (SCD) à partir des PSCD, en remplaçant toutes les occurrences d'un octet ESC suivi d'un octet STUFF par un octet ESC unique. Un codeur crée des PSCD à partir de SCD, en remplaçant toutes les occurrences d'un octet ESC, par un octet ESC suivi d'un octet STUFF. L'utilisation des SCD est décrite en 6.8. Les PSCD sont utilisées pour la définition de la SDE, de préférence aux SCD, pour permettre de situer sans ambiguïté les données d'une bande dans la BID.

Un octet ESC suivi d'un octet ABORT peuvent être utilisés pour effectuer une terminaison prématurée des BID, comme montré dans le Tableau 13.

NOTE – En l'absence d'un tel mécanisme, un codeur confronté à un problème ferait boucler indéfiniment le décodeur associé. Le redémarrage du décodeur serait impossible, car il attendrait, pour se réinitialiser, d'avoir reçu la quantité de données annoncée.

Tableau 13 – Marqueurs de terminaison prématurée de BID

ESC	ABORT
1	1

6.2.6 Segments marqueurs flottants

Les segments marqueurs flottants permettent de contrôler les informations. Ils ne doivent pas figurer dans une SDE. Ils peuvent se présenter entre des SDE ou avant la première SDE. Trois segments marqueurs flottants sont définis: ATMOVE, NEWLEN et COMMENT. (Voir la Note 2 en 6.2.6.2.)

6.2.6.1 Déplacement de gabarit adaptatif (AT)

Le segment marqueur ATMOVE décrit dans le Tableau 14, permet de déplacer le pixel AT.

Tableau 14 – Structure du segment marqueur flottant ATMOVE

ESC	ATMOVE	y_{AT}	τ_X	τ_Y
1	1	4	1	1

Les troisième, quatrième, cinquième et sixième octets définissent la ligne, y_{AT} , à laquelle le gabarit change. Les septième et huitième octets définissent les décalages horizontal et vertical, τ_X et τ_Y du nouveau pixel AT. La ligne y_{AT} doit être décodée comme la somme de 256^3 fois le troisième octet, 256^2 fois le quatrième octet, 256 fois le cinquième octet, plus le sixième octet.

L'estimateur de probabilité n'est pas réinitialisé après un ATMOVE. Le segment marqueur ATMOVE a un effet sur la couche de résolution et le plan binaire de la première SDE qui suit ce segment marqueur. La numérotation des lignes de y_{AT} reprend à 0 pour chaque bande: si un changement prend effet, par exemple, sur la première ligne d'une bande, y_{AT} est égale à 0.

Des compléments sur les pixels AT et les variables y_{AT} , τ_X et τ_Y figurent en 6.7.3.

6.2.6.2 Redéfinition de la longueur d'image

Si VLENGTH vaut 1, il est permis d'utiliser un segment marqueur de nouvelle longueur pour modifier la longueur Y_D de l'image, comme indiqué dans le Tableau 15.

Tableau 15 – Segment marqueur indiquant une nouvelle dimension verticale

ESC	NEWLEN	Y_D
1	1	4

Il ne peut pas figurer plus d'un segment marqueur de nouvelle longueur dans une BIE. Cependant, un segment marqueur peut faire référence à une ligne dans la bande immédiatement précédente du fait d'une interruption inattendue de l'image ou de l'utilisation d'une seule bande. Un tel segment marqueur est immédiatement suivi de ESC + SDNORM/SDRST et la nouvelle Y_D donnée par le segment marqueur de nouvelle longueur peut être inférieure au numéro de ligne de la fin de la bande précédente. Le codeur ne doit pas coder, dans chaque couche, plus que le nombre de lignes correspondant à Y_D . Dans le segment marqueur de nouvelle longueur, Y_D doit être codée sur son champ de quatre octets, exactement comme elle l'est dans son champ de quatre octets du BIH. La nouvelle Y_D ne doit jamais être supérieure à la longueur initiale.

NOTES

1 Le segment marqueur de nouvelle longueur a été défini en sorte qu'il soit possible de commencer le codage d'une image de longueur inconnue. Dans ce cas, la dimension initiale Y_D , placée dans l'en-tête, sert à indiquer la longueur maximale de l'image.

2 Certaines applications peuvent nécessiter que le nouveau segment marqueur de longueur suive immédiatement un PSCD, juste avant ESC + SDNORM/SDRST, pour une interruption prématurée. Cette utilisation est actuellement à l'étude.

6.2.6.3 Segment marqueur de commentaire

Un octet ESC suivi d'un octet COMMENT et d'un entier de quatre octets L_c doit figurer au début d'un segment marqueur de commentaire, comme indiqué dans le Tableau 16.

Le nombre L_c doit être égal à la somme de 256^3 fois le troisième octet, 256^2 fois le quatrième octet, 256 fois le cinquième octet et du sixième octet. Ce nombre donne uniquement la longueur de la partie commentaire privé du segment marqueur de commentaire. Autrement dit, la longueur totale du segment marqueur de commentaire est $L_c + 6$ octets.

Tableau 16 – Segment marqueur de commentaire

ESC	COMMENT	L_c
1	1	4

6.2.7 Octet marqueur réservé

Un octet ESC suivi d'un octet RESERVE est un marqueur réservé. Une utilisation possible de ce marqueur est décrite en 6.8.2.8. Aucune extension future de la présente Spécification ne devra utiliser ce marqueur, pour quelque usage que ce soit. Des codeurs ou décodeurs peuvent l'utiliser pour n'importe quel usage privé. Le marqueur réservé ne doit jamais apparaître dans un train de données public.

6.3 Réduction de résolution

La table de prédiction déterministe par défaut est adaptée à l'algorithme de réduction de résolution suggéré décrit dans ce paragraphe. On peut utiliser, à la place de cet algorithme, un autre algorithme de réduction de résolution, mais dans ce cas, la prédiction déterministe doit être désactivée; sinon, une table de prédiction déterministe adaptée doit être téléchargée dans le décodeur, comme partie du BIH.

L'algorithme de résolution est identique pour toutes les couches de résolution et tous les plans binaires. Le processus de création de l'image de la couche de résolution $d - 1$, à partir d'une image de la couche de résolution d est décrit ci-après.

Si X_d ou Y_d n'est pas pair, une nouvelle image de couche d peut être créée pour pouvoir appliquer l'algorithme de réduction de résolution, en ajoutant une colonne de couleur d'arrière-plan (0) à droite ou une réplication des pixels de la dernière ligne en bas de l'image. Dans la suite de ce paragraphe, X_d et Y_d sont donc supposés pairs.

L'image d'origine peut être divisée en 2×2 blocs de pixels, chacun de ces 2×2 «superpixels» étant en correspondance avec un pixel de basse résolution de l'image de résolution réduite. Ces pixels de basse résolution sont déterminés de gauche à droite et de haut en bas, dans l'ordre normal de balayage. La règle de réduction de résolution suggérée dans la présente Spécification est définie par la Figure 4 et le Tableau 17. Le principe de cette mise en correspondance est exposé dans l'Annexe B (informative).

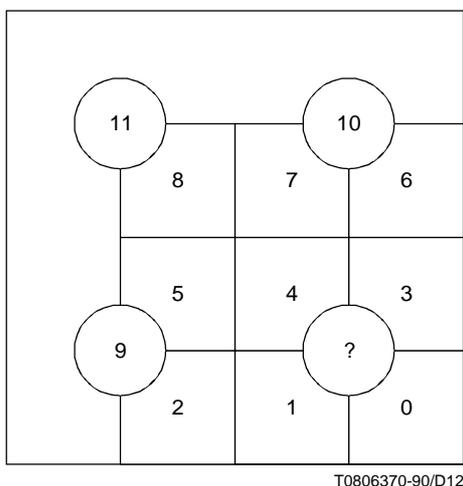


Figure 4 – Pixels utilisés pour déterminer la couleur d'un pixel de basse résolution

Le cercle contenant un «?» représente le pixel de basse résolution dont la couleur est à déterminer. Les cercles et les carrés contenant des nombres correspondent aux pixels utilisés pour cette détermination.

Les pixels numérotés définissent un index, dont chaque bit est déterminé par la couleur du pixel de numéro correspondant. (Le pixel «0» détermine le bit de plus faible poids, le pixel «n» le bit «n», etc.) Lorsqu'un pixel prend la couleur d'avant-plan, le bit correspondant de l'index est mis à 1. L'index donne une entrée du Tableau 17 (indexé de gauche à droite), qui détermine la couleur du pixel «?». Par exemple, les couleurs des pixels correspondant aux index 0 à 7 sont respectivement 0, 0, 0, 1, 0, 0, 0 et 1.

Aux bords d'une image, certains des pixels numérotés de la Figure 4 peuvent ne pas se trouver à l'intérieur de l'image. Pour la définition d'un index, on applique les règles générales relatives aux bords stipulées en 6.1.2.

Au début d'une réduction de résolution, le pixel situé en haut et à gauche de l'image de haute résolution doit être aligné avec le pixel 4 de la Figure 4.

6.4 Prédiction typique de couche différentielle

La prédiction typique de couche différentielle peut être activée ou désactivée par le bit TPDON du champ «options» du BIH. S'il est désactivé (TPDON = 0), le bloc TPD (d'un codeur ou d'un décodeur) sort simplement TPVALUE = 2 pour tous les pixels, pour indiquer au bloc de codage ou de décodage arithmétique qu'aucune prédiction n'est faite. En outre, lorsque la TP de couche différentielle est désactivée, le pseudo-pixel LNTP ne doit être ni codé ni décodé par le (dé)codeur arithmétique. Il est supposé dans les paragraphes suivants que la TP de couche différentielle est activée (TPDON = 1).

Chaque fois qu'il est fait référence à un pixel qui, à cause d'effets des bords, ne se trouve pas effectivement dans la bande courante, la valeur de ce pixel est déterminée par les règles générales relatives aux bords stipulées en 6.1.2.

6.4.1 Traitement dans un codeur

La Figure 5 définit un voisinage de huit pixels, voisins immédiats du pixel «?».

Un pixel de basse résolution donné est dit «atypique» si ce pixel et les huit pixels de son voisinage sont de la même couleur, mais qu'un ou plusieurs des quatre pixels à haute résolution associés ont une couleur différente de cette couleur commune. Une ligne à basse résolution donnée est dite «atypique» (LNTP) si elle contient des pixels «atypiques». Un organigramme de détermination de la LNTP est représenté à la Figure 6, où LPIX désigne un pixel de basse résolution.

NOTE – Des pixels de basse résolution «atypiques», au sens ci-dessus, sont possibles, mais extrêmement rares.

Tableau 17 – Table de détermination de la couleur de basse résolution

Indice	Couleur							
[0, 63]	00010001	01110011	11111111	11111111	00110011	11111111	11111111	11111111
[64, 127]	00000001	01110111	11111111	11111111	00110111	11111111	11111111	11111111
[128, 191]	00110111	11111111	11111111	11111111	01111101	11111111	11111111	11111111
[192, 255]	00110111	11111111	11111111	11111111	11111111	01111101	11111111	11111111
[256, 319]	00000001	00110111	11111101	11111111	00111111	11111111	11111111	11111111
[320, 383]	00110111	01111111	11111111	01111111	01111111	01111111	01111111	11111111
[384, 447]	00110101	11111111	11110111	11111111	11011111	01111111	11111111	11111111
[448, 511]	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
[512, 575]	00000001	00100011	00000101	00111011	00010001	00100011	01110001	11111111
[576, 639]	00000001	01110101	00111011	01111111	00000000	01010011	11111110	11111111
[640, 703]	00000001	01000001	01111111	11111111	00001001	10110111	11111111	11111111
[704, 767]	00000000	01010011	01111111	11111011	10010011	01111001	11111111	11111111
[768, 831]	00000001	00000000	01110011	11111111	00110001	00010011	01110101	11111111
[832, 895]	00000000	01000001	10110111	11101110	00000001	00100001	11111100	11111111
[896, 959]	00000000	10010011	01110101	11111111	00010001	01101011	11110101	11111111
[960, 1023]	11101001	11110111	11111111	11111011	10110111	11111111	11111011	11111111
[1024, 1087]	00000001	00100011	00000001	00111111	00010001	00000001	01110111	11111111
[1088, 1151]	00000001	01110101	01101011	01111111	00000000	01010011	11111110	11111111
[1152, 1215]	00000001	01100001	01111111	11111111	00101001	00110111	11111111	11111111
[1216, 1279]	00000000	01110011	00111111	01111011	10010010	01111101	11111111	11111111
[1280, 1343]	00000001	00000000	01111011	11111110	00101111	00011011	01111111	11111111
[1344, 1407]	00000000	01000001	00110111	11111110	00001001	00110111	01111110	01111111
[1408, 1471]	00000000	11010010	01111111	11111111	00011011	01101111	11111111	11111111
[1472, 1535]	00000000	01110101	01111111	01110111	00100111	01111111	01111011	01111111
[1536, 1599]	00000001	00000011	00000001	00001001	00010001	00000001	01000001	10010011
[1600, 1663]	00000001	01110101	00100001	01010101	00000000	01010001	10000000	11110111
[1664, 1727]	00000001	01000001	01101011	00010011	00000001	00000000	11111011	11111111
[1728, 1791]	00000000	01010001	00000001	01110011	00000000	01000001	10110111	11111111
[1792, 1855]	00000001	00000000	01100001	10000001	00100111	00001001	00011110	10111111
[1856, 1919]	00000000	01000000	00000001	01010110	00001000	00000000	00010000	01111111
[1920, 1983]	00000000	10000000	00100001	01110111	00000011	00000001	00111111	11111111
[1984, 2047]	01101000	11010000	11110011	10110011	00000000	11010011	11111011	11111111
[2048, 2111]	00000001	00000011	00110111	11111111	00110011	00110111	01111111	11111111
[2112, 2175]	00000001	01110111	01111111	11111111	00010001	01111011	11111111	11111111
[2176, 2239]	00000001	11110111	01111111	11111111	00111111	11111111	11111101	11111111
[2240, 2303]	00010010	11110111	11111111	11111111	11111111	11111101	11111111	01111111
[2304, 2367]	00000001	00010010	01111101	11111111	00111111	01111111	11111111	11111111
[2368, 2431]	00000000	01100010	11111111	01111111	00111111	00111111	01111111	11111111
[2432, 2495]	00010000	11111111	11110111	11111111	01111111	11111111	01111111	11111111
[2496, 2559]	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
[2560, 2623]	00000001	00100011	00000001	00011011	00010001	00100011	01110111	11111111
[2624, 2687]	00000001	01110101	00101011	01110111	00000000	01000001	10111110	11111111
[2688, 2751]	00000001	11000001	01011011	01111111	00001001	00110011	01111101	11111111
[2752, 2815]	00000000	01010001	00110111	11111011	10101001	10110001	11111111	11111111
[2816, 2879]	00000001	00000000	01110001	10110111	00100001	00000011	01110101	11111111
[2880, 2943]	00000000	01000000	00010111	01101111	00000000	00000001	01111101	11111111
[2944, 3007]	00000000	11000001	01110101	11111111	00000001	10101011	01010001	11111111
[3008, 3071]	11101000	11010011	11111111	11111011	10111011	11111111	11111011	11111111
[3072, 3135]	00000001	00100011	00000001	00011011	00110001	00000001	01010011	01111111
[3136, 3199]	00000001	01110101	00101001	01111111	00000000	01010001	10110110	11111111
[3200, 3263]	00000001	11100000	01111011	11111111	00001010	00111011	01111111	11111111
[3264, 3327]	00000000	01110001	01111111	11111011	10001000	01110101	11111111	01111111
[3328, 3391]	00000001	00000000	01100001	11110110	00111111	00001001	01111111	11111111
[3392, 3455]	00000000	01000000	00010111	01111111	00001000	00010011	01111110	01111111
[3456, 3519]	00000000	10000000	01110111	11111111	00101011	00101111	01111111	01111111
[3520, 3583]	00000000	01110001	01111111	01110111	00101011	01111111	00111011	01111111
[3584, 3647]	00000001	00000011	00000001	00001001	00010001	00000001	01000001	00000001
[3648, 3711]	00000001	01110101	00100001	01010101	00000000	01010001	10000000	01010011
[3712, 3775]	00000001	01000001	01001001	00000001	00001001	00000000	00000001	00010011
[3776, 3839]	00000000	01010001	00000000	01010011	10000000	01000001	00010011	01111111
[3840, 3903]	00000001	00000000	01100001	10000000	00100001	00000001	00000001	00010011
[3904, 3967]	00000000	01000000	00000000	01000000	00000000	00000000	00000000	00010011
[3968, 4031]	00000000	10000000	00000000	00010011	00000001	00000001	01010001	01111111
[4032, 4095]	00000000	01010000	00000000	01110011	00000001	01010100	00110001	01110111

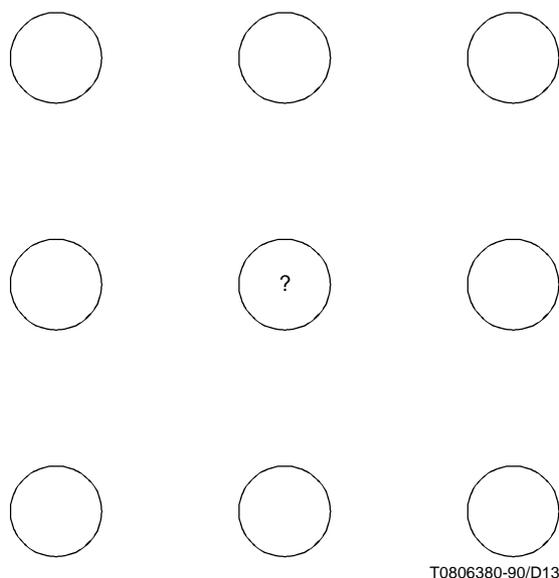


Figure 5 – Définition d'un voisinage de huit pixels

La Figure 7 représente deux lignes à haute résolution, avec la ligne à basse résolution associée. Est également représentée, la position virtuelle du codage de la valeur de la L_{NTP} .

Comme suggéré sur cette figure, la valeur L_{NTP} du pseudo-pixel doit être déterminée par le codeur arithmétique avant le codage de tout pixel de haute résolution normal des deux lignes. Lors du codage de ce pseudo-pixel, TP_{VALUE} et DP_{VALUE} ont toujours la valeur de 2. Le contexte CX à utiliser pour le codage est le même que celui utilisé pour la phase 3 du codage d'un pixel ordinaire, dont l'environnement de pixels est représenté Figure 8. Sur la figure «F» indique l'avant-plan et «B» l'arrière-plan.

NOTE – Ce contexte particulier se présente rarement, c'est pourquoi il a été choisi pour être réutilisé comme contexte de codage de L_{NTP} : pour la plupart des images, la probabilité de codage de l'arrière-plan dans ce contexte est aussi faible que celle que L_{NTP} soit égale à 1. La solution la plus évidente serait de coder L_{NTP} dans son propre contexte, malheureusement elle porterait le nombre total de contextes à une valeur légèrement supérieure à une puissance de 2.

La Figure 9 montre le traitement nécessaire pour produire le signal de sortie TP_{VALUE} . Explicitement, si L_{NTP} est égale à 0 et que le pixel de basse résolution associé à un pixel de haute résolution PIX a la même couleur que tous les pixels de son voisinage de huit, TP_{VALUE} est égale à cette couleur. Autrement, TP_{VALUE} est mise à 2 pour indiquer qu'aucune prédiction n'est possible.

6.4.2 Traitement dans un décodeur

Le pseudo-pixel $TP_{L_{NTP}}$ est décodé au début de chaque couple de lignes à haute résolution (voir la Figure 7). Lors du décodage de L_{NTP} , les entrées TP_{VALUE} et DP_{VALUE} sont égales à 2 et le contexte CX doit être tel que décrit en 6.4.1.

Lors du décodage d'un pixel donné de haute résolution PIX , TP_{VALUE} est générée comme dans un codeur.

6.5 Prédiction typique dans la couche de résolution la plus basse

La TP dans la couche de résolution la plus basse peut être activée ou désactivée par le bit TP_{BON} du champ «options» du BIH. Si la TP est désactivée ($TP_{BON} = 0$), le bloc TPB (d'un codeur ou d'un décodeur) donne simplement une valeur de TP_{VALUE} égale à 2 pour tous les pixels, pour indiquer aux blocs de codage et de décodage qu'aucune prédiction n'est faite. En outre, lorsque la TP est désactivée dans la couche de résolution la plus basse, le pseudo-pixel SL_{NTP} ne doit être ni codé ni décodé par le (dé)codeur arithmétique. On suppose dans la suite de ce paragraphe que la TP dans la couche de résolution la plus basse est activée ($TP_{BON} = 1$).

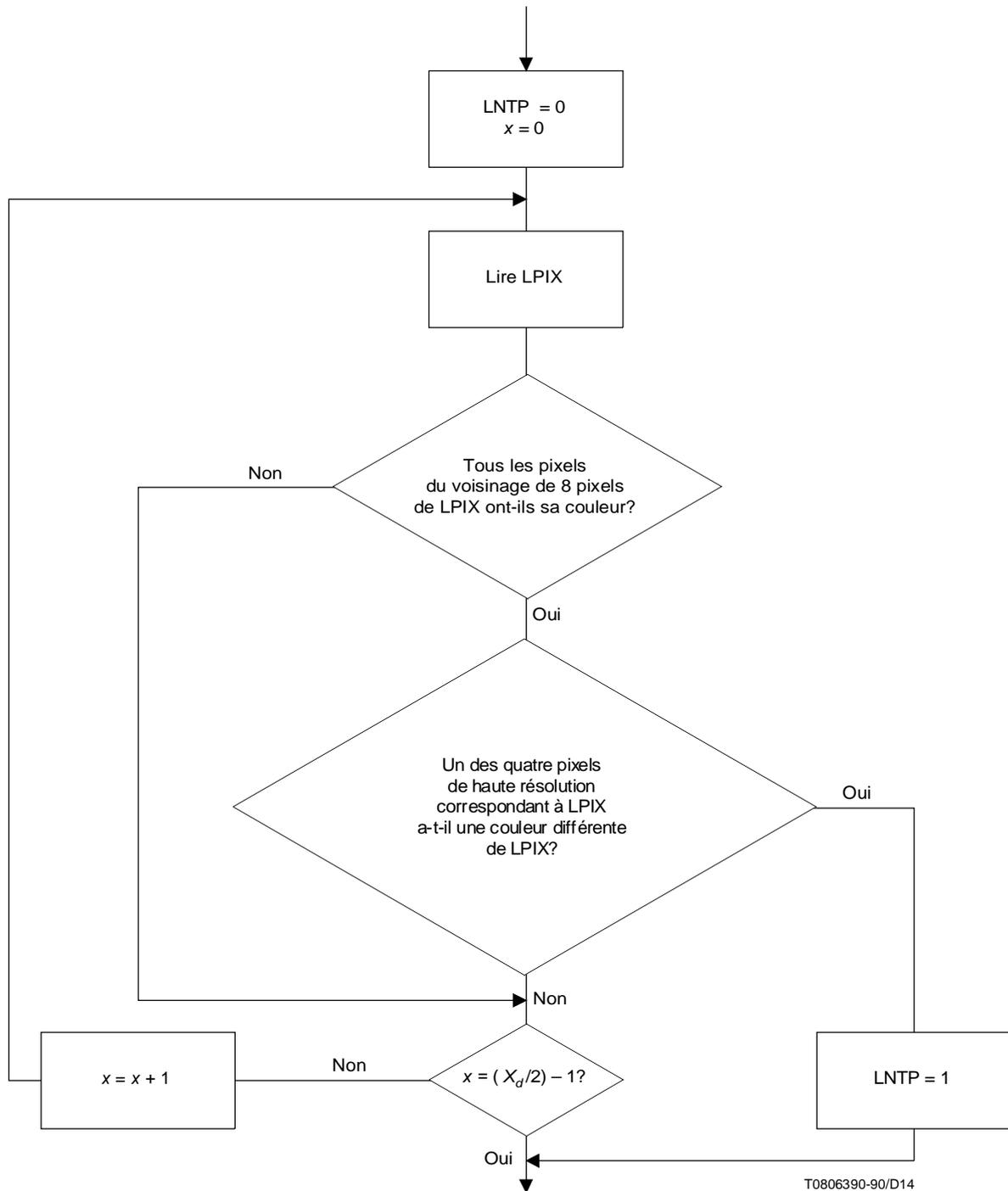


Figure 6 – Traitement de détermination de la LNTP

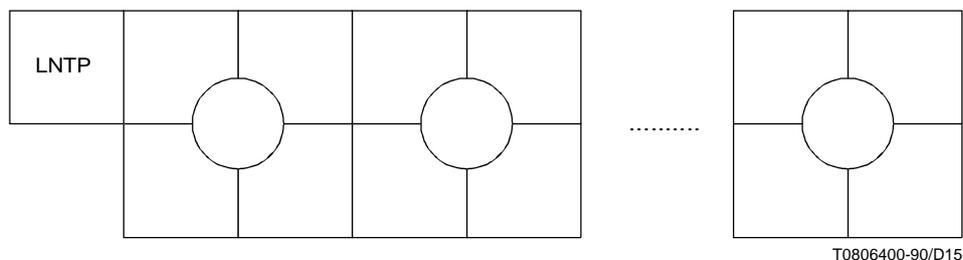


Figure 7 – Position d'un pseudo-pixel par rapport à des pixels ordinaires

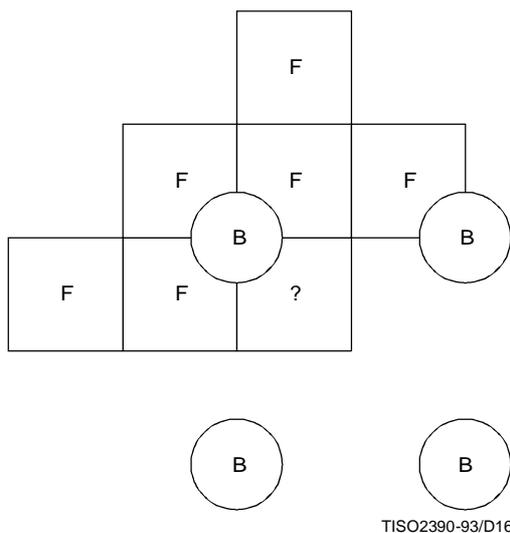


Figure 8 – Contexte réutilisé pour le codage du pseudo-pixel de TP de couche différentielle

6.5.1 Traitement dans un codeur

Soit y la ligne courante: si elle diffère, pour une position de pixel, de la ligne du dessus, $LNTPy$ doit être égale à 1 et la ligne y est dite atypique. Autrement, $LNTPy$ est égale à zéro. Pour déterminer si la toute première ligne d'une image est atypique, il faut supposer que la ligne se trouvant immédiatement au-dessus de l'image est, selon la convention usuelle, dans la couleur d'arrière-plan.

NOTE – Alors que presque toutes les lignes sont «typiques» au sens de la TP de couche différentielle, seule une faible fraction est «typique» au sens de la TP de la couche de résolution la plus basse.

Soit, par définition:

$$SLNTPy = !(LNTPy \oplus LNTPy-1) \tag{5}$$

où le symbole \oplus indique l'opération ou exclusif et le symbole $!$ indique la négation logique. Explicitement, $SLNTPy$ est égale à 1 si et seulement si $LNTPy$ a la même valeur que $LNTPy-1$. Pour la ligne du haut d'une image, $LNTPy-1$ est mise à 1.

Lorsque la TP dans la couche de résolution la plus basse est activée, un pseudo-pixel de valeur égale à $SLNTP$ doit être codé à la position virtuelle indiquée, avant le codage de tout pixel de la ligne y (voir la Figure 10).

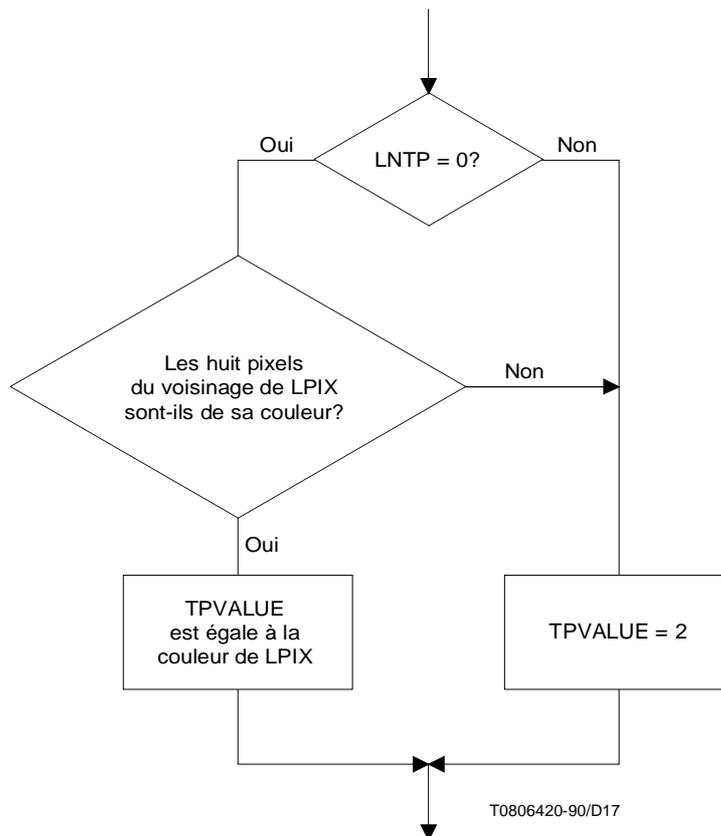


Figure 9 – Traitement de détermination de TPVALUE

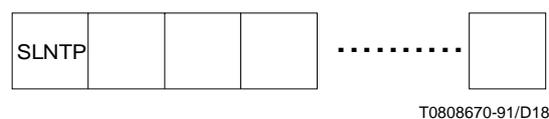


Figure 10 – Position du pseudo-pixel par rapport aux pixels ordinaires

$SLNTP$ doit être codée dans le contexte représenté sur la Figure 11 si $LRLTWO$ est égale à zéro et dans celui représenté sur la Figure 12 si $LRLTWO$ est égale à 1 (voir 6.7.1). Sur ces figures «F» désigne l'avant-plan et «B» l'arrière-plan. Lors du codage de $SLNTP$, $TPVALUE$ est égale à 2. Autrement dit, $SLNTP$ ne peut jamais être prédite par une TP et doit toujours faire l'objet d'un codage arithmétique.

NOTE – Le codage arithmétique des changements de valeur de la $LNTP$ est plus efficace que le codage arithmétique direct de la $LNTP$. Dans la TP de la couche de plus basse résolution, la $LNTP$ n'a pas une forte probabilité de prendre la valeur 1 ou 0, et ne peut donc pas donner lieu à un codage entropique d'une grande efficacité.

Si $LNTP_y$ est égale à 0, le bloc TPB doit donner comme $TPVALUE$ la valeur commune du pixel courant et du pixel du dessus. Autrement, il doit sortir la valeur 2 pour indiquer qu'aucune prédiction ne peut être faite.

6.5.2 Traitement dans le décodeur

Si TPBON est égale à 1, l'indicateur de similitude SLNTP_y doit être décodé (voir la Figure 10). Lors du décodage de SLNTP, TPVALUE doit être égale à 2 et CX doit être le contexte approprié, tel qu'indiqué par la Figure 11 ou la Figure 12.

		B	B	F	
F	F	B	B	F	
B	F	?			TISO2400-93/D19

Figure 11 – Contexte réutilisé pour le codage du pseudo-pixel de la TP de la couche de plus basse résolution (gabarit de trois lignes)

		B	F	F	B	B	F
B	F	B	F	?			TISO2410-93/D20

Figure 12 – Contexte réutilisé pour le codage du pseudo-pixel de la TP de la couche de plus basse résolution (gabarit de deux lignes)

Le décodeur recrée LNTP_y par:

$$LNTP_y = !(SLNTP_y \oplus LNTP_{y-1}) \tag{6}$$

Comme dans le décodeur, cette itération doit être initialisée en mettant LNTP à 1 pour la ligne se trouvant immédiatement au-dessus de la ligne du haut de l'image réelle.

Si LNTP_y est égale à 0, le bloc TPB doit sortir comme valeur de TPVALUE, la valeur du pixel immédiatement supérieur au pixel courant. Autrement, il sort 2 pour indiquer qu'aucune prédiction ne peut être faite.

6.6 Prédiction déterministe (DP)

La prédiction déterministe DP peut être activée ou désactivée par le bit DPON du champ option du BIH. Si la DP est désactivée (DPON = 0), le bloc DP (d'un codeur ou d'un décodeur) sort simplement DPVALUE = 2 pour tous les pixels. Il est supposé dans la suite de ce paragraphe que la DP est activée (DPON = 1).

Si la DP est utilisée lors du codage d'une image, il faut supposer que les tables de DP décrites dans ce paragraphe ont été utilisées à des fins de prédiction, sauf si l'utilisation de tables de DP privées a été signalée comme précisé en 6.2.

6.6.1 Définition des pixels associés

La Figure 13 montre comment sont désignés les pixels des images à basse et haute résolution, utilisés pour décrire l'algorithme de prédiction déterministe. Chaque fois qu'il est fait référence à un pixel ne se trouvant pas réellement dans la bande courante à cause d'effets de bord, la valeur de ce pixel est déterminée par les règles générales s'appliquant aux bords de 6.1.2.

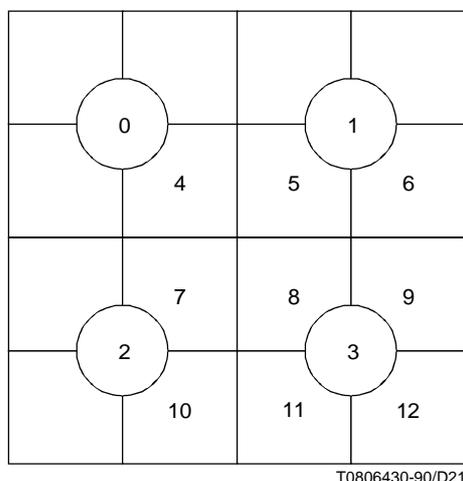


Figure 13 – Désignation des pixels utilisés par la DP

6.6.2 Tables de DP par défaut

Les pixels de voisinage ou «de référence» utilisés pour faire les prédictions relatives à chaque phase spatiale sont indiqués dans le Tableau 18. A noter qu'un ensemble différent de pixels est utilisé pour la prédiction DP de chacune des quatre phases spatiales. Les pixels utilisés pour chaque phase spatiale sont indiqués sur la Figure 13, et sont connus du codeur et du décodeur au début du codage de chaque phase. Figure également dans ce tableau le nombre de combinaisons de pixels de référence donnant lieu à une prédiction DP effective (une «concordance»), lors de l'application des règles de DP ci-après.

Tableau 18 – Pixels DP de chaque phase spatiale

Phase	Pixel cible	Pixels de référence	Nombre de concordances avec la réduction de résolution par défaut
0	8	0, 1, 2, 3, 4, 5, 6, 7,	20
1	9	0, 1, 2, 3, 4, 5, 6, 7, 8	108
2	11	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	526
3	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	1044

Les tables de DP privées ne doivent utiliser, pour aucune phase, d'autres pixels que ceux indiqués dans le Tableau 18. Le nombre de motifs de pixels de référence qui «concorde» sera en général différent des nombres indiqués dans le Tableau 18 pour l'algorithme de réduction de résolution par défaut.

Les Tableaux 19, 20, 21 et 22 définissent la DP pour l'algorithme de réduction de résolution par défaut. Ces quatre tableaux doivent être utilisés pour la détermination de DP_{VALUE} , respectivement pour chacune des quatre phases spatiales 0, 1, 2 et 3. L'index figurant dans les tableaux est créé de la même façon que celui de la table de réduction de résolution du Tableau 17, sauf que le poids du bit doit être tel que défini par les numéros de pixel de la Figure 13 et non de la Figure 4.

Les entrées de ces tableaux donnent DP_{VALUE} et peuvent valoir 0, 1 ou 2. Un «2» indique qu'il n'est pas possible de faire une prédiction déterministe. Un «0» indique une DP «concordante», le pixel cible étant d'arrière-plan (0). Un «1» indique une DP «concordante», le pixel cible étant d'avant-plan (1). Comme dans le Tableau 17, les entrées sont lues de gauche à droite, par valeurs croissantes de l'index.

Tableau 19 – Table DP de prédiction de la phase spatiale 0

Indice	DPVALUE							
[0, 63]	02222222	22222222	22222222	22222222	02222222	22222222	22222222	22222222
[64, 127]	02222222	22222222	22222222	22222222	00222222	22222222	22222222	22222222
[128, 191]	02222222	22222222	00222222	22222222	02020222	22222222	02022222	22222222
[192, 255]	00222222	22222222	22222222	22222221	02020022	22222222	22222222	22222222

Tableau 20 – Table DP de prédiction de la phase spatiale 1

Indice	DPVALUE							
[0, 63]	22222222	22222222	22222222	22000000	02222222	22222222	00222222	22111111
[64, 127]	22222222	22222222	22222222	21111111	02222222	22111111	22222222	22112221
[128, 191]	02222222	22222222	02222222	22222222	00222222	22222200	20222222	22222222
[192, 255]	02222222	22111111	22222222	22222102	11222222	22222212	22220022	22222212
[256, 319]	20222222	22222222	00220222	22222222	20000222	22222222	00000022	22222221
[320, 383]	20222222	22222222	11222222	22222221	22222222	22222221	22221122	22222221
[384, 447]	20020022	22222222	22000022	22222222	20202002	22222222	20220002	22222222
[448, 511]	22000022	22222222	00220022	22222221	21212202	22222222	22220002	22222222

Tableau 21 – Table DP de prédiction de la phase spatiale 2

Indice	DPVALUE							
[0, 63]	22222222	12222222	22222222	22222222	02222222	12222222	02222222	11222222
[64, 127]	22222222	22222222	02222222	12222222	02222222	11222222	22221122	22222222
[128, 191]	00202222	11111111	00200222	11111111	00222222	21122222	10222222	22111222
[192, 255]	02222222	11222222	00222222	21222222	22222222	22202220	22220022	22112222
[256, 319]	20222222	21222222	20020022	22222222	20000222	22222220	22000022	22222212
[320, 383]	20220222	22211111	22020222	22112122	22000022	22122122	22002222	22222212
[384, 447]	20020022	22222200	22000022	22222212	22202022	22222222	20202202	22222212
[448, 511]	22202022	22222200	00002022	22222212	22222202	22222221	22002220	22212221
[512, 575]	02222222	22111122	02222222	11222222	22212122	22220000	22122122	22202000
[576, 639]	00000000	22222222	02222222	22000000	22002222	22222222	22002112	22222222
[640, 703]	20222222	21221122	20222222	22121222	22000022	22112122	02222222	22212222
[704, 767]	20222222	22222022	00222222	21111222	02000022	22222200	22002212	22222222
[768, 831]	22020222	22111122	22222022	22222200	22222022	22222212	22222202	22222221
[832, 895]	22000022	22222222	00201222	22222220	22022202	22222222	22002200	22222222
[896, 959]	22202222	22222222	22222202	22222221	22222222	22222221	22202220	22222222
[960, 1023]	22222222	22222222	22002202	22222221	20202220	22222222	22002220	22222222
[1024, 1087]	22222222	11111111	22222222	11111111	02222222	21111111	22222222	22222222
[1088, 1151]	22222222	11111111	02222222	22222222	02222222	21222222	22222222	22222002
[1152, 1215]	00222222	11111111	22222202	22221121	12222222	22222212	22002202	22221111
[1216, 1279]	02222222	21222222	22222222	22201202	22220222	22101222	22000022	22222221
[1280, 1343]	20222222	22112111	11020222	22211111	22202002	22222222	00202222	22222212
[1344, 1407]	22020022	22222211	22000022	22222212	22222022	22222212	22222202	22222212
[1408, 1471]	22002022	22211211	22222212	22222221	21222102	22222221	22222222	22222121
[1472, 1535]	22121122	22222222	22111122	22220222	22222200	22222221	22000002	22222221
[1536, 1599]	00000000	11111111	02222222	21222222	20222222	22121111	22220222	22121122
[1600, 1663]	00222222	22222222	22222222	21121222	20020222	22111111	22220022	22212122
[1664, 1727]	20220022	22111111	22020022	22221121	22000022	22222122	22220022	22222211
[1728, 1791]	20020222	22111111	22002222	22211122	11122222	22222111	22222202	22222210
[1792, 1855]	22200022	22222222	22122022	22222212	22222202	22222211	22222200	22222221
[1856, 1919]	22222022	22222222	22121222	22222222	22000000	22222211	22000000	22222211
[1920, 1983]	22222202	22222211	22222220	22222221	22222220	22222221	22222222	22222222
[1984, 2047]	22222200	22222221	22222220	22222221	22222222	22222222	22222220	22222222

Tableau 22 – Table DP de prédiction de la phase spatiale 3

Indice	DPVALUE							
[0, 63]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
[64, 127]	22222222	22222222	22222222	22222222	22222222	22222222	22222202	22222212
[128, 191]	22222222	22222222	22222222	22222222	02222222	12222222	20222222	21222222
[192, 255]	22222222	22222222	02222222	12222222	22111121	22000020	22221122	22220022
[256, 319]	22222222	22222222	20000000	21111111	20000000	21111111	22000022	22111122
[320, 383]	20022222	21122222	22221222	22220222	22200222	22211222	22002222	22112222
[384, 447]	20000000	21111111	22000022	22111122	22202022	22212122	20202020	21212121
[448, 511]	22212111	22202000	00002022	11112122	22222212	22222202	22022222	22122222
[512, 575]	02222222	12222222	22222222	22222222	22020200	22121211	22211211	22200200
[576, 639]	00000000	11111111	00000000	11111111	22000000	22111111	22002220	22112221
[640, 703]	22222222	22222222	20222222	21222222	22221222	22220222	02020122	12121022
[704, 767]	20000000	21111111	02222222	12222222	02000100	12111011	22002220	22112221
[768, 831]	22222222	22222222	22222111	22222000	22222022	22222122	22222202	22222212
[832, 895]	22000000	22111111	00202000	11212111	22022200	22122211	22002210	22112201
[896, 959]	22202212	22212202	22222212	22222202	22222222	22222222	22202220	22212221
[960, 1023]	22222220	22222221	22002202	22112212	20202220	21212221	22002220	22112221
[1024, 1087]	22222222	22222222	22222222	22222222	02222222	12222222	02222222	12222222
[1088, 1151]	22222222	22222222	02222222	12222222	02222222	12222222	22221112	22220002
[1152, 1215]	22222222	22222222	22222202	22222212	22111121	22000020	22112222	22002222
[1216, 1279]	02222222	12222222	22112212	22002202	22212122	22202022	22010122	22101022
[1280, 1343]	20220222	21221222	22122222	22022222	22212111	22202000	00200022	11211122
[1344, 1407]	22111122	22000022	22000022	22111122	22222022	22222122	22222222	22222222
[1408, 1471]	22022022	22122122	22220022	22221122	22222212	22222202	22220222	22221222
[1472, 1535]	22202222	22212222	22222222	22222222	22222022	22222212	22111112	22000002
[1536, 1599]	22222222	22222222	02222222	12222222	20222222	21222222	22222222	22222222
[1600, 1663]	00000000	11111111	22222222	22222222	20222222	21222222	22020222	22121222
[1664, 1727]	20222222	21222222	22112212	22002202	22010222	22101222	22221122	22220022
[1728, 1791]	21222222	20222222	22022222	22122222	22201222	22210222	22222220	22222221
[1792, 1855]	22211111	22200000	22202022	22212122	22222222	22222222	22222202	22222212
[1856, 1919]	22222000	22222111	22222202	22222212	22111122	22000022	22001122	22110022
[1920, 1983]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
[1984, 2047]	22222202	22222212	22222222	22222222	22222222	22222222	22222220	22222221
[2048, 2111]	02222222	12222222	22222222	22222222	22222222	22222222	20222222	21222222
[2112, 2175]	22222222	22222222	22222222	22222222	20222222	21222222	22221112	22220002
[2176, 2239]	22020000	22121111	22122111	22022000	22122222	22022222	02010222	12101222
[2240, 2303]	20222222	21222222	22222222	22222222	22020122	22121022	22112222	22002222
[2304, 2367]	22222222	22222222	22212211	22202200	22222012	22222102	22222212	22222202
[2368, 2431]	22112111	22002000	22202022	22212122	22222222	22222222	22222222	22222222
[2432, 2495]	22202222	22212222	22222202	22222212	22222222	22222222	22222020	22222121
[2496, 2559]	22222222	22222222	22222202	22222212	22222220	22222221	22222222	22222222
[2560, 2623]	22111122	22000022	20222222	21222222	22112222	22002222	22020222	22121222
[2624, 2687]	22222222	22222222	21222222	20222222	22220000	22221111	22222000	22222111
[2688, 2751]	22221122	22220022	22020222	22121222	22222222	22222222	22111122	22000022
[2752, 2815]	22000200	22111211	22201222	22210222	22222222	22222222	22222200	22222211
[2816, 2879]	22202022	22212122	22222222	22222222	22222202	22222212	22222220	22222221
[2880, 2943]	22222200	22222211	22221102	22220012	22222220	22222221	22222222	22222222
[2944, 3007]	22222202	22222212	22222220	22222221	22222220	22222221	22222222	22222222
[3008, 3071]	22222220	22222221	22222220	22222221	22222222	22222222	22222222	22222222
[3072, 3135]	00000000	11111111	00000000	11111111	20000000	21111111	02222222	12222222
[3136, 3199]	00000000	11111111	22222222	22222222	21222222	20222222	22220222	22221222
[3200, 3263]	22000000	22111111	22220020	22221121	02010000	12101111	22220020	22221121
[3264, 3327]	20222222	21222222	22020222	22121222	22122022	22022122	22222220	22222221
[3328, 3391]	22000000	22111111	00202000	11212111	22222220	22222221	22220002	22221112
[3392, 3455]	22202200	22212211	22222202	22222212	22222202	22222212	22222222	22222222
[3456, 3519]	22220200	22221211	22220010	22221101	20222020	21222121	22220020	22221121
[3520, 3583]	22111122	22000022	22112022	22002122	22222222	22222222	22222220	22222221
[3584, 3647]	22222222	22222222	21222222	20222222	22020000	22121111	22122022	22022122
[3648, 3711]	22000000	22111111	21020222	20121222	22202000	22212111	22002222	22112222
[3712, 3775]	22002200	22112211	22202200	22212211	22222222	22222222	22222200	22222211
[3776, 3839]	22202000	22212111	22220022	22221122	00000000	11111111	22222222	22222222
[3840, 3903]	22222200	22222211	22112202	22002212	22222220	22222221	22222222	22222222
[3904, 3967]	22222200	22222211	22121212	22020202	22222222	22222222	22222222	22222222
[3968, 4031]	22222220	22222221	22222222	22222222	22222222	22222222	22222222	22222222
[4032, 4095]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222

6.6.3 Format de la table DP

Si $DPON = 1$, $DPPRIV = 1$ et $DPLAST = 0$, la table DP privée doit être codée dans le champ $DPTABLE$ du BIH (voir 6.2). Le champ $DPTABLE$ est défini en mettant côte à côte les quatre tables ci-dessus, qui définissent la DP pour l'algorithme de réduction de résolution par défaut. Les quatre tables sont concaténées, deux bits étant affectés à chacune, en sorte que les quatre entrées soient représentées par un octet. Dans les tables, les entrées imprimées en haut et à gauche sont contenues dans les bits de plus fort poids des premiers octets du champ $DPTABLE$. Lorsqu'il n'est pas possible de faire de prédiction DP, la valeur 2 doit être codée dans le champ de deux bits.

NOTE – Ainsi, le bit de plus fort poids du champ de deux bits devient un indicateur d'impossibilité de prédiction DP.

Comme les pixels de référence 8, 9, 11 et 12 de prédiction des pixels respectifs des phases spatiales 0, 1, 2 et 3, le champ $DPTABLE$ aura une longueur de:

$$1728 = 2 \times (256 + 512 + 2048 + 4096) / 8 \tag{7}$$

octets.

6.7 Gabarits modèles et gabarits adaptatifs

Les gabarits modèles définissent un voisinage autour d'un pixel à coder. Les valeurs des pixels de ce voisinage, complétées par la phase spatiale dans les couches différentielles, définissent un contexte: un codeur arithmétique séparé est utilisé pour chaque contexte (voir 6.8). Un gabarit n'est rien d'autre qu'un motif géométrique de pixels: il est cependant d'usage de dire que les pixels d'un gabarit «prennent des valeurs» lorsque ce gabarit est appliqué sur une partie d'une image.

6.7.1 Couche de plus basse résolution

La Figure 14 montre le gabarit à utiliser lors du codage de la couche de plus basse résolution, lorsque $LRLTWO$ vaut 0.

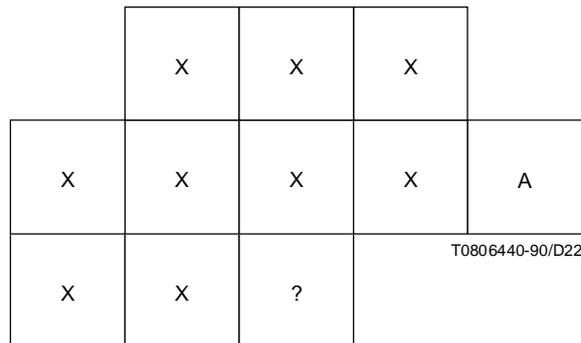


Figure 14 – Gabarit modèle à trois lignes, pour la couche de plus basse résolution

Le pixel «?» correspond au pixel à coder et ne fait pas partie du gabarit. Les pixels «X» correspondent aux pixels ordinaires du gabarit et le pixel «A» est un pixel spécial du gabarit, appelé «pixel adaptatif» ou «pixel AT». Ce pixel est spécial en ce sens que sa position peut changer durant le processus de codage d'une image. Voir 6.7.3 pour une description des pixels AT. «A» indique la position initiale du pixel AT.

Les valeurs des pixels du gabarit sont combinées pour former un contexte. Chaque pixel du gabarit modèle (y compris le pixel adaptatif) correspond à un bit spécifique du contexte (mais les pixels du gabarit peuvent être affectés aux bits du contexte dans n'importe quel ordre). Comme le gabarit comporte 10 pixels, les contextes associés à la couche de plus basse résolution peuvent prendre 1024 valeurs différentes. Ce contexte est utilisé pour déterminer l'adaptateur au codeur arithmétique à utiliser pour le codage du pixel cible, comme décrit en 6.8.

Si $LRLTWO$ est égale à 1, le gabarit modèle de la couche de plus basse résolution est tel que représenté sur la Figure 15.

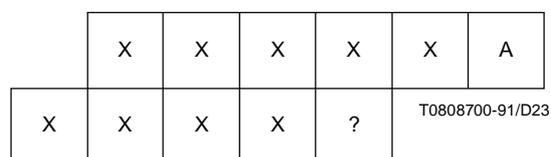


Figure 15 – Gabarit modèle de deux lignes pour la couche de plus basse résolution

Les significations de «X», «A» et «?» sont telles qu'indiquées plus haut.

NOTE – La vitesse du traitement par un logiciel, de la couche de résolution la plus basse sera un peu plus élevée avec le gabarit à 2 lignes qu'avec celui à 3 lignes. La pénalisation entraînée par l'utilisation d'un gabarit à 2 lignes se traduit par une perte d'environ 5% sur l'efficacité de compression.

Chaque fois qu'un des pixels du gabarit de la Figure 14 ou 15 (comme indiqué par $LRLTWO$) se trouve hors des limites de l'image ou de la bande, les règles générales relatives aux bords données en 6.1.2 doivent être appliquées.

6.7.2 Couche différentielle

La Figure 16 représente les gabarits à utiliser lors du codage d'images de couche différentielle. A noter que ces gabarits contiennent des références aux pixels de l'image de niveau de résolution immédiatement inférieur, ainsi qu'à des pixels de l'image en cours de codage, et que le gabarit modèle est différent pour chaque phase. Les symboles «?», «X» et «A» ont la même signification que dans le paragraphe précédent.

Les contextes sont formés à partir de ces gabarits, d'une façon similaire à celle décrite pour le gabarit de la couche de plus basse résolution. Chaque pixel d'un gabarit contribue à un bit du contexte. Lors du codage d'images de couche différentielle, deux bits additionnels sont ajoutés au contexte pour préciser la phase du pixel codé. Comme plus haut, n'importe quel bit peut être utilisé pour décrire l'information de phase, mais l'affectation des pixels et de l'information de phase aux bits du contexte reste fixe durant tout le codage d'une image. Les gabarits de couche différentielle comportant 10 pixels et 2 bits étant utilisés pour décrire l'information de phase, 4096 contextes différents sont possibles lors du traitement d'images différentielles. Le contexte ainsi formé est utilisé pour identifier l'adaptateur du codeur arithmétique à utiliser pour coder le pixel cible.

6.7.3 Pixels de gabarits adaptatifs

Lors du codage des couches différentielles, ainsi que de la couche de plus basse résolution, le gabarit modèle peut évoluer, dans les limites décrites ci-après.

Le seul pixel autorisé à se déplacer est appelé «pixel AT». Sa position initiale (ou par défaut) est indiquée par «A» sur les Figures 14, 15 et 16, représentant respectivement le codage de la couche de plus basse résolution avec un gabarit à trois lignes, le codage de la couche de plus basse résolution avec un gabarit à deux lignes, et un codage de couche différentielle. En général, le pixel AT peut évoluer de façon indépendante dans toutes les couches, jusqu'à n'importe quelle position du champ représenté sur la Figure 17.

La nouvelle position du pixel AT ne doit toutefois coïncider avec aucun pixel normal du gabarit. (Les déplacements permis pour le codage de la couche de plus basse résolution sur deux lignes, le codage de la couche de plus basse résolution sur trois lignes et le codage de couche différentielle sont donc légèrement différents.)

Un déplacement du pixel AT de couche différentielle est effectué simultanément dans les quatre phases. Avec plusieurs couches différentielles, les déplacements dans chaque couche sont indépendants. Cependant, quelle que soit la couche, lorsque le pixel AT a été déplacé pour une bande donnée, les bandes suivantes doivent continuer à utiliser la nouvelle position.

Les paramètres M_X et M_Y définissent la taille du rectangle de la Figure 17. Les limites absolues et les plages minimales suggérées sont indiquées dans les Tableaux 9 et A.1.

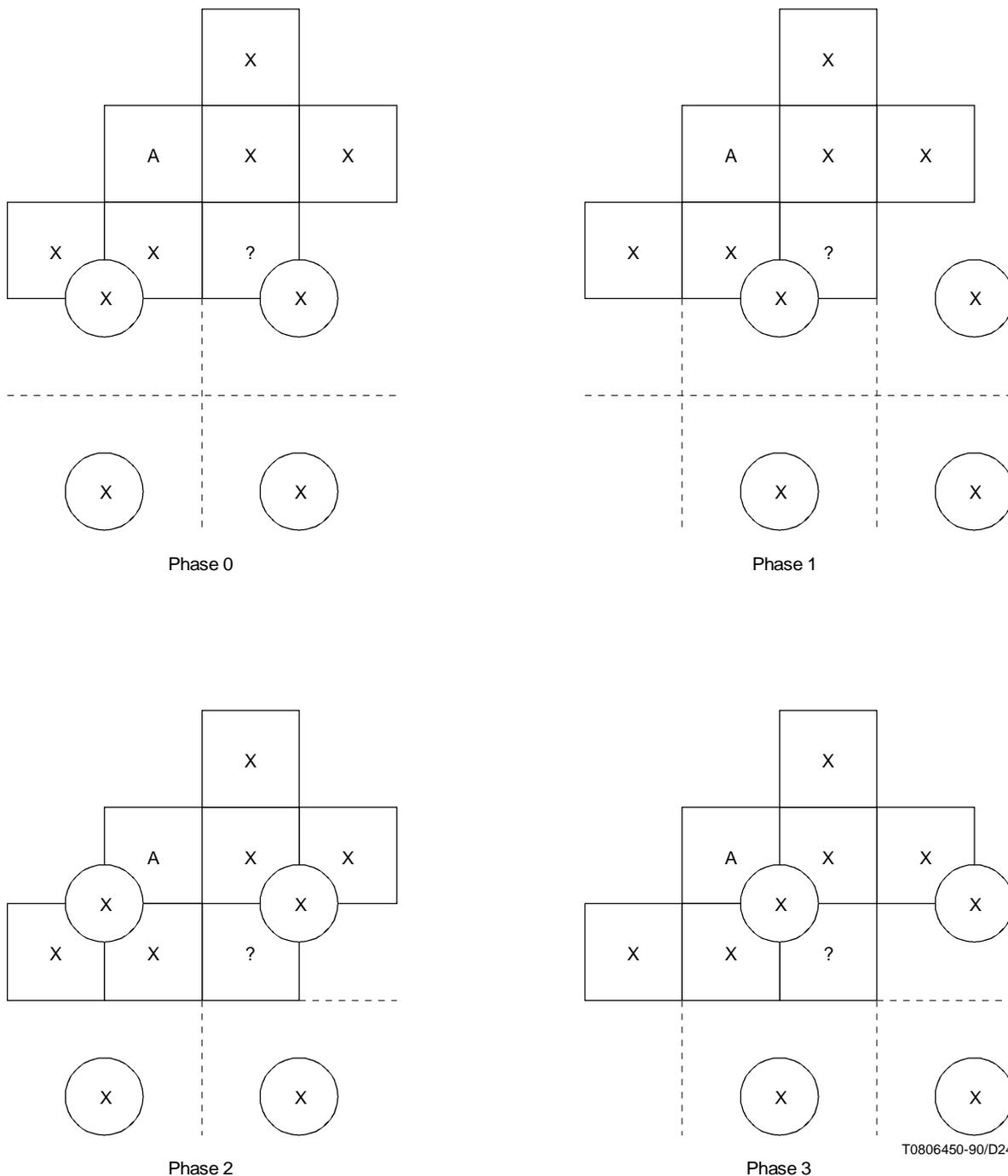


Figure 16 – Gabarits modèles pour le codage de couches différentielles

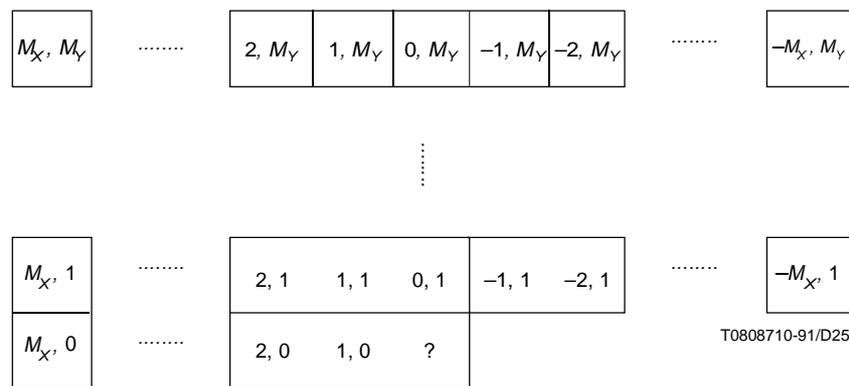


Figure 17 – Champs auxquels sont restreints les déplacements du pixel AT

NOTE – Comme il serait en général plus onéreux, pour le matériel, de prendre en charge des déplacements verticaux du pixel AT, la plage minimale suggérée pour M_Y est restreinte à 0: les déplacements du pixel AT vers un point différent de sa position par défaut sont restreints à la ligne en cours de codage.

Si un codeur désire changer la position du pixel AT, il doit en informer le décodeur, en codant τ_X , τ_Y et y_{AT} comme indiqué dans le Tableau 14. Les nombres codés dans τ_X et τ_Y sont respectivement les décalages horizontal et vertical par rapport au pixel cible, comme indiqué sur la Figure 17. Le nombre τ_X peut être négatif et doit être codé en complément à deux. y_{AT} code le numéro de la ligne à haute résolution au début de laquelle le déplacement doit être effectué. La numérotation des lignes doit reprendre à 0 en haut de chaque bande.

Le pixel AT est autorisé à revenir à sa position initiale (ou par défaut) après un déplacement. La position par défaut du pixel AT est toujours codée par $\tau_X=0$ et $\tau_Y=0$ et non par les coordonnées vraies X et Y.

NOTE – Cette convention est pratique, car les coordonnées vraies sont différentes pour le codage de la couche de plus basse résolution et le codage de couche différentielle. Elle permet en outre à un codeur d'informer un décodeur qu'il n'y aura jamais de déplacement AT, en mettant M_X et M_Y à 0.

L'Annexe C décrit une technique de calcul simple, pour déterminer l'opportunité de déplacement du pixel AT, ainsi que sa position cible.

6.8 Codage arithmétique

Le codeur entropique utilisé dans la présente Spécification est un codeur à compression arithmétique adaptatif. Dans l'ensemble de ce paragraphe, seuls les organigrammes et le Tableau 24 ont un caractère normatif, en ce sens qu'ils définissent une sortie que des réalisations différentes doivent reproduire. Tous les autres éléments et développements des paragraphes suivants ont un caractère informatif.

NOTE – Les opérations de codage arithmétique décrites dans ce paragraphe sont prévues identiques à celles décrites dans la Rec. T.81 du CCITT | ISO/CEI 10918-1. En cas de différence involontaire entre les deux descriptions, la procédure décrite dans la présente Spécification doit être utilisée.

Il est permis d'avoir jusqu'à quatre segments marqueurs *ATMOVE* dans chaque bande. Lorsque plusieurs segments marqueurs *ATMOVE* sont présentés dans une bande, leurs lignes effectives Y_{AT} doivent être distinctes et ordonnées, les premiers marqueurs *ATMOVE* ayant les valeurs numériques Y_{AT} les plus basses.

Pour chaque bande de chaque couche de résolution, le codeur arithmétique doit produire un train d'octets *SCD*. Il reçoit comme entrée 4 trains de données, dont chacun fournit une valeur pour chaque pixel de la bande en cours de codage. Comme montré sur la Figure 18, ces quatre entrées sont un pixel *PIX*, une valeur de contexte *CX*, une indication de *TP* *TPVALUE* et une indication de *DP* *DPVALUE*.

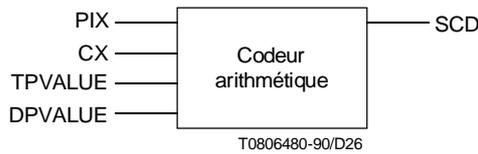


Figure 18 – Entrées et sorties du codeur

Le pixel à coder est généralement un pixel de l'image, mais si la TP de la couche de plus basse résolution ou la TP de couche différentielle sont activées, ce pourra être parfois un pseudo-pixel, LNTP (couche différentielle) ou SLNTP (couche de plus basse résolution). Les entrées CX, TPVALUE et DPVALUE sont générées comme décrit en 6.7, 6.4, 6.5 et 6.6.

Pour chaque bande de chaque couche de résolution, le décodeur arithmétique lit un train d'octets SCD. Comme représenté sur la Figure 19, ce train d'octets, associé aux trains d'entrées par pixel CX, TPVALUE et DPVALUE, doivent être traités pour recréer le train PIX.

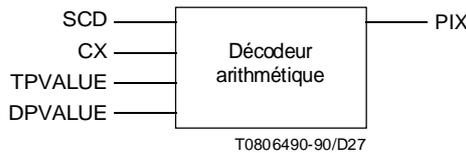


Figure 19 – Entrées et sorties du décodeur

Les entrées CX, TPVALUE et DPVALUE sont identiques à celles utilisées dans le codeur.

Pour décrire le codeur et le décodeur, le plus simple est de donner des exemples de procédures. Ces exemples de procédures sont définis par des organigrammes et une table. Beaucoup de procédures équivalentes existent: certaines présentent des avantages en matière de vitesse, d'utilisation de mémoire ou de simplicité; certaines sont plus appropriées aux mises en œuvre matérielles et d'autres aux mises en œuvre logicielles. Les choix présentés ici ont été retenus pour leur simplicité et leur concision. N'importe quelles procédures de codage ou de décodage produisant les mêmes sorties que les exemples de procédures peuvent être utilisées. Cette équivalence des sorties est la seule prescription.

6.8.1 Concepts fondamentaux du codage arithmétique (informatif)

6.8.1.1 Subdivision d'intervalle

Le codage arithmétique est basé sur la subdivision récursive de l'intervalle de probabilité. Conceptuellement, une séquence d'entrée de symboles est mise en correspondance avec un nombre réel x de l'intervalle $[0, 1)$, le crochet indiquant la possibilité d'égalité et la parenthèse l'inégalité stricte. Le développement binaire de x est transmis ou stocké à la place de la séquence de symboles d'origine.

La Figure 20 montre un exemple de subdivision d'intervalle, la séquence à coder étant 0, 1, 0, 0.

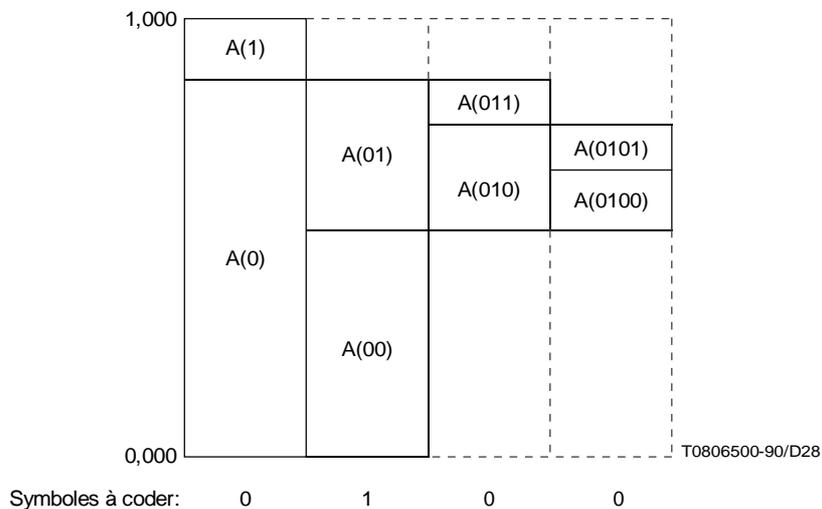


Figure 20 – Subdivision d'intervalle

L'intervalle $[0, 1)$ dans lequel on sait que se trouve x après codage d'une séquence initiale de symboles est appelé intervalle de codage courant. Pour chaque entrée binaire, l'intervalle de codage courant est divisé en deux sous-intervalles, de longueurs proportionnelles aux probabilités relatives des occurrences de valeurs des symboles. Le nouvel intervalle de codage est celui associé à la valeur de symbole qui s'est effectivement présentée. Dans un codeur, la connaissance de l'intervalle de codage courant est détenue sous la forme de deux variables, la première donnant sa longueur et la seconde sa base (limite inférieure). Le train de données de sortie est obtenu à partir de la variable pointant vers la base.

Lors de la division de l'intervalle courant en deux sous-intervalles, le sous-intervalle du symbole le moins probable (LPS) (*less probable symbol*) est ordonné au-dessus du sous-intervalle du symbole le plus probable (MPS) (*more probable symbol*). Ainsi, lorsque le LPS est codé, le sous-intervalle MPS est ajouté à la base. Cette convention de codage nécessite que les symboles soient reconnus comme MPS ou LPS, et non comme 0 ou 1. En conséquence, la longueur de l'intervalle LPS et le sens du MPS de chaque symbole doivent être connus pour pouvoir coder ce symbole.

Comme le train de données codées pointe toujours vers un nombre réel appartenant à l'intervalle de codage courant, le processus de décodage consiste à déterminer, pour chaque décision, vers quel sous-intervalle pointe la chaîne de codes. Cette détermination est également effectuée de façon récursive, selon le même processus de subdivision d'intervalle que dans le codeur. Chaque fois qu'une décision est décodée, le décodeur soustrait un intervalle que le codeur a ajouté au train de données codées. Ainsi, le train de données codées dans le décodeur est un pointeur dans l'intervalle courant, relativement à la base de cet intervalle courant.

Comme le processus de codage implique l'addition de fractions binaires, plutôt que la concaténation de mots codés entiers, les décisions binaires les plus probables peuvent souvent être codées à un coût bien inférieur à celui d'un bit par décision.

6.8.1.2 Conventions de codage et approximations

Il est possible d'effectuer ces opérations de codage en arithmétique entière à précision fixe. Un registre A contient la longueur de l'intervalle de codage courant, normalisé pour appartenir toujours à la plage $[0x8000, 0x10000]$ où le préfixe $0x$ désigne un entier hexadécimal. Chaque fois qu'un symbole A résultant d'un codage devient inférieur à $0x8000$, il est doublé, et ceci de façon récursive, jusqu'à devenir supérieur ou égal à $0x8000$. Ces doublages sont appelés des «renormalisations».

Un second registre, C , contient les bits de tête du train de données codées. Le registre C est doublé chaque fois que A est doublé. Périodiquement, pour empêcher le débordement de C , un octet de données est extrait des bits de plus fort poids du registre C et placé dans une mémoire tampon de chaîne de codes externe. La retenue doit, le cas échéant, être résolue avant de déclencher la sortie du contenu de cette mémoire tampon.

Une approximation arithmétique simple est utilisée pour la subdivision de l'intervalle. Pour un intervalle A , avec une estimation courante p de la probabilité du LPS, un calcul précis du sous-intervalle LPS nécessiterait une multiplication $p \times A$. On utilise à la place l'approximation:

$$p \times A \approx p \times \overline{A} = \text{LSZ} \quad (8)$$

où la barre de surlignement indique une moyenne de la densité de probabilité A et LSZ une quantité stockée égale à la valeur approchée de la longueur de l'intervalle du LPS. A étant restreint à la plage $[0x8000, 0x10000]$, son remplacement par sa moyenne statistique n'introduit pas une erreur trop importante. Il a été constaté empiriquement que la densité de probabilité de A est inversement proportionnelle à sa valeur.

Lors du codage du LPS, la valeur du sous-intervalle $A - \text{LSZ}$ du MPS est ajoutée au registre de code, l'intervalle de codage étant réduit de la valeur LSZ du sous-intervalle du LPS. Lors du codage du MPS, le registre de code n'est pas modifié et l'intervalle est ramené à $A - \text{LSZ}$. Si, lors de ces opérations, A devient inférieur à $0x8000$, il est ramené dans la plage appropriée par renormalisation de A et C .

Avec la procédure décrite ci-dessus, l'approximation associée à la subdivision de l'intervalle peut parfois rendre le sous-intervalle du LPS plus grand que le sous-intervalle du MPS. Par exemple, si la valeur de LSZ est $0,33 \times 0x10000$ et que A a la valeur minimale autorisée $0x8000$, une mise à l'échelle appropriée donne $1/3$ de l'intervalle pour le MPS et $2/3$ pour le LPS. Pour éviter cette inversion, lorsque l'on subdivise l'intervalle en utilisant cette approximation simple, les intervalles affectés au MPS et au LPS sont permutés chaque fois que l'intervalle du LPS est supérieur à l'intervalle du MPS. Cette «permutation conditionnelle» MPS/LPS ne peut avoir lieu que lorsqu'une renormalisation est nécessaire.

Chaque fois qu'une renormalisation est opérée, une procédure d'estimation de probabilité est invoquée pour estimer la nouvelle probabilité du contexte en cours de codage.

6.8.2 Codeur

6.8.2.1 Organigramme du codeur

La procédure de cet organigramme est exécutée pour chacune des bandes de chaque couche de résolution. Les pixels qui ne peuvent faire l'objet ni d'une prédiction typique ni d'une prédiction déterministe sont codés par la procédure ENCODE. La procédure d'initialisation INITENC est appelée à l'entrée, et la procédure de terminaison FLUSH à la sortie (voir la Figure 21).

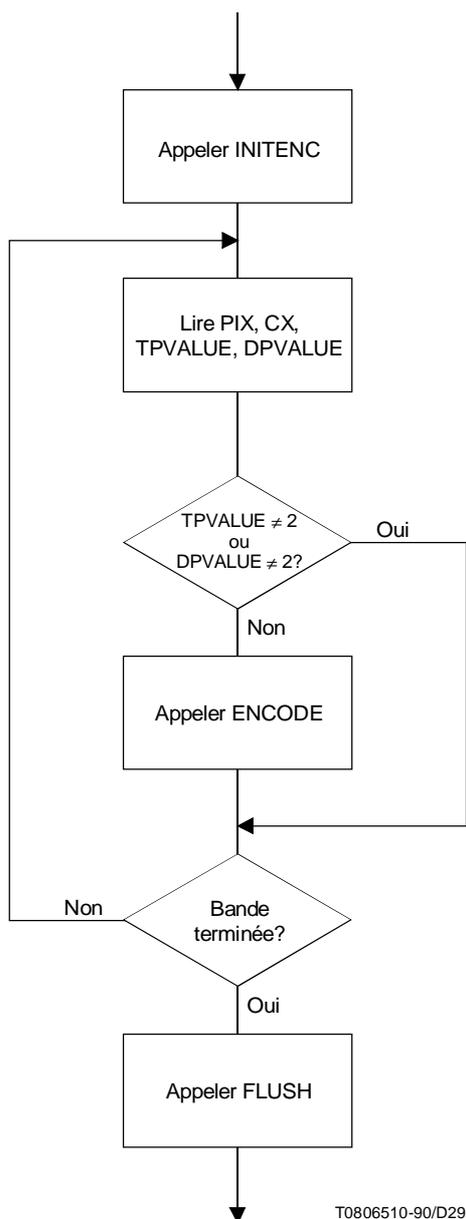


Figure 21 – Organigramme du codeur

6.8.2.2 Conventions relatives au registre de code du codeur

Les organigrammes de ce paragraphe supposent la structure de registre du Tableau 23.

Tableau 23 – Structure du registre du codeur

	msb		lsb	
Registre C	0000cbbb,	bbbbssss,	xxxxxxxx,	xxxxxxxx
Registre A	00000000,	0000000a,	aaaaaaaa,	aaaaaaaa

Les bits «a» sont les bits fractionnels de la valeur courante de l'intervalle, et les bits «x» les bits fractionnels du registre de code. Les bits «s» sont des bits d'espacement, dont au moins un est nécessaire pour forcer la retenue, les bits «b» indiquant les positions binaires à partir desquelles les octets complets de données sont retirés du registre C. Le bit «c» est un bit de retenue. Théoriquement, le 17^e bit du registre A figure: il est donc représenté ici, mais on peut facilement s'en passer pour une mise en œuvre sur 16 bits. Dans ce cas, l'initialisation à 0x0000 au lieu de 0x10000 donne des résultats corrects, à condition qu'un dépassement de capacité soit traité par le matériel ou le logiciel de prise en charge de façon à produire les mêmes 16 bits de bas poids lors d'une soustraction de 0x0000 que lors d'une soustraction de 0x10000. Un tel comportement est usuel.

Ces conventions relatives au registre sont un exemple de mise en œuvre possible. Concernant cet aspect, il existe de nombreuses possibilités.

6.8.2.3 Tables d'estimation de probabilité

Pour chaque valeur possible du contexte CX une valeur $MPS[CX]$ est stockée sur un bit et une valeur $ST[CX]$ sur sept bits, ces deux valeurs représentant complètement l'estimation de probabilité adaptative associée à ce contexte. Le Tableau 24 donne les valeurs de quatre variables indexées par $ST[CX]$.

La couleur MPS est la couleur (estimée) la plus probable du PIX. LSZ est la longueur de l'intervalle LPS, qui pourrait être interprétée en termes de probabilité selon l'équation (8) – mais cette interprétation n'est pas nécessaire, car seul LSZ intervient dans les calculs subséquents.

Les colonnes NLPS et NMPS donnent, respectivement, le prochain état d'estimation de probabilité d'observation du LPS et du MPS. Le déplacement déterminé par NMPS n'a lieu que si une renormalisation est effectuée. Lorsque le déplacement déterminé par NLPS a lieu, une inversion de $MPS[CX]$ doit être opérée si $SWTCH[CX] = 1$.

L'Annexe D (informative) explique comment ont été déterminées les entrées du Tableau 24.

6.8.2.4 Organigramme de la procédure ENCODE

Si le symbole courant PIX est égal à la valeur estimée la plus probable courante, la routine CODEMPS est appelée. Autrement, CODELPS est appelée (voir la Figure 22).

6.8.2.5 Organigramme de la procédure CODELPS

La procédure CODELPS consiste normalement en l'addition du sous-intervalle du MPS $A-LSZ[ST[CX]]$ au train de données codées, suivie d'une réduction du nouvel intervalle de codage à la longueur du sous-intervalle $LSZ[ST[CX]]$. Elle est toujours suivie d'une renormalisation. Si $SWTCH[ST[CX]] = 1$, $MPS[CX]$ est inversé.

Si le sous-intervalle du LPS est plus grand que le sous-intervalle du MPS, la permutation conditionnelle MPS/LPS est opérée, et le codage est appliqué au sous-intervalle du MPS (voir la Figure 23).

6.8.2.6 Organigramme de la procédure CODEMPS

La procédure CODEMPS réduit normalement la longueur de l'intervalle à celle du sous-intervalle du MPS. Toutefois, si le sous-intervalle du LPS est plus grand que le sous-intervalle du MPS, la permutation conditionnelle est opérée et le codage s'applique au sous-intervalle du LPS. A noter que cette inversion de longueur de l'intervalle ne peut avoir lieu sans qu'une renormalisation ne soit prévue après le codage du symbole (voir la Figure 24).

Tableau 24 – Table d'estimation de probabilité

ST	LSZ	NLPS	NMPS	SWTCH	ST	LSZ	NLPS	NMPS	SWTCH
0	0x5a1d	1	1	1	57	0x01a4	55	58	0
1	0x2586	14	2	0	58	0x0160	56	59	0
2	0x1114	16	3	0	59	0x0125	57	60	0
3	0x080b	18	4	0	60	0x00f6	58	61	0
4	0x03d8	20	5	0	61	0x00cb	59	62	0
5	0x01da	23	6	0	62	0x00ab	61	63	0
6	0x00e5	25	7	0	63	0x008f	61	32	0
7	0x006f	28	8	0	64	0x5b12	65	65	1
8	0x0036	30	9	0	65	0x4d04	80	66	0
9	0x001a	33	10	0	66	0x412c	81	67	0
10	0x000d	35	11	0	67	0x37d8	82	68	0
11	0x0006	9	12	0	68	0x2fe8	83	69	0
12	0x0003	10	13	0	69	0x293c	84	70	0
13	0x0001	12	13	0	70	0x2379	86	71	0
14	0x5a7f	15	15	1	71	0x1edf	87	72	0
15	0x3f25	36	16	0	72	0x1aa9	87	73	0
16	0x2cf2	38	17	0	73	0x174e	72	74	0
17	0x207c	39	18	0	74	0x1424	72	75	0
18	0x17b9	40	19	0	75	0x119c	74	76	0
19	0x1182	42	20	0	76	0x0f6b	74	77	0
20	0x0cef	43	21	0	77	0x0d51	75	78	0
21	0x09a1	45	22	0	78	0x0bb6	77	79	0
22	0x072f	46	23	0	79	0x0a40	77	48	0
23	0x055c	48	24	0	80	0x5832	80	81	1
24	0x0406	49	25	0	81	0x4d1c	88	82	0
25	0x0303	51	26	0	82	0x438e	89	83	0
26	0x0240	52	27	0	83	0x3bdd	90	84	0
27	0x01b1	54	28	0	84	0x34ee	91	85	0
28	0x0144	56	29	0	85	0x2eae	92	86	0
29	0x00f5	57	30	0	86	0x299a	93	87	0
30	0x00b7	59	31	0	87	0x2516	86	71	0
31	0x008a	60	32	0	88	0x5570	88	89	1
32	0x0068	62	33	0	89	0x4ca9	95	90	0
33	0x004e	63	34	0	90	0x44d9	96	91	0
34	0x003b	32	35	0	91	0x3e22	97	92	0
35	0x002c	33	9	0	92	0x3824	99	93	0
36	0x5ae1	37	37	1	93	0x32b4	99	94	0
37	0x484c	64	38	0	94	0x2e17	93	86	0
38	0x3a0d	65	39	0	95	0x56a8	95	96	1
39	0x2ef1	67	40	0	96	0x4f46	101	97	0
40	0x261f	68	41	0	97	0x47e5	102	98	0
41	0x1f33	69	42	0	98	0x41cf	103	99	0
42	0x19a8	70	43	0	99	0x3c3d	104	100	0
43	0x1518	72	44	0	100	0x375e	99	93	0
44	0x1177	73	45	0	101	0x5231	105	102	0
45	0x0e74	74	46	0	102	0x4c0f	106	103	0
46	0x0bfb	75	47	0	103	0x4639	107	104	0
47	0x09f8	77	48	0	104	0x415e	103	99	0
48	0x0861	78	49	0	105	0x5627	105	106	1
49	0x0706	79	50	0	106	0x50e7	108	107	0
50	0x05cd	48	51	0	107	0x4b85	109	103	0
51	0x04de	50	52	0	108	0x5597	110	109	0
52	0x040f	50	53	0	109	0x504f	111	107	0
53	0x0363	51	54	0	110	0x5a10	110	111	1
54	0x02d4	52	55	0	111	0x5522	112	109	0
55	0x025c	53	56	0	112	0x59eb	112	111	1
56	0x01f8	54	57	0					

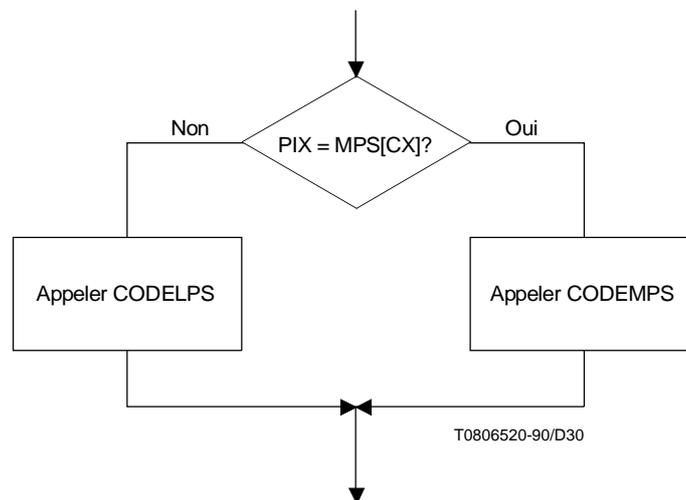


Figure 22 – Organigramme de la procédure ENCODE

6.8.2.7 Organigramme de la procédure RENORME

Le registre d'intervalle A et le registre de code C sont décalés, bit par bit. Le nombre de décalages est compté par le compteur CT: lorsque CT atteint zéro, un octet de données comprimées est extrait de C par la procédure BYTEOUT. La renormalisation continue tant que A n'est pas inférieur à 0x8000 (voir la Figure 25).

6.8.2.8 Organigramme de la procédure BYTEOUT

La procédure BYTEOUT est appelée par RENORME. TEMP est une variable temporaire qui contient l'octet de plus fort poids du registre C à présenter à la sortie avec l'indication de retenue. La variable BUFFER contient la plus récente sortie différente de 0xff. Le compteur SC contient le nombre d'octets 0xff écrits depuis la sortie de l'octet contenu dans BUFFER (voir la Figure 26).

Le décalage du registre de code de 19 bits aligne les bits de sortie «b» avec les bits de faible poids de TEMP. Un premier test détermine alors si une retenue a eu lieu. Dans l'affirmative, la retenue doit être ajoutée à l'octet de sortie contenu dans BUFFER avant sa présentation définitive à la sortie. Tous les octets de sortie empilés (convertis en zéros par la retenue) sont alors sortis. Enfin, le nouvel octet d'essai de sortie BUFFER est mis égal à TEMP moins la retenue.

Si aucune retenue n'a été générée, on vérifie si l'octet de sortie est égal à 0xff. Dans l'affirmative, le compteur de piles SC est incrémenté, car la sortie doit être retardée jusqu'à la résolution de la retenue. Dans la négative, la retenue a été résolue, et tous les octets 0xff peuvent être sortis.

NOTE – La probabilité que le compteur SC atteigne un entier donné n décroît rapidement, comme 2^{-8n} : en pratique, des valeurs de SC dépassant 3 ou 4 sont rarement observées dans le codage d'une image. Toutefois, SC peut, en principe, devenir aussi grand que le nombre d'octets du fichier d'entrée SCD. Chaque fois qu'une résolution de retenue est opérée, l'image d'entrée ne peut pas être traitée pendant la sortie des 0x00 ou 0xff octets de SC. Comme SC peut devenir en principe très grand, cette interruption peut être théoriquement très longue.

Il importe, pour une réalisation déterminée, de limiter la durée d'interruption du traitement de l'image d'entrée: le marqueur réservé peut être utilisé à cette fin de deux façons. Une première façon d'utiliser le marqueur réservé est de l'insérer chaque fois que SC atteint une petite valeur, par exemple 8, puis de décrémenter SC de 8. Lors d'une étape de post-traitement dans le codeur, chacun de ces marqueurs est remplacé, comme approprié, par 8 octets 0xff ou 8 octets 0x00. Ce post-traitement doit être effectué par le codeur, car aucun décodeur n'est censé avoir connaissance de cette utilisation du marqueur réservé – qui n'est possible que sur des données PSCD créées dynamiquement à partir des données SCD lors de leur génération.

Une deuxième façon d'utiliser le marqueur réservé à cette même fin est de permettre à SC d'atteindre des valeurs arbitrairement grandes. Si SC est supérieur à un certain nombre (par exemple 8) lors de la résolution finale de la retenue, un octet 0x00 ou 0xff lui est accolé pour indiquer la valeur de la retenue. La valeur effective de SC est codée à l'aide d'octets additionnels. Là encore, un post-traitement au niveau du codeur est nécessaire pour remplacer ce marqueur par le nombre approprié d'octets 0xff ou 0x00.

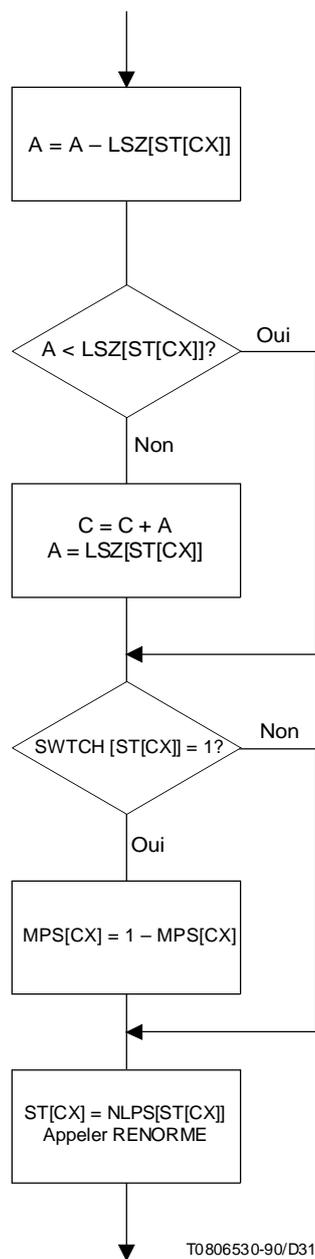


Figure 23 – Organigramme de procédure CODELPS

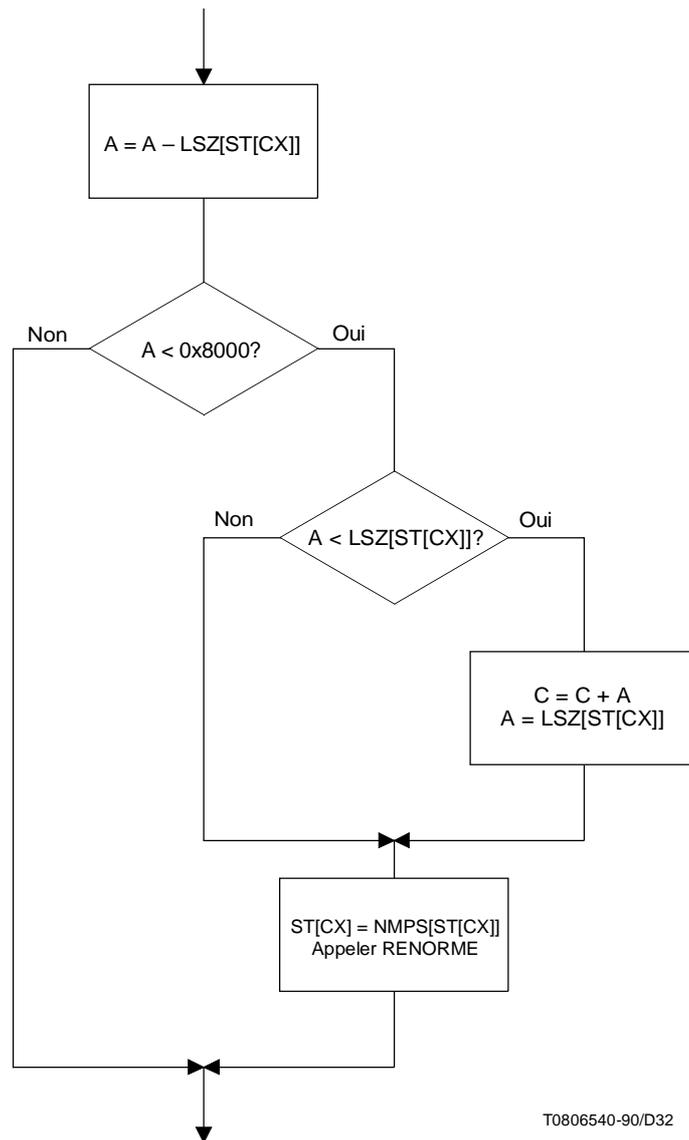


Figure 24 – Organigramme de la procédure CODEMPS

6.8.2.9 Organigramme de la procédure INITENC

Si la bande se trouve en haut de l'image, les états d'estimation de probabilité de toutes les valeurs possibles de CX sont mis à zéro (c'est-à-dire à l'état équiprobable). Autrement, ils sont remis aux valeurs qu'ils avaient à la fin de la dernière bande de même résolution. Le compteur de pile SC et le registre de code C sont remis à zéro. Le compteur CT est mis à 11 (un octet plus les 3 bits d'espacement). Le registre A d'intervalle de codage est mis à $0x10000$. Il est également possible, dans des réalisations sur 16 bits, de le mettre à $0x0000$, sous réserve que le matériel ou le logiciel donnent le même résultat de 16 bits pour la soustraction d'une quantité de 16 bits de 0, que de $0x10000$ – ce qui sera presque toujours le cas (voir la Figure 27).

6.8.2.10 Organigramme de la procédure FLUSH

Deux sous-procédures sont d'abord appelées. Ensuite, le premier octet écrit dans le train de données SCD est retiré, ainsi que, le cas échéant, certains ou tous les $0x00$ octets se trouvant à la fin de SCD , jusqu'à obtenir un octet différent de $0x00$. Les mises en œuvre matérielles et logicielles de qualité, positionneront des variables auxiliaires en sorte que ces octets ne soient jamais écrits en première position. La réalisation décrite ici a été choisie à des fins de simplicité et de concision (voir la Figure 28).

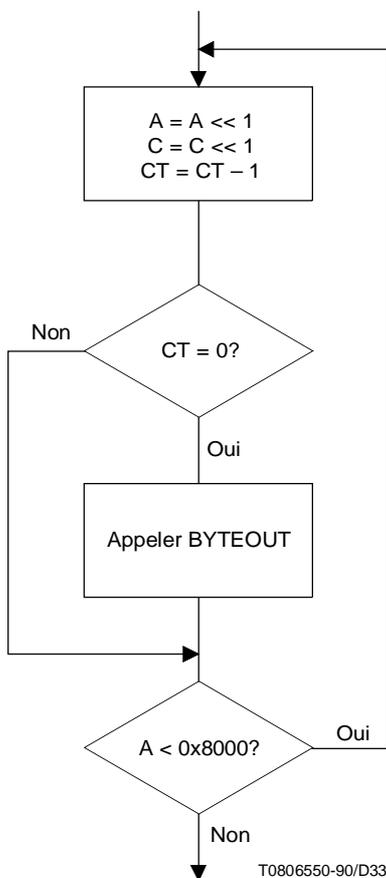


Figure 25 – Organigramme de la procédure RENORME

6.8.2.11 Organigramme de la procédure CLEARBITS

Le registre de code C est mis à la valeur de l'intervalle [C, C+A-1] qui se termine par le plus grand nombre possible de bits zéro (voir la Figure 29).

6.8.2.12 Organigramme de la procédure FINALWRITES

La résolution de retenue finale est effectuée et deux octets de C sont écrits (voir la Figure 30).

6.8.3 Décodeur

6.8.3.1 Organigramme du décodeur

Cet organigramme est exécuté pour chacune des bandes de chaque couche de résolution. Les pixels qui ne peuvent faire l'objet ni d'une prédiction typique ni d'une prédiction déterministe, sont décodés par la procédure DECODE. La procédure d'initialisation INITDEC est appelée à l'entrée (voir la Figure 31).

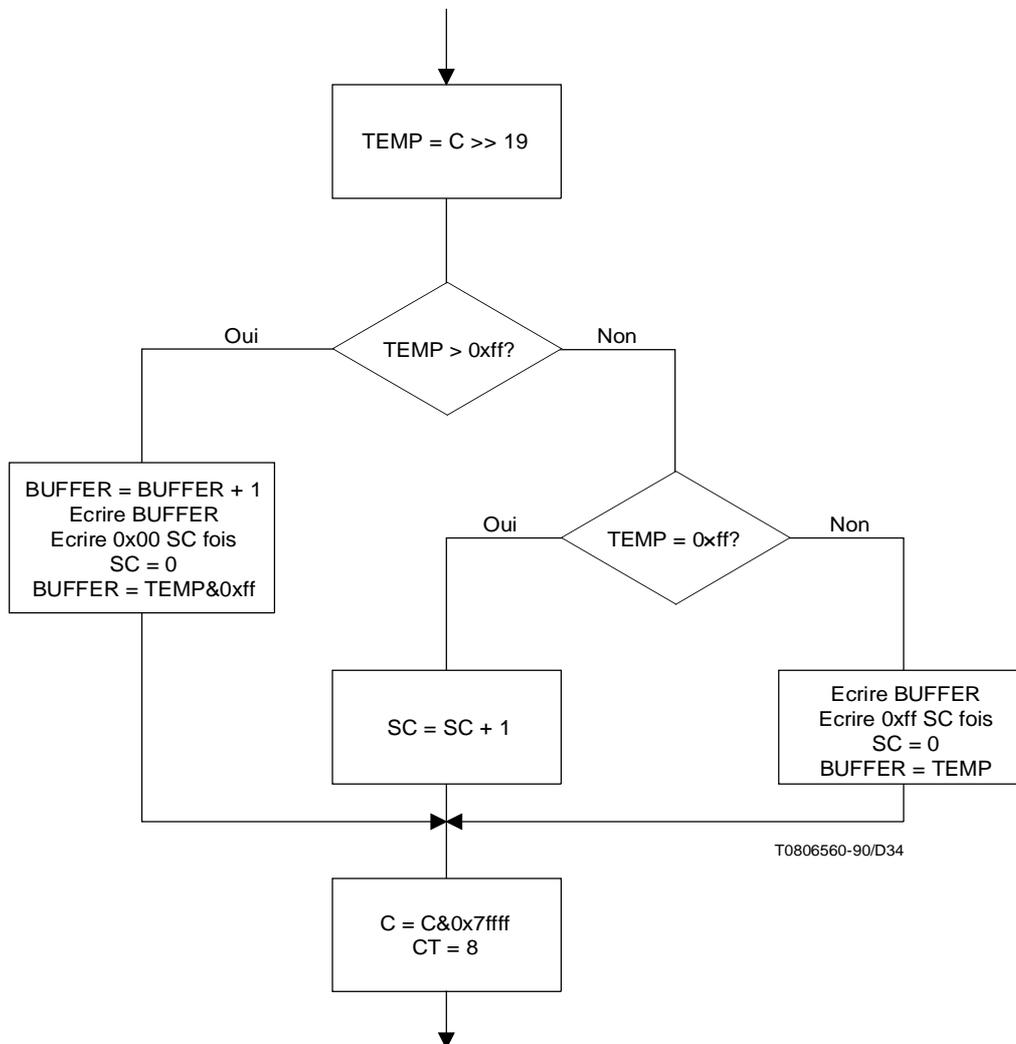


Figure 26 – Organigramme de la procédure BYTEOUT

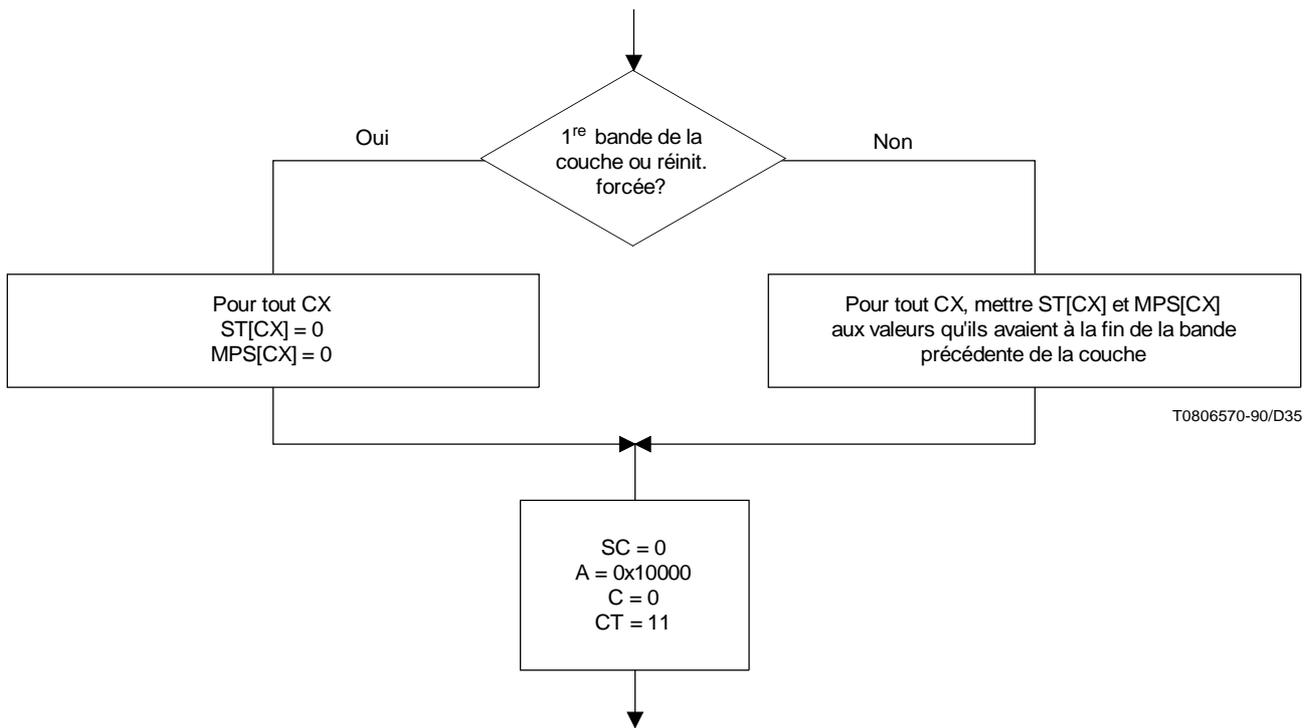


Figure 27 – Organigramme de la procédure INTENC

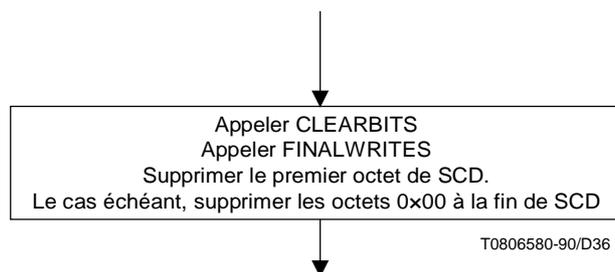


Figure 28 – Organigramme de la procédure FLUSH

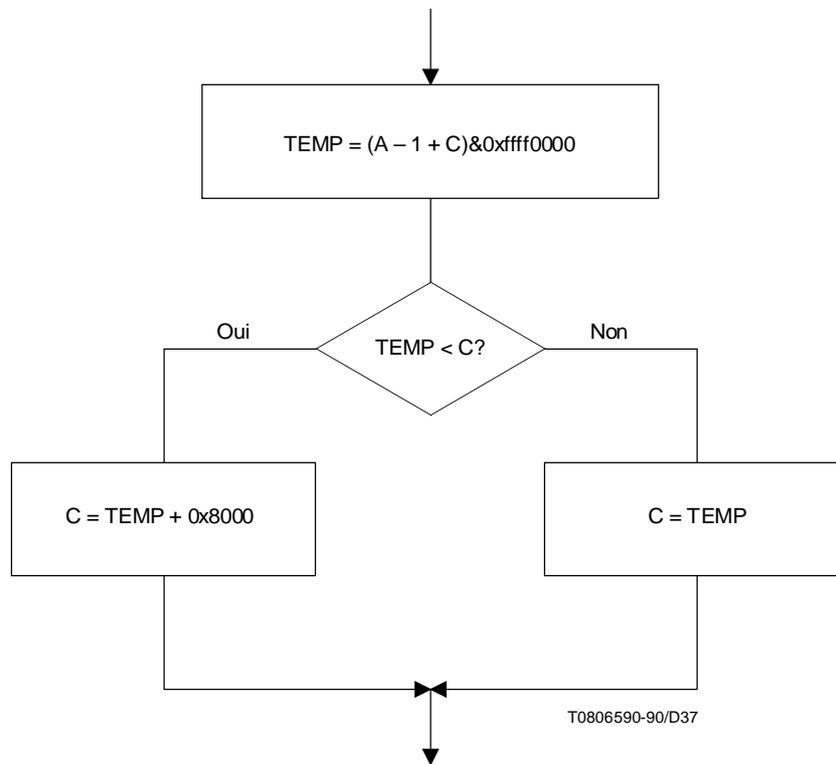


Figure 29 – Organigramme de la procédure CLEARBITS

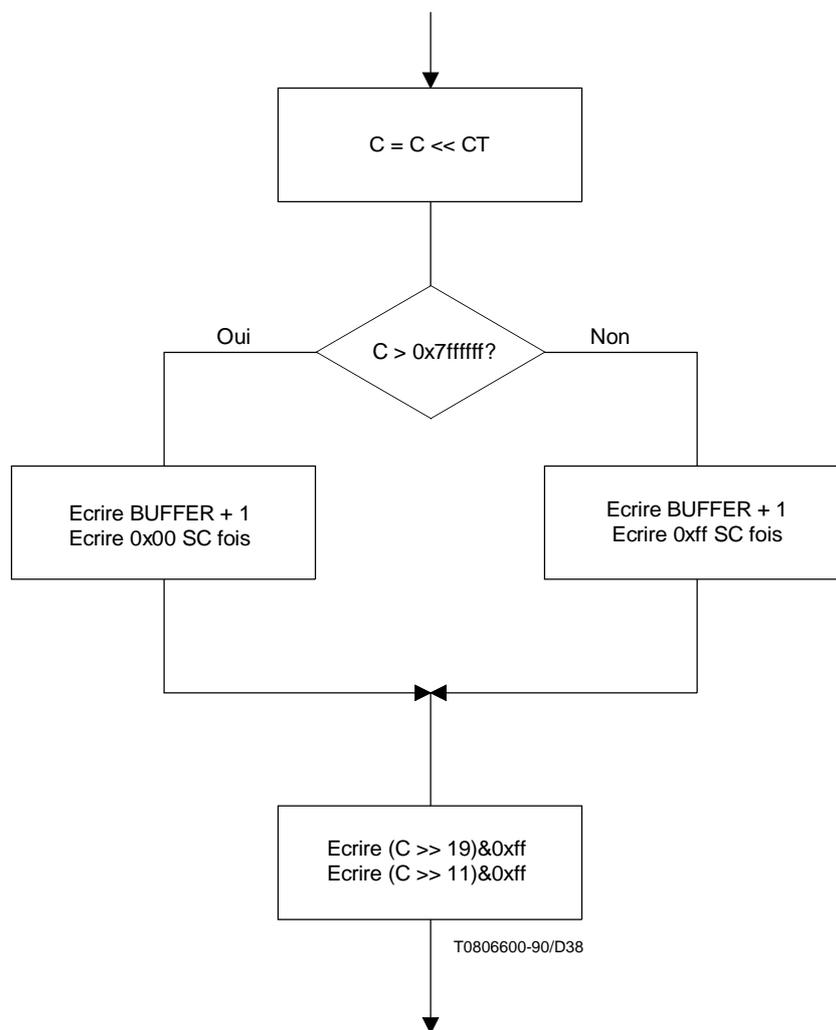


Figure 30 – Organigramme de la procédure FINALWRITES

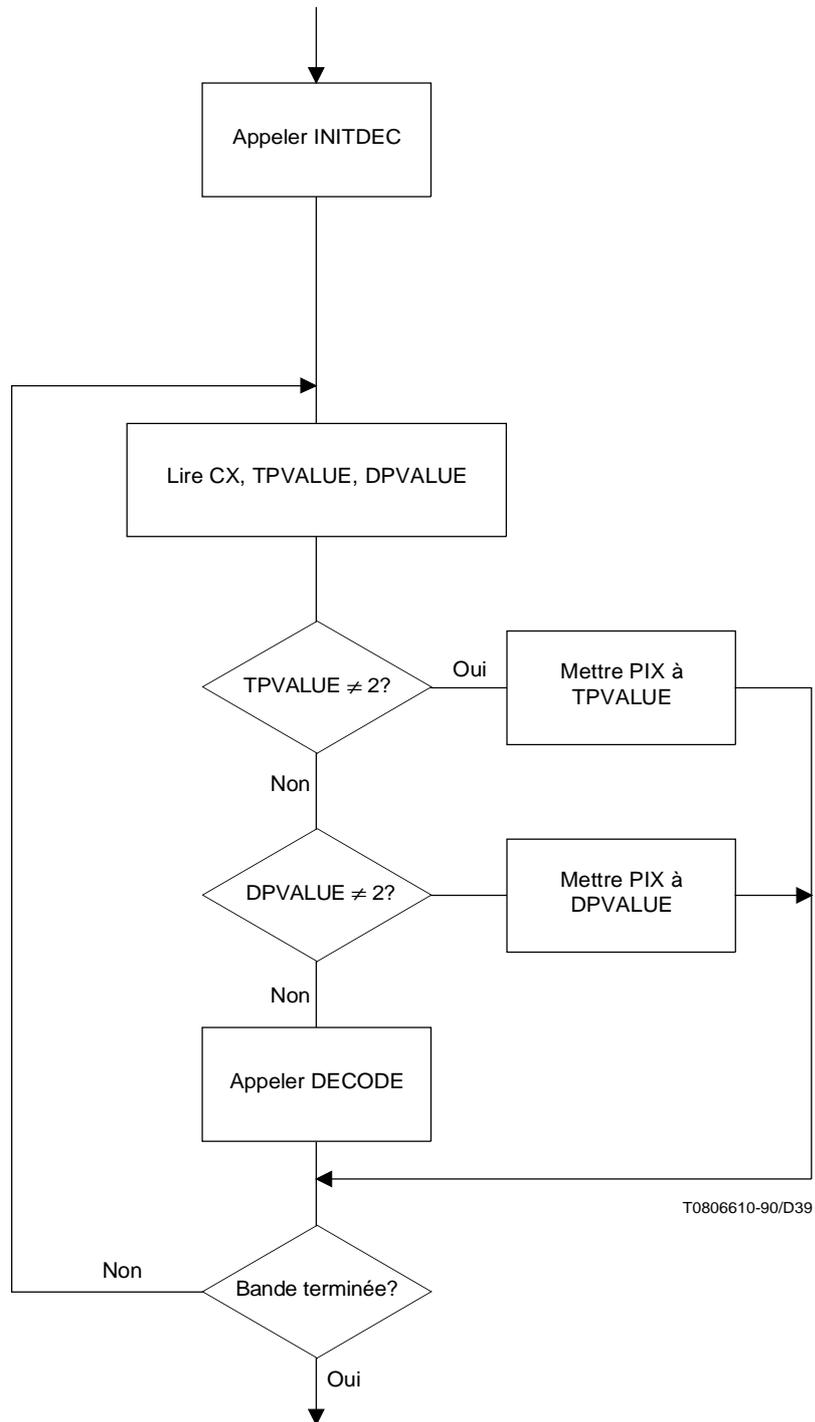


Figure 31 – Organigramme du décodeur

6.8.3.2 Conventions relatives au registre de code du décodeur

Les organigrammes de ce paragraphe supposent la structure de registre du Tableau 25.

Tableau 25 – Structure du registre du décodeur

	msb	lsb
Registre CHIGH	xxxxxxx,	xxxxxxx
Registre CLOW	bbbbbbb,	0000000
Registre A	a, aaaaaaa,	aaaaaaa

CHIGH et CLOW peuvent être considérés comme un registre C unique de 32 bits, en ce sens que la renormalisation de C décale d'un bit les nouvelles données du bit 15 (le plus à gauche) de CLOW vers le bit 0 (le plus à droite) de CHIGH. Néanmoins, les comparaisons opérées pour le décodage utilisent CHIGH seul. Les nouvelles données sont insérées dans les bits «b» de CLOW, octet par octet. Comme dans le codeur, le 17^e bit du registre A existe théoriquement, mais il est facile de s'en passer dans les réalisations.

6.8.3.3 Tables d'estimation de probabilité

Les tables d'estimation de probabilité utilisées pour le décodage sont identiques à celles du codage.

6.8.3.4 Organigramme de la procédure DECODE

La permutation conditionnelle MPS/LPS ne peut avoir lieu que lorsqu'une renormalisation est nécessaire, et seulement dans ce cas (voir la Figure 32).

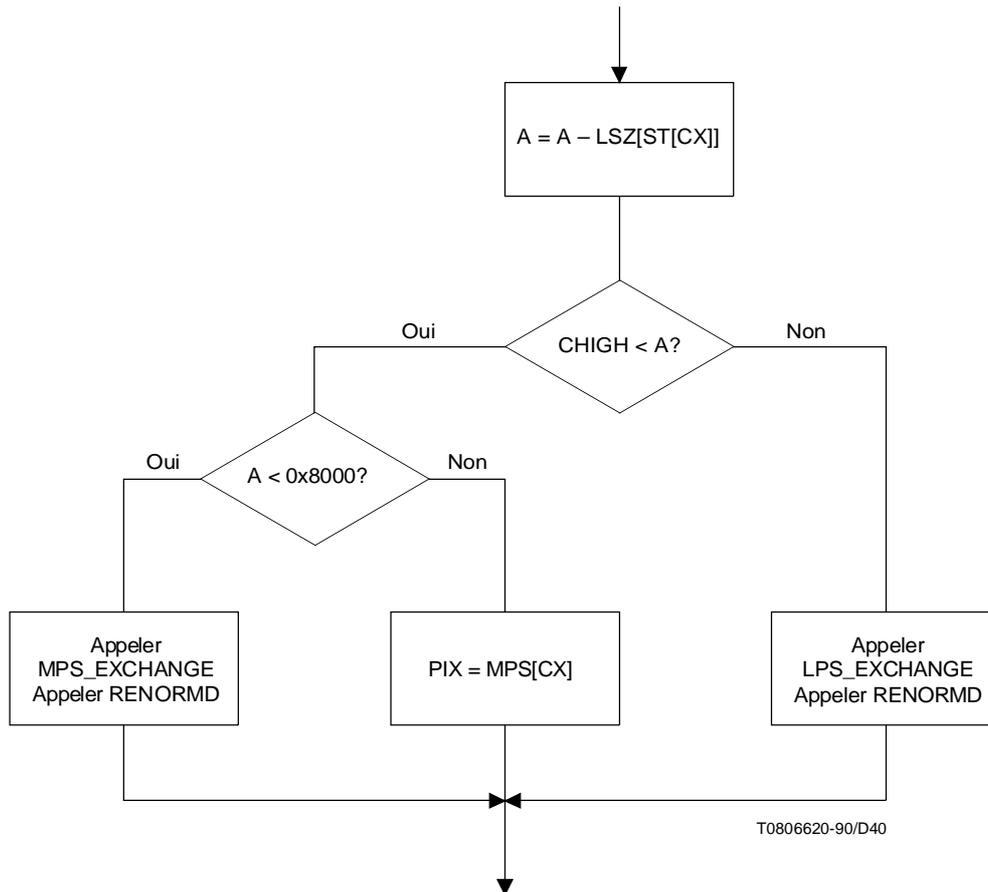


Figure 32 – Organigramme de la procédure DECODE

6.8.3.5 Organigramme de la procédure LPS_EXCHANGE

Voir la Figure 33.

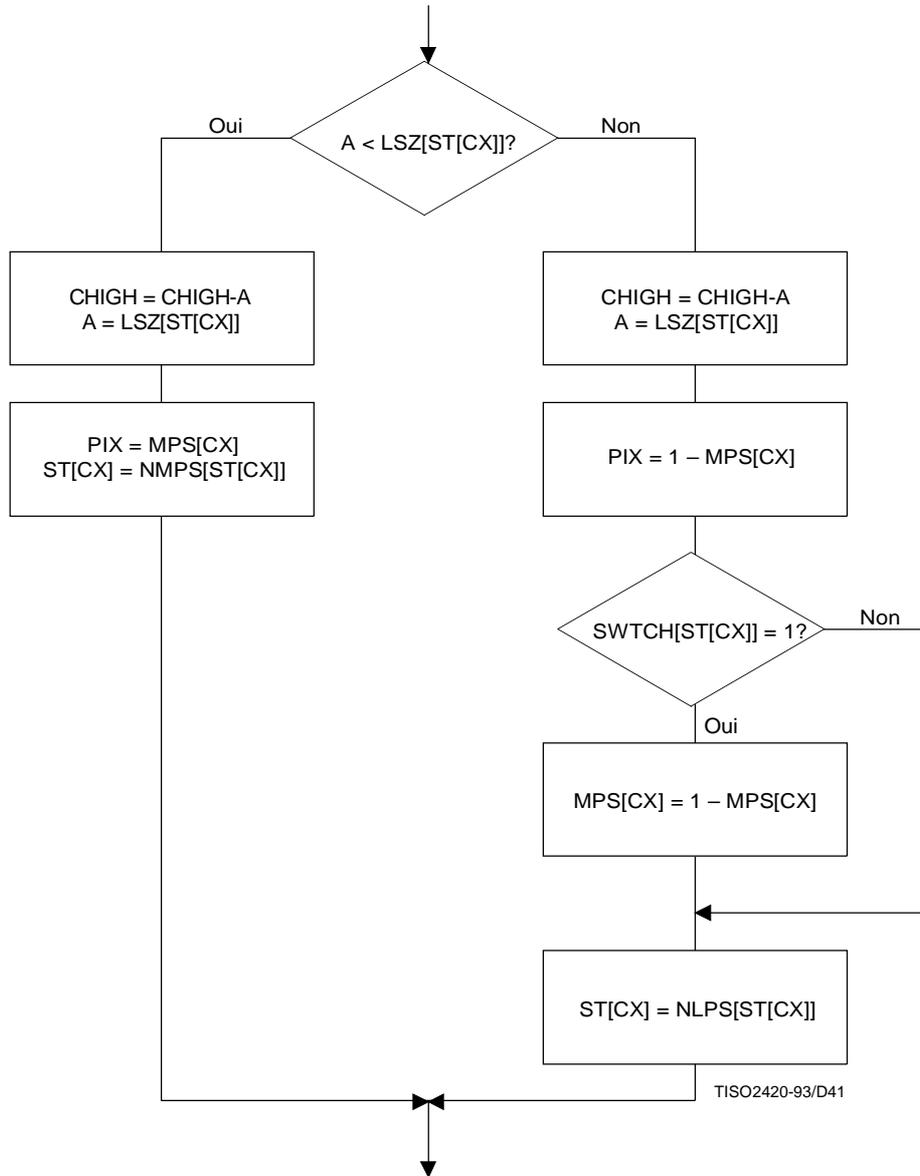


Figure 33 – Organigramme de la procédure LPS_EXCHANGE

6.8.3.6 Organigramme de la procédure MPS_EXCHANGE

Voir la Figure 34.

6.8.3.7 Organigramme de la procédure RENORMD

CT est un compteur qui enregistre le nombre de bits comprimés dans la section CLOW du registre C. Quand CT passe à zéro, un nouvel octet est inséré dans CLOW.

Le registre d'intervalle A et le registre de code C sont décalés tous deux, bit par bit, jusqu'à ce que A ne soit pas plus long que 0x8000 (voir la Figure 35).

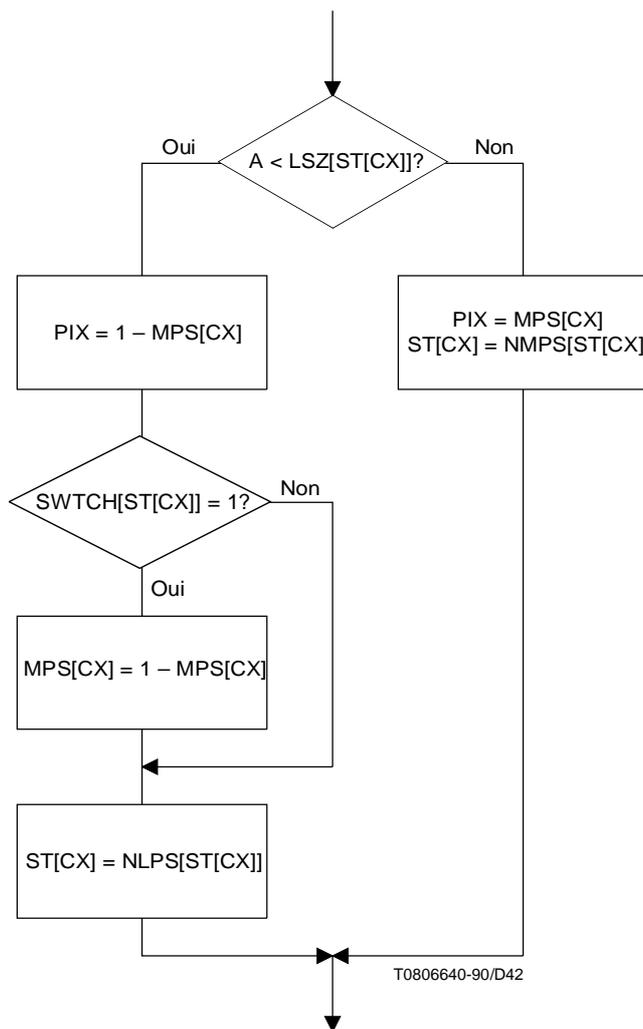


Figure 34 – Organigramme de la procédure MPS_EXCHANGE

6.8.3.8 Organigramme de la procédure BYTEIN

Les octets sont lus dans SCD jusqu'à épuisement, puis complétés pour les lectures suivantes par des valeurs 0x00. Les octets lus sont insérés dans les 8 bits de plus fort poids de CLOW. Le compteur CT est remis à 8 (voir la Figure 36).

6.8.3.9 Organigramme de la procédure INITDEC

Si la bande se trouve en haut de l'image, les états d'estimation de probabilité pour toutes les valeurs possibles de CX sont mis à zéro. Autrement, ils sont remis aux valeurs qu'ils avaient à la fin de la dernière bande de même résolution. Trois octets sont lus dans le registre C (voir la Figure 37).

7 Exemples de méthodes et de suites de données de test

Cet article normatif décrit des méthodes de test pour l'algorithme spécifié dans la présente Spécification. Comme de nombreux paramétrages (choix de valeurs et plages de valeurs des paramètres) sont possibles, cet article se limite aux moyens de test de certains paramétrages considérés comme utiles pour la mise au point de réalisations.

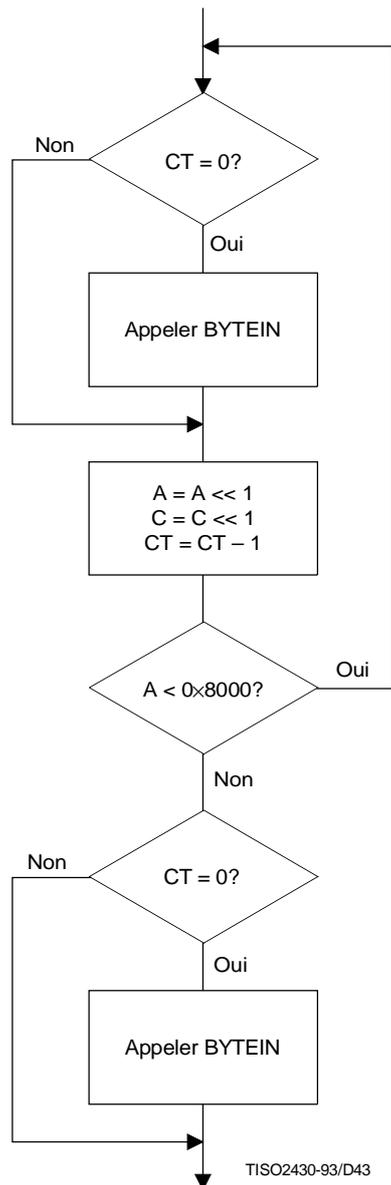


Figure 35 – Organigramme de la procédure RENORMD

7.1 Codage arithmétique

Dans ce paragraphe, un petit ensemble de données est fourni pour tester le codeur et le décodeur arithmétiques. Cet ensemble de données est supposé représenter les données à traiter pour une bande, dans l'ordre du balayage en allant du MSB au LSB. Le test est structuré de façon à tester un grand nombre de trajets logiques dans le codeur et dans le décodeur. Comme il est impossible de les tester tous au cours d'une séquence de test courte, un accord avec les résultats de ce test ne garantit malheureusement pas qu'une mise en œuvre est complètement correcte.

PIX: 05e0 0000 8b00 01c4 1700 0034 7fff 1a3f 951b 05d8 1d17 e770 0000 0000 0656 0e6a

CX: 0fe0 0000 0f00 00f0 ff00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

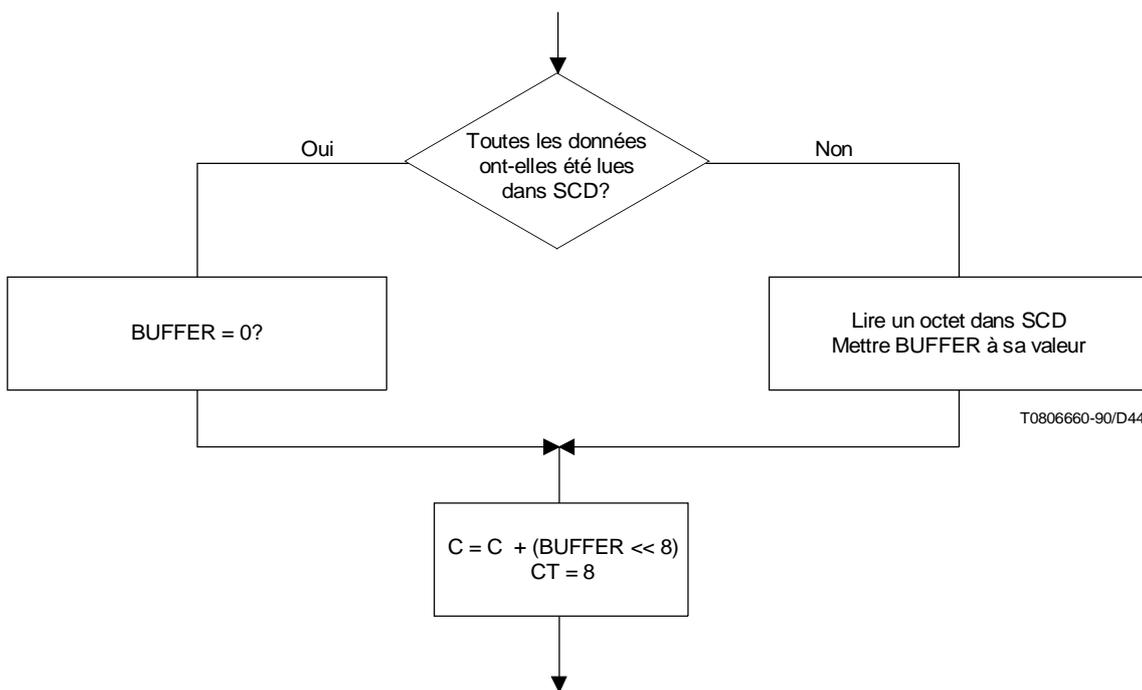


Figure 36 – Organigramme de la procédure BYTEIN

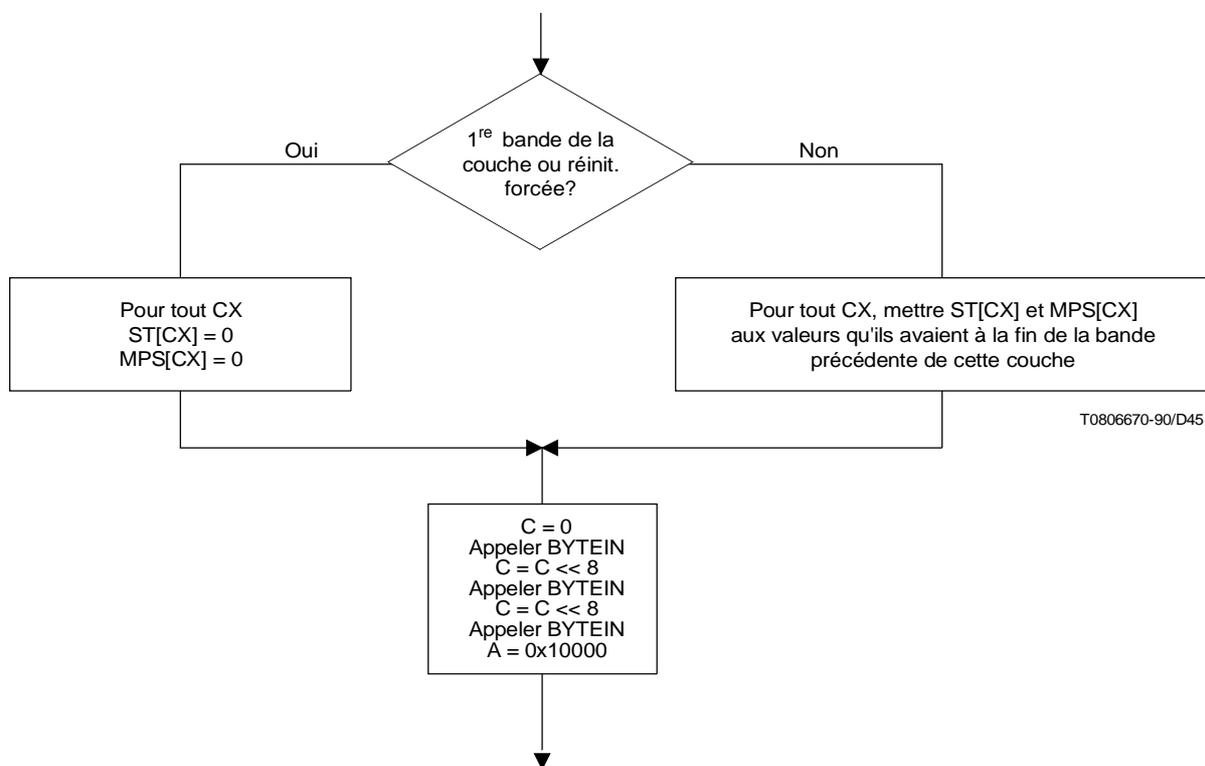


Figure 37 – Organigramme de la procédure INITDEC

Le codeur a quatre entrées (voir la Figure 18). La première séquence décrit l'entrée PIX, et la deuxième décrit l'entrée correspondante CX. Pour simplifier, la variable d'entrée CX de ce test ne peut prendre que deux valeurs 0 et 1, et non les 4096 valeurs qu'elle prend pour le codeur/décodeur JBIG. Les deux autres entrées TPVALUE et DPVALUE sont supposées être toujours égales à 2 (pour le codeur et le décodeur). La sortie du codeur (SCD), comprenant 200 bits (25 octets), est représentée ci-après en hexadécimal:

```
SCD: 6989 995c 32ea faa0 d5ff 527f ffff ffc0 0000 003f ff2d 2082 91
```

Pour le décodeur (voir la Figure 19), les entrées SCD et CX sont données à partir des séquences ci-dessus, tandis que PIX devient l'entrée.

Le Tableau 26 donne une liste de données, symbole par symbole, décrivant le fonctionnement du codeur et du décodeur arithmétiques. La première ligne de ce tableau correspond aux procédures INITENC et INITDEC, le BUFFER étant initialisé à 0x00. La dernière ligne du Tableau 26 correspond à la procédure FINALWRITES du codeur. La première colonne est le compteur d'éléments (EC), la deuxième la valeur de l'élément binaire PIX à coder et décoder, la troisième est la valeur correspondante de la variable d'entrée CX. La colonne MPS indique le sens du MPS, et la colonne CE indique si la permutation conditionnelle (voir 6.8.2.5, 6.8.2.6 et 6.8.3.4) aura lieu lors du codage (ou du décodage) de l'élément binaire courant PIX. L'état courant (voir le Tableau 24) et la longueur correspondante LPS, sont représentés dans les colonnes ST et LSZ. La colonne suivante est la liste des valeurs du registre A avant codage (ou décodage) de l'élément. A noter que le registre A est toujours supérieur ou égal à 0x8000.

Les variables précédentes sont communes au codeur et au décodeur. Les cinq colonnes suivantes (C, CT, SC, BUF, OUT) concernent uniquement le codeur et les trois dernières (C, CT, IN) le décodeur. Les entrées du codeur sont (PIX, CX) et sa sortie est donnée dans la colonne OUT, alors que les entrées du décodeur sont (CX, IN) et que sa sortie est donnée dans PIX. Les valeurs du registre C (colonne C) sont données avant codage (ou décodage) de l'élément courant. Pour le codeur, CT est un compteur indiquant quand un octet est prêt à être sorti du registre C. SC est le nombre d'octets 0xff empilés dans le codeur, en attente de résolution de retenue. La colonne BUF présente l'octet de la variable BUFFER, attendant d'être émis. Cet octet peut parfois changer sous l'effet d'une retenue. Enfin, les octets de code de sortie du codeur figurent dans la colonne OUT. Ces octets sont envoyés à la sortie lors du codage de l'élément courant. Si plusieurs octets figurent, ils ont également été sortis lors du codage de l'élément courant et générés par la remise à zéro du compteur SC.

Pour le décodeur, les valeurs du registre C sont données avant le décodage de l'élément, et sont indiquées dans la colonne C. Le compteur CT du décodeur, représenté dans la colonne suivante, indique quand il faut entrer l'octet suivant du train de données codées. Dans la dernière colonne, figurent les octets de code qui ont été lus dans le registre de code à la fin de l'élément courant.

La dernière ligne du tableau montre la sortie générée par la procédure FINALWRITES. Cette procédure génère également cinq octets additionnels 0x00, deux provenant de la remise à zéro du compteur SC et trois du décalage du registre C. Ces octets finals 0x00 ne sont pas inclus dans les octets de code SCD, car la possibilité de retirer tous les octets 0x00 se trouvant à la fin du train de données codées d'une bande (voir procédure FLUSH, Figure 28) a été utilisée. A noter qu'il est permis de laisser n'importe quel nombre de ces octets finals 0x00 à la fin du train de données codées SCD. Lorsque le décodeur atteint la fin du train de données codées, il lit les octets 0x00 de son registre C, jusqu'au décodage du nombre de pixels désiré (256).

Pour générer le train de données codées PSCD (voir le Tableau 12), l'octet STUFF (0x00) doit ensuite être inséré après chaque octet ESC (0xff) de SCD.

```
PSCD: 6989 995c 32ea faa0 d5ff 0052 7fff 00ff 00ff 00c0 0000 003f ff00 2d20 8291
```

Enfin, comme il a été supposé que les données (CX, PIX) sont les données brutes d'une bande, l'entité de données de bande (SDE) peut être générée en concaténant les octets ESC (0xff) et SDNORM (0x02).

```
SDE: 6989 995c 32ea faa0 d5ff 0052 7fff 00ff 00ff 00c0 0000 003f ff00 2d20 8291 ff02
```

Tableau 26 – Données de trace du codeur et du décodeur

								CODEUR					DÉCODEUR		
EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
0													00000000	0	698999
1	0	0	0	0	0	5a1d	10000	00000000	11	0	00		69899900	8	
2	0	0	0	1	0	5a1d	0a5e3	00000000	11	0	00		69899900	8	
3	0	0	0	0	1	2586	0b43a	0000978c	10	0	00		3b873200	7	
4	0	0	0	0	1	2586	08eb4	0000978c	10	0	00		3b873200	7	
5	0	1	0	0	0	5a1d	0d25c	00012f18	9	0	00		770e6400	6	
6	1	1	0	0	1	2586	0f07e	00025e30	8	0	00		ee1cc800	5	
7	0	1	0	1	14	5a7f	0961e	000ca4a0	6	0	00		8c932000	3	
8	1	1	0	0	15	3f25	0b4fe	0019c072	5	0	00		a1f44000	2	5c
9	1	1	0	0	36	5ae1	0fc94	0068d92c	3	0	00		b06d5c00	8	
10	1	1	1	0	37	484c	0b5c2	00d2f5be	2	0	00		1d74b800	7	
11	1	1	1	0	38	3a0d	0daec	01a5eb7c	1	0	00		3ae97000	6	
12	0	0	0	0	2	1114	0a0df	01a5eb7c	1	0	00		3ae97000	6	
13	0	0	0	0	2	1114	08fcb	01a5eb7c	1	0	00		3ae97000	6	
14	0	0	0	0	3	080b	0fd6e	0003d6f8	8	0	69		75d2e000	5	
15	0	0	0	0	3	080b	0f563	0003d6f8	8	0	69		75d2e000	5	
16	0	0	0	0	3	080b	0ed58	0003d6f8	8	0	69		75d2e000	5	
17	0	0	0	0	3	080b	0e54d	0003d6f8	8	0	69		75d2e000	5	
18	0	0	0	0	3	080b	0dd42	0003d6f8	8	0	69		75d2e000	5	
19	0	0	0	0	3	080b	0d537	0003d6f8	8	0	69		75d2e000	5	
20	0	0	0	0	3	080b	0cd2c	0003d6f8	8	0	69		75d2e000	5	
21	0	0	0	0	3	080b	0c521	0003d6f8	8	0	69		75d2e000	5	
22	0	0	0	0	3	080b	0bd16	0003d6f8	8	0	69		75d2e000	5	
23	0	0	0	0	3	080b	0b50b	0003d6f8	8	0	69		75d2e000	5	
24	0	0	0	0	3	080b	0ad00	0003d6f8	8	0	69		75d2e000	5	
25	0	0	0	0	3	080b	0a4f5	0003d6f8	8	0	69		75d2e000	5	
26	0	0	0	0	3	080b	09cea	0003d6f8	8	0	69		75d2e000	5	
27	0	0	0	0	3	080b	094df	0003d6f8	8	0	69		75d2e000	5	
28	0	0	0	0	3	080b	08cd4	0003d6f8	8	0	69		75d2e000	5	
29	0	0	0	0	3	080b	084c9	0003d6f8	8	0	69		75d2e000	5	
30	0	0	0	0	4	03d8	0f97c	0007adf0	7	0	69		eba5c000	4	
31	0	0	0	0	4	03d8	0f5a4	0007adf0	7	0	69		eba5c000	4	
32	0	0	0	0	4	03d8	0f1cc	0007adf0	7	0	69		eba5c000	4	
33	1	0	0	0	4	03d8	0edf4	0007adf0	7	0	69		eba5c000	4	32
34	0	0	0	0	20	0cef	0f600	02260300	1	0	69		6270c800	6	
35	0	0	0	0	20	0cef	0e911	02260300	1	0	69		6270c800	6	
36	0	0	0	0	20	0cef	0dc22	02260300	1	0	69		6270c800	6	
37	1	1	1	0	38	3a0d	0cf33	02260300	1	0	69		6270c800	6	
38	0	1	1	0	38	3a0d	09526	02260300	1	0	69	69	6270c800	6	
39	1	1	1	0	65	4d04	0e834	00097864	7	0	89		1d5f2000	4	
40	1	1	1	0	65	4d04	09b30	00097864	7	0	89		1d5f2000	4	
41	0	0	0	0	20	0cef	09c58	0012f0c8	6	0	89		3abe4000	3	
42	0	0	0	0	20	0cef	08f69	0012f0c8	6	0	89		3abe4000	3	
43	0	0	0	0	20	0cef	0827a	0012f0c8	6	0	89		3abe4000	3	
44	0	0	0	0	21	09a1	0eb16	0025e190	5	0	89		757c8000	2	
45	0	0	0	0	21	09a1	0e175	0025e190	5	0	89		757c8000	2	
46	0	0	0	0	21	09a1	0d7d4	0025e190	5	0	89		757c8000	2	
47	0	0	0	0	21	09a1	0ce33	0025e190	5	0	89		757c8000	2	
48	0	0	0	0	21	09a1	0c492	0025e190	5	0	89		757c8000	2	
49	0	0	0	0	21	09a1	0baf1	0025e190	5	0	89		757c8000	2	
50	0	0	0	0	21	09a1	0b150	0025e190	5	0	89		757c8000	2	
51	0	0	0	0	21	09a1	0a7af	0025e190	5	0	89		757c8000	2	
52	0	0	0	0	21	09a1	09e0e	0025e190	5	0	89		757c8000	2	
53	0	0	0	0	21	09a1	0946d	0025e190	5	0	89		757c8000	2	
54	0	0	0	0	21	09a1	08acc	0025e190	5	0	89		757c8000	2	
55	0	0	0	0	21	09a1	0812b	0025e190	5	0	89		757c8000	2	
56	1	0	0	0	22	072f	0ef14	004bc320	4	0	89	89	eaf90000	1	ea

Tableau 26 (suite)

EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	CODEUR					DÉCODEUR			
								C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex	
57	1	1	1	0	66	412c	0e5e0	000560a0	7	0	99			628ea000	4	
58	1	1	1	0	66	412c	0a4b4	000560a0	7	0	99			628ea000	4	
59	0	1	1	0	67	37d8	0c710	000ac140	6	0	99			c51d4000	3	
60	0	1	1	0	82	438e	0df60	002d41e0	4	0	99			d7950000	1	fa
61	0	0	0	0	46	0bfb	0871c	005bbb64	3	0	99			7786fa00	8	
62	1	0	0	0	47	09f8	0f642	00b776c8	2	0	99	99		ef0df400	7	
63	0	0	0	0	77	0d51	09f80	00063120	6	0	5c			2c3f4000	3	
64	0	0	0	0	77	0d51	0922f	00063120	6	0	5c			2c3f4000	3	
65	0	1	1	1	89	4ca9	084de	00063120	6	0	5c			2c3f4000	3	
66	0	1	1	0	95	56a8	0e0d4	0018c480	4	0	5c			b0fd0000	1	a0
67	0	1	0	0	95	56a8	0ad50	00329d58	3	0	5c			4da2a000	8	
68	1	1	0	0	96	4f46	0ad50	00653ab0	2	0	5c			9b454000	7	
69	0	1	0	1	101	5231	09e8c	00cb3174	1	0	5c	5c		7a768000	6	
70	1	1	0	0	102	4c0f	0a462	0006fb9e	8	0	32			5c370000	5	
71	1	1	0	1	106	50e7	0981e	000ea7e2	7	0	32			07c80000	4	
72	1	1	0	1	108	5597	08e6e	001d4fc4	6	0	32			0f900000	3	
73	0	0	0	0	77	0d51	0e35c	00753f10	4	0	32			3e400000	1	
74	0	0	0	0	77	0d51	0d60b	00753f10	4	0	32			3e400000	1	
75	0	0	0	0	77	0d51	0c8ba	00753f10	4	0	32			3e400000	1	
76	0	0	0	0	77	0d51	0bb69	00753f10	4	0	32			3e400000	1	
77	0	0	0	0	77	0d51	0ae18	00753f10	4	0	32			3e400000	1	
78	0	0	0	0	77	0d51	0a0c7	00753f10	4	0	32			3e400000	1	
79	0	0	0	0	77	0d51	09376	00753f10	4	0	32			3e400000	1	
80	0	0	0	0	77	0d51	08625	00753f10	4	0	32			3e400000	1	d5
81	0	0	0	0	78	0bb6	0f1a8	00ea7e20	3	0	32			7c80d500	8	
82	0	0	0	0	78	0bb6	0e5f2	00ea7e20	3	0	32			7c80d500	8	
83	0	0	0	0	78	0bb6	0da3c	00ea7e20	3	0	32			7c80d500	8	
84	0	0	0	0	78	0bb6	0ce86	00ea7e20	3	0	32			7c80d500	8	
85	0	0	0	0	78	0bb6	0c2d0	00ea7e20	3	0	32			7c80d500	8	
86	0	0	0	0	78	0bb6	0b71a	00ea7e20	3	0	32			7c80d500	8	
87	0	0	0	0	78	0bb6	0ab64	00ea7e20	3	0	32			7c80d500	8	
88	0	0	0	0	78	0bb6	09fae	00ea7e20	3	0	32			7c80d500	8	
89	0	0	0	0	78	0bb6	093f8	00ea7e20	3	0	32			7c80d500	8	
90	0	0	0	0	78	0bb6	08842	00ea7e20	3	0	32			7c80d500	8	
91	1	0	0	0	79	0a40	0f918	01d4fc40	2	0	32	32		f901aa00	7	
92	1	0	0	0	77	0d51	0a400	001eb180	6	0	ea			a29aa000	3	ff
93	0	0	0	0	75	119c	0d510	01f482f0	2	0	ea			bebbfe00	7	
94	1	0	0	0	75	119c	0c374	01f482f0	2	0	ea	ea		bebbfe00	7	
95	0	0	0	0	74	1424	08ce0	0009a640	7	0	fa			671ff000	4	
96	0	0	0	0	75	119c	0f178	00134c80	6	0	fa			ce3fe000	3	
97	0	0	0	0	75	119c	0dfd0	00134c80	6	0	fa			ce3fe000	3	
98	1	0	0	0	75	119c	0ce40	00134c80	6	0	fa			ce3fe000	3	52
99	1	0	0	0	74	1424	08ce0	00a04920	3	0	fa	fa		8cdf5200	8	
100	1	0	0	0	72	1aa9	0a120	00060ee0	8	0	a0			a11a9000	5	
101	1	0	0	0	87	2516	0d548	0034aab8	5	0	a0			d51c8000	2	7f
102	1	0	0	0	86	299a	09458	00d56ba8	3	0	a0			93aa7f00	8	
103	1	0	0	0	93	32b4	0a668	03575998	1	0	a0	a0		a3b1fc00	6	
104	1	0	0	0	99	3c3d	0cad0	000f3530	7	0	d5			bff7f000	4	
105	1	0	0	0	104	415e	0f0f4	003f0f0c	5	0	d5			c593c000	2	
106	1	0	0	1	103	4639	082bc	007f7d44	4	0	d5			2bfb8000	1	ff
107	1	0	0	0	107	4b85	0f20c	01fd510	2	0	d5			afeffe00	7	
108	1	0	0	1	109	504f	0970a	03fd372e	1	0	d5			12d1fc00	6	
109	1	0	0	1	111	5522	08d76	00026e5c	8	1	d5			25a3f800	5	
110	1	0	0	0	112	59eb	0e150	0009b970	6	1	d5			968fe000	3	
111	1	0	1	0	112	59eb	0b3d6	001481aa	5	1	d5			1e55c000	2	
112	1	0	1	0	111	5522	0b3d6	00290354	4	1	d5			3cab8000	1	ff

Tableau 26 (suite)

								CODEUR					DÉCODEUR		
EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
113	0	0	1	0	109	504f	0bd68	005206a8	3	1	d5		7957ff00	8	
114	0	0	1	1	111	5522	0a09e	00a4e782	2	1	d5		187dfe00	7	
115	0	0	1	1	112	59eb	096f8	0149cf04	1	1	d5	d5ff	30fbfc00	6	
116	1	0	0	0	112	59eb	0f434	00073c10	7	0	52		c3eff000	4	
117	1	0	1	0	112	59eb	0b3d6	000facb2	6	0	52		534de000	3	
118	0	0	1	0	111	5522	0b3d6	001f5964	5	0	52		a69bc000	2	
119	1	0	1	1	112	59eb	0aa44	003f7030	4	0	52		8fcf8000	1	ff
120	0	0	1	0	111	5522	0b3d6	007f8112	3	0	52		7eedff00	8	
121	0	0	1	1	112	59eb	0aa44	00ffb8c	2	0	52		4073fe00	7	
122	0	0	0	1	112	59eb	0a0b2	01ff7f18	1	0	52	52	80e7fc00	6	
123	1	0	0	0	111	5522	0b3d6	00078bbe	8	0	7f		7441f800	5	
124	1	0	0	1	112	59eb	0aa44	000fd4e4	7	0	7f		2b1bf000	4	
125	1	0	1	1	112	59eb	0a0b2	001fa9c8	6	0	7f		5637e000	3	
126	1	0	1	0	111	5522	0b3d6	003fe11e	5	0	7f		1ee1c000	2	
127	1	0	1	0	109	504f	0bd68	007fc23c	4	0	7f		3dc38000	1	c0
128	1	0	1	0	107	4b85	0da32	00ff8478	3	0	7f		7b87c000	8	
129	1	0	1	1	107	4b85	08ead	00ff8478	3	0	7f		7b87c000	8	
130	0	0	1	0	103	4639	0970a	01ff8f40	2	0	7f		70bf8000	7	
131	0	0	1	1	107	4b85	08c72	03ffc022	1	0	7f		3fdd0000	6	
132	1	0	1	1	109	504f	081da	00078044	8	1	7f		7fba0000	5	
133	0	0	1	0	107	4b85	0a09e	000f639e	7	1	7f		9c5e0000	4	
134	1	0	1	1	109	504f	0970a	001f716e	6	1	7f		8e8a0000	3	
135	0	0	1	0	107	4b85	0a09e	003f7052	5	1	7f		8f9e0000	2	
136	1	0	1	1	109	504f	0970a	007f8ad6	4	1	7f		750a0000	1	00
137	0	0	1	0	107	4b85	0a09e	00ffa322	3	1	7f		5c9e0000	8	
138	0	0	1	1	109	504f	0970a	01fff076	2	1	7f		0f0a0000	7	
139	0	0	1	1	111	5522	08d76	03ffe0ec	1	1	7f		1e140000	6	
140	1	0	1	0	112	59eb	0e150	000f83b0	7	2	7f		78500000	4	
141	1	0	1	1	112	59eb	08765	000f83b0	7	2	7f		78500000	4	
142	0	0	1	0	111	5522	0b3d6	001f6254	6	2	7f		95ac0000	3	
143	1	0	1	1	112	59eb	0aa44	003f8210	5	2	7f		6df00000	2	
144	1	0	1	0	111	5522	0b3d6	007fa4d2	4	2	7f		3b2e0000	1	00
145	0	0	1	0	109	504f	0bd68	00ff49a4	3	2	7f		765c0000	8	
146	0	0	1	1	111	5522	0a09e	01ff6d7a	2	2	7f		12860000	7	
147	0	0	1	1	112	59eb	096f8	03fedaf4	1	2	7f		250c0000	6	
148	0	0	0	0	112	59eb	0f434	000b6bd0	7	3	7f		94300000	4	
149	0	0	0	1	112	59eb	09a49	000b6bd0	7	3	7f		94300000	4	
150	1	0	0	0	111	5522	0b3d6	0017585c	6	3	7f		a7a40000	3	
151	0	0	0	1	112	59eb	0aa44	002f6e20	5	3	7f		91e00000	2	
152	1	0	0	0	111	5522	0b3d6	005f7cf2	4	3	7f		830e0000	1	00
153	1	0	0	1	112	59eb	0aa44	00bfb74c	3	3	7f		48b40000	8	
154	1	0	1	1	112	59eb	0a0b2	017f6e98	2	3	7f		91680000	7	
155	0	0	1	0	111	5522	0b3d6	02ff6abe	1	3	7f	7fffffff	95420000	6	
156	1	0	1	1	112	59eb	0aa44	000792e4	8	0	bf		6d1c0000	5	
157	1	0	1	0	111	5522	0b3d6	000fc67a	7	0	bf		39860000	4	
158	0	0	1	0	109	504f	0bd68	001f8cf4	6	0	bf		730c0000	3	
159	0	0	1	1	111	5522	0a09e	003ff41a	5	0	bf		0be60000	2	
160	0	0	1	1	112	59eb	096f8	007fe834	4	0	bf		17cc0000	1	3f
161	0	0	0	0	112	59eb	0f434	01ffa0d0	2	0	bf		5f307e00	7	
162	0	0	0	1	112	59eb	09a49	01ffa0d0	2	0	bf		5f307e00	7	
163	0	0	0	0	111	5522	0b3d6	03ffc25c	1	0	bf		3da4fc00	6	
164	1	0	0	0	109	504f	0bd68	000784b8	8	1	bf		7b49f800	5	
165	1	0	0	1	111	5522	0a09e	000fe3a2	7	1	bf		1c61f000	4	
166	1	0	0	1	112	59eb	096f8	001fc744	6	1	bf		38c3e000	3	
167	0	0	1	0	112	59eb	0f434	007f1d10	4	1	bf		e30f8000	1	ff
168	1	0	0	0	112	59eb	0b3d6	00ff6eb2	3	1	bf		918dff00	8	

Tableau 26 (suite)

								CODEUR					DÉCODEUR		
EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
169	0	0	1	0	112	59eb	0b3d6	01ff913a	2	1	bf		6f45fe00	7	
170	0	0	0	0	112	59eb	0b3d6	03ffd64a	1	1	bf		2ab5fc00	6	
171	0	0	0	0	111	5522	0b3d6	0007ac94	8	2	bf		556bf800	5	
172	1	0	0	0	109	504f	0bd68	000f5928	7	2	bf		aad7f000	4	
173	0	0	0	1	111	5522	0a09e	001f8c82	6	2	bf		7b7de000	3	
174	1	0	0	0	109	504f	0aa44	003faffc	5	2	bf		6003c000	2	
175	1	0	0	1	111	5522	0a09e	008013e2	4	2	bf		0c1d8000	1	2d
176	1	0	0	1	112	59eb	096f8	010027c4	3	2	bf		183b2d00	8	
177	1	0	1	0	112	59eb	0f434	04009f10	1	2	bf		60ecb400	6	
178	1	0	1	1	112	59eb	09a49	04009f10	1	2	bf	c00000	60ecb400	6	
179	1	0	1	0	111	5522	0b3d6	0001bedc	8	0	00		411d6800	5	
180	0	0	1	0	109	504f	0bd68	00037db8	7	0	00		823ad000	4	
181	0	0	1	1	111	5522	0a09e	0007d5a2	6	0	00		2a43a000	3	
182	1	0	1	1	112	59eb	096f8	000fab44	5	0	00		54874000	2	
183	1	0	1	0	111	5522	0b3d6	001fd0a2	4	0	00		2ef48000	1	20
184	1	0	1	0	109	504f	0bd68	003fa144	3	0	00		5de92000	8	
185	0	0	1	0	107	4b85	0da32	007f4288	2	0	00		bbd24000	7	
186	1	0	1	1	109	504f	0970a	00ffa26a	1	0	00	00	5a4a8000	6	
187	1	0	1	0	107	4b85	0a09e	0007d24a	8	0	3f		271f0000	5	
188	1	0	1	0	103	4639	0aa32	000fa494	7	0	3f		4e3e0000	4	
189	0	0	1	0	104	415e	0c7f2	001f4928	6	0	3f		9c7c0000	3	
190	0	0	1	1	103	4639	082bc	003f9f78	5	0	3f		2bd00000	2	82
191	0	0	1	0	107	4b85	0f20c	00fe7de0	3	0	3f		af408200	8	
192	0	0	1	1	109	504f	0970a	01fe48ce	2	0	3f		11730400	7	
193	0	0	1	1	111	5522	08d76	03fc919c	1	0	3f		22e60800	6	
194	0	0	1	0	112	59eb	0e150	00024670	7	1	3f		8b982000	4	
195	0	0	0	0	112	59eb	0b3d6	00059baa	6	1	3f		08664000	3	
196	0	0	0	0	111	5522	0b3d6	000b3754	5	1	3f		10cc8000	2	
197	0	0	0	0	109	504f	0bd68	00166ea8	4	1	3f		21990000	1	91
198	0	0	0	0	107	4b85	0da32	002cdd50	3	1	3f		43329100	8	
199	0	0	0	1	107	4b85	08ead	002cdd50	3	1	3f		43329100	8	
200	0	0	0	0	103	4639	0970a	005a40f0	2	1	3f		00152200	7	
201	0	0	0	0	104	415e	0a1a2	00b481e0	1	1	3f	3fff	002a4400	6	
202	0	0	0	0	99	3c3d	0c088	000103c0	8	0	2d		00548800	5	
203	0	0	0	0	99	3c3d	0844b	000103c0	8	0	2d		00548800	5	
204	0	0	0	0	100	375e	0901c	00020780	7	0	2d		00a91000	4	
205	0	0	0	0	93	32b4	0b17c	00040f00	6	0	2d		01522000	3	
206	0	0	0	0	94	2e17	0fd90	00081e00	5	0	2d		02a44000	2	
207	0	0	0	0	94	2e17	0cf79	00081e00	5	0	2d		02a44000	2	
208	0	0	0	0	94	2e17	0a162	00081e00	5	0	2d		02a44000	2	
209	0	0	0	0	86	299a	0e696	00103c00	4	0	2d		05488000	1	
210	0	0	0	0	86	299a	0bcfc	00103c00	4	0	2d		05488000	1	
211	0	0	0	0	86	299a	09362	00103c00	4	0	2d		05488000	1	00
212	0	0	0	0	87	2516	0d390	00207800	3	0	2d		0a910000	8	
213	0	0	0	0	87	2516	0ae7a	00207800	3	0	2d		0a910000	8	
214	0	0	0	0	87	2516	08964	00207800	3	0	2d		0a910000	8	
215	0	0	0	0	71	1edf	0c89c	0040f000	2	0	2d		15220000	7	
216	0	0	0	0	71	1edf	0a9bd	0040f000	2	0	2d		15220000	7	
217	0	0	0	0	71	1edf	08ade	0040f000	2	0	2d		15220000	7	
218	0	0	0	0	72	1aa9	0d7fe	0081e000	1	0	2d		2a440000	6	
219	0	0	0	0	72	1aa9	0bd55	0081e000	1	0	2d		2a440000	6	
220	0	0	0	0	72	1aa9	0a2ac	0081e000	1	0	2d		2a440000	6	
221	0	0	0	0	72	1aa9	08803	0081e000	1	0	2d	2d	2a440000	6	
222	0	0	0	0	73	174e	0dab4	0003c000	8	0	20		54880000	5	
223	0	0	0	0	73	174e	0c366	0003c000	8	0	20		54880000	5	

Tableau 26 (fin)

								CODEUR				DÉCODEUR			
EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
224	0	0	0	0	73	174e	0ac18	0003c000	8	0	20		54880000	5	
225	0	0	0	0	73	174e	094ca	0003c000	8	0	20		54880000	5	
226	0	0	0	0	74	1424	0faf8	00078000	7	0	20		a9100000	4	
227	0	0	0	0	74	1424	0e6d4	00078000	7	0	20		a9100000	4	
228	0	0	0	0	74	1424	0d2b0	00078000	7	0	20		a9100000	4	
229	0	0	0	0	74	1424	0be8c	00078000	7	0	20		a9100000	4	
230	1	0	0	0	74	1424	0aa68	00078000	7	0	20		a9100000	4	
231	1	0	0	0	72	1aa9	0a120	0040b220	4	0	20		96600000	1	00
232	0	0	0	0	87	2516	0d548	0209c4b8	1	0	20		7f480000	6	
233	0	0	0	0	87	2516	0b032	0209c4b8	1	0	20		7f480000	6	
234	1	0	0	0	87	2516	08b1c	0209c4b8	1	0	20	20	7f480000	6	
235	0	0	0	0	86	299a	09458	0008aaf8	7	0	82		65080000	4	
236	1	0	0	0	87	2516	0d57c	001155f0	6	0	82		ca100000	3	
237	0	0	0	0	86	299a	09458	00481958	4	0	82		66a80000	1	00
238	1	0	0	0	87	2516	0d57c	009032b0	3	0	82		cd500000	8	
239	1	0	0	0	86	299a	09458	02438c58	1	0	82	82	73a80000	6	
240	0	0	0	0	93	32b4	0a668	000fdc58	7	0	90		23a80000	4	
241	0	0	0	0	94	2e17	0e768	001fb8b0	6	0	90		47500000	3	
242	0	0	0	0	94	2e17	0b951	001fb8b0	6	0	90		47500000	3	
243	0	0	0	0	94	2e17	08b3a	001fb8b0	6	0	90		47500000	3	
244	0	0	0	0	86	299a	0ba46	003f7160	5	0	90		8ea00000	2	
245	1	0	0	0	86	299a	090ac	003f7160	5	0	90		8ea00000	2	00
246	1	0	0	0	93	32b4	0a668	00ff61c8	3	0	90		9e380000	8	
247	1	0	0	0	99	3c3d	0cad0	03ff55f0	1	0	90		aa100000	6	
248	0	0	0	0	104	415e	0f0f4	000f920c	7	1	90		6df40000	4	
249	0	0	0	0	104	415e	0af96	000f920c	7	1	90		6df40000	4	
250	1	0	0	0	99	3c3d	0dc70	001f2418	6	1	90		dbe80000	3	
251	1	0	0	0	104	415e	0f0f4	007f112c	4	1	90		eed40000	1	00
252	0	0	0	1	103	4639	082bc	00ff8184	3	1	90		7e7c0000	8	
253	1	0	0	0	104	415e	08c72	01ff7c0e	2	1	90		83f20000	7	
254	0	0	0	1	103	4639	082bc	03ff8e44	1	1	90		71bc0000	6	
255	1	0	0	0	104	415e	08c72	0007958e	8	2	90		6a720000	5	
256	0	0	0	1	103	4639	082bc	000fc144	7	2	90		3ebc0000	4	
257							08c72	08000000	6	2	90	91			

7.2 Algorithme paramétré

Ce paragraphe normatif décrit des méthodes de test pour l'algorithme spécifié dans les premiers articles de la présente Spécification. Comme de nombreux paramétrages (choix de valeurs et plages de valeurs des paramètres) sont possibles, ce paragraphe se limite aux moyens de test de certains paramétrages considérés comme utiles pour la mise au point de réalisations. Si une mise en œuvre du codeur est déclarée prendre en charge un paramétrage aussi étendu ou plus étendu que celui de toute configuration pour laquelle des données de test sont fournies dans le présent paragraphe, ladite mise en œuvre doit générer exactement les nombres d'octets indiqués pour ces données de test. Si une mise en œuvre du décodeur est déclarée prendre en charge un paramétrage aussi étendu ou plus étendu que celui de toute configuration pour laquelle des données de test sont fournies dans le présent paragraphe, ladite mise en œuvre doit décoder ces données de test (générées par une réalisation du codeur qui satisfasse les prescriptions de mise en œuvre du codeur stipulées ci-dessus) et générer exactement l'image artificielle décrite en 7.2.1. Un codeur prenant en charge les gabarits adaptatifs, mais n'utilisant pas l'algorithme suggéré dans l'Annexe C pour déterminer les déplacements du pixel AT, doit imposer artificiellement des déplacements de pixel AT identiques à ceux décrits ici. En outre, s'il n'a pas été choisi pour un codeur de retirer tous les octets 0x00 possibles se trouvant à la fin de toutes les SDE, un post-traitement de cette nature est nécessaire pour dupliquer les nombres d'octets donnés.

7.2.1 Image artificielle

Les divers tests de l'algorithme complet utilisent une image générée artificiellement par l'organigramme de la Figure 38. Elle comporte 1960 pixels par ligne, 1951 lignes et contient 861 965 pixels d'avant-plan et 2 961 995 pixels d'arrière-plan.

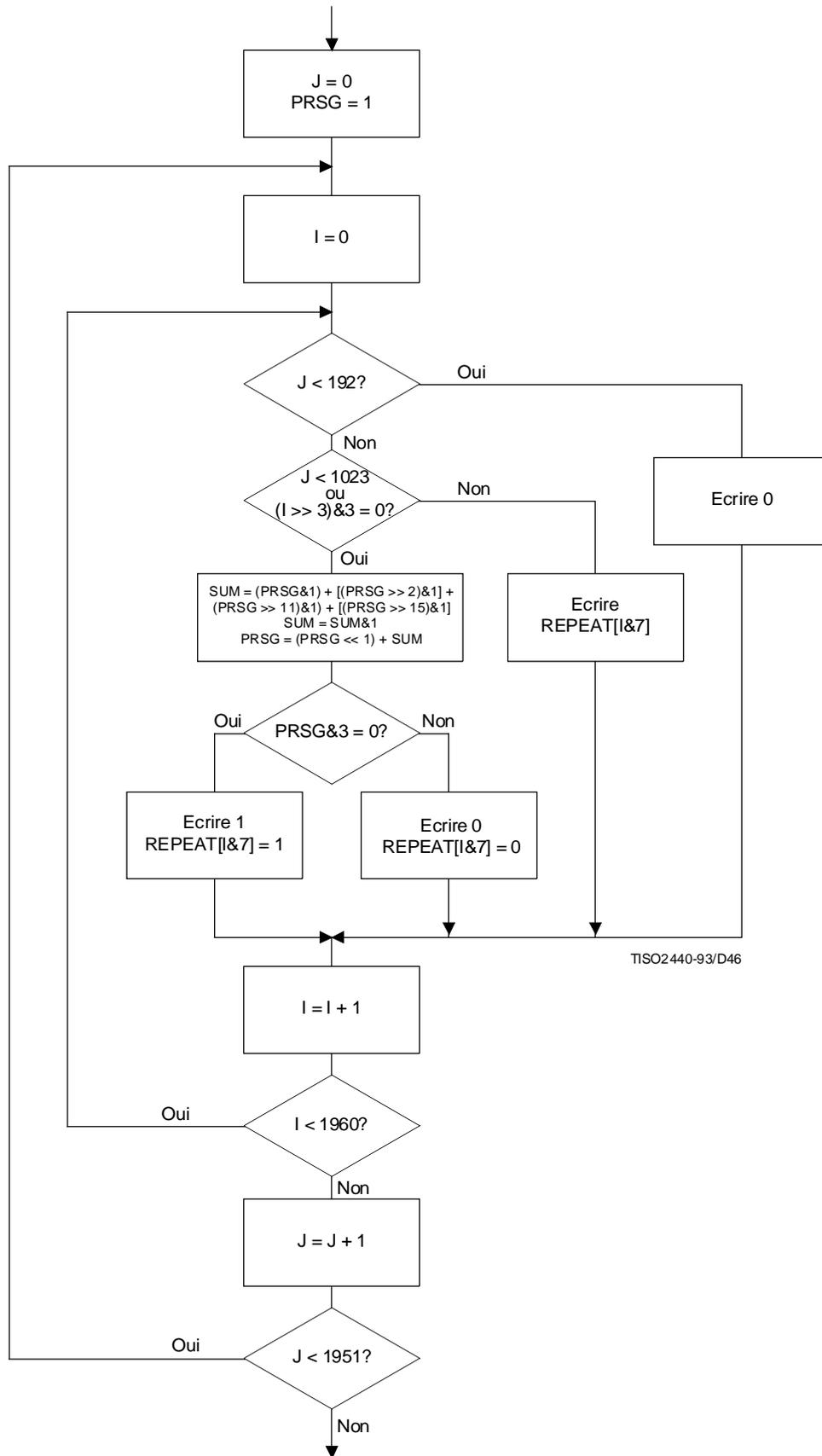


Figure 38 – Procédure de génération d'une image de test

Construite pour tester un maximum de fonctions, cette image n'est en aucun sens une image typique: les résultats de sa compression ne sont pas représentatifs.

7.2.2 Tests du codage séquentiel simple

Pour les trois tests de ce paragraphe, $D_L = 0$, $D = 0$, $P = 1$, $X_0 = 1960$, $Y_0 = 1951$ et $M_Y = 0$. Les valeurs de HITOLO, SEQ, ILEAVE, SMID, VLENGTH, TPDON, DPON, DPPRIV, DPLAST n'ont pas d'importance. Les quatre paramètres restants, L_0 , M_X , LRLTWO et TPBON varient comme indiqué dans le Tableau 27. Les trois dernières colonnes de ce tableau donnent les données de trace obtenues lorsque l'image d'entrée est l'image artificielle du paragraphe précédent. Pour chacun des deux premiers tests, il n'y a qu'un seul train de SCD, dans la longueur et le nombre d'octets indiqués. Le test final, pour $L_0 = 128$, produit 16 SCD, dont la somme des longueurs est le nombre d'octets indiqué. Dans tous les cas, tous les éventuels octets de fin 0x00 sont retirés de la fin de chaque SCD (voir 6.8.2.10 et Figure 28).

Tableau 27 – Paramètres de trace des tests du codage séquentiel simple

TPBON	M_X	LRLTWO	L_0	Pixels TP	Pixels codés	Octets codés
0	0	0	1951	0	3 823 960	316 094
0	0	1	1951	0	3 823 960	315 887
1	8	0	128	376 320	3 447 640	252 557

Dans les deux premiers tests, AT est désactivé puisque M_X est mis à zéro. Le test final réactive AT, ainsi que la TP de la couche de plus basse résolution. Le traitement AT est réalisé comme décrit dans l'Annexe C en se conformant à la suggestion de différer tous les changements d'AT jusqu'au début de la prochaine bande. Les données de la dernière ligne du Tableau 28 ont été obtenues en utilisant SDNORM, qui implique que l'estimateur de probabilité n'a pas été réinitialisé. Le Tableau 29 donne des informations utiles pour la mise au point du traitement AT. Les deux premières colonnes donnent la bande et la ligne auxquelles le changement d'AT devient effectif (la numérotation des lignes et des bandes commence à zéro), et la troisième colonne donne le décalage τ_X de la nouvelle position du pixel AT. La colonne finale 8 donne les valeurs des compteurs décrits dans l'Annexe C lorsque le déplacement de pixels AT est déclenché.

Tableau 28 – Informations relatives au changement d'AT pour le troisième test du codage séquentiel simple

Bande	Ligne	τ_X	C_{all}	C_0	C_3	C_4	C_5	C_6	C_7	C_8
9	0	8	3900	2336	2456	2472	2446	2442	2730	3534

Tableau 29 – Nombre d'octets des tests du codage séquentiel simple

TPBON	M_X	LRLTWO	L_0	SCD	PSCD	SDE	BID	BIE
0	0	0	1951	316 094	317 362	317 364	317 364	317 384
0	0	1	1951	315 887	317 110	317 112	317 112	317 132
1	8	0	128	252 557	253 593	253 625	253 633	253 653

D'autres données pour ces trois tests sont données dans le Tableau 30. Les entrées de la colonne SCD reproduisent la dernière colonne du Tableau 27. Les données codées de bande protégées (PSCD) (*protected stripe coded data*) sont obtenues à partir des données codées de bande (SCD) (*stripe coded data*) en remplaçant chaque octet calé à 0xff par 0xff (ESC) et 0x00 (STUFF). L'entité de données de bande (SDE) (*stripe data entity*) est obtenue à partir des PSCD, en concaténant un octet 0xff (ESC) et un octet 0x02 (SDNORM) à la fin de chaque PSCD. Les données d'image binaire ont le même nombre d'octets que dans la SDE, sauf pour le test final, où elles ont 8 octets de plus, à cause du segment marqueur de changement d'AT. Enfin, la BIE a une longueur supérieure de 20 octets (taille de l'en-tête) à celle des BID.

Tableau 30 – Paramètres de trace du codage de l'image artificielle

Couche	X_d	Y_d	L_d	Lignes TP	Exception sTP	Pixels TP	Pixels DP	Pixels codés	Octets codés
6	1960	1951	128	137	7033	375 520	589 344	2 859 096	188 817
5	980	976	64	186	1442	93 120	128 642	734 718	65 584
4	490	488	32	181	135	22 792	30 230	186 098	16 565
3	245	244	16	117	8	5406	7246	47 128	4994
2	123	122	8	61	0	1238	1769	11 999	1430
1	62	61	4	31	0	248	452	3082	370
0	31	31	2	3	–	93	–	868	113

7.2.3 Test du codage séquentiel progressif et du codage séquentiel compatible progressif

Pour le test de ce paragraphe, l'image d'entrée est encore l'image artificielle, mais cette fois, les paramètres sont positionnés comme suit (voir le Tableau 9): $D_L = 0$, $D = 6$, $P = 1$, $X_6 = 1960$, $Y_6 = 1951$, $L_0 = 2$, $M_X = 8$, $M_Y = 0$, HITOLO = 0 (sans importance), SEQ = 0 (sans importance), ILEAVE = 0 (sans importance), SMID = 0 (sans importance), LRLTWO = 0, VLENGTH = 0 (sans importance), TPDON = 1, TPBON = 1, DPPRIV = 0 et DPLAST = 0 (sans importance). Bien que le paramétrage ci-dessus soit à codage progressif, la valeur de SEQ est sans importance et le même nombre d'octets est généré en codage séquentiel compatible progressif. Les présentes données de test s'appliquent donc aux deux modes. Le traitement AT est encore réalisé comme suggéré dans l'Annexe C, tous les changements d'AT qu'il est désirable de différer jusqu'au début de la prochaine bande étant différés.

Le Tableau 31 donne les données de trace pour le codage de l'image artificielle avec les paramètres ci-dessus. Le nombre d'octets codés indiqué dans la dernière colonne est la somme des nombres d'octets des 16 SCD de la couche. Comme plus haut, tous les octets 0x00 présents sont, si possible, retirés.

Deux changements de gabarit adaptatif ont lieu, un à la couche 6 et un autre à la couche 5. Les données correspondant à ces deux changements sont données dans le Tableau 31.

Tableau 31 – Informations relatives au changement d'AT

Couche	Bande	Ligne	τ_X	C_{all}	C_0	C_3	C_4	C_5	C_6	C_7	C_8
6	9	0	8	3243	1984	2014	2055	2031	2001	2212	2924
5	10	0	4	2580	1323	1401	2259	1440	1447	1426	1966

Les nombres d'octets par couche des 16 PSCD et SDE sont indiqués dans le Tableau 32, ainsi que le nombre total d'octets des BID et de la BIE (voir 6.2). Les données codées de bande protégées (PSCD) sont obtenues à partir des données codées de bande (SCD) en remplaçant chaque octet calé à 0xff par 0xff (ESC) et 0x00 (STUFF). L'entité de données de bande (SDE) est obtenue à partir des PSCD en concaténant un octet 0xff (ESC) et un octet 0x02 (SDNORM) à la fin de chaque PSCD. Enfin, le champ de données d'image binaire BID est obtenu en concaténant les 112 SDE (7 couches de 16 bandes) et les deux segments marqueurs ATMOVE, la BIE étant obtenue en concaténant les BID aux BIH de vingt octets.

Tableau 32 – Nombre d'octets pour l'image artificielle

Couche	SCD	PSCD	SDE	BID	BIE
6	188 817	189 584	189 616		
5	65 584	65 905	65 937		
4	16 565	16 634	16 666		
3	4994	5010	5042		
2	1430	1434	1466		
1	370	373	405		
0	113	114	146		
Total	277 873	279 054	279 278	279 294	279 314

7.3 Exemples de suites de données

Voici un exemple de BIE pour le codage séquentiel simple d'une image binaire de largeur 1728 pixels et de hauteur 2376 pixels, comportant une seule bande pour toute l'image, et dont tous les paramètres binaires sont mis à zéro (en hexadécimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x09 0x48|0x00|0x00|0x00|0x00|
|      PSCD pour image entière |0xff|0x02|
```

(Les barres verticales déterminent des regroupements logiques, sans autre signification que celle de l'espacement normal.)

Voici un exemple de BIE pour un codage séquentiel compatible progressif d'une image binaire de largeur 1728 pixels et de hauteur 2376 pixels, comportant une couche différentielle, 64 lignes par bande dans l'image réduite et dont tous les paramètres binaires, sauf SEQ, sont mis à zéro (en hexadécimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x00 0x40|0x00|0x00|0x04|0x00|
| PSCD pour bande 0 et couche 0 (lignes image de base 0 à 63)|0xff|0x02|
| PSCD pour bande 0 et couche 1 (lignes image diff. 0 à 127)|0xff|0x02|
| PSCD pour bande 1 et couche 0 (lignes image de base 64 à 127)|0xff|0x02|
| PSCD pour bande 1 et couche 1 (lignes image diff. 128 à 255)|0xff|0x02|
.
.
.
| PSCD pour bande 17 et couche 0 (lignes image de base 1088 à 1151)|0xff|0x02|
| PSCD pour bande 17 et couche 1 (lignes image diff. 2176 à 2303)|0xff|0x02|
| PSCD pour bande 18 et couche 0 (lignes image de base 1152 à 1187)|0xff|0x02|
| PSCD pour bande 18 et couche 1 (lignes image diff. 2304 à 2375)|0xff|0x02|
```

Les commentaires entre parenthèses indiquent les lignes de l'image de base et des images différentielles qui sont codées dans chaque bande.

Voici un exemple de BIE pour un codage progressif d'une image binaire de largeur 1728 pixels et de hauteur 2376 pixels, comportant une couche différentielle, 64 lignes par bande dans l'image réduite et dont tous les paramètres binaires sont mis à zéro (en hexadécimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x00 0x40|0x00|0x00|0x00|0x00|
| PSCD pour bande 0 et couche 0 (lignes image de base 0 à 63)|0xff|0x02|
| PSCD pour bande 1 et couche 0 (lignes image de base 64 à 127)|0xff|0x02|
.
.
.
| PSCD pour bande 17 et couche 0 (lignes image de base 1088 à 1151)|0xff|0x02|
| PSCD pour bande 18 et couche 0 (lignes image de base 1152 à 1187)|0xff|0x02|
| PSCD pour bande 0 et couche 1 (lignes image diff. 0 à 127)|0xff|0x02|
| PSCD pour bande 1 et couche 1 (lignes image diff. 128 à 255)|0xff|0x02|
.
.
.
| PSCD pour bande 17 et couche 1 (lignes image diff. 2176 à 2303)|0xff|0x02|
| PSCD pour bande 18 et couche 1 (lignes image diff. 2304 à 2375)|0xff|0x02|
```

Annexe A

Plages minimales suggérées pour les paramètres libres

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

Pour les applications, les 19 paramètres libres peuvent être positionnés à toutes les valeurs des plages spécifiées dans le Tableau 9. Les plages minimales suggérées dans la présente annexe ont pour objet de permettre à une large gamme d'applications de partager un même matériel et d'échanger des données d'image décodables. Sauf nécessité absolue, il est déconseillé de choisir, pour des applications, des valeurs de paramètres extérieures aux plages minimales suggérées.

Le Tableau A.1 donne les plages minimales suggérées du décodeur. Les matériels et logiciels destinés à une utilisation générale, dotés de divers dispositifs de sortie, doivent être conçus pour pouvoir prendre en charge, s'ils comportent une mémoire externe suffisante, des choix de paramètres dans les plages indiquées. Un décodeur complet, doté d'un dispositif de sortie spécifique, doit assurer la prise en charge de ces mêmes plages, avec des exceptions suivantes évidentes, dont la prise en charge n'est pas nécessaire:

- dimensions d'image X_D et Y_D au-delà des capacités du dispositif de sortie particulier utilisé;
- valeurs de P au-delà de la capacité du dispositif de sortie particulier utilisé;
- autre ordre de bande que celui défini par SEQ, ILEAVE et SMID.

NOTE – Dans les applications à codage séquentiel simple, la plage maximale de D est zéro, la plage maximale de L_0 est Y_D , et aucune prise en charge n'est requise pour les paramètres HITOLO, SEQ, TPDON, DPON, DPPRIV et DPLAST.

Tableau A.1 – Plages minimales suggérées pour les paramètres libres

Paramètre	Minimum	Maximum
D_L	0	D
D	D_L	6
P	1	4
X_D	1	5184
Y_D	1	8192
L_0	1	$128/2^D$, pour $D > 0$ Y_D , pour $D = 0$
M_X	0	8, pour codeurs 16, pour décodeurs
M_Y	0	0
HITOLO	0	0
SEQ	0	1
ILEAVE	0	1
SMID	0	1
LRLTWO	0	1
VLENGTH	0	1
TPDON	0	1
TPBON	0	1
DPON	0	1
DPPRIV	0	1
DPLAST	0	1

Annexe B

Conception de la table de réduction de résolution

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

B.1 Filtrage

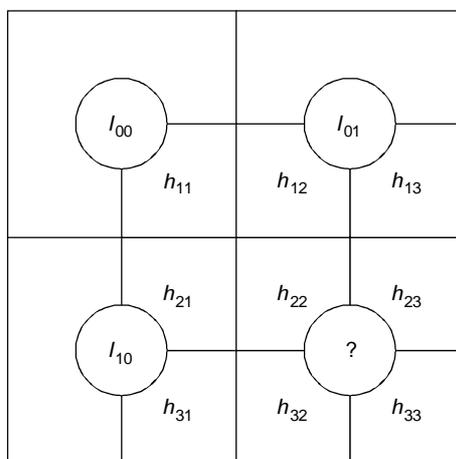
Le principe sous-jacent à l'algorithme de réduction de résolution est la préservation de la densité, par l'utilisation d'un filtrage (une équation aux différences). Mais il est parfois nécessaire de remplacer la sortie du filtre par une autre valeur, pour préserver les bords, les traits, les motifs périodiques et les motifs tramés. Ces remplacements sont appelés «exceptions» (à la règle générale).

La Figure B.1 reproduit la Figure 4, avec quelques différences pour les noms des divers pixels. La couleur du pixel cible «?» est déterminée en utilisant non seulement les quatre pixels de haute résolution correspondants, h_{22} , h_{23} , h_{32} , h_{33} , mais également les pixels périphériques h_{11} , h_{12} , h_{13} , h_{21} , h_{31} , ainsi que les pixels de basse résolution déjà connus l_{00} , l_{01} , l_{10} . Si la valeur de l_{00} n'est pas égale à celle de h_{11} , cette différence de valeur affecte la décision concernant la valeur courante. De même, des différences de valeur entre l_{10} et h_{21} , entre l_{10} et h_{31} , entre l_{01} et h_{12} et entre l_{01} et h_{13} affectent la décision concernant la valeur courante. On est ainsi conduit à calculer l'expression suivante:

$$4h_{22} + 2(h_{23} + h_{32}) + h_{33} + (h_{11} - l_{00}) + 2(h_{21} - l_{10}) + (h_{31} - l_{10}) + 2(h_{12} - l_{01}) + (h_{13} - l_{01}) \tag{B-1}$$

équivalent à:

$$4h_{22} + 2(h_{12} + h_{21} + h_{23} + h_{32}) + (h_{11} + h_{13} + h_{31} + h_{33}) - 3(l_{01} + l_{10}) - l_{00} \tag{B-2}$$



T0806690-90/D47

Figure B.1 – Pixels utilisés pour déterminer la couleur d'un pixel de basse résolution

La Figure B.2 donne une représentation graphique des pondérations correspondant à la dernière forme de cette expression.

En supposant une équiprobabilité de l'avant-plan et de l'arrière-plan et une indépendance statistique des pixels, la valeur prévue de l'expression (B-2) est 4,5. En principe, la valeur 1 est attribuée au pixel l_{11} , si et seulement si l'expression (B-2) a une valeur supérieure à 4,5 (ce doit être un entier).

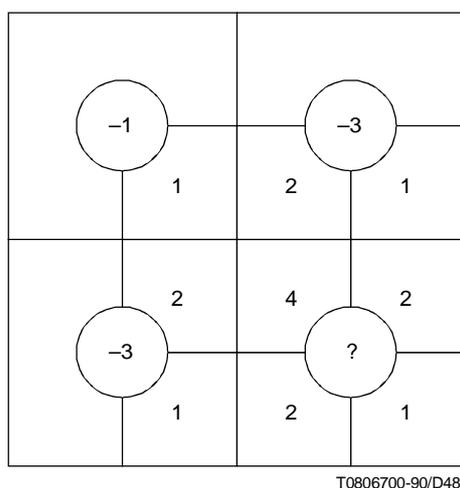


Figure B.2 – Pondération des pixels

B.2 Exceptions

B.2.1 Préservation des bords

La préservation des bords est la raison la plus importante pour introduire des exceptions. Les exceptions définies pour les bords permettent d'avoir des limites de zone de couleur uniforme franches et non en zigzag, que ces limites coïncident avec une ligne ou une colonne de haute résolution, paire ou impaire. Les contextes des 132 exceptions relatives aux bords sont indiqués ci-après en hexadécimal. Les couleurs des pixels sont mises en correspondance avec ces contextes entiers par la table de signification des bits de la Figure 4. Pour tous ces 132 contextes, la couleur effective choisie est inversée par rapport à celle donnée par la règle générale.

```

0x007 0x207 0x407 0x607 0x807 0xa07 0xc07 0xe07 0x20f 0x40f 0x60f 0xa0f
0xc0f 0xe0f 0x617 0xa17 0xc17 0xe17 0x61f 0xe1f 0x227 0x427 0x627 0xa27
0xc27 0xe27 0x62f 0xa2f 0xc2f 0xe2f 0x637 0xe37 0x247 0x447 0x647 0x847
0xa47 0xc47 0xe47 0x049 0x249 0x449 0x649 0x849 0xa49 0xc49 0xe49 0x24b
0x44b 0x64b 0xa4b 0xc4b 0xe4b 0x24d 0x44d 0x64d 0x84d 0xa4d 0xc4d 0xe4d
0x659 0xa59 0xc59 0xe59 0x65b 0xe5b 0x269 0x469 0x669 0xa69 0xc69 0xe69
0x287 0x487 0x687 0xa87 0xc87 0xe87 0x8b6 0xab6 0x2c9 0x4c9 0x6c9 0xac9
0xcc9 0xec9 0x6cb 0xacb 0xccb 0xecb 0x6d9 0xed9 0x8f8 0xcf8 0x307 0x507
0x707 0x907 0xb07 0xd07 0xf07 0x334 0x336 0xb36 0x349 0x549 0x749 0x949
0xb49 0xd49 0xf49 0x578 0xd78 0x396 0xb96 0x3a6 0xbb2 0x3b4 0xbb4 0x3b6
0xbb6 0xfb6 0xdb8 0x5d0 0x5d8 0xdd8 0x5e8 0x5f0 0xdf0 0x5f8 0xdf8 0xff8

```

B.2.2 Préservation des traits

Les traits verticaux et horizontaux sont préservés par la règle générale, complétée des exceptions relatives aux bords. Les exceptions définies pour les traits permettent de préserver la continuité des lignes obliques. Les exceptions relatives aux traits sont importantes pour la qualité des images de textes. Les 420 exceptions relatives aux traits sont indiquées ci-après:

```

0x003 0x009 0x00a 0x022 0x088 0x0a6 0x0e8 0x0ee 0x116 0x124 0x158 0x168
0x170 0x186 0x194 0x1a2 0x1a8 0x1c0 0x20a 0x20e 0x215 0x21c 0x21d 0x223
0x22a 0x22e 0x231 0x235 0x236 0x24a 0x24f 0x252 0x254 0x270 0x277 0x289
0x291 0x294 0x2a4 0x2a8 0x2dd 0x2e0 0x2ed 0x2ee 0x311 0x312 0x318 0x322

```

ISO/CEI 11544:1993(F)

0x323 0x327 0x350 0x35b 0x35f 0x36b 0x36e 0x376 0x377 0x388 0x394 0x3ab
0x3ad 0x3c0 0x3c1 0x3c2 0x3c4 0x3c8 0x3c9 0x3dd 0x3e0 0x3f5 0x40a 0x40e
0x41c 0x423 0x431 0x44a 0x44f 0x451 0x452 0x453 0x454 0x470 0x477 0x489
0x48a 0x491 0x494 0x4a2 0x4a4 0x4ca 0x4d1 0x4d8 0x4dd 0x4e0 0x4e7 0x4ee
0x511 0x512 0x514 0x518 0x51f 0x522 0x524 0x525 0x526 0x527 0x52c 0x55f
0x564 0x570 0x577 0x588 0x589 0x58f 0x5a4 0x5ab 0x5c9 0x5ce 0x5dc 0x5e3
0x5f5 0x60e 0x61c 0x623 0x631 0x64a 0x64f 0x652 0x657 0x65d 0x66b 0x66f
0x670 0x677 0x678 0x679 0x689 0x68f 0x691 0x694 0x696 0x697 0x6a7 0x6b0
0x6b1 0x6b2 0x6b4 0x6b8 0x6cf 0x6dd 0x6e9 0x6f0 0x711 0x712 0x717 0x718
0x722 0x725 0x726 0x727 0x72c 0x72f 0x733 0x734 0x735 0x736 0x737 0x738
0x73c 0x757 0x759 0x75d 0x75f 0x764 0x776 0x777 0x788 0x792 0x799 0x7a6
0x7a7 0x7b4 0x7c1 0x7c2 0x7c4 0x7c8 0x7c9 0x7cb 0x7d0 0x7d1 0x7d2 0x7d8
0x7d9 0x7dc 0x7dd 0x7e8 0x7e9 0x7f0 0x7f5 0x80e 0x822 0x823 0x826 0x830
0x84a 0x866 0x86d 0x888 0x889 0x890 0x8a4 0x8c7 0x8c8 0x8cc 0x8e0 0x8ee
0x90f 0x916 0x922 0x924 0x925 0x947 0x94b 0x94d 0x94f 0x958 0x964 0x968
0x969 0x970 0x986 0x987 0x988 0x994 0x9b0 0x9c0 0x9c1 0x9c4 0xa0a 0xa0e
0xa1c 0xa23 0xa2a 0xa2e 0xa31 0xa32 0xa35 0xa4a 0xa4f 0xa52 0xa54 0xa56
0xa70 0xa74 0xa77 0xa88 0xa89 0xa8f 0xa91 0xa94 0xaa4 0xaa7 0xadd 0xae0
0xae2 0xae4 0xae8 0xaed 0xaee 0xb11 0xb12 0xb13 0xb18 0xb22 0xb27 0xb2e
0xb31 0xb5b 0xb6b 0xb6e 0xb76 0xb88 0xb89 0xb91 0xba8 0xbab 0xbac 0xbad
0xbb0 0xbb5 0xbc0 0xbc1 0xbc2 0xbc4 0xbc8 0xbc9 0xbd0 0bdd 0xbe0 0xbe2
0xbe4 0xbe8 0xbf5 0xc0a 0xc0e 0xc1c 0xc22 0xc23 0xc31 0xc4a 0xc4f 0xc52
0xc54 0xc5c 0xc6b 0xc70 0xc77 0xc88 0xc89 0xc8a 0xc91 0xc94 0xc98 0xca4
0xca6 0xcac 0cca 0xcd1 0xcd4 0xcd 0xce0 0xce4 0xce7 0xce9 0cee 0xd11
0xd12 0xd18 0xd19 0xd1f 0xd22 0xd23 0xd24 0xd25 0xd26 0xd27 0xd2c 0xd31
0xd34 0xd64 0xd77 0xd88 0xd8f 0xd91 0xda2 0xda4 0xda6 0xdab 0xdac 0xdb0
0xdc9 0xdca 0xdce 0xddc 0xde2 0xde3 0xde4 0xdf1 0xdf5 0xe0e 0xe1c 0xe23
0xe31 0xe4a 0xe4f 0xe52 0xe57 0xe5d 0xe5f 0xe6b 0xe6f 0xe70 0xe79 0xe89
0xe8f 0xe91 0xe94 0xe97 0xea4 0xea7 0xecf 0xee0 0xee9 0xeef 0xf11 0xf12
0xf17 0xf18 0xf22 0xf27 0xf2f 0xf37 0xf59 0xf5f 0xf77 0xf88 0xfa7 0xfaf
0xfb1 0xfc9 0xfcb 0xfd9 0xdd 0xfe7 0xfe9 0xfeb 0xfed 0xff5 0xff6 0xffc

B.2.3 Préservation des motifs périodiques

La plupart des motifs périodiques sont préservés par la règle générale. Certaines exceptions sont toutefois nécessaires pour améliorer les résultats dans des zones de transition vers ou à partir de motifs périodiques. Les 10 exceptions relatives aux motifs périodiques sont indiquées ci-après:

0x638 0xa38 0x692 0xc92 0xaaa 0xcaa 0xb55 0xd55 0x36d 0x5c7

B.2.4 Préservation des motifs tramés

Les 12 exceptions relatives aux motifs tramés contribuent à préserver la juxtaposition de points de couleurs différentes de très basse densité ou de très haute densité, c'est-à-dire de pixels d'arrière-plan ou d'avant-plan isolés. Elles sont indiquées ci-après:

0x010 0x028 0x082 0x085 0xeba 0xebd 0x142 0x145 0xf7a 0xf7d 0xfd7 0xfef

Annexe C

Changements de gabarit adaptatif

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

C.1 Considérations générales

La technique décrite dans la présente annexe implique des calculs simples et permet une bonne détermination de l'opportunité d'un déplacement de pixel AT. Elle procure un gain de codage substantiel, atteignant parfois 80%, sur des images contenant des demi-teintes.

La description suppose que $M_Y = 0$, c'est-à-dire que le pixel AT n'est autorisé à se déplacer que sur la ligne. (La plage minimale suggérée prend en charge uniquement cette situation.) La généralisation à M_Y différent de zéro serait évidente, mais l'algorithme résultant n'a pas été testé et cette généralisation impliquerait un traitement substantiel par le codeur pour les grandes valeurs de M_Y .

L'algorithme détermine, au début de chaque bande, le pixel AT ayant la plus grande valeur de prédiction pour le pixel cible. Lorsque l'algorithme trouve un pixel AT n'appartenant pas au gabarit et ayant une valeur de prédiction beaucoup plus grande que celle du pixel AT qui y est configurée, une permutation peut être souhaitable. Il est toutefois essentiel que ces permutations demeurent peu fréquentes et soient fortement justifiées. A chaque permutation de gabarits, l'estimation de probabilité conservée par le codeur arithmétique se dégrade, pendant le délai nécessaire pour une nouvelle adaptation.

C.2 Couches différentielles

Les Figures C.1 et C.2 donnent un organigramme de traitement d'AT d'une couche différentielle. Un vecteur de compteurs est utilisé pour déterminer la valeur de prédiction. Le compteur c_n , ($n = 0$ ou $3 \leq n \leq M_X$), compte le nombre de coïncidences de polarité entre le pixel cible et le pixel AT candidat n . Le pixel AT par défaut est le pixel AT candidat 0. Les pixels candidats n , $3 \leq n \leq M_X$ sont positionnés à $\tau_X = n$ et $\tau_Y = 0$. On dit qu'une coïncidence de polarité se produit lorsque deux pixels sont d'arrière-plan ou d'avant-plan.

L'organigramme des Figures C.1 et C.2 est exécuté une fois au début de chaque bande. Les compteurs sont remis à 0 et les coïncidences sont comptées jusqu'à la fin d'une ligne, le nombre maximal possible de ces coïncidences étant supérieur ou égal à 2048. Lorsque cette valeur est atteinte, il faut vérifier si un ensemble de conditions imbriquées est satisfait: si toutes ces conditions sont satisfaites, la permutation de gabarits est opérée. Autrement, le gabarit est laissé tel quel, au moins jusqu'à la fin de la bande.

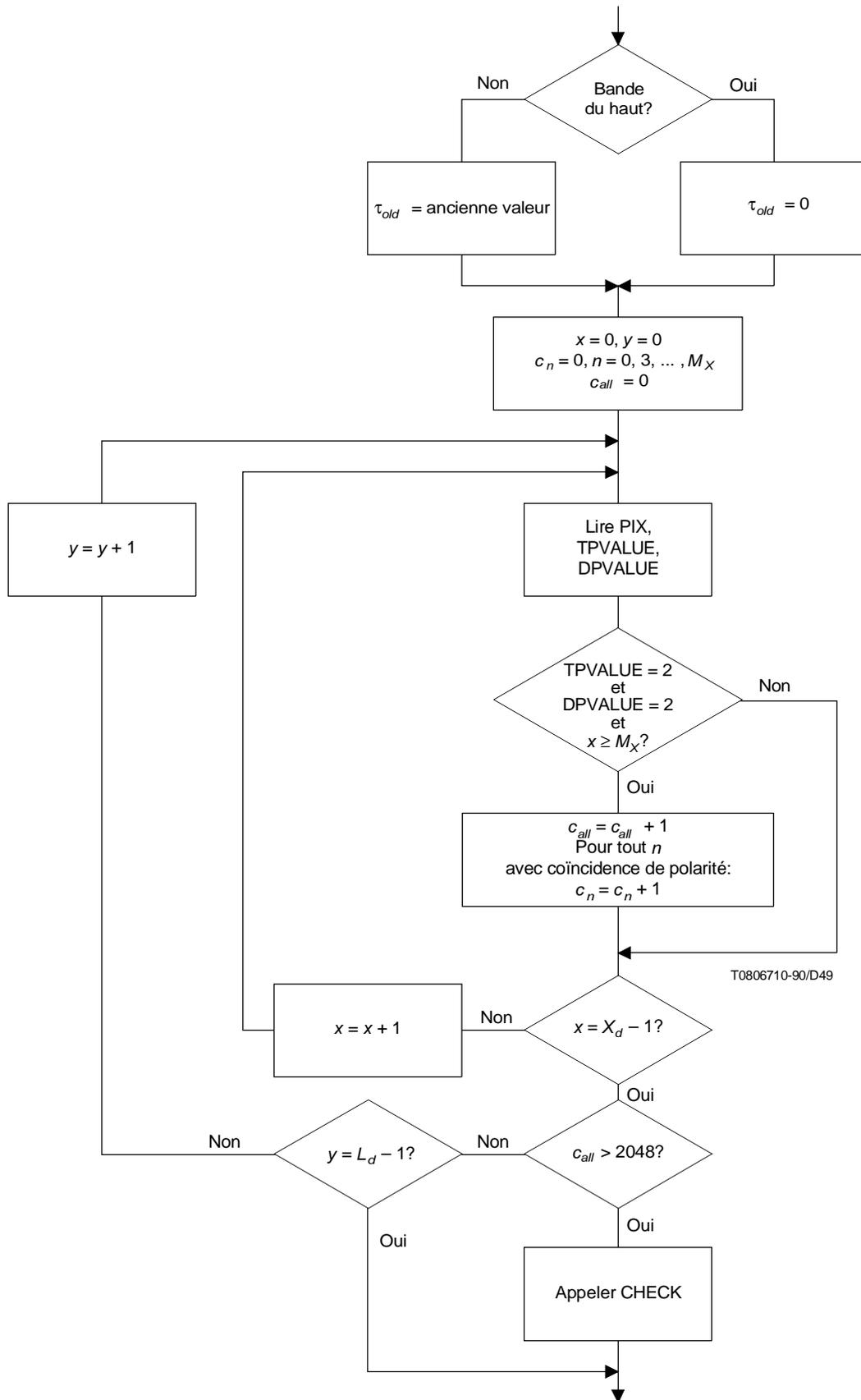
Une solution raisonnable pour économiser les calculs, si le nombre de bandes par image n'est pas nettement inférieur au nombre suggéré 35, est de se contenter de vérifier une fois par bande si un changement de gabarit est possible. Avec quelques bandes par image – notamment avec une seule bande par image – une détermination continue ou périodique, sur chaque bande, de l'opportunité d'une permutation de gabarits, peut être nécessaire.

Aux très basses résolutions, une bande peut comporter moins de 2048 pixels: le présent algorithme n'opère alors aucun changement de gabarit AT. Ce comportement est probablement avantageux: avec un trop petit nombre de pixels, les données sont entachées de bruit et ne pourraient pas donner d'indication fiable quant au point vers lequel il conviendrait de déplacer le pixel AT.

S'il a été déterminé qu'un changement d'AT est souhaitable, deux moments peuvent raisonnablement être choisis pour l'effectuer. Une solution consiste à opérer le changement au début de la prochaine ligne. Cette méthode procure un gain de codage plus élevé, mais nécessite soit un traitement AT lors d'une première passe, soit la mise en mémoire tampon par le codeur des SCD d'une bande donnée, pour pouvoir les précéder d'un segment marqueur `ATMOVE`, lorsqu'un changement d'AT est estimé souhaitable lors du codage de cette bande. L'autre possibilité, qui implique une très légère perte d'efficacité de codage, (en supposant encore un nombre raisonnable de bandes par image), est d'effectuer le changement au début ($y_{AT} = 0$) de la bande suivante. Dans ce cas, le segment marqueur AT n'a pas besoin d'apparaître avant le début de cette bande, et il est encore possible d'utiliser le codeur de façon itérative (en «pipeline»).

C.3 Couche de plus basse résolution

Le traitement AT de la couche de plus basse résolution peut être effectué d'une façon presque identique à celle décrite au paragraphe précédent pour les couches différentielles. Seulement trois modifications doivent être apportées à la Figure C.1: premièrement, supprimer la condition `DPVALUE = 2` du bloc central; deuxièmement, remplacer la condition $x \geq M_X$ par $M_X \leq x < X_d - 2$; enfin, changer de façon appropriée la plage d'évolution de l'indice n du compteur.



T0806710-90/D49

Figure C.1 – Organigramme du traitement AT

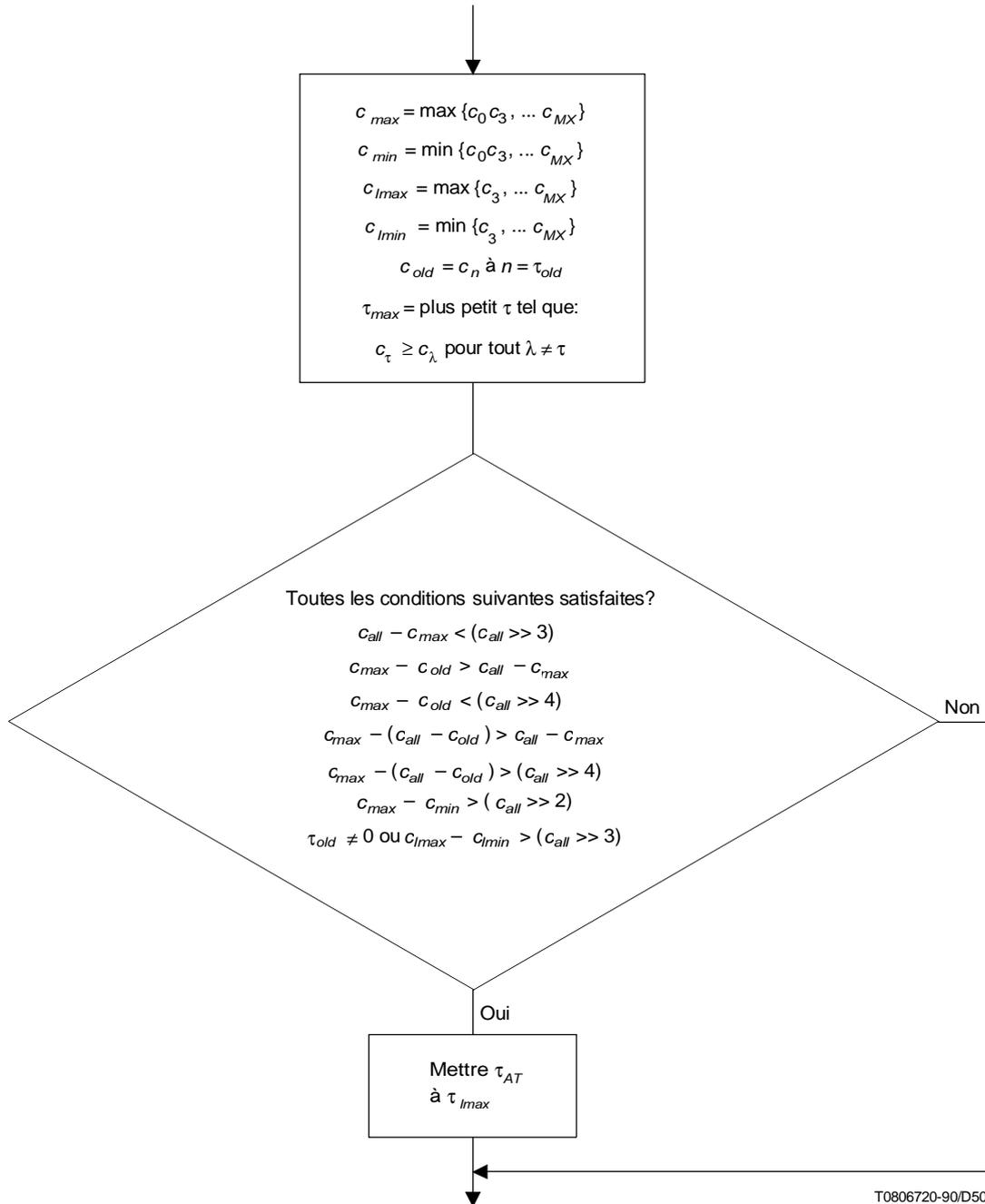


Figure C.2 – Organigramme de la procédure CHECK

Annexe D

Conception de la table d'estimation de probabilité

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

D.1 Estimation bayésienne

Soit x_0, x_1, \dots une séquence de variables aléatoires binaires indépendantes à distribution identique, prenant les valeurs 1 et 0 avec les probabilités p_1 et $p_0 = (1 - p_1)$. Soit $n_1(k)$ le nombre de un de la séquence x_0, x_1, \dots, x_{k-1} et $n_0(k)$ le nombre de zéros de la même séquence. (On a donc $n_0(k) + n_1(k) = k$.) Si p_1 est lui-même une variable aléatoire de distribution uniforme sur $[0, 1]$, l'estimation bayésienne $\hat{p}_1(k)$ de p_1 pour une observation donnée x_0, x_1, \dots, x_{k-1} , est donnée par:

$$\hat{p}_1(k) = \frac{n_1(k) + 1}{n_1(k) + 1 + n_0(k) + 1} \quad (\text{D-1})$$

L'estimation

$$\hat{p}_1(k) = \frac{n_1(k) + \delta}{n_1(k) + \delta + n_0(k) + \delta} \quad (\text{D-2})$$

avec $\delta \in (0,1)$, est également une estimation bayésienne. Toutefois, la distribution a priori particulière pour laquelle elle est bayésienne est celle qui donne une probabilité de valeurs de p_1 voisines de 0 et 1, plutôt que de 0,5. Plus la valeur de δ est petite, plus cette probabilité est biaisée vers 0 et 1.

D.2 Environnements multicontextes

Le codeur arithmétique de la présente Spécification fonctionne dans un environnement à plusieurs contextes. Lors du codage de $\text{PIX}(k)$, un contexte $\text{CX}(k)$ est fourni. Une estimation bayésienne de la probabilité $p_1(k)$ pour que $\text{PIX}(k)$ soit égal à 1 est donnée par:

$$\hat{p}_1(k) = \frac{n_{1,\text{CX}}(k) + \delta}{n_{1,\text{CX}}(k) + \delta + n_{0,\text{CX}}(k) + \delta} \quad (\text{D-3})$$

où δ est encore un paramètre libre dans $(0,1)$, $n_{1,\text{CX}}(k)$ indique combien de fois $\text{PIX}(i)$, avec $i \in [0, k-1]$, a pris la valeur 1 dans le contexte $\text{CX}(k)$, et $n_{0,\text{CX}}(k)$ est défini de façon similaire pour la valeur 0.

Pour faciliter la présentation de la suite de cette annexe, un environnement à contexte unique est supposé, mais tous les concepts et formules développés sont généralisables de façon triviale à des environnements à plusieurs contextes, de la même façon que l'équation (D-3) généralise l'équation (D-2).

D.3 Paramétrage MPS/LPS

Il est pratique de reparamétriser l'équation (D-2). On met $\text{MPS}(k)$ à 1 ou 0, selon le symbole majoritaire dans la séquence x_0, x_1, \dots, x_{k-1} . (Si le nombre de zéros et de un est le même, $\text{MPS}(k)$ peut être défini comme étant à 0 ou 1.) Alors:

$$\hat{p}_1(k) = \begin{cases} \hat{p}_{LPS}(k), & \text{si } \text{MPS}(k) = 0 \\ 1 - \hat{p}_{LPS}(k), & \text{si } \text{MPS}(k) = 1 \end{cases} \quad (\text{D-4})$$

où

$$\hat{p}_{LPS}(k) = \frac{n_{LPS}(k) + \delta}{n_{MPS}(k) + \delta + n_{LPS}(k) + \delta} \quad (\text{D-5})$$

$n_{LPS}(k)$ et $n_{MPS}(k)$ étant respectivement le nombre d'occurrences du symbole le moins probable et le nombre d'occurrences du symbole le plus probable dans x_0, x_1, \dots, x_{k-1} .

Les itérations de $n_{LPS}(k)$, de $n_{MPS}(k)$ et de $MPS(k)$ sont données par:

$$n_{LPS}(k+1) = \begin{cases} n_{LPS}(k) + 1, & \text{si } x_k \neq MPS(k) \text{ et } n_{LPS}(k) \neq n_{MPS}(k) \\ n_{LPS}(k), & \text{autrement} \end{cases} \quad (D-6)$$

$$n_{MPS}(k+1) = \begin{cases} n_{MPS}(k) + 1, & \text{si } x_k = MPS(k) \text{ et } n_{LPS}(k) = n_{MPS}(k) \\ n_{MPS}(k), & \text{autrement} \end{cases} \quad (D-7)$$

et

$$MPS(k+1) = \begin{cases} 1 - MPS(k), & \text{si } x_k \neq MPS(k) \text{ et } n_{LPS}(k) = n_{MPS}(k) \\ MPS(k), & \text{autrement} \end{cases} \quad (D-8)$$

D.4 Réponse rapide

L'estimation bayésienne de p_{LPS} donnée par (D-5) est excellente, s'il est supposé dans son calcul que la séquence x_0, x_1, \dots est stationnaire. Toutefois, dans l'application du codage arithmétique de la présente Spécification, l'entrée peut ne pas être stationnaire et changer de nature statistique lors du codage de différentes parties de l'image. Pour une bonne efficacité de codage, il est important que l'estimateur de probabilité suive l'évolution de la structure statistique de l'entrée. En environnement non stationnaire, l'inconvénient de l'estimation (D-5) de p_{LPS} est qu'elle tend à «coller». Quand n_{LPS} et n_{MPS} ont atteint des valeurs importantes, il faut beaucoup d'observations contraires pour changer de façon appréciable p_{LPS} .

Bien sûr, on souhaite en même temps une estimation précise (sans bruit) en environnement stationnaire. Un excellent compromis entre une réponse rapide et une estimation précise en environnement stationnaire est obtenu par troncature de n_{LPS} . Lorsque l'itération (D-6), (D-7), (D-8) conduit n_{LPS} à dépasser un certain seuil, une réduction d'échelle proportionnelle est opérée sur n_{LPS} et n_{MPS} . Etant proportionnelle, cette réduction d'échelle n'affecte pas p_{LPS} . Elle maintient toutefois n_{LPS} et n_{MPS} à des valeurs suffisamment petites pour avoir une réponse rapide à une évolution de la structure statistique. Le positionnement du seuil de déclenchement de la réduction d'échelle proportionnelle, permet de déterminer un compromis entre réponse rapide et précision d'estimation. Des seuils bas favorisent une réponse rapide et les seuils élevés la précision d'estimation.

D.5 Réduction de la charge de calcul

Le Tableau 24 définit un estimateur de probabilité. Cet estimateur «imite» la version tronquée de l'itération (D-6), (D-7), (D-8) décrite ci-dessus. A noter que cette imitation est opérée en sorte de réduire au minimum la charge de calcul.

Les Figures D.1A et D.1B représentent le même diagramme, mais avec des résolutions verticales différentes d'un facteur de 1000. Ces diagrammes représentent graphiquement, dans l'espace n_{LPS}, n_{MPS} , les données représentées dans le Tableau 24. Le tracé comprend 113 carrés pleins, dont chacun correspond à un des états du Tableau 24. A chacun de ces états une estimation de probabilité \hat{P}_{LPS} est associée par l'équation (D-5). La valeur de δ a été choisie par optimisation expérimentale à 0,45. Pour faciliter la représentation graphique, le point $(-\delta, -\delta)$ est représenté par un cercle sur les Figures D.1A et D.1B. Tous les traits rayonnant à partir de ce point sont des droites d'égalité de probabilité.

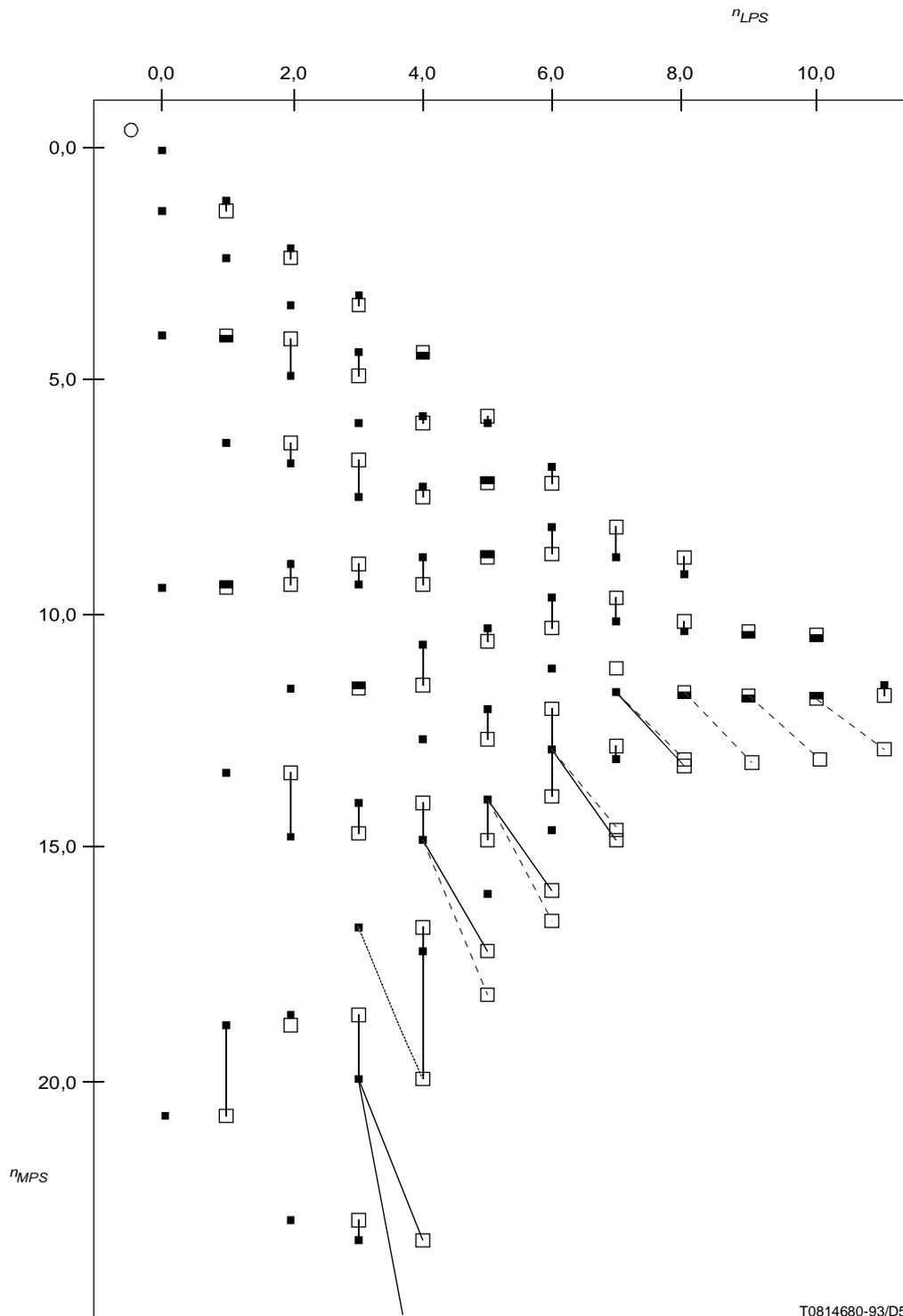
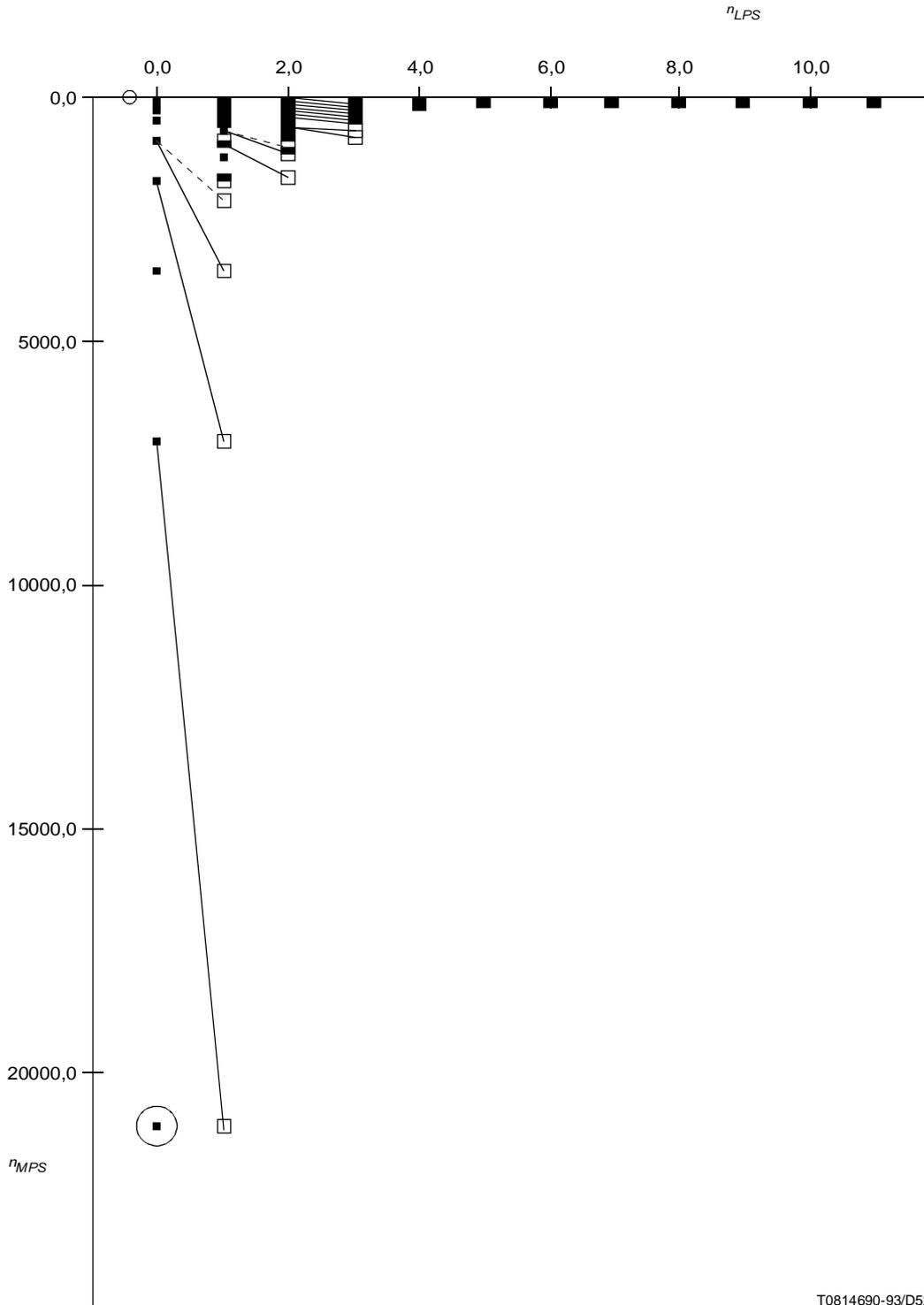


Figure D.1A – Etats de l'estimateur de probabilité dans l'espace $n_{LPS} - n_{MPS}$.
Valeur maximale de n_{MPS} représentée = 25



T0814690-93/D52

Figure D.1B – Etats de l'estimateur de probabilité dans l'espace $n_{LPS} - n_{MPS}$.
Valeur maximale de n_{MPS} représentée = 25 000

La colonne LSZ du Tableau 24 est obtenue en première approximation en utilisant l'équation (8) (voir 6.8.1.2). En bonne approximation, la densité de probabilité de l'intervalle de codage A est inversement proportionnelle à A , en sorte que \bar{A} de l'équation (8) (voir 6.8.1.2) vaut:

$$0,721 \times 0x10000 = 0xb893 \quad (D-9)$$

Les entrées effectives du Tableau 24 diffèrent de $p_{LPS} \times 0xb893$ de quelques pour-cent, car une optimisation expérimentale complémentaire a été effectuée.

A la réception du MPS, n_{MPS} devrait théoriquement être augmenté d'une unité, comme dans l'équation (D-7). Mais cette méthode conduirait à un nombre d'états déraisonnablement grand. Le mouvement descendant du diagramme (D-1) est donc conditionné, comme s'il y avait renormalisation. Soit:

$$a = A / 0x10000 \quad (D-10)$$

indiquant la longueur normalisée du registre A . La probabilité P_{RN} d'une renormalisation déterminée par le MPS est donnée par:

$$\begin{aligned} P_{RN} &= P_R \{(1-p)a < (1/2)\} \\ &= P_R \{a < (1/2)/(1-p)\} \\ &= \log_2(1/(1-p)) \end{aligned} \quad (D-11)$$

où la variable aléatoire a a encore été supposée avoir une densité sur $[1/2,1)$ inversement proportionnelle à la valeur de a . Dans n'importe quelle colonne verticale, la distance entre deux états est égale à $1/P_{RN}$ où P_{RN} est donné par (D-11), avec $p = p_{LPS}$ pour le plus élevé des deux états. Cet incrément important compense exactement le fait que n_{MPS} n'est modifié qu'avec la probabilité P_{RN} .

Lorsque l'état courant est en bas de sa colonne, il est bien sûr impossible de descendre plus bas d'une hauteur $1/P_{RN}$. Sur le diagramme (D-1), un carré vide, situé immédiatement en dessous de ces états, marque les points où l'actualisation du MPS devrait commander n_{MPS} . A noter que les traits ponctués convergeant tous vers le cercle centré sur $(-\delta, -\delta)$, le positionnement change aussi peu que possible p_{LPS} .

Lorsque la valeur du LPS est reçue, une normalisation est toujours effectuée, et le déplacement nominal dans l'espace $n_{LPS} - n_{MPS}$ s'effectue vers le point $(n_{LPS} + 1, n_{MPS})$. Ces points sont également représentés sous forme de carré vide dans le diagramme (D-1). Dans tous les cas, ces carrés vides ne coïncident avec aucun état disponible et doivent être repositionnés. Lorsque l'actualisation nominale est «interne» à la zone de l'espace $n_{LPS} - n_{MPS}$ couvert par les états disponibles, le repositionnement est effectué par un déplacement vertical vers l'état disponible le plus voisin (au sens p_{LPS}). Ces repositionnements sont représentés par des traits continus. Lorsque l'actualisation nominale est «externe», une troncature telle que décrite en D.4 est souhaitable, et le repositionnement se fait vers un état disponible, dans la colonne décalée d'une unité à gauche du point nominal. Là encore, le point particulier choisi dans cette colonne est le plus voisin du point nominal au sens p_{LPS} . Des repositionnements externes tels que celui-ci sont également représentés par des traits continus sur le diagramme (D-1).

Il existe un état qui se repositionne sur lui-même à réception d'un MPS. Il est représenté par un cercle entourant un carré plein, et figure uniquement sur le diagramme à grande échelle de la Figure D.1B. Cet état a la plus petite des probabilités associée à tous les états (environ 0,00002) et il n'existe pas de meilleur point vers lequel la transition puisse se faire à réception du MPS.

Le seuil de troncature effectif est une fonction faible de n_{MPS} et donc de p_{LPS} . Les valeurs de p_{LPS} voisines de 1/2 sont estimées avec moins de bruit, et donc une réponse moins rapide que les valeurs proches de zéro. Ce comportement a été trouvé souhaitable lors de la conception d'un codeur arithmétique commun à la présente Spécification et à la Rec. T.81 du CCITT | ISO/CEI 10918-1. Pour les applications de la présente Spécification, une réponse rapide est relativement plus importante, alors que pour la Rec. T.81 du CCITT | ISO/CEI 10918-1, la qualité d'estimation est relativement plus importante. Fait avantageux, les valeurs de p_{LPS} rencontrées dans l'application de la présente Spécification sont beaucoup plus petites que celles généralement rencontrées dans l'application de la Rec. T.81 du CCITT | ISO/CEI 10918-1. Mettre le seuil de troncature légèrement en dessous des petites valeurs de p_{LPS} procure automatiquement le comportement désiré lors de l'application de ces deux Spécifications.

Annexe E

Brevets

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

E.1 Remarques d'introduction

L'attention de l'utilisateur est attirée sur le fait que la conformité à la présente Spécification peut nécessiter l'utilisation d'une invention couverte par des brevets.

La publication de la présente Spécification n'implique aucune prise de position quant à la validité dudit titre de propriété industrielle ou de tout autre brevet ayant rapport avec celui-ci. Toutefois, pour chacun des brevets mentionnés dans la présente annexe, le titulaire du brevet a déposé auprès du Comité responsable des technologies de l'information (ITTF) (*information technology task force*) de l'ISO/CEI et du Bureau de la normalisation des télécommunications (TSB), une déclaration aux termes de laquelle il est disposé à concéder une licence de ses droits selon des termes et conditions raisonnables et non discriminatoires à tous ceux qui exprimeront le désir d'obtenir une telle licence.

Les critères d'insertion de brevet dans la présente annexe sont:

- 1) Le brevet a été identifié par un expert familiarisé avec les domaines techniques concernés par la présente Spécification, qui considère que l'utilisation de l'invention couverte par le brevet est requise pour la mise en œuvre d'un ou plusieurs des procédés de codage spécifiés.
- 2) Le titulaire du brevet a écrit une lettre à l'ITTF et au TSB déclarant qu'il est disposé à concéder une licence à un nombre illimité de postulants, du monde entier, selon des termes et conditions raisonnables, manifestement exempts de toute discrimination non équitable.

Durant la maintenance de la présente Spécification, la liste sera mise à jour, le cas échéant, lors de la publication de chaque révision de la Recommandation | Norme internationale.

E.2 Liste des brevets

Seuls les brevets déposés dans le pays des entreprises titulaires de brevets sont indiqués. Dans beaucoup de cas, des dépôts ont été effectués à l'étranger.

- 1) IBM, *A method and means for pipeline decoding of the high to low order pairwise combined digits of a decodable set of relatively shifted finite number of strings* (Méthode et moyens de décodage en pipeline de chiffres combinés par paires, d'ordre élevé à bas, d'un ensemble décodable de chaînes en nombre fini à décalage relatif), US 4 295 125, 13 octobre 1981.
- 2) IBM, *A method and means for carry-over control in a high order to low order combining of digits of a decodable set of relatively shifted finite number strings* (Méthode et moyens de traitement de la retenue dans des combinaisons de chiffres d'ordre élevé à faible, d'un ensemble décodable de chaînes de nombre finies à décalage relatif), US 4 463 342, 31 juillet 1984.
- 3) IBM, *High-speed arithmetic compression using concurrent value updating* (Compression arithmétique à haute vitesse avec mise à jour des valeurs simultanées) US 4 467 317, 21 août 1984.
- 4) IBM, *Method and means for arithmetic coding using a reduced number of operations* (Méthode et moyens de codage utilisant un nombre réduit d'opérations), US 4 286 256, 25 août 1981.
- 5) IBM, *A multiplication-free multi-alphabet arithmetic code* (Code arithmétique multi-alphabet sans multiplication), US 4 652 856, 4 février 1986.
- 6) IBM, *Symmetrical adaptive data compression/decompression system* (Système de compression/décompression de données adaptatif symétrique), US 4 633 490, 30 décembre 1986.
- 7) IBM, *Arithmetic coding data compression/decompression by selectively employed, diverse arithmetic encoders and decoders* (Compression/décompression de données à codage arithmétique, par utilisation sélective de divers codeurs et décodeurs arithmétiques), US 4 891 643, 2 janvier 1990.
- 8) IBM, *System for compression bi-level data* (Système de compression de données en deux tons), US 4 901 363, 13 février 1990.
- 9) IBM, *Arithmetic coding encoder and decoder system* (Système de codeur et de décodeur pour codage arithmétique), US 4 905 297, 27 février 1990.
- 10) IBM, *Probability adaptation for arithmetic coders* (Adaptation de probabilité pour codeurs arithmétiques), US 4 935 882, 19 juin 1990.

- 11) IBM, *Probability adaptation for arithmetic coders* (Estimation de probabilité basée sur un historique des décisions), demande déposée aux Etats-Unis, US 5 099 440, 24 mars 1992.
- 12) IBM, *Method and apparatus for processing pel signals of an image* (Méthode et équipement de traitement des signaux de pixels d'une image), US 4 982 292, 1^{er} janvier 1991.
- 13) AT&T, *Progressive transmission of high resolution two-tone facsimile images* (Transmission progressive d'images de télécopie en deux tons à haute résolution), US 4 870 497, 26 septembre 1989.
- 14) AT&T, *Edge decomposition for the transmission of high resolution facsimile images* (Décomposition des bords pour la transmission d'images de télécopie à haute résolution), US 4 873 577, 10 octobre 1989.
- 15) AT&T, *Adaptive probability estimator for entropy encoder/decoder* (Estimateur de probabilité adaptatif pour codeur/décodeur entropique), US 5 025 258, 18 juin 1991.
- 16) AT&T, *Efficient encoding/decoding in the decomposition and recomposition of a high resolution image utilizing its low resolution replica* (Codage/décodage efficace dans la décomposition et la recomposition d'une image à haute résolution utilisant sa réplique à basse résolution), US 4 979 049, 18 décembre 1990.
- 17) AT&T, *Efficient encoding/decoding in the decomposition and recomposition of a high resolution image utilizing pixel clusters* (Codage/décodage efficace dans la décomposition et la recomposition d'une image à haute résolution utilisant des groupes de pixels), US 5 031 053, 9 juillet 1991.
- 18) AT&T, *Entropy encoder/decoder including a context extractor* (Codeur/décodeur entropique incorporant un extracteur de contexte), US 5 023 611, 11 juin 1991.
- 19) AT&T, *Method and apparatus for carry-over control in arithmetic entropy coding* (Méthode et équipements de traitement de retenue en codage arithmétique entropique), US 4 973 961, 27 novembre 1990.
- 20) KDD, *Methods for reduced-sized images* (Méthodes pour des images de taille réduite), Japan Application N° 63-212 432, demande de brevet déposée au Japon.
- 21) KDD, *Image reduction system* (Système de réduction d'image), Japan Application N° 1-167 033, conjointement avec Canon, demande de brevet déposée au Japon.
- 22) Mitsubishi, *Facsimile encoding communication system* (Système de communication par codage de télécopie), Japan 1 251 403, 6 juillet 1984.
- 23) Mitsubishi, *Encoding method* (Méthode de codage), demande de brevet déposée au Japon.
- 24) Canon, *Image reduction system* (Système de réduction d'image), Japan Application N° 1-167 033, conjointement avec KDD, demande de brevet déposée au Japon.

E.3 Adresses à contacter pour des renseignements sur les brevets

Directeur du Bureau de la normalisation des télécommunications (ancien CCITT)

Union Internationale des Télécommunications

Place des Nations

CH-1211 Genève 20

Suisse

Tél: +41 (22) 730 5111

Fax: +41 (22) 730 5853

ITTF

Organisation Internationale de Normalisation

1, rue de Varembe

CH-1211 Genève 20

Suisse

Tél: +41 (22) 734 0150

Fax: +41 (22) 733 3843

Program Manager, Licensing
Intellectual Property and Licensing Services
IBM Corporation
208 Harbor Drive
P.O. Box 10501
Stamford, Connecticut 08904-2501
Tél: +1 (203) 973 7935
Fax: +1 (203) 973 7981 ou +1 (203) 973 7982

Mitsubishi Electric Corp.
Intellectual Property License Department
1-2-3 Morunouchi, Chiyoda-ku
Tokyo 100, Japan
Tél: +81 (3) 3218 3465
Fax: +81 (3) 3215 3842

International Affairs Department
Kokusai Denshin Denwa Co. Ltd.
3-2 Nishishinjuku 2-chome
Shinjuku-ku
Tokyo 163, Japan
Tél: +81 (3) 3347 6457
Tél: +81 (3) 3347 6470

AT&T Intellectual Property Division Manager
Room 3A21
10 Independence Blvd.
Warren, NJ 07059, USA
Tél: +1 (908) 580 5392
Fax: +1 (908) 580 6355

Senior General Manager,
Corporate Intellectual Property and Legal Headquarters
Canon Inc.,
30-2 Shimomaruko 3-chome
Ohta-ku, Tokyo 146, Japan
Tél: +81 (3) 3758 2111
Fax: +81 (3) 3756 0947

Annexe F

Bibliographie

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

ISO/CEI 9281-1, *Technologie de l'information – Méthodes de codage des images – Partie 1: Identification.*

Recommandation T.81 du CCITT (1992) | ISO/CEI 10918-1:1993, *Technologie de l'information – Compression numérique et codage des images fixes de nature photographique – Spécifications et directives.*

Recommandation T.4 du CCITT, *Normalisation des télécopieurs du groupe 3 pour la transmission de documents.*

Recommandation T.6 du CCITT, *Schémas de codage et fonctions de commande de codage de la télécopie pour les télécopieurs du groupe 4.*

WITTEN (I.H.), NEAL (R.M.) et CLEARY (J.G.): Arithmetic coding for data compression (Codage arithmétique pour compression de données), *Communications of the ACM*, vol. 30, n° 6, p. 520-540, juin 1987.

ARPS (R.B.), TRUONG (T.J.), LU (D.J.), PASCO (R.C.) et FRIEDMAN (T.D.): A multi-purpose VLSI chip for adaptative data compression of bilevel images (Puces VLSI multi-usages pour compression de données adaptatives d'images en deux tons), *IBM Journal of Research and Development*, vol. 32, n° 6, p. 775-795, novembre 1988.

PENNEBAKER (W.B.), MITCHELL (J.L.), LANGDON (G.G.), Jr. et ARPS (R.B.): An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder (Présentation générale des principes de base du codeur arithmétique binaire adaptatif Q-Coder), *IBM Journal of Research and Development*, vol.32, n° 6, p. 717-726, novembre 1988.

ONO (F.), KINO (S.), YOSHIDA (M.) et KIMURA (T.): Bi-level image coding with Melcode – Comparison of block type code and arithmetic type code (Codage d'image en deux tons avec Melcode – Comparaison du code de type bloc et du code de type arithmétique), *IEEE Global Telecommunications Conference*, p. 255-260, novembre 1989.

DUTTWEILER (D.) et CHAMZAS (C.): Probability estimation in arithmetic and adaptive-Huffman entropy coders (Estimation de probabilité dans les codeurs arithmétiques), à paraître dans *IEEE Trans. on Image processing*.

SHEINWALD (D.) et PASCO (R.): Deriving deterministic prediction rules from reduction schemes (Détermination de règles de prédiction déterministe d'algorithmes de réduction), à paraître dans *IEEE Trans. Information Theory*.

LANGDON (G.): Method for carry-over control in a FIFO arithmetic code string («Méthode de traitement de la retenue dans une chaîne de code arithmétique FIFO»), *IBM Technical Disclosure Bulletin*, vol. 23, n° 1, p 310-312, juin 1980.

JONES (C.): An efficient coding system for long source sequences (Système de codage efficace pour des séquences source longues), *IEEE Trans. Information Theory*, vol. IT-27, p. 280-291, mai 1981.

LANGDON (G.): An introduction to arithmetic coding (Introduction au codage arithmétique), *IBM Journal of Research and Development*, vol. 28, p. 135-149, mars 1984.