



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.262

Amendement 1
(11/2000)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructure des services audiovisuels – Codage des
images vidéo animées

Technologies de l'information – Codage générique
des images animées et du son associé: données
vidéo

**Amendement 1: Données de description du
contenu du flux vidéo élémentaire**

Recommandation UIT-T H.262 – Amendement 1

(Antérieurement Recommandation du CCITT)

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

CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
SYSTÈMES ET ÉQUIPEMENTS TERMINAUX POUR LES SERVICES AUDIOVISUELS	H.300–H.399
SERVICES COMPLÉMENTAIRES EN MULTIMÉDIA	H.450–H.499

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

**NORME INTERNATIONALE ISO/CEI 13818-2
RECOMMANDATION UIT-T H.262**

**TECHNOLOGIES DE L'INFORMATION – CODAGE GÉNÉRIQUE DES IMAGES
ANIMÉES ET DU SON ASSOCIÉ: DONNÉES VIDÉO**

AMENDEMENT 1

Données de description du contenu du flux vidéo élémentaire

Résumé

Cet amendement offre la possibilité d'envoyer des "données de description de contenu" complémentaires dans des flux vidéo élémentaires. Les données de description de contenu qui peuvent être transportées comprennent des informations de rythme de capture d'image, des paramètres additionnels de panoramique, une indication de zone visuelle active dans l'image vidéo et une représentation codée de la taille de l'image en octets.

Source

L'Amendement 1 de la Recommandation H.262 de l'UIT-T, élaboré par la Commission d'études 16 (2001-2004) de l'UIT-T, a été approuvé le 17 novembre 2000. Un texte identique est publié comme Norme Internationale ISO/CEI 13818-2, Amendement 1.

AVANT-PROPOS

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

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

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2001

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

TABLE DES MATIÈRES

		<i>Page</i>
1)	Paragraphe 6.2.3	1
2)	Nouveau § 6.2.3.7.3	2
3)	Nouveau § 6.2.3.7.3.1	2
4)	Nouveau § 6.2.3.7.3.2	3
5)	Nouveau § 6.2.3.7.3.2.1	4
6)	Nouveau § 6.2.3.7.3.3	5
7)	Nouveau § 6.2.3.7.3.4	6
8)	Nouveau § 6.2.3.7.3.5	6
9)	Paragraphe 6.3.9	7
10)	Nouveau § 6.3.21	7
11)	Nouveau § 6.3.21.1	8
12)	Nouveau § 6.3.21.2	8
13)	Nouveau § 6.3.21.2.1	9
14)	Nouveau § 6.3.21.3	11
15)	Nouveau § 6.3.21.4	12
16)	Nouveau § 6.3.21.5	13
17)	Paragraphe E.1	14
18)	Nouvelle Annexe K	15
	K.1 Codage à balayage progressif et non progressif	15
	K.2 Syntaxe d'information relative à la fréquence d'échantillonnage de la source vidéo	15
	K.3 Méthodes de production de contenu	16
	K.4 Montage post-codage du fanion de bitrame à balayage progressif dans des flux binaires de données vidéo	18
	K.5 Post-traitement des systèmes dotés d'un affichage à balayage progressif	18
	K.6 Utilisation des informations du code temporel d'acquisition	19

**NORME INTERNATIONALE
RECOMMANDATION UIT-T**

**TECHNOLOGIES DE L'INFORMATION – CODAGE GÉNÉRIQUE DES IMAGES
ANIMÉES ET DU SON ASSOCIÉ: DONNÉES VIDÉO**

AMENDEMENT 1

Données de description du contenu du flux vidéo élémentaire

1) Paragraphe 6.2.3

Remplacer le § 6.2.3 par:

6.2.3 En-tête d'image

picture_header() {	Nombre de bits	Mnémonique
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if (picture_coding_type == 2 picture_coding_type == 3) {		
full_pel_forward_vector	1	bslbf
forward_f_code	3	bslbf
}		
if (picture_coding_type == 3) {		
full_pel_backward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while (nextbits() == '1') {		
extra_bit_picture /* with the value '1' */	1	uimsbf
content_description_data() /* with every 9 th bit having the value '1' */		
}		
extra_bit_picture /* with the value '0' */	1	uimsbf
next_start_code()		
}		

2) Nouveau § 6.2.3.7.3

Ajouter le nouveau § 6.2.3.7.3 suivant:

6.2.3.7.3 Données de description de contenu

content_description_data() {	Nombre de bits	Mnémonique
data_type_upper	8	uimsbf
marker_bit	1	bslbf
data_type_lower	8	
marker_bit	1	bslbf
data_length	8	uimsbf
if (data_type == "Padding Bytes")		
padding_bytes()		
else if (data_type == "Capture Timecode")		
capture_timecode()		
else if (data_type == "Additional Pan-Scan Parameters")		
additional_pan_scan_parameters()		
else if (data_type == "Active Region Window")		
active_region_window()		
else if (data_type == "Coded Picture Length")		
coded_picture_length()		
else		
for (i = 0; i < data_length; i ++) {		
marker_bit	1	bslbf
reserved_content_description_data	8	uimsbf
}		
}		

3) Nouveau § 6.2.3.7.3.1

Ajouter le nouveau § 6.2.3.7.3.1 suivant:

6.2.3.7.3.1 Octets de paquet de bourrage

padding_bytes() {	Nombre de bits	Mnémonique
for (i = 0; i < data_length; i ++) {		
marker_bit	1	bslbf
padding_byte	8	bslbf
}		
}		

4) **Nouveau § 6.2.3.7.3.2**

Ajouter le nouveau § 6.2.3.7.3.2 suivant:

6.2.3.7.3.2 Code temporel d'acquisition

capture_timecode() {	Nombre de bits	Mnémonique
marker_bit	1	bslbf
timecode_type	2	uimsbf
counting_type	3	uimsbf
reserved_bit	1	uimsbf
reserved_bit	1	uimsbf
reserved_bit	1	uimsbf
if (counting_type != 0) {		
marker_bit	1	bslbf
nframes_conversion_code	1	uimsbf
clock_divisor	7	uimsbf
marker_bit	1	bslbf
nframes_multiplier_upper	8	uimsbf
marker_bit	1	bslbf
nframes_multiplier_lower	8	
}		
frame_or_field_capture_timestamp()		
if (timecode_type == '11')		
frame_or_field_capture_timestamp()		
}		

5) Nouveau § 6.2.3.7.3.2.1

Ajouter le nouveau § 6.2.3.7.3.2.1 suivant:

6.2.3.7.3.2.1 Marqueur temporel d'acquisition de bitrame ou trame

frame_or_field_capture_timestamp() {	Nombre de bits	Mnémonique
if (counting_type != 0) {		
marker_bit	1	bslbf
nframes	8	uimsbf
}		
marker_bit	1	bslbf
time_discontinuity	1	uimsbf
prior_count_dropped	1	uimsbf
time_offset_part_a	6	simsbf
marker_bit	1	bslbf
time_offset_part_b	8	
marker_bit	1	bslbf
time_offset_part_c	8	
marker_bit	1	bslbf
time_offset_part_d	8	
marker_bit	1	bslbf
units_of_seconds	4	uimsbf
tens_of_seconds	4	uimsbf
marker_bit	1	bslbf
units_of_minutes	4	uimsbf
tens_of_minutes	4	uimsbf
marker_bit	1	bslbf
units_of_hours	4	uimsbf
tens_of_hours	4	uimsbf
}		

6) Nouveau § 6.2.3.7.3.3

Ajouter le nouveau § 6.2.3.7.3.3 suivant:

6.2.3.7.3.3 Paramètres de recadrage plein écran supplémentaires

additional_pan_scan_parameters() {	Nombre de bits	Mnémonique
marker_bit	1	bslbf
aspect_ratio_information	4	uimsbf
reserved_bit	1	bslbf
reserved_bit	1	bslbf
reserved_bit	1	bslbf
display_size_present	1	bslbf
if (display_size_present == '1') {		
marker_bit	1	bslbf
reserved_bit	1	bslbf
reserved_bit	1	bslbf
display_horizontal_size_upper	6	uimsbf
marker_bit	1	bslbf
display_horizontal_size_lower	8	
marker_bit	1	bslbf
reserved_bit	1	bslbf
reserved_bit	1	bslbf
display_vertical_size_upper	6	uimsbf
marker_bit	1	bslbf
display_vertical_size_lower	8	
}		
for (i = 0; i < number_of_frame_centre_offsets; i ++) {		
marker_bit	1	bslbf
frame_centre_horizontal_offset_upper	8	simsbf
marker_bit	1	bslbf
frame_centre_horizontal_offset_lower	8	
marker_bit	1	bslbf
frame_centre_vertical_offset_upper	8	simsbf
marker_bit	1	bslbf
frame_centre_vertical_offset_lower	8	
}		
}		

7) **Nouveau § 6.2.3.7.3.4**

Ajouter le nouveau § 6.2.3.7.3.4 suivant:

6.2.3.7.3.4 Fenêtre de région active

active_region_window() {	Nombre de bits	Mnémonique
marker_bit	1	bslbf
top_left_x_upper	8	uimsbf
marker_bit	1	bslbf
top_left_x_lower	8	
marker_bit	1	bslbf
top_left_y_upper	8	uimsbf
marker_bit	1	bslbf
top_left_y_lower	8	
marker_bit	1	bslbf
active_horizontal_size_upper	8	uimsbf
marker_bit	1	bslbf
active_horizontal_size_lower	8	
marker_bit	1	bslbf
active_vertical_size_upper	8	uimsbf
marker_bit	1	bslbf
active_vertical_size_lower	8	
}		

8) **Nouveau § 6.2.3.7.3.5**

Ajouter le nouveau § 6.2.3.7.3.5 suivant:

6.2.3.7.3.5 Longueur d'image codée

coded_picture_length() {	Nombre de bits	Mnémonique
marker_bit	1	bslbf
picture_byte_count_part_a	8	uimsbf
marker_bit	1	bslbf
picture_byte_count_part_b	8	
marker_bit	1	bslbf
picture_byte_count_part_c	8	
marker_bit	1	bslbf
picture_byte_count_part_d	8	
}		

9) Paragraphe 6.3.9

Remplacer la sémantique du § 6.3.9 en ce qui concerne les structures `extra_bit_picture` et `extra_information_picture` par la sémantique suivante (en supprimant la sémantique de la structure `extra_information_picture`):

image par bit supplémentaire (`extra_bit_picture`) – Ce fanion indique la présence ou l'absence des informations supplémentaires suivantes. S'il est mis à '1', des données de description de contenu [`content_description_data()`] le suivront. S'il est mis à '0', aucune donnée de description de contenu ne le suivra dans cet en-tête d'image.

10) Nouveau § 6.3.21

Ajouter le nouveau § 6.3.21 suivant:

6.3.21 Données de description de contenu

type de données supérieures, type de données inférieures (`data_type_upper`, `data_type_lower`) – Deux valeurs d'entier non signé à 8 bits contenant les bits de poids le plus fort et les bits de poids le plus faible, correspondant à la valeur de l'entier non signé à 16 bits `data_type` définissant le type de données de description de contenu. La sémantique de la structure `data_type` est définie dans le Tableau 6-21.

Tableau 6-21 – Valeurs de type de données (`data_type` values)

Valeur	Signification
0000 0000 0000 0000	Valeur réservée
0000 0000 0000 0001	Octets de paquet de bourrage
0000 0000 0000 0010	Code temporel d'acquisition
0000 0000 0000 0011	Paramètres de recadrage plein écran supplémentaires
0000 0000 0000 0100	Fenêtre de région active
0000 0000 0000 0101	Longueur d'image codée
0000 0000 0000 0110	Valeur réservée
...	Valeur réservée
1111 1111 1111 1111	Valeur réservée

longueur de données (`data_length`) – Entier non signé à 8 bits spécifiant la quantité restante de données devant suivre à l'intérieur du reste de la structure de données de description de contenu, exprimé en unités de 9 bits. Le nombre de bits de données qui suit à l'intérieur du reste de cette structure sera égal à `data_length * 9`.

données de description de contenu réservées (`reserved_content_description_data`) – Entier non signé à 8 bits réservés. Un décodeur rencontrant la structure `reserved_content_description_data` dans un flux binaire doit l'ignorer (c'est-à-dire la supprimer du flux binaire et la rejeter). Un flux binaire conforme à la présente Spécification ne doit pas contenir cet élément syntaxique.

Au cas où un décodeur rencontrerait un entier non signé `data_type` décrit comme "réservé" dans le Tableau 6-21, il doit rejeter les paires successives des structures `marker_bit` et `reserved_content_description_data` qui suivent la structure `data_length` dans le flux binaire. Le nombre de ces paires sera égal à la structure `data_length`. Cette exigence permet la définition ultérieure d'extensions conformes à la présente Spécification.

bit réservé (`reserved_bit`) – Entier non signé à 1 bit réservé. Doit être égal à '0' dans des flux binaires conformes à la présente spécification. La valeur '1' est réservée pour une utilisation ultérieure à compatibilité amont par l'UIT-T | ISO/CEI. Un décodeur conforme à la présente Spécification autorisera une valeur '0' ou '1' pour la structure `reserved_bit`.

11) Nouveau § 6.3.21.1

Ajouter le nouveau paragraphe 6.3.21.1 suivant:

6.3.21.1 Octets de paquet de bourrage

octet de paquet de bourrage (padding_byte) – Chaîne à 8 bits qui doit être égale à '0000 0000'. Toutes les autres valeurs sont interdites.

NOTE – Les octets de paquet de bourrage permettent l'inclusion d'un certain nombre d'octets de données pour des calculs sur la vérification de la mémoire vidéo (*VBV, video buffering verifier*).

12) Nouveau § 6.3.21.2

Ajouter le nouveau § 6.3.21.2 suivant:

6.3.21.2 Code temporel d'acquisition

Le code temporel d'acquisition décrit le temps initial d'acquisition ou de création des trames ou bitrames du contenu.

Il contient des marqueurs temporels absolus pour les bitrames ou trames associées. Un seul code temporel d'acquisition pour chaque image doit être présent dans le flux binaire. Ce code temporel ne doit pas avoir la priorité sur un code temporel quelconque spécifié en vue de la présentation ou du décodage au niveau du multiplexeur du système, par exemple les marqueurs temporels de présentation ou de décodage définis dans la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (Systèmes).

type de code temporel (timecode_type) – Entier à 2 bits indiquant le nombre de marqueurs temporels associés à cette image, tel qu'il est défini dans le Tableau 6-22. Les valeurs '00', '10' et '11' doivent uniquement être utilisées lorsque la structure d'image (*picture_structure*) est égale à "image bitrame". La valeur '00' indique que les deux trames qui composent l'image bitrame présentent le même instant d'acquisition. Lorsque le type de code temporel est '11', le premier marqueur temporel se rapporte à la première trame de l'image bitrame et le second se rapporte à la seconde trame de l'image bitrame.

Tableau 6-22 – Valeurs de type de code temporel (timecode_type values)

Valeur	Signification
00	Un marqueur temporel pour la bitrame
01	Un marqueur temporel pour la première ou l'unique trame
10	Un marqueur temporel pour la seconde trame
11	Deux marqueurs temporels, un pour chacune des deux trames

type de comptage (counting_type) – Entier à 3 bits indiquant la méthode utilisée pour compenser la valeur du paramètre de comptage de nframes des marqueurs temporels d'acquisition de trame ou d'image bitrame dans le but de corriger le décalage par rapport au temps réel dans les paramètres restants de chaque marqueur temporel.

Tableau 6-23 – Valeurs de counting_type

Valeur	Signification
000	Paramètre nframes pas utilisé
001	Pas d'abandon de valeurs de comptage nframes
010	Abandon de valeurs zéro individuelles de comptage nframes
011	Abandon de valeurs max_nframes individuelles de comptage nframes
100	Abandon des deux valeurs de comptage nframes les plus faibles (0 et 1) quand les unités de seconde et de dix secondes sont égales à zéro et que les unités de minutes sont différentes de zéro
101	Abandon de valeurs de comptage nframes individuelles non spécifiées
110	Abandon de nombres non spécifiés de valeurs de comptage nframes non spécifiées
111	Réservé

code de conversion de nframes (nframes_conversion_code) – Entier non signé de 1 bit qui indique le facteur de conversion à utiliser pour déterminer la durée indiquée par le paramètre nframes de chaque marqueur temporel d'acquisition de trame ou d'image bitrame. Le facteur spécifié est 1000 + code de conversion de nframes.

diviseur d'horloge (clock_divisor) – Entier non signé de 7 bits contenant le nombre de divisions de l'horloge de système à 27 MHz qu'il convient d'appliquer pour générer le marqueur temporel équivalent pour le marqueur temporel d'acquisition de chaque trame et de chaque image bitrame.

multiplicateur nframes fort, multiplicateur nframes faible (nframes_multiplier_upper, nframes_multiplier_lower) – Respectivement le bit de poids fort et le bit de poids faible du multiplicateur de nframes.

multiplicateur de nframes (nframes_multiplier) – Entier non signé utilisé pour générer le marqueur temporel équivalent du marqueur temporel d'acquisition de chaque trame et de chaque image bitrame, tel que spécifié par nframes_multiplier_upper et nframes_multiplier_lower.

13) Nouveau § 6.3.21.2.1

Ajouter le nouveau paragraphe 6.3.21.2.1 suivant:

6.3.21.2.1 Marqueur temporel d'acquisition de bitrame ou de trame

nframes – Entier non signé à 8 bits contenant le nombre d'incrément temporels d'image bitrame qu'il convient d'ajouter dans le calcul du marqueur temporel équivalent. La valeur de nframes ne doit pas être supérieure au nombre maximum de nframes tel qu'il est calculé au moyen de la formule suivante:

$$\text{max_nframes} = (26\ 999\ 999) / (\text{nframes_multiplier} * (1000 + \text{nframes_conversion_code}) * \text{clock_divisor})$$

où "/" est l'opérateur de division défini au § 4.1.

discontinuité temporelle (time_discontinuity) – Fanion à 1 bit indiquant si une discontinuité temporelle ou de base temporelle s'est produite entre le marqueur temporel précédent et le marqueur temporel en cours. Si ce fanion est mis à '0', la différence de temps pouvant être calculée entre les marqueurs temporels en cours et précédent représente la durée d'affichage idéale de la trame ou image bitrame précédente. S'il est mis à '1', la différence de temps entre les marqueurs temporels précédents et en cours pouvant être calculée n'a pas de sens précise. Si, lors du montage, de telles discontinuités se produisent, ou si le marqueur temporel précédent de la trame ou de l'image bitrame est indisponible, le bit de la structure time_discontinuity doit être mis à '1'.

suppression de comptage préalable (prior_count_dropped) – Fanion à 1 bit indiquant si une ou plusieurs valeurs du paramètre nframes ont été supprimées dans le but de freiner l'accumulation de dérive dans les autres paramètres du marqueur temporel. Doit être zéro si le type de comptage est '001', si le type de comptage est '010' avec nframes différent de 1, si le type de comptage est '011' avec nframes différent de 0 et si le type de comptage est '100' avec nframes différent de 2.

décalage temporel partie a (time_offset_part_a) – Entier à 6 bits contenant les bits de poids le plus fort de time_offset.

décalage temporel partie b (time_offset_part_b) – Entier non signé à 8 bits contenant les bits de poids le plus fort venant en deuxième position de time_offset.

décalage temporel partie c (time_offset_part_c) – Entier non signé à 8 bits contenant les bits de poids le plus fort venant en troisième position de time_offset.

décalage temporel partie d (time_offset_part_d) – Entier non signé à 8 bits contenant les bits de poids le plus faible de time_offset.

décalage temporel (time_offset) – Entier signé à 30 bits en format de complément à deux qui est le nombre de cycles d'horloge (de système original à 27 MHz ou avec une fréquence d'horloge modifiée par un diviseur d'horloge) décalés à partir de l'instant spécifié par les autres paramètres du marqueur temporel de trame ou d'image bitrame afin de spécifier le marqueur temporel équivalent correspondant à l'instant où la trame ou l'image bitrame en cours a été acquise. Lorsque le type de comptage est 0, la valeur de time_offset sera limitée par le codeur qui imposera des valeurs inférieures à 27 000 000 en magnitude.

unités de secondes (units_of_seconds) – Entier non signé à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en secondes modulo 10. Le Tableau 6-24 définit la gamme de valeurs autorisée.

Tableau 6-24 – Valeurs d'unités de secondes (units_of_seconds values)

Valeur	Signification
0000-1001	Nombre de secondes modulo 10
1010-1111	Valeur interdite

dizaines de secondes (tens_of_seconds) – Entier non signé à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en secondes divisées par 10. Le Tableau 6-25 définit la gamme de valeurs autorisée.

Tableau 6-25 – Valeurs de dizaines de secondes (tens_of_seconds values)

Valeur	Signification
0000-0101	Nombre de secondes/10
0110-1111	Valeur interdite

unités de minutes (units_of_minutes) – Entier à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en minutes modulo 10. Le Tableau 6-26 définit la gamme de valeurs autorisée.

Tableau 6-26 – Valeurs d'unités de minutes (units_of_minutes values)

Valeur	Signification
0000-1001	Nombre de minutes modulo 10
1010-1111	Valeur interdite

dizaines de minutes (tens_of_minutes) – Entier à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en minutes divisées par 10. Le Tableau 6-27 définit la gamme de valeurs autorisée.

Tableau 6-27 – Valeurs de dizaines de minutes (tens_of_minutes values)

Valeur	Signification
0000-0101	Nombre de minutes/10
0110-1111	Valeur interdite

unités d'heures (units_of_hours) – Entier à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en heures modulo 10. Le Tableau 6-28 définit la gamme de valeurs autorisée. Il ne dépassera pas la valeur '3' si la structure tens_of_hours est égale à '2'.

Tableau 6-28 – Valeurs d'unités d'heures (units_of_hours values)

Valeur	Signification
0000-1001	Nombre d'heures modulo 10
1010-1111	Valeur interdite

dizaines d'heures (tens_of_hours) – Entier à 4 bits utilisé pour le calcul du marqueur temporel équivalent. Il représente la partie du marqueur temporel de cette trame ou bitrame exprimée en heures divisées par 10. Le Tableau 6-29 définit la gamme de valeurs autorisée.

Tableau 6-29 – Valeurs de dizaines d'heures (tens_of_hours values)

Valeur	Signification
0000–0010	Nombre d'heures/10
0011–1111	Valeur interdite

Lorsque le type de comptage est 0, un marqueur temporel équivalent exprimé en cycles d'horloge de système à 27 MHz est défini à partir de la formule suivante:

$$\text{marqueur temporel équivalent (equivalent_timestamp)} = (60 * (60 * (\text{units_of_hours} + 10 * \text{tens_of_hours}) + (\text{units_of_minutes} + 10 * \text{tens_of_minutes})) + \text{units_of_seconds} + 10 * \text{tens_of_seconds}) * 27\,000\,000 + \text{time_offset}$$

Lorsque le type de comptage est 0, les valeurs des paramètres à l'intérieur du marqueur temporel seront contraintes par le codeur de façon que equivalent_timestamp ne soit pas inférieur à 0 et ne dépasse pas 2 332 799 999 999.

Lorsque le type de comptage est différent de 0, un marqueur temporel équivalent exprimé en cycles d'horloge de système à 27 MHz est défini à partir de la formule suivante:

$$\text{equivalent_timestamp} = (60 * (60 * (\text{units_of_hours} + 10 * \text{tens_of_hours}) + (\text{units_of_minutes} + 10 * \text{tens_of_minutes})) + \text{units_of_seconds} + 10 * \text{tens_of_seconds}) * 27\,000\,000 + (\text{nframes} * (\text{nframes_multiplier} * (1000 + \text{nframes_conversion_code})) + \text{time_offset}) * \text{clock_divisor}$$

Lorsque le type de comptage est différent de 0, les valeurs des paramètres dans le code temporel seront contraintes par le codeur de façon que equivalent_timestamp ne soit pas inférieur à 0.

Deux structures equivalent_timestamp identiques calculées sur des bitrames ou trames consécutives sans une structure time_discontinuity intermédiaire indiquent que des bitrames ou trames ont été acquises ou créées au même moment.

14) Nouveau § 6.3.21.3

Ajouter le nouveau § 6.3.21.3 suivant:

6.3.21.3 Paramètres de recadrage plein écran (pan-scan) supplémentaires

L'ajout de paramètres de recadrage plein écran permet le transport d'informations de recadrage plein écran pour plus d'un type d'affichage. Par exemple, si des informations codées en vue d'un processus de recadrage plein écran dans l'en-tête de séquence, l'extension d'affichage de séquence et l'extension d'affichage d'image sont utilisées pour la définition des paramètres nécessaires pour l'affichage au format 3:4, les paramètres de recadrage plein écran supplémentaires peuvent définir les paramètres nécessaires pour l'affichage au format 9:16.

informations liées au format (aspect_ratio_information) – Valeur d'entier à 4 bits définie au § 6.3.3 (en-tête de séquence). Cette valeur ne doit pas être la même que celle spécifiée dans la séquence d'en-tête (sequence_header()).

présence de la taille d'affichage (display_size_present) – Fanion à 1 bit qui, lorsqu'il est mis à '1', indique la présence des paramètres display_horizontal_size_upper, display_horizontal_size_lower, display_vertical_size_upper et display_vertical_size_lower. Lorsqu'il est mis à '0', les valeurs précédentes de display_horizontal_size et de display_vertical_size correspondant à la valeur de l'information de format seront utilisées. Pour un format donné, cette trame devrait être mise à '1' dans le premier en-tête d'image après une séquence d'en-tête (sequence_header()) quelconque. A la suite d'un en-tête de séquence, la valeur des paramètres display_horizontal_size et display_vertical_size sera celle qui est définie dans l'extension d'affichage de séquence (sequence_display_extension()).

dimension horizontale d'affichage supérieure (display_horizontal_size_upper) – Les 6 bits de poids le plus fort du paramètre display_horizontal_size.

dimension horizontale d'affichage inférieure (display_horizontal_size_lower) – Les 8 bits de poids le plus faible du paramètre display_horizontal_size.

dimension horizontale d'affichage (display_horizontal_size) – Valeur d'entier à 14 bits définie au § 6.3.6 (extension d'affichage de séquence). Pour une valeur donnée d'information de format, la valeur de ce paramètre restera la même pour la séquence.

dimension verticale d'affichage supérieure (display_vertical_size_upper) – Les 6 bits de poids le plus fort du paramètre display_vertical_size.

dimension verticale d'affichage inférieure (display_vertical_size_lower) – Les 8 bits de poids le plus faible du paramètre display_vertical_size.

dimension verticale d'affichage (display_vertical_size) – Valeur d'entier à 14 bits définie au § 6.3.6 (extension d'affichage de séquence). Pour une valeur donnée d'information de format, la valeur de ce paramètre restera la même pour la séquence.

décalage horizontal du centre de l'image supérieur, décalage horizontal du centre de l'image inférieur (frame_centre_horizontal_offset_upper, frame_centre_horizontal_offset_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre frame_centre_horizontal_offset.

décalage horizontal du centre de l'image (frame_centre_horizontal_offset) – Entier signé à 16 bits défini au § 6.3.12 (extension d'affichage de l'image).

décalage vertical du centre de l'image supérieur, décalage vertical du centre de l'image inférieur (frame_centre_vertical_offset_upper, frame_centre_vertical_offset_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre frame_centre_vertical_offset.

décalage vertical du centre de l'image (frame_centre_vertical_offset) – Entier signé à 16 bits défini au § 6.3.12 (extension d'affichage de l'image). A la suite d'un en-tête de séquence (sequence_header), la valeur zéro sera utilisée pour tous les décalages du centre de l'image jusqu'à ce qu'une extension d'affichage de l'image (picture_display_extension()) définisse des valeurs autres que zéro.

nombre de décalages du centre de l'image (number_of_frame_centre_offsets) – Entier défini au § 6.3.12. A la suite d'un en-tête de séquence, la valeur zéro sera utilisée pour tous les décalages du centre de l'image jusqu'à ce qu'une extension d'affichage de l'image définisse des valeurs autres que zéro.

15) Nouveau § 6.3.21.4

Ajouter le nouveau § 6.3.21.4 suivant:

6.3.21.4 Fenêtre de région active

La fenêtre de région active contient des entiers définissant le rectangle de l'image reconstruite devant être affichée. Cette fenêtre ne doit pas être plus grande que le rectangle défini par les structures horizontal_size et vertical_size définies au § 6.3.3. Pas plus d'une fenêtre de région active (active_region_window) sera présente dans le flux binaire pour chaque image. Lorsqu'une image est codée en tant qu'image à deux trames, la fenêtre de région active ne sera pas présente dans la seconde trame.

abscisse du coin gauche supérieur, abscisse du coin gauche inférieur (top_left_x_upper, top_left_x_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre top_left_x.

abscisse du coin gauche supérieur (top_left_x) – Entier à 16 bits définissant le nombre d'échantillons à l'intérieur d'une ligne de la composante de luminance dans l'image reconstruite, et qui, conjointement avec le paramètre top_left_y, spécifie le coin gauche supérieur du rectangle de la fenêtre de région active (active_region_window).

ordonnée du coin gauche supérieur, ordonnée du coin gauche inférieur (top_left_y_upper, top_left_y_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre top_left_y.

ordonnée du coin gauche supérieur (top_left_y) – Entier à 16 bits définissant le nombre de lignes de la composante de luminance dans l'image reconstruite, et qui, conjointement avec le paramètre top_left_x, spécifie le coin gauche supérieur du rectangle de la fenêtre de région active (active_region_window).

dimension horizontale de région active supérieure, dimension horizontale de région active inférieure (active_region_horizontal_size_upper, active_region_horizontal_size_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre active_region_horizontal_size.

dimension horizontale de région active (active_region_horizontal_size) – Entier à 16 bits qui, conjointement avec la structure active_region_vertical_size, définit un rectangle à l'intérieur de la composante de luminance, qui peut être considéré comme la région active. Si ce rectangle est plus petit que l'image bitrame codée, le processus d'affichage ne devrait afficher qu'une partie de cette image. Cette valeur ne sera pas plus grande que la dimension horizontale (horizontal_size) de l'image bitrame codée. La valeur '0' indique que la dimension est inconnue.

dimension verticale de région active supérieure, dimension verticale de région active inférieure (active_region_vertical_size_upper, active_region_vertical_size_lower) – Les 8 bits de poids le plus fort et de poids le plus faible correspondant au paramètre active_region_vertical_size.

dimension verticale de région active (active_region_vertical_size) – Voir la définition de la structure active_region_horizontal_size. Cette valeur ne sera pas plus grande que la dimension verticale (vertical_size) de l'image bitrame codée. La valeur '0' indique que la dimension est inconnue.

Au cas où une image bitrame donnée ne posséderait pas de fenêtre de région active dans le flux binaire, la dernière fenêtre de région active décodée sera utilisée. A la suite d'un en-tête de séquence, les paramètres de fenêtre de région active active_region_horizontal_size et active_region_vertical_size seront remis aux valeurs des paramètres horizontal_size et vertical_size tels que définis dans l'en-tête de séquence, et les paramètres top_left_x et top_left_y seront remis à 0.

16) Nouveau § 6.3.21.5

Ajouter le nouveau § 6.3.21.5 suivant:

6.3.21.5 Longueur d'image codée

La longueur d'image codée spécifie le nombre d'octets inclus depuis le premier octet qui suit immédiatement le premier code de déclenchement de tranche (slice_start_code) d'une image jusqu'au premier octet du préfixe de code de déclenchement qui suit immédiatement le dernier macrobloc de l'image. Le flux binaire ne contiendra pas plus d'une longueur d'image codée pour chaque image.

comptage des octets de l'image partie a, comptage des octets de l'image partie b, comptage des octets de l'image partie c, comptage des octets de l'image partie d (picture_byte_count_part_a, picture_byte_count_part_b, picture_byte_count_part_c, picture_byte_count_part_d) – Les 8 bits: ceux de poids le plus fort, ceux qui viennent en deuxième position, ceux qui viennent en troisième position et ceux de poids le plus faible, correspondant au paramètre picture_byte_count.

comptage des octets de l'image (picture_byte_count) – Entier non signé à 32 bits indiquant le nombre d'octets depuis le premier octet du premier code de déclenchement de tranche (slice_start_code) de l'image en cours jusqu'à l'octet précédant le préfixe du code de déclenchement qui suit immédiatement le dernier macrobloc de cette image. La valeur '0' est autorisée. Celle-ci indique que la longueur de l'image est inconnue.

17) Paragraphe E.1

Remplacer le Tableau E.7 par le suivant:

Tableau E-7 – En-tête d'image

#	Mode								Type	Observations	
	Eléments syntaxiques	SIMPLE	PRINCIPAL	SNR	SPATIAL	SUPERIEUR	4:2:2	Multiformat			
01	temporal_reference	x	x	x	x	x	x	x	x	I	
02	picture_coding_type	x	x	x	x	x	x	x	x	I	Profil simple: I et P à niveau principal I, P et B à bas niveau Profil principal, SNR, spatial, supérieur et multiformat: I, P et B
03	vbv_delay	x	x	x	x	x	x	x	x	I	
04	full_pel_forward_vector	x	x	x	x	x	x	x	x	I	Prend la valeur "0" pour la Rec. UIT-T H.262 ISO/CEI 13818-2
05	forward_f_code	x	x	x	x	x	x	x	x	I	Prend la valeur "111" pour la Rec. UIT-T H.262 ISO/CEI 13818-2
06	full_pel_backward_vector	x	x	x	x	x	x	x	x	I	Prend la valeur "0" pour la Rec. UIT-T H.262 ISO/CEI 13818-2
07	backward_f_code	x	x	x	x	x	x	x	x	I	Prend la valeur "111" pour la Rec. UIT-T H.262 ISO/CEI 13818-2
08	content_description_data()	x	x	x	x	x	x	x	x	I	
09	picture_coding_extension()	x	x	x	x	x	x	x	x	I	
10	quant_matrix_extension()	x	x	x	x	x	x	x	x	I	
11	picture_display_extension()	x	x	x	x	x	x	x	x	P	
12	picture_spatial_scalable_extension()	o	o	o	x	x	o	o	o	I	
13	picture_temporal_scalable_extension()	o	o	o	o	o	o	o	x	I	
14	camera_parameters_extension()	o	o	o	o	o	o	o	x	P	

18) Nouvelle Annexe K

Ajouter la nouvelle Annexe K suivante:

Annexe K**Impact des méthodes de codage liées à l'utilisation de flux binaires de séquences à balayage non progressif sur l'affichage à balayage progressif**

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

K.1 Codage à balayage progressif et non progressif

La présente annexe traite de l'impact des méthodes de codage liées à l'utilisation de séquences vidéo à balayage non progressif selon la Rec. UIT-T H.262 | ISO/CEI 13818-2 sur les systèmes d'affichage à balayage progressif. Elle a été établie principalement dans le but d'encourager les producteurs de contenu à coder des données de sorte qu'elles ne présentent pas d'artéfacts inutiles lorsqu'elles sont présentées sur des systèmes d'affichage à balayage progressif. Bien que le processus d'affichage n'entre pas dans le cadre de la présente Recommandation | Norme internationale, un certain nombre d'éléments syntaxiques, inclus dans le flux binaire, peuvent faciliter le processus d'affichage, tels que l'extension d'affichage de séquence ou l'extension d'affichage d'image. La présente annexe traite de l'optimisation de l'usage de la syntaxe en ce qui concerne l'impact de celle-ci sur le processus d'affichage.

La sémantique normative des fanions `progressive_frame` décrit la relation temporelle d'origine entre les différentes trames situées à l'intérieur d'une image codée d'une séquence à balayage non progressif. Les décodeurs qui affichent le contenu sur des dispositifs à balayage progressif se servent généralement de ce fanion pour coupler des trames en vue de leur présentation.

La méthode d'affichage la plus couramment utilisée est la suivante: si une image est codée sous la forme d'une bitrame à balayage progressif, les deux trames sont entrelacées en vue de leur présentation sur un dispositif à balayage progressif; dans le cas contraire, on effectue une conversion balayage entrelacé/balayage progressif pour convertir les données de trame sorties en données de bitrame en vue de leur affichage. Si l'image est vraiment produite au moyen d'un balayage progressif mais codée à l'aide d'une indication temporelle d'origine à balayage non progressif incorrecte, le processus de conversion balayage entrelacé/balayage progressif ne pourra être appliqué correctement et risque de provoquer de graves artéfacts ainsi qu'une perte de la résolution verticale sur le dispositif d'affichage.

K.2 Syntaxe d'information relative à la fréquence d'échantillonnage de la source vidéo

La fréquence d'échantillonnage de la source vidéo représentée pour des images contenues dans des séquences à balayage non progressif (lorsque le paramètre `progressive_sequence` a la valeur '0') dépend du fanion `progressive_frame` dans l'extension de codage de l'image définie au § 6.3.10. (Voir également les Figures 6-2, 6-3 et 6-4). Il est important de souligner que dans les images bitrames de ces séquences (lorsque le paramètre `picture_structure` a la valeur '11'), le paramètre `progressive_frame` peut avoir la valeur '0' ou '1' sans que le processus de décodage soit affecté; ce paramètre sert donc uniquement à indiquer la fréquence d'échantillonnage de la source.

La structure temporelle d'échantillonnage de la source représentée dans une séquence à balayage non progressif comprend un décalage temporel entre les trames de l'image à n'importe quel moment où le fanion `progressive_frame` a un bit a la valeur '0', dans les cas suivants:

- lorsque le paramètre `picture_structure` a la valeur '01' (trame supérieure) ou '10' (trame inférieure), auquel cas le paramètre `progressive_frame` doit avoir la valeur '0', ou
- lorsque le paramètre `picture_structure` a la valeur '11' (image bitrame) et le paramètre `progressive_frame` a la valeur '0' (balayage non progressif).

La fréquence d'échantillonnage de la source représentée est celle d'une image bitrame échantillonnée à un instant unique dans le cas suivant:

- lorsque le paramètre `picture_structure` a la valeur '11' (image bitrame) et le paramètre `progressive_frame` a la valeur '1' (balayage progressif).

Dans ce dernier cas, l'image est indiquée comme étant à balayage progressif, comme il serait le cas si le paramètre `progressive_sequence` avait la valeur '1'.

Le processus d'affichage à balayage progressif des bitrames à balayage progressif fait intervenir normalement simplement toutes les lignes de l'image décodée avec entrelacement des deux trames. Le processus d'affichage à balayage progressif de bitrames à balayage non progressif est généralement très différent du processus précédent à simple entrelacement des trames.

K.3 Méthodes de production de contenu

Si les données d'origine qui doivent être codées sont échantillonnées sous la forme de bitrames complètes de contenu à balayage progressif, il est important que la nature progressive des données d'origine soit correctement représentée dans le flux binaire de données vidéo. Le contenu à balayage progressif devrait par conséquent être codé au moyen d'une représentation à balayage progressif de couplage approprié. Si cette méthode n'est pas employée, un grand nombre d'artéfacts inutiles peuvent se produire sur des systèmes utilisant un affichage à balayage progressif. Il est également important de s'assurer que les données véritablement entrelacées soient codées avec le paramètre `progressive_frame = '0'`, de manière à éviter un traitement d'affichage incorrect sur des systèmes utilisant un affichage à balayage progressif.

Si une séquence d'origine complète est composée de bitrames à balayage progressif, la séquence sera alors, si possible, simplement codée sous la forme de bitrames à balayage progressif avec le paramètre `progressive_sequence` portant la valeur '1'. Dans des séquences à balayage non progressif (lorsque le paramètre `progressive_sequence` a la valeur '0'), la nature progressive peut toujours être représentée par le codage d'images à balayage progressif sous la forme d'images bitrames avec le paramètre `progressive_frame` égal à '1'.

L'expérience montre que les producteurs de contenu omettent parfois de signaler correctement la nature progressive des bitrames à balayage progressif codées à l'intérieur de séquences à balayage non progressif. L'emploi de certaines méthodes de montage vidéo peut également entraîner dans une certaine mesure la perte de la nature progressive d'une source à balayage progressif et empêcher ainsi son codage correct sous la forme de bitrames à balayage progressif. Le but de la présente annexe est principalement d'éviter que les réalisateurs de contenu créent des flux binaires de données vidéo qui produisent des artéfacts inutiles résultant de ces problèmes.

K.3.1 Prétraitement de conversion de fréquence image

Les données d'origine produites à une fréquence image donnée sont généralement converties en vue de leur codage sous la forme d'un flux binaire de données vidéo à une fréquence image différente. Si la fréquence image d'origine est relativement plus basse que la fréquence image codée, on ajoute la plupart du temps dans des séquences à balayage non progressif des monotrames répétées de contenu codé avec le paramètre `progressive_frame = '1'` et le paramètre `repeat_first_field = '1'`.

La méthode la plus couramment employée actuellement consiste à convertir des images à balayage progressif extraites de films à 24 images/seconde en représentation vidéo 30 000/1001 bitrames par seconde par le procédé de décimation à 3:2 (appelé également décimation à 2:3). Dans ce processus, on convertit chaque ensemble de quatre images consécutives analysées par balayage progressif et désignées A, B, C et D, en dix trames de contenu vidéo en répétant la première trame des deuxième et quatrième images (images B et D). Le même processus est répété pour chaque ensemble subséquent de quatre images consécutives.

Chaque image de film est scannée en vue de produire deux trames de lignes en alternance et de transmettre, dans chaque séquence de quatre images, la première trame des deuxième et quatrième images après l'envoi des première et deuxième trames de l'image et avoir choisi la trame à envoyer en premier pour conserver le modèle à trames alternées, convertissant ainsi chaque séquence de quatre images du film à 24 Hz en dix trames vidéo à 29,97 Hz, de la manière suivante:

- envoi de la trame supérieure de la première image du film (image A), puis
- envoi de la trame inférieure de la première image du film (image A), puis
- envoi de la trame supérieure de la deuxième image du film (image B), puis
- envoi de la trame inférieure de la deuxième image du film (image B), puis
- répétition de l'envoi de la trame supérieure de la deuxième image du film (image B) (généralement en mettant `repeat_first_field` à 1), puis
- envoi de la trame inférieure de la troisième image du film (image C), puis
- envoi de la trame supérieure de la troisième image du film (image C), puis
- envoi de la trame inférieure de la quatrième image du film (image D), puis
- envoi de la trame supérieure de la quatrième image du film (image D), puis
- répétition de l'envoi de la trame inférieure de la quatrième image du film (image D) (généralement en mettant `repeat_first_field` à 1), puis
- le processus ci-dessus se répète modulo 10 pour les trames subséquentes.

Ce processus diminue la durée totale d'un facteur 1001/1000 (créant une source de 23,976 images de film par seconde environ) et propose le film sous une forme qui se prête à la lecture par des systèmes à affichage entrelacé. Dans le cas d'une décimation à 3:2, la méthode préférée est la représentation de chaque image d'origine (A, B, C et D) dans le flux binaire sous la forme d'une image codée distincte. En d'autres termes, les images d'origine B et D du modèle devraient être codées sous la forme d'images distinctes avec le paramètre `progressive_frame = '1'` et le paramètre `repeat_first_field = '1'`. Un cas d'utilisation pauvre de la syntaxe serait de coder les images d'origine A et B comme les deux premières images codées (chacune avec le paramètre `repeat_first_field = '0'`), de coder ensemble la première trame répétée de l'image d'origine B et la première des deux trames de l'image d'origine C comme la troisième image codée, puis de coder ensemble la seconde trame de l'image d'origine C et la première trame de l'image d'origine D comme la quatrième image codée, et enfin de coder les deux trames de l'image d'origine D comme la cinquième image codée; et de répéter ce schéma pour chaque ensemble de quatre images d'origine. Cette pauvre utilisation de la syntaxe se traduira probablement par l'apparition d'artéfacts significatifs dans un système d'affichage à balayage progressif, étant donné que le processus d'affichage sera sans doute incapable de récupérer la paire de trames et l'information de structure temporelle appropriées nécessaires pour reconstruire correctement les images à balayage progressif.

En outre, il est très important que même les images d'origine qui ne possèdent pas une première trame répétée contiennent une représentation appropriée de leur nature progressive. Cela nécessite que les images d'origine A et C soient codées sous la forme d'images bitrames avec le paramètre `progressive_frame = '1'` (et avec le paramètre `repeat_first_field = '0'`). Dans les cas où le codeur doit essayer d'identifier la présence d'une décimation à 3:2 à l'intérieur d'une série de trames de source vidéo, la présence de trames répétées peut être utile pour déterminer que les données d'origine sont à balayage progressif et pour définir l'association correcte des trames aux bitrames (la qualité des données sorties serait néanmoins bien meilleure si le codeur pouvait connaître la vraie nature de la source au lieu d'avoir à l'identifier au moyen d'un processus de détection si imparfait).

L'application de ces méthodes de codage préférées exige que la nature progressive des bitrames d'origine et que l'association correcte des trames aux bitrames soient conservées tout au long du processus de codage.

K.3.2 Méthodes de montage préjudiciables liées aux trames

Certaines méthodes de montage vidéo que l'on emploie pour former des trames à balayage progressif, sans connaître le couplage de trames correct, peuvent être préjudiciables à la nature progressive des données d'origine. Il convient, dans la mesure du possible, de les éviter en raison des artéfacts qu'elles peuvent provoquer sur les systèmes utilisant un affichage à balayage progressif. Afin d'éviter ces problèmes, la chaîne complète des processus de production, de montage et de codage devrait être conçue de manière à garantir le maintien des informations correctes concernant la nature progressive ou entrelacée de chaque partie du contenu vidéo. Dans le traitement par décimation à 3:2, il convient d'employer des méthodes de montage basées sur le même modèle de traitement à paires de trames que celui utilisé pour les données d'origine sous-jacentes, afin d'éviter ces problèmes.

K.3.2.1 Coupures de scène liées aux trames

Une méthode de montage préjudiciable consisterait par exemple à commuter entre deux sources vidéo à balayage progressif au cours d'un traitement basé sur les trames, entre les deux trames d'une bitrame à balayage progressif, en appliquant par exemple une coupure de scène juste après la première trame de l'image d'origine C dans une série de décimations à 3:2, tel qu'il est décrit au § K.3.1. Une coupure de scène à cet endroit se traduira par la création d'une "trame torsadée" du contenu des données vidéo d'origine, c'est-à-dire une trame isolée des données vidéo d'origine qui ne peut être correctement couplée à une autre bitrame en vue de créer une bitrame codée à balayage progressif du contenu des données vidéo d'origine. Si une telle condition existe et doit être codée dans un flux binaire de données vidéo, il est important que le paramètre `progressive_frame` soit mis à '0' sur l'image spécifique qui contient la trame torsadée afin de signaler correctement sa nature non progressive.

K.3.2.2 Superpositions et composition d'images structurées en trames

Un autre cas de montage basé sur les trames, qui peut être préjudiciable à la détermination de la nature progressive d'une image, est l'insertion de superpositions de données textuelles animées ou d'autres types de contenu. Si ces données sont insérées dans une scène à balayage non progressif au moyen d'un procédé de montage basé sur l'entrelacement, les données d'origine ne peuvent plus être correctement caractérisées comme étant réellement à balayage progressif ou à balayage entrelacé. Par conséquent, il n'apparaît pas clairement si une image doit être marquée avec le paramètre `progressive_frame = '0'` ou '1'. En outre, les systèmes utilisant un affichage à balayage progressif peuvent avoir de grosses difficultés à déterminer la présentation des images codées en vue de leur affichage. Afin d'éviter ces problèmes, ces superpositions et compositions d'images devraient être réalisées, dans la mesure du possible, au moyen de représentations à balayage progressif.

K.3.2.3 Fondus et transitions entre les scènes liés aux trames

Si l'on réalise des transitions graduelles entre différents éléments de scène à balayage progressif, par exemple, un "fondu au blanc" (*fade in*), un "fondu au noir" (*fade out*), une commutation de scène graduelle ou une transition "par volet" (*wipe*), il conviendra d'employer dans toute la mesure du possible, un traitement à balayage progressif basé sur les trames (à partir des bitrames réellement à balayage progressif couplées de façon appropriée). Si cette méthode n'est pas employée, les données d'origine ne peuvent plus être correctement caractérisées comme étant réellement à balayage progressif ou à balayage entrelacé. Par conséquent, il n'apparaît pas clairement si une image doit être marquée avec le paramètre `progressive_frame = '0'` ou `'1'`. En outre, les systèmes utilisant un affichage à balayage progressif peuvent avoir de grosses difficultés à déterminer la présentation des images codées en vue de leur affichage.

K.3.2.4 Effets spéciaux liés aux trames

Des effets spéciaux, tels qu'un zoom qui peut avoir été employé après l'acquisition initiale des données vidéo, devront être réalisés de la même façon, avec le plus grand soin, en ce qui concerne la nature progressive des données. Ces effets devront être appliqués en cours d'exécution sur des images bitrames à balayage progressif correctement couplées plutôt que d'une façon arbitraire, de manière à préserver la nature progressive des données.

K.3.2.5 Ajustements de la vitesse liés aux trames

Si l'on modifie le nombre d'intervalles de trames associés aux images d'origine à balayage progressif en vue d'ajuster la vitesse de mouvement dans une scène, il faudra bien s'assurer lors d'un traitement ultérieur (par exemple, la détection d'une décimation à 3:2 dans un codeur, tel que décrit au § K.3.1), que la nature progressive des images ainsi que l'association correcte des trames aux images soient préservées.

K.3.2.6 Centrage d'image

Si le centre de l'image à signaler dans une structure `picture_display_extension()` indique des valeurs différentes de la structure `frame_centre_vertical_offset` ou de la structure `frame_centre_horizontal_offset` pour différentes trames dans la même image, une forme de changement temporel entre les trames autre que celle des images à balayage progressif sera alors indiquée. La fréquence image sous-jacente des images d'origine à balayage progressif serait alors réellement modifiée par ce processus, ce qui pourrait entraîner la création d'une séquence affichée présentant des changements axés sur les trames pour chaque image à balayage progressif de ce type.

K.4 Montage post-codage du fanion de bitrame à balayage progressif dans des flux binaires de données vidéo

Il est important de souligner que dans des images bitrames de séquences à balayage non progressif (lorsque le paramètre `progressive_sequence` est mis à `'0'`), la valeur du paramètre `progressive_frame` n'a aucun effet sur le processus de décodage et sert donc uniquement à indiquer la fréquence d'échantillonnage d'origine.

En raison de l'importance de la valeur du paramètre `progressive_frame` pour les systèmes utilisant un affichage à balayage progressif, il peut être conseillé dans certains cas de modifier la valeur de ce bit dans des flux binaires qui ont été créés au moyen de paramètres incorrects de ce fanion. Les conditions de modification de cette valeur garantissant le bon fonctionnement du processus de décodage sont les suivantes:

- le paramètre `progressive_sequence` doit être mis à `'0'` (séquence non progressive);
- le paramètre `picture_structure` doit être mis à `'11'` (image bitrame);
- le paramètre `progressive_frame` ne peut pas passer de `'1'` à `'0'` si le paramètre `repeat_first_field` est mis à `'1'`; et
- le paramètre `progressive_frame` ne peut pas être modifié à moins que le paramètre `chroma_420_format` soit également modifié de sorte qu'il ait la même valeur que le paramètre `progressive_frame`.

NOTE – Ces déclarations ne sont données qu'à titre indicatif (comme toutes les déclarations indiquées dans la présente annexe, ces déclarations ne modifient d'aucune façon les spécifications figurant dans les parties intégrantes de la présente Recommandation [Norme internationale).

K.5 Post-traitement des systèmes dotés d'un affichage à balayage progressif

Si le système de décodage détecte la présence d'une bitrame à balayage non progressif isolée dans une série de bitrames à balayage progressif, cela pourrait indiquer la présence d'une trame torsadée (se reporter au paragraphe K.3.2.1). Les concepteurs d'affichage devront envisager des mesures particulières pour régler ce problème.

Si le système de décodage détecte la présence d'un modèle répétitif de bitrames à balayage progressif avec la structure `repeat_first_field = '1'` combinée à des bitrames à balayage non-progressif, cela pourrait indiquer qu'un codeur n'est pas informé de la nature progressive des bitrames du contenu d'origine et qu'il peut détecter qu'une bitrame est à balayage

progressif uniquement si sa première trame est répétée dans la séquence vidéo d'origine. Les concepteurs d'affichage devraient envisager même de considérer les bitrames de structure `repeat_first_field = '0'` et de structure `progressive_frame = '0'` comme étant des bitrames réellement à balayage progressif si un modèle répétitif persistant de ce phénomène était rencontré dans le flux binaire.

K.6 Utilisation des informations du code temporel d'acquisition

La structure temporelle correcte des trames et images bitrames vidéo pour la visualisation peut être indiquée en ajoutant au flux vidéo les informations du code temporel d'acquisition. On trouvera ci-après quelques exemples d'utilisation de ces informations.

K.6.1 Exemple: Vidéo 525/60 (29,97 Hz) avec comptage en mode non-temps réel (NDF)

La vidéo entrelacée habituelle 525/60 peut être représentée dans la structure de données du code temporel d'acquisition sans utilisation de la structure temporelle en temps réel (DF) par le réglage suivant des paramètres de base de temps:

```
counting_type = '001' (sans abandon d'unités de nframes)
nframes_conversion_code = '1'
clock_divisor = 45
nframes_multiplifier = 20
```

Etant donné que le type de comptage est '001', `prior_count_dropped` sera toujours égal à zéro.

Le nombre `max_nframes` est donc $(26\ 999\ 999) / (20 * (1000 + 1) * 45) = 29$.

Définir une variable X de calcul du décalage temporel (`time_offset`), avec une valeur initiale X_0 (par exemple $X_0 = 0$).

Si le processus démarre avec `nframes = 1` et `time_offset = X` pour le marqueur temporel de la première trame, la deuxième trame aura les mêmes valeurs pour tous les paramètres, sauf `time_offset`, qui sera $X + 10010 = 10010$. Pour la première trame de la deuxième image, `time_offset` sera à nouveau X, et pour la deuxième trame, à nouveau $X + 10010$, pour se répéter jusqu'à ce que la valeur du comptage `nframes` dépasse 29.

Lorsque le comptage de `nframes` dépasse 29, `nframes` est mis à 0, les unités de secondes sont incrémentées (avec les reports éventuels sur les dizaines de secondes, les unités de minutes etc.), X est augmenté de 600 et `time_offset` est mis à X. Pour 30 images, `time_offset` sera de X pour la première trame de chaque image et de $X + 10010$ pour la deuxième trame de chaque image (au moyen de la valeur incrémentée de X) jusqu'à ce que le comptage de `nframes` atteigne à nouveau 29; à ce moment, X sera à nouveau incrémenté de 600 et `time_offset` sera X pour la première trame avec `nframes = 0`, `time_offset` sera $X + 10010$ pour la deuxième trame avec `nframes = 0`, X pour la première trame avec `nframes = 1`, et sera $X + 10010$ pour la deuxième trame avec `nframes = 1`, etc.

Dans le présent exemple, la différence entre toute paire de marqueurs temporels adjacents équivalents sera toujours $10010 * 45 = 450\ 450$. On le constate en examinant les marqueurs temporels de la deuxième trame de la 29^e image:

```
units_of_seconds = 0
nframes = 29
time_offset = X0 + 10010
equivalent_timestamp = 0 + (29 * 20 * (1000 + 1) + X0 + 10010) * 45
                    = X0 + 26 576 550
```

et le marqueur temporel de la première trame de la 30^e image:

```
units_of_seconds = 1
nframes = 0
time_offset = 600
equivalent_timestamp = 27 000 000 + (0 * 20 * (1000 + 1) + X0 + 600) * 45
                    = X0 + 27 027 000
```

La soustraction fait apparaître une différence de:

$$27\ 027\ 000 - 26\ 576\ 550 = 450\ 450$$

soit la valeur escomptée.

K.6.2 Exemple: Vidéo 525/60 (29,97 Hz) avec comptage en mode temps réel (DF)

La vidéo entrelacée classique 525/60 peut être représentée dans la structure de données de code temporel d'acquisition avec le comptage en mode temps réel, d'usage courant, avec les valeurs de paramètres de base de temps suivantes:

```
counting_type = '100' (abandon de deux unités de nframes)
nframes_conversion_code = '1'
clock_divisor = 45
nframes_multiplier = 20
```

Ce processus commence d'une manière pratiquement identique à celle du § K.6.1, sauf pour la valeur de `counting_type`. Le processus de comptage est également le même jusqu'à ce que l'unité des minutes passe à 1; à ce moment, au lieu de `nframes` passant à 0 et `X` étant incrémenté de 600 d'une image à la suivante, `prior_frame_dropped` est mis à '1', `nframes` est mis à 2 au lieu de 0 et `X` est diminué de 39440 au lieu d'être augmenté de 600. Le même ajustement se produit chaque fois que les unités de minutes augmentent et que la valeur résultante des unités de minutes est différente de zéro.

K.6.3 Exemple: Structure temporelle vidéo 625/50 (25 Hz)

La vidéo entrelacée classique 625/50 peut être représentée dans la structure de données de code temporel d'acquisition avec les valeurs de paramètres de la base de temps suivantes:

```
counting_type = '001' (pas d'abandon d'unités de nframes)
nframes_conversion_code = '0'
clock_divisor = 45
nframes_multiplier = 24
```

Etant donné que `counting_type = '001'`, `prior_count_dropped` sera toujours égal à zéro.

Cela se traduit par un nombre `max_nframes` de $(26\ 999\ 999) / (24 * (1000 + 0) * 45) = 24$.

Dans ce cas, le nombre de `nframes` augmente de 0 à 24, puis les unités de secondes augmentent et `nframes` devient à nouveau 0. Le paramètre `time_offset` reste à sa valeur initiale $X = X_0$ (par exemple, $X_0 = 0$) pour tous les marqueurs temporels subséquents avec le même type de structure temporelle.

K.6.4 Exemple: Vidéo 525/60 (29,97 Hz) à 23,976 Hz par décimation à 3:2

Il est courant de convertir des images à balayage progressif extraites de films à 24 Hz en représentation vidéo 525/60 par le procédé de décimation à 3:2 (voir § K.3.1). Cela nécessite le scannage de chaque image à balayage progressif pour produire des trames à lignes alternées puis de représenter 10 trames vidéo codées par séquence de 4 images à balayage progressif de la source. Dans un tel cas, le code temporel d'acquisition représente la nature progressive du film original qui sous-tend la séquence de trames convertie en indiquant le même instant d'échantillonnage pour les trames provenant de la même image de film originale.

Dans un tel cas, le code temporel d'acquisition permet au décodeur de rétablir la structure de l'image non entrelacée sous-jacente en identifiant les trames associées par leur instant d'échantillonnage. Ces informations temporelles additionnelles peuvent être très utiles en visualisation à balayage progressif.

K.6.4.1 Exemple: Vidéo de 525/60 (23,976 Hz) avec comptage en mode non-temps réel (NDF)

La structure temporelle de l'échantillonnage des images sous-jacente, portée à 23,976 Hz, peut être indiquée comme étant sous-jacente à un code temporel NDF de la manière suivante:

```
counting_type = '001' (pas d'abandon d'unités de nframes)
nframes_conversion_code = '1'
clock_divisor = 45
nframes_multiplier = 20
```

Etant donné que `counting_type = '001'`, `prior_count_dropped` sera toujours égal à zéro.

Cela se traduit par un nombre `max_nframes` de $(26\ 999\ 999) / (20 * (1000 + 1) * 45) = 29$.

Ensuite il convient de définir l'ensemble des 10 éléments (indiqués de 0 à 9):

$$Y[10] = \{0, 0, 5005, 5005, -15015, 10010, -10010, 15015, -5005, -5005\}$$

Si le processus commence par $nframes = 1$ et $time_offset = X + Y[0] = X_0$ (par exemple $X_0 = 0$) pour le marqueur temporel de la première trame de la première image, la deuxième trame aura les mêmes valeurs pour tous les paramètres sauf pour $time_offset$, qui sera $X + Y[1] = 0$. Pour la première trame de la deuxième image, $time_offset$ sera $X + Y[2]$, et pour la deuxième trame il sera $X + Y[3]$, etc. Le processus continue jusqu'à ce que l'indice de l'ensemble Y atteigne le nombre, modulo 10, de la trame en cours de transmission (0 représentant l'indice de la première trame pour $nframes = 1$; 1 étant la deuxième trame pour $nframes = 1$ et 2 étant la première trame pour $nframes = 2$, etc.).

Cette répétition se poursuit jusqu'à ce que la valeur du comptage $nframes$ atteigne 29; à ce moment, la valeur de X est ajustée comme indiqué dans le § K.6.4.1, le nombre d'unités de secondes est augmenté, $nframes$ est mis à 0, et le processus continue. L'indice de Y continue d'augmenter modulo 10 (il n'est pas remis à 0 lorsque les unités de secondes sont augmentées).

Lorsque les marqueurs temporels sont générés de cette manière (omission du marqueur temporel de toute trame qui n'est pas envoyée comme la troisième trame d'une image avec $repeat_first_field = 1$ étant donné que celles-ci ont un marqueur temporel équivalent sous-entendu égal à celui de la première trame de toute image de cette nature), la structure temporelle à 23,976 Hz à balayage progressif qui sous-tend la fréquence de trame 525/60 est indiquée. Les valeurs de $equivalent_timestamp$ des trames adjacentes diffèrent de 25025 lorsque les trames proviennent d'images de film à balayage progressif différentes et de 0 lorsqu'elles proviennent de la même image.

K.6.4.2 Exemple: Vidéo 525/60 (23,976 Hz) avec comptage en mode temps réel (DF)

La structure temporelle de l'échantillonnage de l'image sous-jacente, telle qu'elle est portée à 23,976 Hz environ, peut être indiquée comme étant sous-jacente au code temporel en mode temps réel, d'usage courant, conformément à ce qui suit:

```
counting_type = '100' (suppression de deux unités de nframes)
nframes_conversion_code = '1'
clock_divisor = 45
nframes_multiplier = 20
```

La définition de l'ensemble à 10 éléments est la même qu'au § K.6.4.1, le processus est le même qu'au § K.6.4.1, à cela près que les valeurs de X et de $nframes$ sont calculées comme au § K.6.2 au lieu du § K.6.1. La valeur de $time_offset$ est mise à $X + Y[k]$, et k est l'indice de trame tel qu'il est utilisé dans le § K.6.4.1.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication