Union internationale des télécommunications

UIT-R

Secteur des Radiocommunications de l'UIT

Recommandation UIT-R BT.1885

(03/2011)

Techniques de mesure objective de la qualité vidéo perçue pour la télédiffusion numérique à définition normale en présence d'une largeur de bande réduite

Série BT

Service de radiodiffusion télévisuelle

ITU Union internationale des télécommunications

Avant-propos

Le rôle du Secteur des radiocommunications est d'assurer l'utilisation rationnelle, équitable, efficace et économique du spectre radioélectrique par tous les services de radiocommunication, y compris les services par satellite, et de procéder à des études pour toutes les gammes de fréquences, à partir desquelles les Recommandations seront élaborées et adoptées.

Les fonctions réglementaires et politiques du Secteur des radiocommunications sont remplies par les Conférences mondiales et régionales des radiocommunications et par les Assemblées des radiocommunications assistées par les Commissions d'études.

Politique en matière de droits de propriété intellectuelle (IPR)

La politique de l'UIT-R en matière de droits de propriété intellectuelle est décrite dans la «Politique commune de l'UIT-T, l'UIT-R, l'ISO et la CEI en matière de brevets», dont il est question dans l'Annexe 1 de la Résolution UIT-R 1. Les formulaires que les titulaires de brevets doivent utiliser pour soumettre les déclarations de brevet et d'octroi de licence sont accessibles à l'adresse http://www.itu.int/ITU-R/go/patents/fr, où l'on trouvera également les Lignes directrices pour la mise en oeuvre de la politique commune en matière de brevets de l'UIT-T, l'UIT-R, l'ISO et la CEI et la base de données en matière de brevets de l'UIT-R.

Séries des Recommandations UIT-R

(Egalement disponible en ligne: http://www.itu.int/publ/R-REC/fr)

Séries	Titre
BO	Diffusion par satellite
BR	Enregistrement pour la production, l'archivage et la diffusion; films pour la télévision
BS	Service de radiodiffusion sonore
BT	**Service de radiodiffusion télévisuelle**
F	Service fixe
M	Services mobile, de radiorepérage et d'amateur y compris les services par satellite associés
P	Propagation des ondes radioélectriques
RA	Radio astronomie
RS	Systèmes de télédétection
S	Service fixe par satellite
SA	Applications spatiales et météorologie
SF	Partage des fréquences et coordination entre les systèmes du service fixe par satellite et du service fixe
SM	Gestion du spectre
SNG	Reportage d'actualités par satellite
TF	Emissions de fréquences étalon et de signaux horaires
V	Vocabulaire et sujets associés

Note: *Cette Recommandation UIT-R a été approuvée en anglais aux termes de la procédure détaillée dans la Résolution UIT-R 1.*

Publication électronique
Genève, 2011

© UIT 2011

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

RECOMMANDATION UIT-R BT.1885

Techniques de mesure objective de la qualité vidéo perçue pour la télédiffusion numérique à définition normale en présence d'une largeur de bande réduite

(2011)

Domaine d'application

La présente Recommandation décrit des méthodes d'évaluation objective de la qualité vidéo pour la télédiffusion numérique à définition normale qui permettent de mesurer la qualité vidéo perçue dans des conditions de réception mobiles et fixes, lorsque les caractéristiques extraites du signal vidéo de référence sont directement disponibles au point de mesure.

L'Assemblée des radiocommunications de l'UIT,

considérant

a) qu'il est souhaitable de pouvoir mesurer automatiquement les dégradations de signaux vidéo diffusés;

b) que la qualité vidéo perçue de la réception mobile peut varier de façon dynamique en fonction des conditions de réception;

c) que les mesures objectives de la qualité vidéo perçue peuvent utilement compléter les méthodes d'évaluation subjective;

d) que trois méthodes de mesure objective de la qualité vidéo pour la télédiffusion numérique à définition normale en présence d'une largeur de bande réduite ont été proposées à l'UIT-R, et qu'il a été établi que ces méthodes donnaient des résultats équivalents et concordants;

e) que des techniques de mesure de la qualité vidéo perçue pour la télédiffusion numérique à définition normale en présence d'une référence de largeur de bande complète ont été définies dans la Recommandation UIT-R BT.1683,

recommande

1 d'utiliser les modèles d'évaluation objective de la qualité vidéo décrits dans l'Annexe 1 pour effectuer des mesures objectives de la qualité vidéo perçue pour la télédiffusion numérique à définition normale en présence d'une largeur de bande réduite.

1 Introduction

Le présent test RRNR-TV porte sur les images définies dans la Recommandation UIT-R BT.601-6 et sur deux types de modèles: le modèle avec image de référence réduite (RR), et le modèle sans image de référence (NR). Les modèles RR ont un accès limité en largeur de bande à la séquence vidéo source et les modèles NR n'ont pas accès à cette séquence.

Dans chaque essai, les circuits HRC comprenaient à la fois un codage produisant des défauts (artéfacts) seulement et un codage avec des erreurs de transmission. Les systèmes de codage examinés étaient les systèmes MPEG-2 et H.264 (MPEG-4 partie 10). Les codeurs MPEG-2 ont été utilisés à différents débits binaires, compris entre 1,0 et 5,5 Mbit/s. les codeurs H.264 ont été utilisés à différents débits binaires, compris entre 1,0 et 3,98 Mbit/s. Chaque essai comportait 12 séquences source, dont deux étaient des codes source secrets. Chaque essai comprenait 34 circuits HRC, et 156 séquences vidéo traitées (PVS), dont 40 contenaient des erreurs de transmission et 116 uniquement du codage.

1.1 Application

La présente Recommandation donne des estimations de la qualité vidéo pour les classes vidéo TV3 à MM5B définies dans l'Annexe B de la Recommandation UIT-T P.911. Les applications des modèles d'estimation décrits dans la présente Recommandation sont notamment les suivantes:

1) contrôle de la qualité pendant le service, éventuellement en temps réel, à la source;
2) télécontrôle de la qualité au point de destination lorsqu'on dispose de canaux latéraux pour les caractéristiques extraites de la séquence vidéo source;
3) mesures de qualité d'un système d'archivage ou de transmission qui utilise des techniques de compression ou de décompression vidéo, par passage unique ou concaténation de telles techniques;
4) essais en laboratoire de systèmes vidéo.

1.2 Limites

Les modèles d'estimation décrits dans la présente Recommandation ne peuvent être utilisés pour remplacer intégralement les essais subjectifs. Les valeurs de corrélation entre deux essais subjectifs conçus et exécutés avec soin (par exemple dans deux laboratoires différents) se situent normalement dans la fourchette 0,95-0,98. Si la présente Recommandation est utilisée pour comparer différents codecs, il est recommandé d'utiliser une méthode quantitative (telle celle décrite dans la Recommandation UIT-T J.149) pour déterminer l'exactitude d'un modèle pour ce contexte particulier.

Les modèles décrits dans la présente Recommandation ont été validés par des mesures de la qualité vidéo objective présentant un gel de trame pouvant durer jusqu'à 2 s.

Les modèles décrits dans la présente Recommandation n'ont pas été validés pour des mesures objectives de la qualité vidéo présentant un retard augmentant régulièrement (par exemple une vidéo dans laquelle les trames manquantes ne sont pas éliminées après un gel de trame).

Il convient de noter que, dans le cas de nouvelles technologies de codage et de transmission introduisant des défauts (artéfacts) qui n'ont pas été incluses dans la présente évaluation, les modèles de mesure objective pourraient donner des résultats erronés. Dans ce cas, une évaluation subjective est indispensable.

2 Références

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

2.1 Références normatives

Recommandation UIT-R BT.500-12 – Méthodologie d'évaluation subjective de la qualité des images de télévision.

Recommandation UIT-T P.910 (2008) – Méthodes subjectives d'évaluation de la qualité vidéographique pour les applications multimédias.

Recommandation UIT-T P.911 (1998) – Méthodes subjectives d'évaluation de la qualité audiovisuelle pour applications multimédias.

Recommandation UIT-T J.143 (2000) – Prescriptions d'utilisateur relatives aux mesures objectives de la qualité vidéo perçue en télévision numérique par câble.

Recommandation UIT-T J.244 (2008) – Méthodes d'étalonnage du désalignement constant des domaines spatial et temporel avec un gain et un décalage constant.

2.2 Références informatives

Recommandation UIT-T J.149 (1998) – Méthodes subjectives d'évaluation de la qualité audiovisuelle pour applications multimédias.

Recommandation UIT-T J.144 (2001) – Techniques de mesure objective de la qualité vidéo perçue pour la télévision numérique par câble en présence d'un signal de référence complet.

Recommandation UIT-T P.931 (1998) – Mesure du temps de transmission, de la synchronisation et du débit de trames dans les communications multimédias.

Recommandation UIT-T J.148 (2003) – Prescriptions pour un modèle objectif de qualité multimédia perçue.

Recommandation UIT-T H.261 (1993) – Codec vidéo pour services audiovisuels à p x 64 kbits.

Recommandation UIT-T H.263 (1996) – Codage vidéo pour communications à faible débit.

Recommandation UIT-T H.263 (1998) – Codage vidéo pour communications à faible débit (H.263+).

Recommandation UIT-T H.264 (2003) – Codage vidéo évolué pour les services audiovisuels génériques VQEG – Groupe d'experts sur la qualité vidéo (Video Quality Experts Group) Validation de modèles objectifs avec référence réduite et absence de référence pour la télévision à définition normale , Phase I, 2009.

3 Définitions

3.1 Termes définis ailleurs

La présente Recommandation utilise les termes suivants définis ailleurs:

3.1.1 évaluation subjective (image) (Recommandation UIT-T J.144): définition citée à titre facultatif.

3.1.2 mesure perceptuelle objective (image) (Recommandation UIT-T J.144): définition citée à titre facultatif.

3.1.3 proposant (Recommandation UIT-T J.144): (Recommandation UIT-T J.144): définition citée à titre facultatif.

3.2 Termes définis dans la présente Recommandation

La présente Recommandation définit les termes suivants:

3.2.1 répétition de trames anormale: cas dans lequel le circuit HRC produit une seule trame de manière répétée en réponse à un événement inhabituel ou présentant un caractère anormal. La répétition de trames anormale comprend, sans toutefois s'y limiter, les types de cas suivants: erreur dans le canal de transmission, variation du retard via le canal de transmission, ressources informatiques limitées ayant une incidence sur la qualité de fonctionnement du décodeur ou sur l'affichage du signal vidéo.

3.2.2 saut d'image constant: cas dans lequel le circuit HRC produit des trames avec un contenu actualisé à une fréquence de trame effective qui est fixe et inférieure à la fréquence de trame de la source.

3.2.3 fréquence de trame effective: nombre de trames unique (c'est-à-dire nombre total de trames – nombre de trames répétées) par seconde.

3.2.4 fréquence de trame: nombre de trames unique (c'est-à-dire nombre total de trames – nombre de trames répétées) par seconde.

3.2.5 fréquence de trame recherchée: nombre de trames vidéo par seconde stockées physiquement pour une représentation donnée d'une séquence vidéo. La fréquence de trame doit être constante Deux exemples de fréquences de trame recherchées sont une cassette BetacamSP® contenant 25 fps et un fichier YUV à 625 lignes conforme au signal VQEG FR-TV Phase I contenant 25 fps, qui ont tous deux une fréquence de trame recherchée de 25 fps.

3.2.6 conditions actives dans le réseau: erreurs imposées au flux binaire vidéo numérique en raisons de conditions actives dans le réseau.

3.2.7 pause avec sauts: cas dans lesquels le signal vidéo marque une pause pendant un certain temps, puis repart moyennant une perte d'informations vidéo. En cas de pause avec sauts, le décalage temporel dans le système variera comme un retard de système moyen, tantôt en hausse, tantôt en baisse. A titre d'exemple, on peut citer le cas d'une paire de visiophones IP, dans lequel il y a un bref arrêt de l'image sur l'écran du visiophone IP en raison d'un trafic intense sur le réseau. Le saut d'image constant et le saut d'image variable sont des sous-ensembles de la pause avec sauts. Une séquence vidéo traitée contenant une pause avec sauts aura plus ou moins la même durée que la séquence vidéo d'origine associée.

3.2.8 pause sans sauts: événement pendant lequel le signal vidéo marque une pause pendant un certain temps, puis repart sans aucune perte d'informations vidéo. En conséquence, le décalage temporel dans le système doit augmenter.

3.2.9 fréquence de régénération: fréquence à laquelle l'affichage est mis à jour.

3.2.10 erreurs de transmission simulées: erreurs imposées au flux binaire vidéo numérique dans un environnement très contrôlé. On citera à titre d'exemple les taux de perte de paquets et les erreurs binaires simulées.

3.2.11 fréquence de trame de la source (SFR): fréquence de trame recherchée des séquences vidéo source d'origine. La fréquence de trame de la source est constante. Pour le test VQEG RRNR-TV, la fréquence SFR est de 25 fps ou 30 fps.

3.2.12 erreurs de transmission: erreur imposée à la transmission vidéo. Comme exemples de types d'erreurs, on citera les erreurs de transmission simulées et les conditions actives dans le réseau.

3.2.13 saut de trame variable: cas dans lequel le circuit HRC produit des trames avec un contenu actualisé à une fréquence de trame effective qui varie en fonction du temps. Le décalage temporel dans le système augmentera et diminuera en fonction du temps et variera comme un retard de système moyen. Une séquence vidéo traitée contenant un saut d'image variable sera plus ou moins de même durée que la séquence vidéo d'origine associée.

4 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

ACR évaluation par catégories absolues (voir la Recommandation UIT-T P.910) (*absolute category rating*)

ACR-HR évaluation par catégories absolues avec référence dissimulée (voir la Recommandation UIT-T P.910) (*absolute category rating with hidden reference*)

AVI entrelacement audio vidéo (*audio video interleave*)

DMOS note d'opinion moyenne de dégradation (*difference mean opinion score*)

FR image de référence complète (*full reference*)

FRTV télévision avec image de référence complète (*full reference television*)

HRC circuit de référence hypothétique (*hypothetical reference circuit*)

NR absence de signal de référence (*ou signal de référence nul*) (*no (or zero) reference*)

PSNR rapport signal-bruit de crête (*peak signal-to-noise ratio*)

PVS séquence vidéo traitée (*processed video sequence*)

RMSE erreur quadratique moyenne (*root mean square error*)

RR référence réduite (*reduced reference*)

SFR fréquence de trame à la source (*source frame rate*)

SRC canal ou circuit de référence à la source (*source reference channel or circuit*)

VQEG groupe d'experts sur la qualité vidéo (*video quality experts group*)

YUV espace de couleurs (*colour space*)

5 Conventions

Néant.

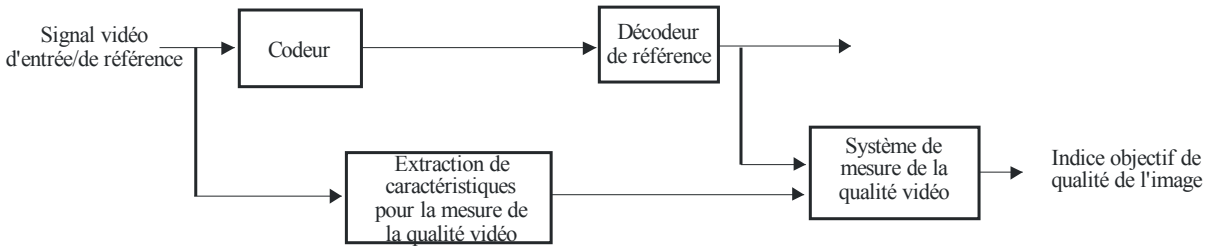
6 Description de la méthode de mesure avec référence réduite

La méthode de mesure aux deux extrémités avec référence complète, servant à mesurer de façon objective la qualité vidéo perçue, permet d'évaluer la performance de systèmes en établissant une comparaison entre le signal vidéo d'entrée non distordu, ou de référence, à l'entrée du système et le signal dégradé à la sortie du système (Fig. 1).

La Fig. 1 montre un exemple d'application de la méthode avec référence complète pour tester un codec en laboratoire.

FIGURE 1

Application de la méthode de mesure de la qualité perçue avec référence réduite pour tester un codec en laboratoire



La comparaison entre le signal d'entrée et le signal de sortie peut nécessiter un processus d'alignement spatial et temporel pour compenser les éventuels déplacements d'image verticaux ou horizontaux ou les éventuels recadrages. Elle peut aussi nécessiter la correction des éventuels décalages et des éventuelles différences de gain dans les canaux de luminance et de chrominance. On calcule alors l'indice objectif de qualité de l'image, généralement en appliquant un modèle de perception de la vision humaine.

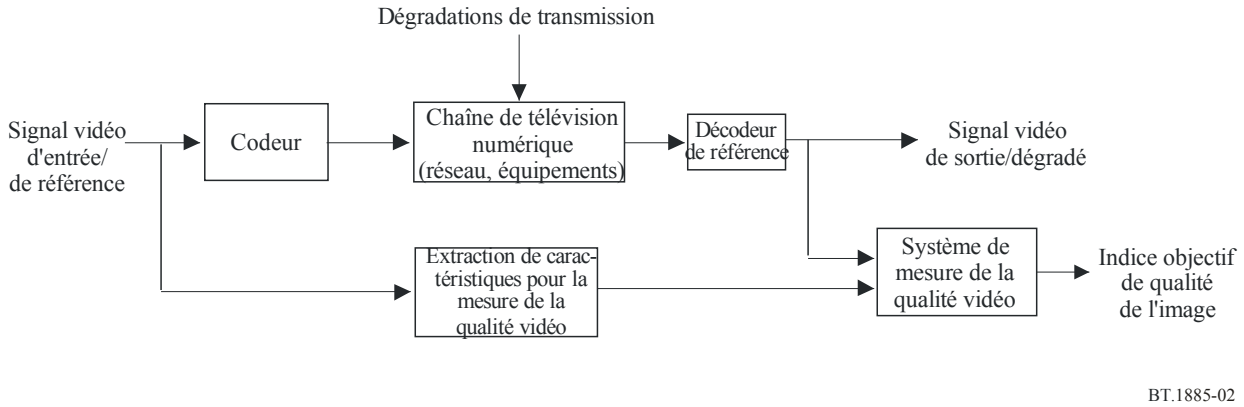
La normalisation désigne l'alignement et l'ajustement du gain. Cette opération est nécessaire puisque la plupart des méthodes avec référence complète compare les images traitées et les images de référence effectivement pixel par pixel. Le calcul de la valeur de crête du rapport signal sur bruit (PSNR, *peak signal to noise ratio*) en constitue un exemple. On élimine uniquement les changements statiques stationnaires de la vidéo, alors que les changements dynamiques dus aux processus de compression et de décompression sont mesurés dans le cadre du calcul de l'évaluation de la qualité. Les Recommandations UIT-T J.244 et J.144 fournissent des méthodes normalisées permettant d'indiquer les valeurs nécessaires à la normalisation du signal vidéo avant l'évaluation objective de la qualité. Les mesures de la qualité vidéo décrites dans l'Annexe de la présente Recommandation font état des méthodes de normalisation correspondantes. Le recours à d'autres méthodes de normalisation est possible en ce qui concerne les mesures de la qualité vidéo décrite en annexe, à condition qu'elles offrent la précision de normalisation requise.

Comme la mesure de la qualité vidéo est fondée sur un modèle de la vision humaine et non sur la mesure d'artéfacts de codage particuliers, elle est en principe valable aussi bien pour les systèmes analogiques que pour les systèmes numériques. Elle est aussi valable en principe pour les chaînes dans lesquelles des systèmes analogiques et des systèmes numériques sont mélangés ou dans lesquelles des systèmes de compression numérique sont concaténés.

La Fig. 2 montre un exemple d'application de la méthode avec référence réduite pour tester une chaîne de transmission.

FIGURE 2

Application de la méthode de mesure de la qualité perçue avec référence réduite pour tester une chaîne de transmission



Dans ce cas, un décodeur de référence est alimenté depuis divers points dans la chaîne de transmission; le décodeur peut par exemple être situé en un point du réseau comme sur la Fig. 2, ou directement à la sortie du codeur comme sur la Fig. 1. Si la chaîne de transmission numérique est transparente, la mesure de l'indice objectif de qualité de l'image à la source est égale à la mesure en n'importe quel point ultérieur dans la chaîne.

Il est généralement admis que la méthode avec référence réduite offre la meilleure précision en ce qui concerne les mesures de la qualité d'image perçue. La méthode s'est avérée pouvoir offrir une forte corrélation avec les évaluations subjectives faites conformément aux méthodes ACR-HR spécifiées dans la Rec. UIT-T P.910.

7 Conclusions du Groupe d'experts sur la qualité vidéo (VQEG)

Un groupe informel, le Groupe d'experts sur la qualité vidéo (VQEG, *video quality expert group*), fait des études sur les mesures de la qualité vidéo perçue et fait rapport aux Commissions d'études 9 et 12 de l'UIT-T et à la Commission d'études 6 de l'UIT-R. Le test RRNR-TV du VQEG récemment mené à bien a permis d'évaluer les performances des algorithmes proposés de mesure de la qualité vidéo perceptuelle en présence d'un signal de référence réduit pour les formats d'image de la Recommandation UIT-R 601-6.

Sur la base des données actuellement disponibles, six méthodes RR peuvent être recommandées actuellement par l'UIT-T, à savoir Model_A 15k, Model_A 80k, Model_A 256k, Model_C 80k, Model_B 80k (525 lignes seulement), Model_B 256k (525 lignes seulement).

Les descriptions techniques de ces modèles figurent aux Annexes A à C respectivement. Il convient de signaler que l'ordre des annexes est purement arbitraire et ne revêt aucun caractère indicatif quant aux performances escomptées.

Les Tableaux 1 et 2 représentent des test de signification dans le test RRNR-TV du VQEG. Pour le format à 525 lignes, quatre modèles (Model_A 15k, Model_A 80k, Model_A 256k, Model_C 80k) sont statistiquement meilleurs que le PSNR et deux modèles (Model_B 80k, Model_B 256k) sont statistiquement équivalents au PSNR. Il convient de noter que le PSNR a été calculé par la NTIA au moyen d'une recherche détaillée des limites d'étalonnage. Pour le format à 625 lignes, quatre modèles (Model_A 15k, Model_A 80k, Model_A 256k, Model_C 80k) sont statistiquement équivalents et sont statistiquement meilleurs que le PSNR.

TABLEAU 1

Test de signification pour le format à 525 lignes

Format à 525 lignes	Meilleure comparaison	Comparaison avec le PSNR	Corrélation
Model_A 15k	1	1	0,906
Model_A 80k	1	1	0,903
Model_A 256k	1	1	0,903
Model_C 80k	1	1	0,882
Model_B 80k	0	1	0,795
Model_B 256k	0	1	0,803
PSNR_NTIA	0	1	0,826

NOTE 1 – «1» dans «Meilleure comparaison» indique que ce modèle est statistiquement équivalent au modèle donnant les meilleurs résultats. «0» indique que ce modèle n'est pas statistiquement équivalent au modèle donnant les meilleurs résultats. «1» dans «Comparaison avec le PSNR» indique que ce modèle est statistiquement équivalent au modèle donnant les meilleurs résultats. «0» indique que ce modèle n'est pas statistiquement équivalent au modèle donnant les meilleurs résultats.

TABLEAU 2

Test de signification pour le format à 625 lignes

Format à 525 lignes	Meilleure comparaison	Comparaison avec le PSNR	Corrélation
Model_A 15k	1	1	0,894
Model_A 80k	1	1	0,899
Model_A 256k	1	1	0,898
Model_C 80k	1	1	0,866
PSNR_NTIA	0	1	0,857

NOTE 1 – «1» dans «Meilleure comparaison» indique que ce modèle est statistiquement équivalent au modèle donnant les meilleurs résultats. «0» indique que ce modèle n'est pas statistiquement équivalent au modèle donnant les meilleurs résultats. «1» dans «Comparaison avec le PSNR» indique que ce modèle est statistiquement équivalent au modèle donnant les meilleurs résultats. «0» indique que ce modèle n'est pas statistiquement équivalent au modèle donnant les meilleurs résultats.

Les Tableaux 3 et 4 donnent des renseignements détaillés sur les résultats obtenus avec les modèles dans le test RRNR-TV du VQEG.

TABLEAU 3

Description pour information des résultats du modèle avec le test RRNR-TV du groupe VQEG (format à 525 lignes)

Format à 525 lignes	Corrélation	RMSE	OR
Model_A 15k	0,906	0,418	0,385
Model_A 80k	0,903	0,423	0,378
Model_A 256k	0,903	0,424	0,378
Model_B 80k	0,795	0,598	0,667
Model_B 256k	0,803	0,587	0,647
Model_C 80k	0,882	0,465	0,513
PSNR_NTIA	0,826	0,556	0,571

TABLEAU 4

Description pour information des résultats du modèle avec le test RRNR-TV du groupe VQEG (Format à 625 lignes)

Format à 625 lignes	Corrélation	RMSE	OR
Model_A 15k	0,894	0,524	0,468
Model_A 80k	0,899	0,513	0,462
Model_A 256k	0,898	0,516	0,468
Model_C_80k	0,866	0,585	0,583
PSNR_NTIA	0,857	0,605	0,564

Annexe A

Modèle A: Méthode à référence réduite de l'Université de Yonsei

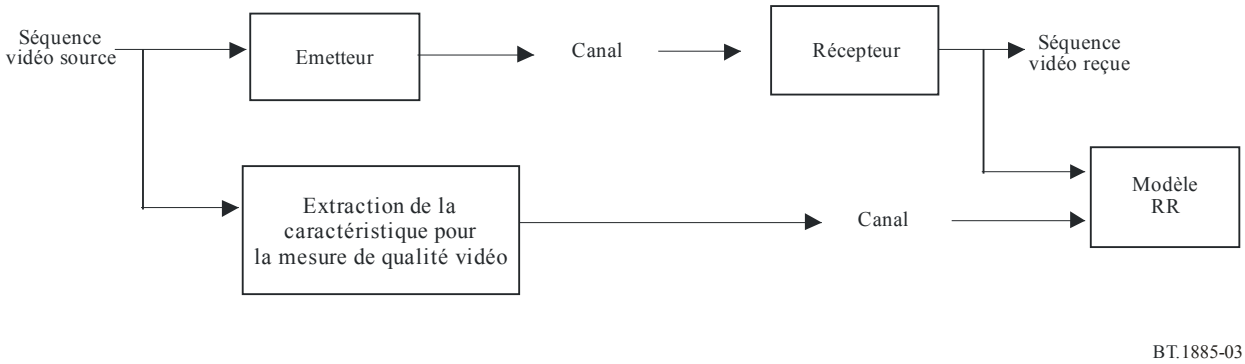
1 Introduction

Bien que le rapport PSNR soit largement utilisé pour la mesure de la qualité vidéo objective, on signale qu'il ne représente pas bien la qualité vidéo perceptuelle. En analysant comment l'être humain perçoit la qualité vidéo, on constate que le système visuel humain est sensible aux dégradations sur les contours. En d'autres termes, lorsque les pixels sur les contours d'une séquence vidéo sont flous, les évaluateurs tendent à donner des notes faibles à la séquence vidéo même si le rapport PSNR est élevé. A partir de cette observation, on a mis au point des modèles à référence réduite qui mesurent principalement les dégradations sur les contours.

La Fig. 3 illustre la façon dont le modèle à référence réduite fonctionne. Les caractéristiques qui seront utilisées pour mesurer la qualité vidéo au point de mesure sont extraites de la séquence vidéo source et transmises. Le Tableau 5 montre les largeurs de bande d'un canal latéral pour les caractéristiques, qui ont été testées lors du test RRNR-TV du VQEG.

FIGURE 3

Schéma synoptique d'un modèle à référence réduite



BT.1885-03

TABLEAU 5

Largeurs de bande du canal latéral

Format vidéo	Largeurs de bande testées
Format à 525 lignes	15 kbit/s, 80 kbit/s, 256 kbit/s
Format à 625 lignes	15 kbit/s, 80 kbit/s, 256 kbit/s

2 Rapport EPSNR pour les modèles à référence réduite

2.1 Rapport PNSR pour les contours

Les modèles à référence réduite (RR) permettent de mesurer principalement les dégradations sur les contours. Dans ces modèles, un algorithme de détection des contours est d'abord appliqué à la séquence vidéo source pour localiser les pixels des contours. Puis, la dégradation de ces pixels des contours est mesurée en calculant l'erreur quadratique moyenne. A partir de cette erreur quadratique moyenne, on calcule le rapport EPNSR (EPNSR: Edge PNSR) (PNSR des contours).

On peut utiliser tout algorithme de détection des contours, bien que ces algorithmes puissent donner des différences mineures dans les résultats. Par exemple, on peut utiliser un opérateur gradient pour localiser les contours. Un certain nombre d'opérateurs gradients ont été proposés. Dans de nombreux algorithmes de détection des contours, l'image en gradients horizontaux $g_{horizontal}(m,n)$ et l'image en gradients verticaux $g_{vertical}(m,n)$ ont d'abord été calculées au moyen d'opérateurs gradients. On peut alors calculer la valeur de l'image en gradients de valeurs g(m,n) comme suit:

$$g(m,n) = \left|g_{horizontal}(m,n)\right| + \left|g_{vertical}(m,n)\right|$$

Enfin, on applique une opération de seuillage à l'image en gradient de valeurs $g(m,n)$ pour trouver les pixels des contours. En d'autres termes, les pixels dont les gradients de valeur sont supérieurs à une valeur seuil sont considérés comme étant des pixels de contour.

Les Fig. 4 à 8 illustrent la procédure. La Fig. 4 montre une image source. La Fig. 5 montre une image en gradients horizontaux $g_{horizontal}(m,n)$, qui est obtenue en appliquant un opérateur gradient horizontal à l'image source de la Fig. 4. La Fig. 6 montre une image en gradients verticaux $g_{vertical}(m,n)$, qui est obtenue en appliquant un opérateur gradient vertical à l'image source de la Fig. 4. La Fig. 7 montre l'image en gradients de valeurs (image des contours) et la Fig. 8 montre une image binaire (image de masque) des contours obtenue en appliquant un seuillage à l'image en gradients de valeurs de la Fig. 7.

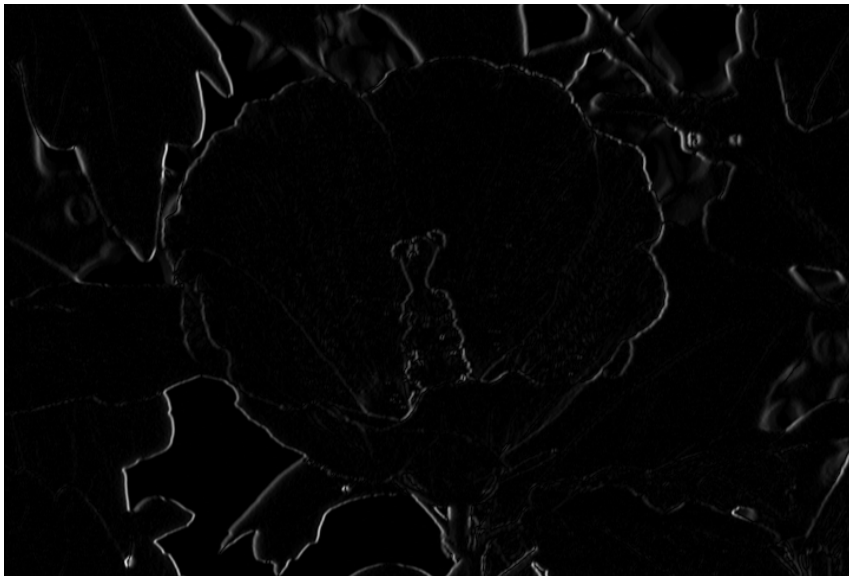
FIGURE 4

Image source (image originale)

BT.1885-04

FIGURE 5

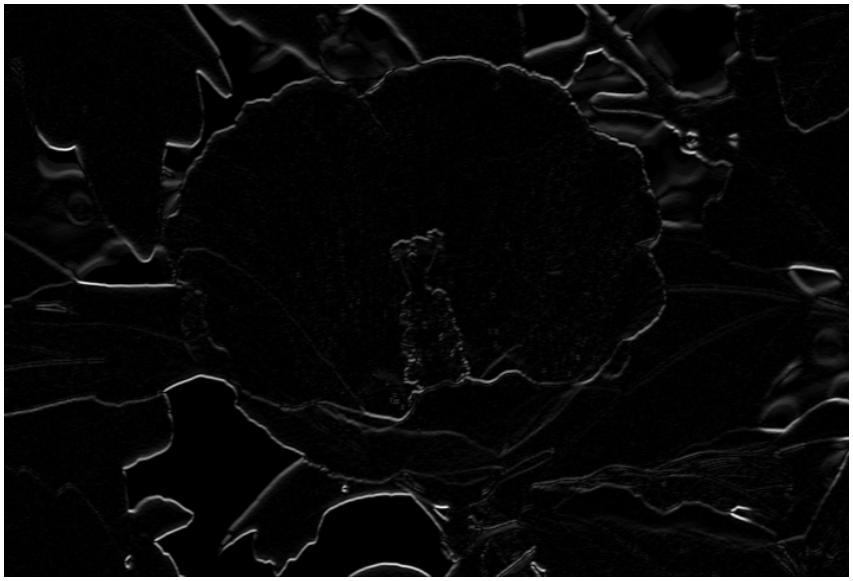
Image en gradients horizontaux obtenue en appliquant un opérateur gradient horizontal à l'image source de la Fig. 4



BT.1885-05

FIGURE 6

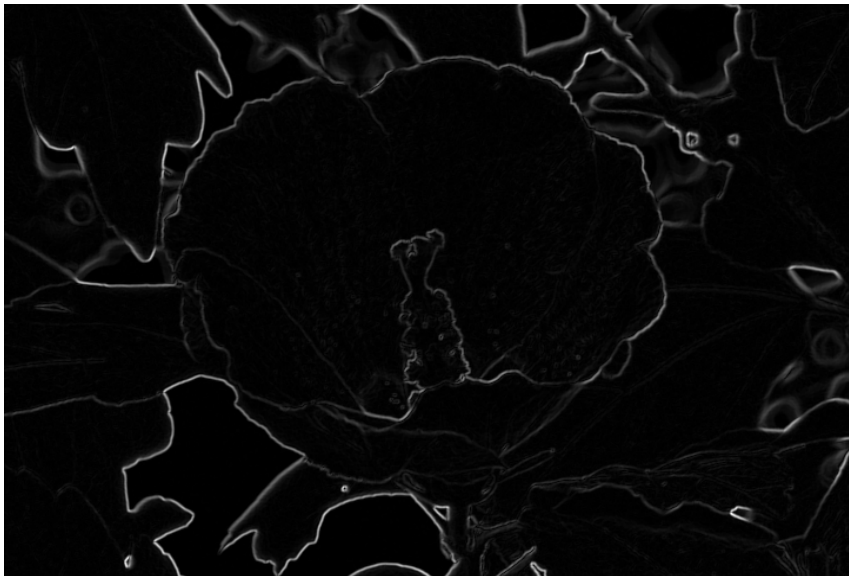
Image en gradients verticaux obtenue en appliquant un opérateur gradient vertical à l'image source de la Fig. 4



BT.1885-06

FIGURE 7

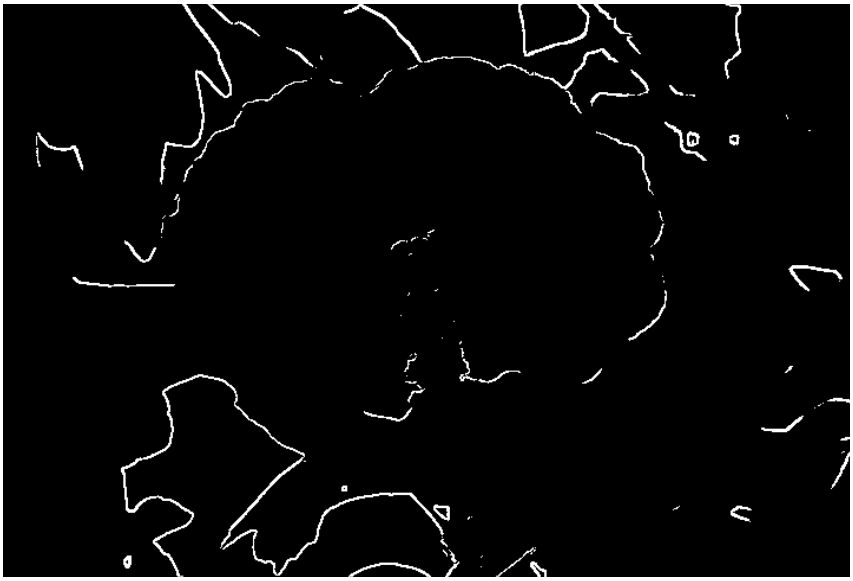
Image en gradients de valeur



BT.1885-07

FIGURE 8

Image à contours binaire (image de masque) obtenue en appliquant un seuillage à l'image en gradients de valeurs de la Fig. 7

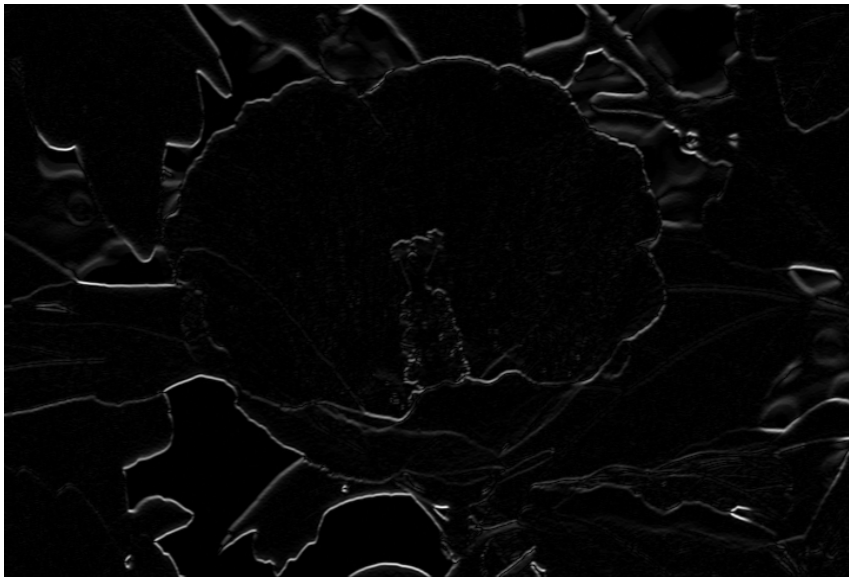


BT.1885-08

On peut également utiliser une procédure modifiée pour localiser les pixels de contour. Par exemple, on peut tout d'abord appliquer un opérateur gradient vertical à l'image source, ce qui donne une image en gradients verticaux. On applique ensuite un opérateur gradient horizontal à l'image en gradients verticaux qui donne une image en gradients successifs (image en gradients horizontaux et en gradients verticaux). On peut alors appliquer un seuillage à l'image en gradients successifs pour trouver les pixels des contours. En d'autres termes, les pixels de l'image en gradients successifs modifiée qui dépassent une valeur seuil sont considérés comme pixels de contour. Les Fig. 9 à 12 illustrent la procédure modifiée. La Fig. 9 montre une image en gradients verticaux $g_{vertical}(m,n)$, laquelle est obtenue par application d'un opérateur gradient vertical à l'image source de la Fig. 4. La Fig. 10 montre une image en gradients successifs modifiée (image en gradients horizontaux et en gradients verticaux), laquelle est obtenue par application d'un opérateur gradient horizontal à l'image en gradients verticaux de la Fig. 9. La Fig. 11 montre l'image binaire des contours (image masque) obtenue en appliquant un seuillage à l'image en gradients successifs modifié de la Fig. 10.

FIGURE 9

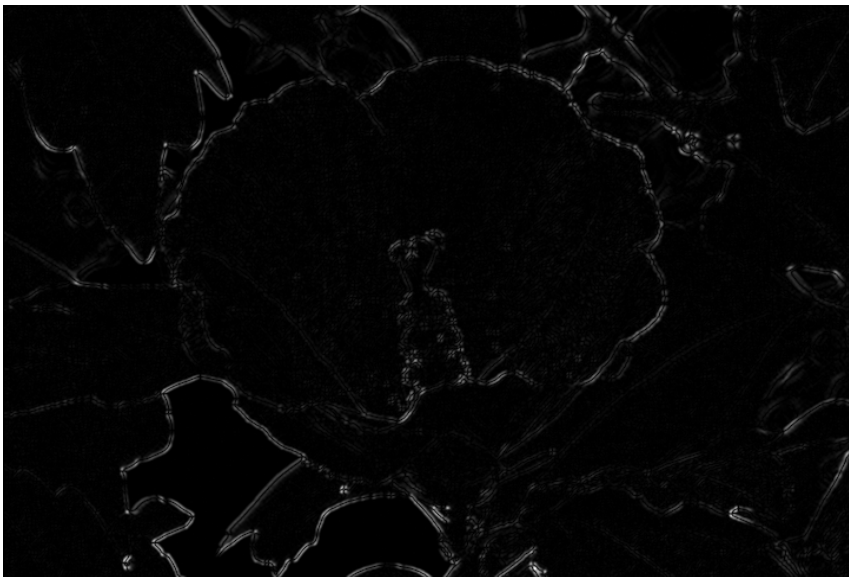
Image en gradients verticaux obtenue en appliquant un opérateur gradient vertical à l'image source de la Fig. 4



BT.1885-09

FIGURE 10

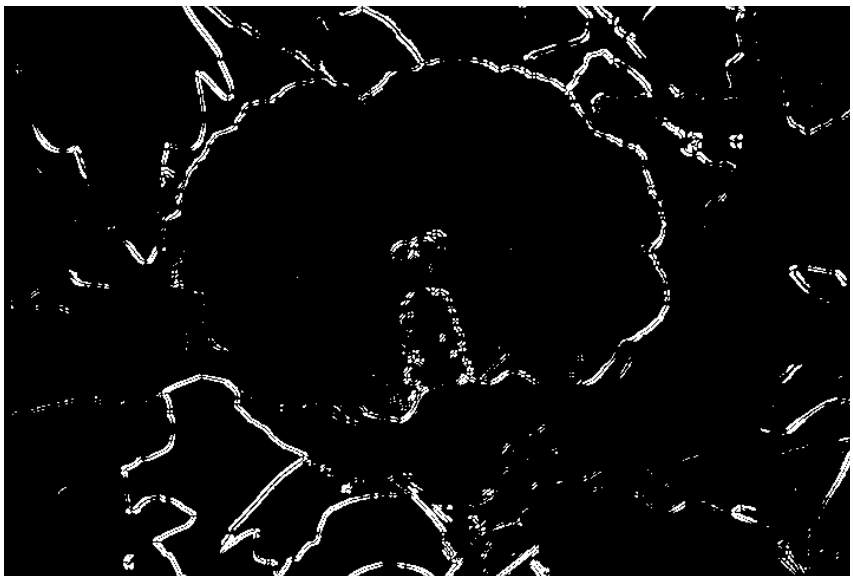
Image en gradients successifs (image du gradient horizontal et du gradient vertical) obtenue en appliquant un opérateur gradient horizontal à l'image du gradient horizontal de la Fig. 9



BT.1885-10

FIGURE 11

Image à contours binaire (image de masque) obtenue en appliquant un seuillage à l'image en gradients successifs modifiée de la Fig. 10



BT.1885-11

Il est à noter que les deux méthodes peuvent être considérées comme étant des algorithmes de détection des contours. On peut choisir tout algorithme de détection des contours selon la nature des séquences vidéo et les algorithmes de compression. Toutefois, certaines méthodes peuvent donner de meilleurs résultats que d'autres.

Ainsi, dans le modèle, on applique tout d'abord un opérateur de détection des contours qui donne des images des contours (Fig. 7 et 10). Puis, on produit une image de masque (image binaire des contours) en appliquant un seuillage à l'image des contours (Fig. 8 et 11). En d'autres termes, les pixels de l'image des contours dont la valeur est inférieure au seuil t_e sont mis à zéro et les pixels dont la valeur est égale ou supérieure au seuil sont mis à une valeur non nulle. Les Fig. 8 et 11 montrent certaines images de masque. Etant donné qu'une séquence vidéo peut être considérée comme une séquence de trames ou d'images, la procédure précitée peut être appliquée à chaque trame ou image des séquences vidéo. Puisque le modèle peut être utilisé pour les séquences vidéo à trames ou à images, le terme «image» sera utilisé pour désigner indifféremment une image ou une trame.

2.2 Choix des caractéristiques à partir des séquences vidéo source

Etant donné que le modèle est un modèle à référence réduite (RR), un ensemble de caractéristiques peut être extrait de chaque image d'une séquence vidéo source. Dans le modèle EPSNR RR, un certain nombre de pixels de contour est choisi sur chaque image. Les positions et les valeurs des pixels de contour sont ensuite codées et transmises. Toutefois, pour certaines séquences vidéo, le nombre de pixels de contour peut être très faible lorsqu'on utilise une valeur seuil fixe. Dans le scénario le plus défavorable, ce nombre peut être nul (images supprimées ou images à cadence très faible). Afin de traiter du problème, si le nombre de pixels de contour d'une image est inférieur à une valeur donnée, l'utilisateur peut abaisser le seuil jusqu'à ce que le nombre de pixels de contour soit supérieur à une certaine valeur. On peut aussi choisir les pixels de contour qui correspondent aux plus grandes valeurs de l'image en gradients horizontaux et en gradients verticaux . Lorsqu'il n'y a pas de pixels d'image (image supprimée par exemple) dans une trame, on peut choisir aléatoirement le nombre de pixels requis ou sauter la trame. Par exemple, si 10 pixels de contour doivent être extraits de chaque trame, on peut trier les pixels de l'image en gradients horizontaux et

en gradients verticaux selon leurs valeurs et retenir les 10 plus grandes valeurs. Toutefois, cette procédure peut produire plusieurs pixels de contour sur des positions identiques. Pour traiter ce problème, on peut d'abord choisir plusieurs fois le nombre souhaité de pixels de l'image en gradients horizontaux et verticaux et choisir ensuite aléatoirement le nombre souhaité de pixels parmi les pixels de l'image en gradients horizontaux et en gradients verticaux. Dans les modèles testés lors du test multimédia VQEG, le nombre de pixels souhaité est choisi aléatoirement parmi un grand pool de pixels de contour. Le pool des pixels de contour est obtenu en appliquant une opération de seuillage de l'image en gradients.

Dans les modèles EPSNR RR, les positions et les valeurs des pixels de contour sont codés, après application d'une filtre passe-bas gaussien aux emplacements de pixels choisis. Bien que l'on ait utilisé le filtre LPF gaussien (5 × 3) dans le teste RRNR-TV du VQEG, on peut utiliser différents filtres passe-bas selon le format vidéo. Il est à noter que pendant le processus de codage, on peut appliquer un recadrage. Afin d'éviter le choix des pixels de contour dans les zones recadrées, le modèle choisit les pixels de contour dans la zone centrale (Fig. 12). Le Tableau 6 montre les dimensions après recadrage. Ce tableau montre aussi le nombre de bits nécessaire pour coder la position et la valeur de pixel d'un pixel de contour.

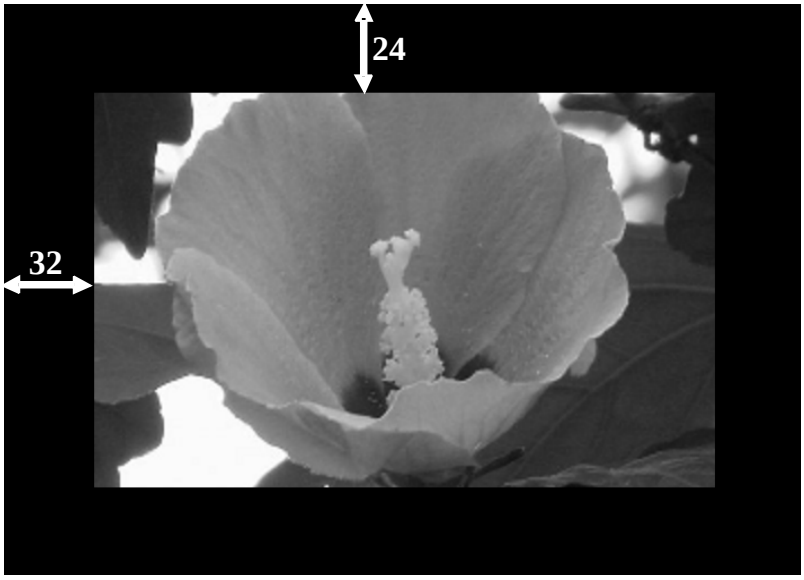
TABLEAU 6

Bits nécessaires par pixel de contour

Format vidéo	Dimensions	Dimensions après recadrage	Bits de position	Bits de valeur de pixel	Nbre total de bits par pixel
525	720 × 486	656 × 438	19	8	27
625	720 × 576	656 × 528	19	8	27

FIGURE 12

Exemple de recadrage (VGA) et zone centrale



BT.1885-12

Le modèle choisit les pixels de contour dans chaque trame conformément à la largeur de bande autorisée (Tableau 5). Le Tableau 7 montre le nombre de pixels de contour par trame qui peut être transmis pour les largeurs de bande testées.

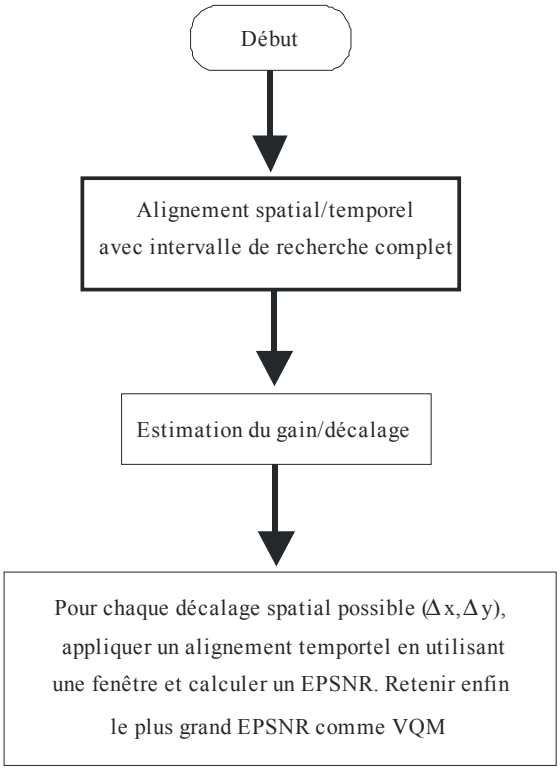
TABLEAU 7

Nombre de pixels de contour par trame

Format vidéo	15 kbit/s	80 kbit/s	256 kbit/s
525	16	74	238
625	20	92	286

FIGURE 13

Organigramme du modèle



BT.1885-13

2.3 Alignement spatial/temporel et réglage du gain/décalage

Avant de calculer la différence entre les pixels de contour de la séquence vidéo source et ceux de la séquence vidéo traitée qui est la séquence vidéo reçue sur le récepteur, le modèle procède d'abord à un alignement spatial/temporel et à un réglage du gain/décalage. On a utilisé la méthode d'étalonnage (Annexe B) de la Recommandation UIT-T J.244. Pour transmettre les caractéristiques de gain et de décalage de cette Recommandation (Annexe B), on a utilisé 30% de la largeur de bande disponible dans le test RRNR-TV du VQEG. Etant donné que la séquence vidéo est entrelacée, on applique la méthode d'étalonnage trois fois: trames paires, trames impaires et trames combinées. Si la différence entre l'erreur de trame paire (PSNR) et l'erreur de trame impaire est supérieure à un seuil, les résultats de l'alignement (x-shift, y-shift) ayant la plus petite valeur de

PSNR ont été utilisés. Dans le cas contraire, on a utilisé les résultats de l'alignement avec les trames combinées. Dans le test RRNR-TV du VQEG, le seuil a été fixé à 2 dB.

Au point de monitorage, la séquence vidéo traitée doit être ajustée avec les pixels de contour extraits de la séquence vidéo source. Toutefois, si la largeur de bande des canaux latéraux est petite, seuls quelques pixels de contour de la séquence vidéo source sont disponibles (Fig. 14). En conséquence, l'alignement temporel peut être faussé si l'alignement temporel est effectué en utilisant une seule trame (Fig. 15). Pour résoudre ce problème, le modèle fait appel à une fenêtre pour l'alignement temporel. Au lieu d'utiliser une seule trame de la séquence vidéo traitée, le modèle construit une fenêtre qui est composée de trames adjacentes pour trouver le décalage temporel optimal. La Fig. 16 illustre la procédure. L'erreur quadratique moyenne dans la fenêtre est calculée comme suit:

$$MSE_{window} = \frac{1}{N_{win}} \sum \left(E_{SRC}(i) - E_{PVS}(i) \right)^2$$

où:

MSE_{window} : l'erreur quadratique moyenne dans la fenêtre

$E_{SRC}(i)$: un pixel de contour à l'intérieur de la fenêtre qui a un pixel correspondant dans la séquence vidéo traitée

$E_{PVS}(i)$: un pixel de la séquence vidéo traitée correspondant au pixel de contour

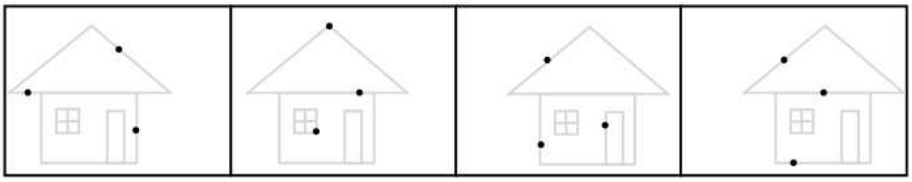
N_{win} : le nombre total de pixels de contour utilisé pour calculer MSE_{window}.

L'erreur quadratique moyenne est utilisée comme différence entre une trame de la séquence vidéo traitée et la trame de la séquence vidéo source correspondante.

La longueur de la fenêtre vidéo peut être déterminée en prenant en considération de la nature de la séquence vidéo traitée. Pour une application type, une fenêtre correspondant à deux secondes est recommandée. On peut aussi appliquer différentes longueurs de fenêtre et utiliser la meilleure de celles-ci, c'est à dire celle qui donne la plus faible erreur quadratique moyenne. En outre, on peut utiliser différents centres de fenêtre pour tenir compte du saut de trames dû à des erreurs de transmission (Fig. 20).

FIGURE 14

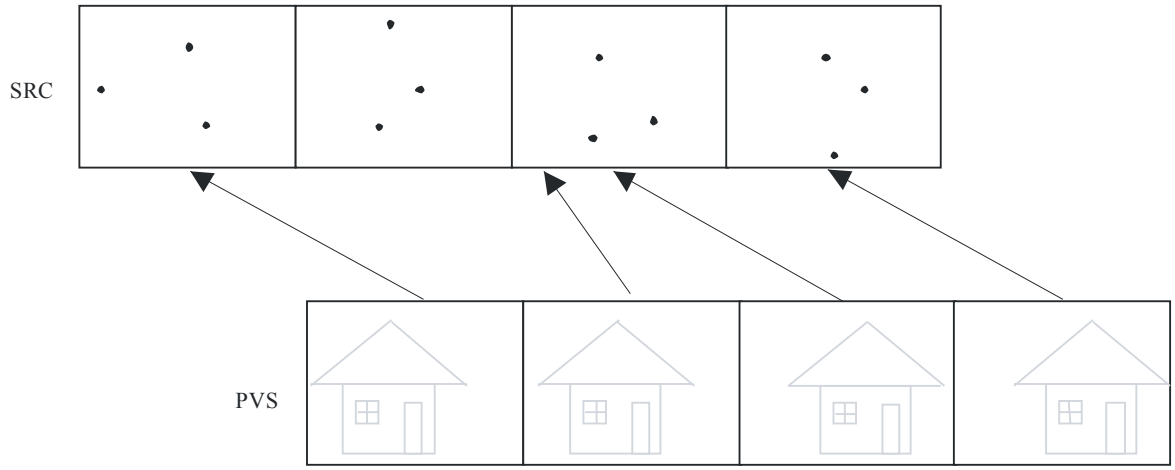
Choix des pixels de contour dans une séquence vidéo source



BT.1885-14

FIGURE 15

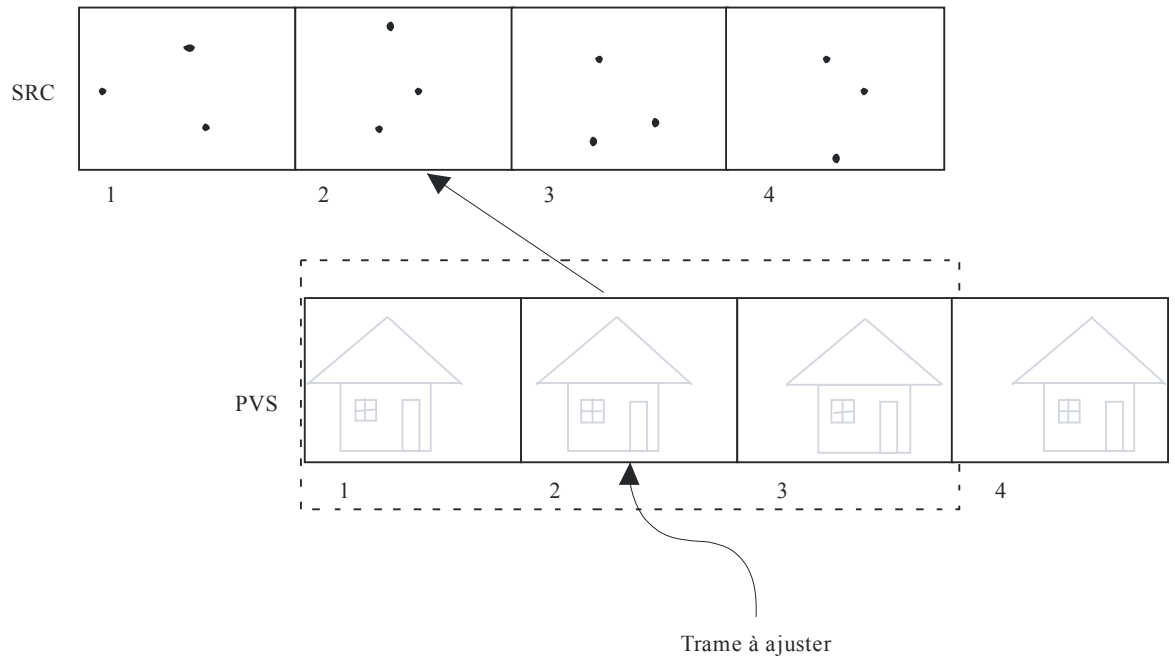
Ajustement de la séquence vidéo traitée sur les pixels de contour de la séquence vidéo source



BT.1885-15

FIGURE 16

Ajustement de la séquence vidéo traitée sur les pixels de contour en utilisant une fenêtre



BT.1885-16

Lorsque la séquence vidéo source est codée avec des taux de compression élevés, le codeur doit réduire le nombre de trames par seconde et la séquence vidéo traitée comporte des trames répétées (Fig. 17). Dans la Fig. 17, la séquence vidéo traitée ne comporte pas de trames correspondant à certaines trames de la séquence vidéo source (2ème, 4ème, 6ème et 8ème trame). Dans ce cas, le modèle n'utilise pas les trames répétées dans le calcul de l'erreur quadratique moyenne. En d'autres termes, le modèle effectue un alignement temporel en utilisant la première trame (trame valide) de chaque bloc répété. Ainsi, sur la Fig.18, seules trois trames (3ème, 5ème et 6ème trame) dans la fenêtre sont utilisées pour l'alignement temporel.

FIGURE 17

Exemple de trames répétées

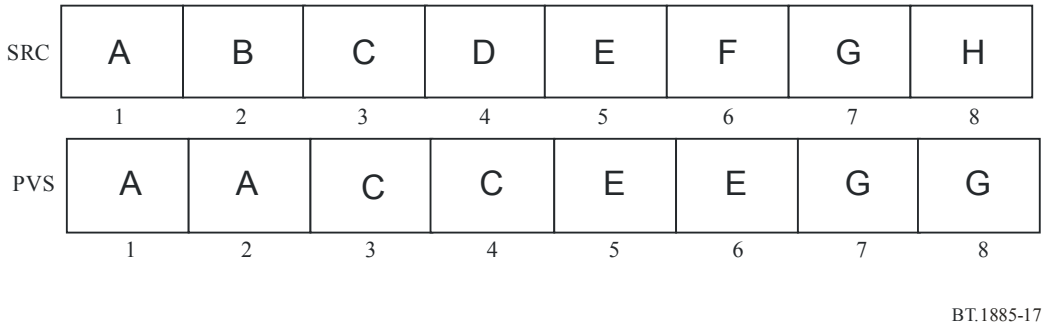
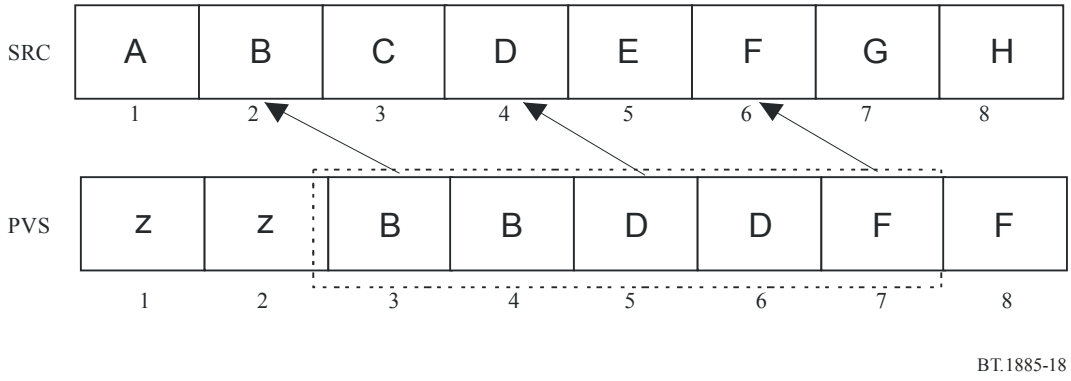


FIGURE 18

Traitement de trames répétées



Il est possible d'avoir une séquence vidéo traitée avec une répétition irrégulière de trames, ce qui peut aboutir à ce que la méthode d'alignement temporel utilisant une fenêtre, donne des résultats erronés. Pour résoudre ce problème, il est possible d'ajuster localement chaque trame de la fenêtre dans les limites d'une certaine valeur (±1 par exemple) comme le montre la Fig. 21 après l'alignement temporel au moyen d'une fenêtre. Puis, l'ajustement local qui donne la plus petite erreur quadratique moyenne est utilisé pour calculer l'EPSNR.

FIGURE 19

Fenêtre de longueurs diverses

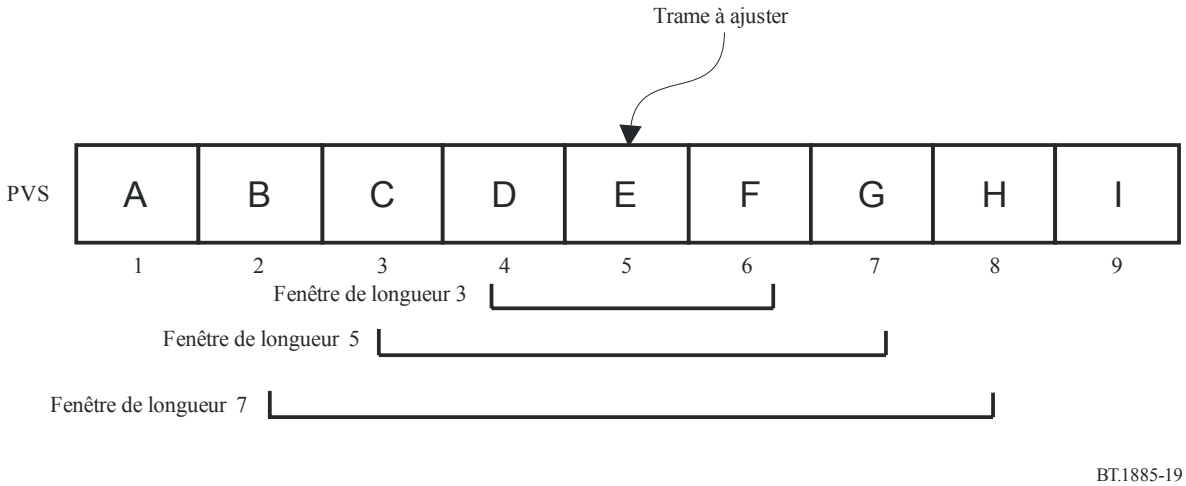


FIGURE 20

Centre des fenêtres

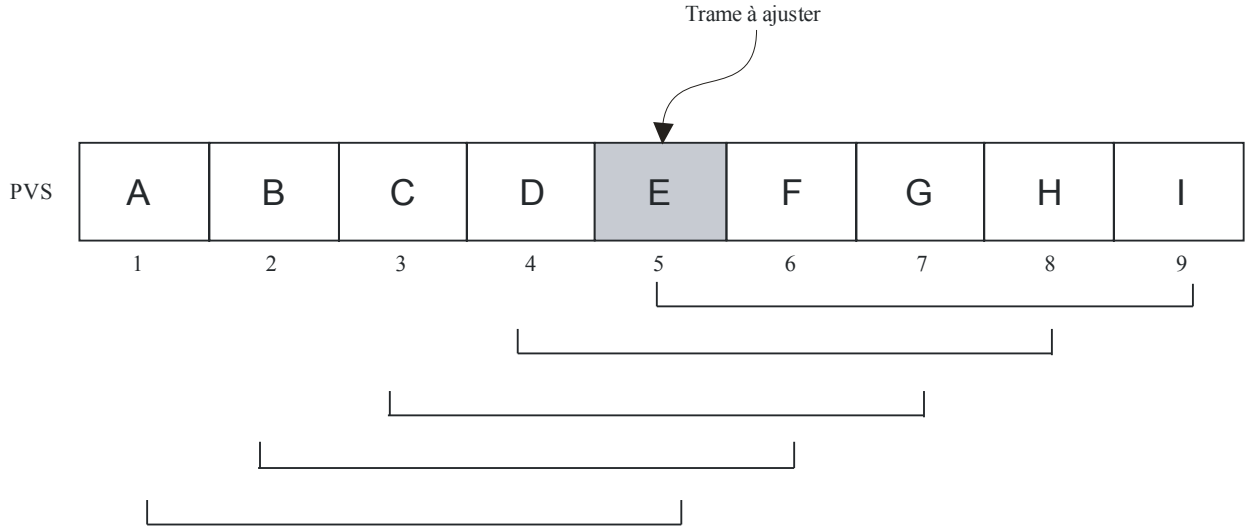
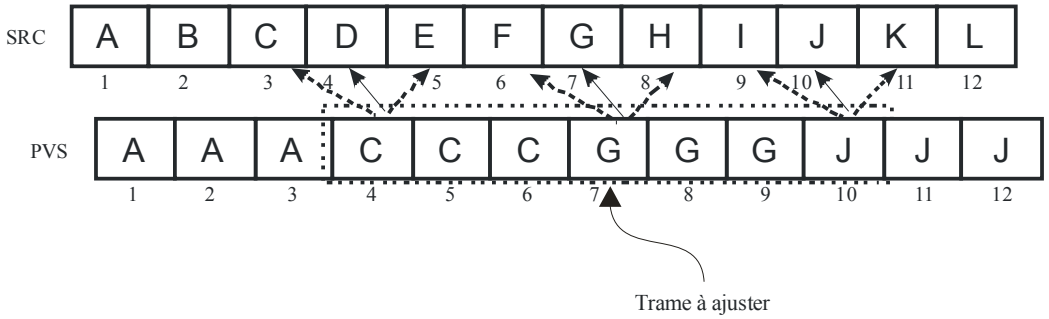


FIGURE 21

Ajustement local pour l'alignement temporel en utilisant une fenêtre



2.4 Calcul de l'EPSNR et post-traitement

Après l'alignement temporel, on calcule la moyenne des différences entre les pixels de contour de la séquence vidéo source et les pixels correspondants de la séquence vidéo traitée, différence qui peut être interprétée comme étant l'erreur quadratique moyenne des contours de la séquence vidéo traitée (MSE_{edge}). On calcule enfin l'EPSNR (PSNR des contours) comme suit:

$$EPSNR = 10 \log_{10}\left(\frac{P^2}{MSE_{edge}}\right)$$

où p est la valeur crête de l'image.

1) Trames gelées

Il peut y avoir une répétition de trames causée par des cadences de trame réduites et un gel de trame dus à des erreurs de transmission, ce qui entraîne une dégradation de la qualité vidéo perceptuelle. Afin de remédier à cet effet, le modèle applique l'ajustage suivant avant de calculer l'EPSNR:

$$MSE_{freezed\_frame\_considered} = MSE_{edge} \times \frac{K \times N_{total\_frame}}{N_{total\_frame} - N_{total\_freezed\_frame}}$$

où:

$MSE_{freezed\_frame\_considered}$: l'erreur quadratique moyenne qui tient compte des trames répétées et gelées

N_{total_frame} : le nombre total de trames, $N_{total\_freezed\_frame}$

K: une constante.

Dans le modèle testé lors du test multimédia VQEG, K a été fixé à 1.

2) Hautes fréquences et mouvements rapides

Si la séquence vidéo contient un grand nombre de hautes fréquences et de mouvements rapides, la qualité perceptuelle tend à augmenter pour le même MSE. Pour tenir compte de cette conséquence, la différence de trame normalisée (NFD) et l'énergie haute fréquence normalisée (NHFE) sont définies ci-après:

$$NFD = \frac{FD}{average\ energy\ \ per\ \ pixel}$$

où: $FD = \frac{1}{N_F} \sum_i \sum_{k=1}^{height} \sum_{j=1}^{width} (Frame_i[j,k] - Frame_{i-1}[j,k])^2$ et N_F est le nombre de trames utilisées dans la sommation. Il convient de noter que les trois différences de trames les plus importantes sont exclues du calcul de FD, afin d'exclure les changements de scène lors du calcul de la différence de trame moyenne, en prenant pour hypothèse des séquences vidéo de 8 secondes. On calcule l'énergie haute fréquence normalisée (NHFE) en calculant la moyenne des énergies hautes fréquences (voir la Fig. 22) après avoir appliqué la transformée de Fourier à deux dimensions:

$$NHFE = \frac{average\ \ high\ \ frequency\ energies}{average\ energy\ \ per\ \ pixel}$$

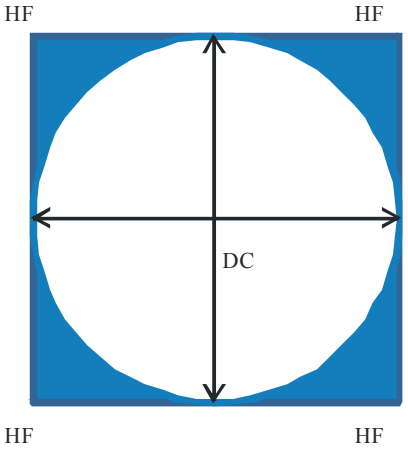
Enfin, on utilise les formules suivantes:

```
IF(SNFD > 0.35 && SNHFE > 2.5)  {
                    IF(EPSNR < 20) EPSNR = EPSNR+3
                    ELSE IF(EPSNR < 35) EPSNR = EPSNR+5
}
ELSE IF((SNFD > 0.2 && SNHFE > 1.5) || (SNFD>0.27) && SNHFE > 1.3))       {
      I             F(28 < EPSNR < 40) EPSNR = EPSNR + 3
                    IF(EPSNR > 40) EPSNR = 40
}
```

où SNFD est la NFD source et SNHFE est la NHFE source. Il convient de noter que l'on calcule les SNFD et SNHFE à partir du SRC et qu'on les transmet en tant que données caractéristiques (1 octet chacune).

FIGURE 22

Calcul de l'énergie haute fréquence normalisée (NHEE). Les énergies hautes fréquence sont calculées à partir des données ombrées



BT.1885-22

3) Flou

Pour tenir compte des effets du flou, on utilise les formules suivantes:

```
IF (NHFE/SNHFE < 0.5)
                IF(EPSNR>26)      EPSNR = 26
ELSE IF (NHFE/SNHFE < 0.6)
                IF(EPSNR>32)      EPSNR = 32
ELSE IF (NHFE/SNHFE < 0.7)
                IF(EPSNR>36)      EPSNR = 36
ELSE IF (NHFE/SNHFE > 1.2)
                IF(EPSNR>23)      EPSNR = 23
ELSE IF (NHFE/SNHFE > 1.1)
                IF (EPSNR>25)     EPSNR = 25
```

où NHFE est la NHFE de la PVS.

4) Subdivision en blocs

Pour tenir compte des effets de la subdivision en blocs, on calcule les différences moyennes entre les colonnes. En prenant pour hypothèse le modulo 8, la note de subdivision en blocs pour la énième trame est calculée de la façon suivante:

$$Blk[i] = \frac{largest\ column\ difference}{second\ largest\ column\ difference}$$

La note finale de subdivision en blocs (*blocking*) est calculée en faisant la moyenne des notes de subdivision en blocs des trames:

$$Blocking = \frac{1}{number\ of\ \ frames} \sum_{i} Blk[i]$$

Enfin, on utilise les formules suivantes:

```
IF(BLOCKING > 1.4)      {
                IF (20≤EPSNR<25) EPSNR = EPSNR-1.086094*BLOCKING-0.601316
                ELSE IF (EPSNR<30) EPSNR = EPSNR-0.577891*BLOCKING-3.158586
                ELSE IF (EPSNR<35) EPSNR = EPSNR-0.223573*BLOCKING-3.125441
}
```

5) Nombre maximal de trames gelées

Les erreurs de transmission peuvent avoir pour conséquence des trames gelées de longue durée. Pour tenir compte de ce phénomène, on utilise les formules suivantes:

```
IF(MAX_FREEZE > 22 AND EPSNR>28) EPSNR = 28
ELSE IF(MAX_FREEZE > 10 AND EPSNR>34) EPSNR = 34
```

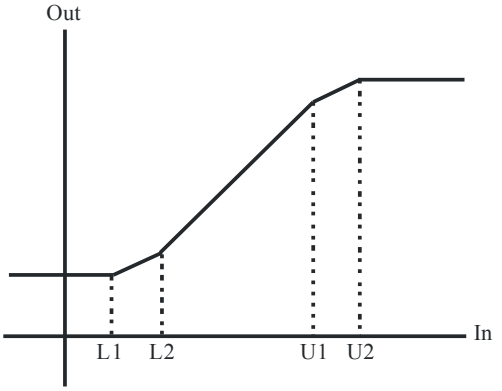
où MAX_FREEZE est la trame gelée de plus grande durée. Il convient de noter que si la séquence vidéo n'est pas de 8 s, il conviendra d'utiliser des seuils différents.

6) Mise à l'échelle linéaire par paliers

Lorsque l'EPSNR est supérieur à une certaine valeur, la qualité vidéo perçue devient saturée. Dans ce cas, il est possible de fixer la limite supérieure de l'EPSNR. De plus, lorsqu'une relation linéaire entre l'EPSNR et la note d'opinion moyenne différentielle (DMOS, *difference mean opinion score*) est souhaitable, on peut appliquer une fonction linéaire par paliers comme le montre la Fig. 23. Dans le modèle testé lors du test RRNR-TV du VQEG multimédia, la limite supérieure a été fixée à 48 et la limite inférieure a été fixée à 15.

FIGURE 23

Fonction linéaire par paliers pour la relation linéaire entre l'EPSNR et la DMOS



BT.1885-23

Les modèles d'EPSNR à référence réduite pour la mesure objective de la qualité vidéo sont fondés sur la dégradation des contours. Ces modèles peuvent être mis en œuvre en temps réel avec utilisation modérée de la puissance de calcul. Les modèles sont bien adaptés aux applications qui nécessitent un monitorage en temps réel de la qualité vidéo lorsque des canaux latéraux sont disponibles.

Annexe B

Modèle B: Méthode de référence réduite de la NEC

On trouvera dans la présente annexe une description fonctionnelle complète du modèle RR. Selon ce modèle, on transmet côté client les valeurs d'activité au lieu des valeurs de pixel pour différents blocs de pixel d'une taille donnée. On évalue la qualité vidéo sur la base de la différence d'activité entre le canal de référence source (SRC) et la séquence vidéo traitée (PVS). Les pondérations psychovisuelles en ce qui concerne la différence d'activité visent à améliorer la précision de l'évaluation.

Ce modèle nécessite très peu de calculs de l'alignement spatial et de l'alignement du gain et du décalage. En outre, il peut être mis en oeuvre par un programme à 30 lignes et un programme à 250 lignes côté serveur et coté client respectivement. En conséquence, il se prête bien à la surveillance de la qualité vidéo en temps réel dans les services de radiodiffusion qui bénéficient le plus d'une complexité réduite et d'une mise en oeuvre facilitée.

1 Résumé

Selon le modèle RR, on transmet côté client les valeurs d'activité au lieu des valeurs de pixel pour différents blocs de pixels d'une taille donnée. On évalue la qualité vidéo sur la base de la différence d'activité entre le canal de référence source (SRC) et la séquence vidéo traitée (PVS). Les pondérations psychovisuelles en ce qui concerne la différence d'activité visent à améliorer la précision de l'évaluation.

Ce modèle nécessite très peu de calculs de l'alignement spatial et de l'alignement du gain et du décalage. En outre, il peut être mis en oeuvre par un programme à 30 lignes et un programme à 250 lignes côté serveur et côté client respectivement. En conséquence, il se prête bien à la surveillance de la qualité vidéo en temps réel dans les services de radiodiffusion qui bénéficient le plus d'une complexité réduite et d'une mise en oeuvre facilitée.

2 Définitions

Activité: valeur moyenne de la différence absolue entre chaque valeur de luminance et la moyenne des valeurs de luminance pour un bloc d'une taille donnée.

Bloc: ensemble de pixels M x N (M-colonne par N-rangée)

Image: une image de télévision complète.

Gain: facteur multiplicatif appliqué par le circuit fictif de référence (HRC) à tous les pixels d'un plan d'image donné (par exemple luminance, chrominance). Le gain du signal de luminance est généralement appelé contraste.

Circuit fictif de référence (HRC, *hypothetical reference circuit*): système vidéo testé, par exemple un codec ou un système de transmission vidéonumérique.

Luminance (Y): partie du signal vidéo qui achemine avant tout l'information de luminance (c'est-à-dire la partie en noir et blanc de l'image).

Système NTSC (*national television systems committee*): système couleur de vidéo composite analogique à 525 lignes [1].

Décalage ou décalage de niveau: facteur additif appliqué par le circuit fictif de référence à tous les pixels d'un plan d'image donné (par exemple luminance, chrominance). Le décalage du signal de luminance est généralement appelé brillance.

Rapport signal-bruit de crête (PSNR): rapport entre la valeur maximale possible de la puissance d'un signal et la puissance d'un bruit altéré.

Système PAL: système couleur de vidéo composite analogique à 625 lignes.

Balayage matriciel: mise en correspondance (mappage) d'un canevas rectangulaire bidimensionnel avec un canevas unidimensionnel, de telle sorte que les premières entrées dans le canevas unidimensionnel proviennent de la première rangée supérieure du canevas bidimensionnel balayé de gauche à droite, suivi de la même façon par la seconde, la troisième, etc., rangée du canevas (en descendant), chacune d'elle étant balayée de gauche à droite.

Référence réduite (RR): méthode de mesure de la qualité vidéo qui utilise des caractéristiques de faible largeur de bande extraites des flux vidéo d'origine et traité, par opposition à une méthode fondée sur l'image de référence complète pour laquelle il faut connaître entièrement les flux vidéo d'origine et traité [2]. Les méthodes fondées sur une référence réduite présentent des avantages quant à la surveillance de qualité de bout en bout en service étant donné que les informations de référence réduite sont transmises facilement sur les réseaux de télécommunications du monde entier.

Région d'intérêt (ROI, *region of interest*): grille d'image (spécifiée en coordonnées de rectangle) utilisée pour désigner une sous-région particulière d'une trame ou d'une image vidéo.

Scène: séquence d'images vidéo.

Alignement spatial: processus utilisé pour évaluer et corriger les décalages spatiaux de la séquence vidéo traitée par rapport à la séquence vidéo d'origine.

Alignement temporel: processus utilisé pour évaluer et corriger le décalage temporel (c'est-à-dire le retard vidéo) de la séquence vidéo traitée par rapport à la séquence vidéo d'origine.

Mesure de la qualité vidéo: mesure globale de la dégradation de la qualité vidéo. La qualité VQM est un nombre unique dont la plage nominale est comprise entre zéro et un, zéro correspondant à aucune dégradation perçue et un à la dégradation maximale perçue.

3 Aperçu général du calcul de la qualité VQM

Le modèle RR transmet les valeurs d'activité pour différents blocs de pixels d'une taille donnée côté client. Cette valeur indique l'écart des valeurs de luminance dans le bloc. La Fig. 1 résume le modèle RR. Comme indiqué sur la Fig. 24, on évalue la qualité vidéo sur la base de la différence d'activité entre le SRC et la PVS. En outre, on applique des pondérations psychovisuelles pour la différence d'activité, afin d'obtenir une plus grande précision dans l'évaluation. On procède à l'évaluation de la qualité vidéo en suivant les étapes ci-après:

1) La valeur d'activité pour chaque bloc de 16 × 16 pixels de luminance du SRC est calculée côté serveur. Toutes les valeurs d'activité sont ensuite transmises côté client. La valeur d'activité d'un bloc est définie comme étant la différence moyenne absolue des différentes valeurs et leur valeur moyenne.
2) Les valeurs d'activité correspondante sont calculées côté client en ce qui concerne la PVS.
3) Côté client, on évalue tout d'abord chaque bloc avec son erreur quadratique, à savoir la différence quadratique entre les valeurs d'activité du SRC et de la PVS.

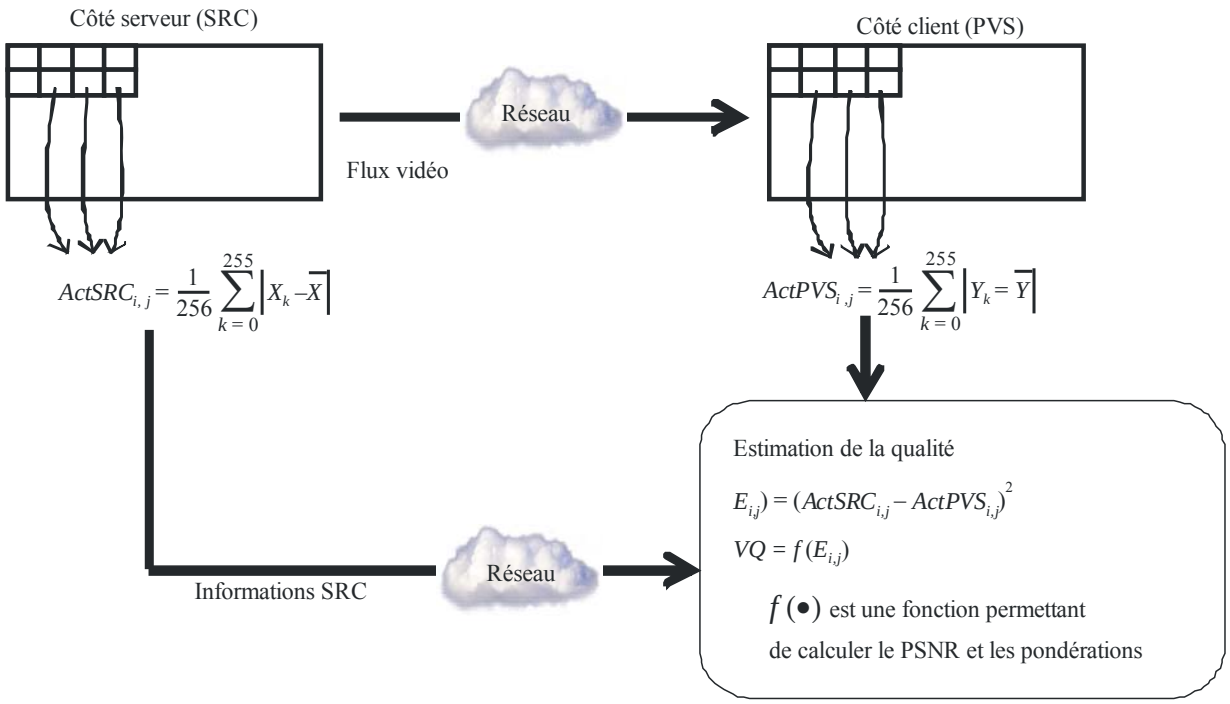
4) On applique des pondérations psychovisuelles aux erreurs quadratiques dans les blocs ayant une grande quantité de composantes de fréquences spatiales, ainsi qu'une couleur spécifique, une différence intertrames importante et un changement de scène.

5) On calcule une note provisoire de la qualité vidéo d'après la somme des erreurs quadratiques pondérées, de la même manière que dans le calcul du rapport PSNR.

6) On modifie la note pour tenir compte des dégradations perceptuelles intenses résultant du tuilage et de dégradations locales. Enfin, la note modifiée représente la qualité vidéo mesurée de la PVS dans le modèle RR.

FIGURE 24

Evaluation de la qualité vidéo sur la base de la différence d'activité



BT.1885-24

4 Algorithme détaillé

4.1 Côté serveur

1) Les pixels de luminance d'un SRC sont subdivisés en blocs de 16 × 16 pixels dans chaque trame, au bout d'une seconde après le début de la séquence vidéo. Au cours de la première seconde, les informations SRC ne sont pas transmises, car il est difficile pour le système visuel humain de déceler une dégradation de la qualité vidéo dans les scènes suivants immédiatement la première trame.

2) Dans chaque bloc, sauf ceux qui sont situés dans la bordure de la trame, les valeurs d'activité (SRC-activity: $ActSRC_{i,j}$) sont calculées. La Fig. 25 décrit les blocs que les valeurs d'activité calculent et transmettent. L'activité SRC-activity est calculée à l'aide de la formule:

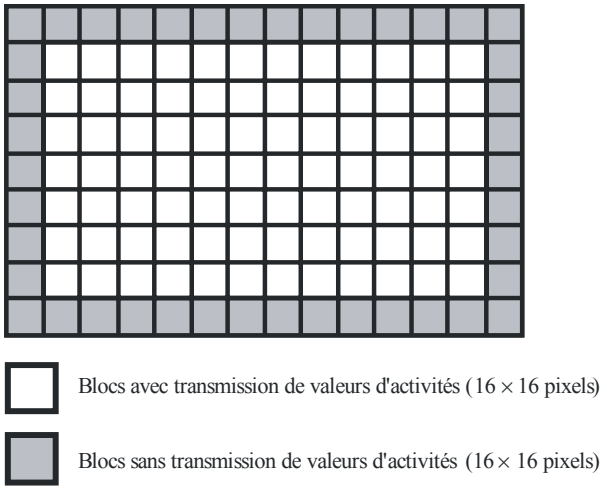
$$ActSRC_{i,j} = \frac{1}{256}\sum_{k=0}^{255}\left|X_k - \overline{X}\right|$$

où X_k est une valeur de luminance dans un bloc de taille donnée du SRC, $\overline{X}$ est sa moyenne, *i* est un numéro de trame, et *j* est un numéro de bloc dans la trame.

3) Les valeurs d'activité, qui sont exprimées à l'aide de 8 bits par bloc, sont transmises côtés client dans l'ordre d'exploration par balayage, au bout d'une seconde après le début de la séquence vidéo. Pour la transmission d'informations SRC à un débit de 256 kbit/s, les valeurs d'activité sont transmises dans toutes les trames. Lorsque le débit est ramené à 80 kbit/s, les valeurs d'activité sont transmises toutes les quatre trames.

FIGURE 25

Blocs avec et sans transmission de valeurs d'activité



BT.1885-25

4.2 Côté client

4.2.1 Calcul de l'erreur quadratique des valeurs d'activité

1) Les pixels de luminance d'une PVS sont subdivisés en blocs de 16 × 16 pixels dans chaque trame après une seconde suivant le début de la séquence vidéo.

2) Dans chaque bloc, sauf ceux qui sont situés dans la bordure de la trame, les valeurs d'activité (PVS-activity: $ActPVS_{i,j}$) sont calculées. Pour la transmission d'informations SRC à un débit de 256 kbit/s, les valeurs d'activité sont calculées dans toutes les trames. Lorsque le débit des informations SRC est ramené à 80 kbit/s, les valeurs d'activité sont calculées toutes les quatre trames.

$$ActPVS_{i,j} = \frac{1}{256}\sum_{k=0}^{255}\left|Y_k - \overline{Y}\right|$$

où:

Y_k: une valeur de luminance dans un bloc de taille donnée de la PVS

$\overline{Y}$: sa moyenne

i: un numéro de trame

j: un numéro de bloc dans la trame.

3) On calcule les erreurs quadratiques entre les activités SRC et PVS à l'aide de la formule:

$$E_{i,j} = \left(ActSRC_{i,j} - ActPVS_{i,j}\right)^2$$

4.2.2 Pondérations psychovisuelles pour l'erreur quadratique

On applique trois types de pondérations, à savoir la pondération pour la différence de fréquence spatiale, la pondération pour la différence de région de la couleur spécifique et la pondération pour la différence de luminance intertrames à $E_{i,j}$, à fin de tenir compte des caractéristiques du système visuel humain.

1) Pondération pour la différence de fréquence spatiale

On utilise un facteur de pondération W_{SF} et un seuil Th_{SF} pour cette pondération (voir le Tableau 8 pour les valeurs de W_{SF} and Th_{SF}.)

$$E_{i,j} \Leftarrow \begin{cases} E_{i,j} \times W_{SF}, & ActPVS_{i,j} > Th_{SF} \\ E_{i,j}, & otherwise \end{cases}$$

2) Pondération pour la différence de région de la couleur spécifique

Pour un bloc donné et les huit blocs qui l'entourent, si le nombre de pixels (*NumROIPixels*) dans $48 \le Y \le 224$, $104 \le Cb \le 125$ et $135 \le Cr \le 171$ est supérieur à un seuil, on effectue la pondération suivante en utilisant un facteur de pondération W_{CR} et un seuil Th_{CR}.

$$E_{i,j} \Leftarrow \begin{cases} E_{i,j} \times W_{CR}, & NumROIPixels > Th_{CR} \\ E_{i,j}, & otherwise \end{cases}$$

Voir le Tableau 8 pour les valeurs de W_{CR} et Th_{CR}.

3) Pondération pour la différence de luminance intertrames

On calcule la différence moyenne absolue ($MAD_{i,j}$) de la luminance entre un bloc donné et celle de la trame précédente. $MAD_{i,j}$ est défini de la façon suivante:

$$MAD_{i,j} = \frac{1}{256} \sum_{k=0}^{255} \left| Y_k - Y'_k \right|$$

où Y_k est une valeur de luminance dans un bloc de 16 × 16 pixels de la PVS et Y'_k est une valeur de luminance à la même position dans la trame précédente.

On effectue la pondération suivante au moyen des facteurs de pondération W_{MAD1}, W_{MAD2} et des seuils Th_{MAD1}, Th_{MAD2}.

$$E_{i,j} \Leftarrow \begin{cases} E_{i,j} \times W_{MAD\,1}, & MAD_{i,j} > Th_{MAD\,1} \\ E_{i,j} \times W_{MAD\,2}, & MAD_{i,j} \le Th_{MAD\,2} \\ E_{i,j}, & otherwise \end{cases}$$

Voir le Tableau 8 pour les valeurs de W_{MAD1}, W_{MAD2}, Th_{MAD1} et Th_{MAD2}.

4.2.3 Pondération en cas de détection de changement de scène

On calcule une moyenne de $MAD_{i,j}$ ($MADAve_i$) pour chaque trame à l'aide de la formule suivante:

$$MADAve_i = \frac{1}{M}\sum_{j=0}^{M-1} MAD_{i,j}$$

où M et le nombre de blocs dans une trame.

Si $MADAve_i$ est supérieur à un seuil Th_{SC}, on considère qu'il s'agit d'un changement de scène. Lorsqu'un changement de scène est détecté, $E_{i,j}$ est mis à 0 pour 15 trames après le changement de scène.

$$SceneChange = \begin{cases} TRUE, & MADAve_i > Th_{SC} \\ FALSE & otherwise \end{cases}$$

$$E_{i,j} \Leftarrow \begin{cases} E_{i,j} \times W_{SC} & 15\ frames\ after\ SceneChang\ e = TRUE \\ E_{i,j}, & otherwise \end{cases}$$

Voir le Tableau 8 pour les valeurs de W_{SC} etTh_{SC}

4.2.4 Rapport PSNR fondé sur l'erreur quadratique de l'activité

On calcule un rapport PSNR sur la base de la différence d'activité à l'aide de la formule suivante:

$$VQ = 10 \times \log_{10} \frac{255 \times 255}{E_{Ave}}$$

$$E_{Ave} = \frac{1}{N \times M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} E_{i,j}$$

où N et M sont le nombre de trames et de blocs utilisés pour le calcul du rapport PSNR

4.2.5 Pondération pour les artéfacts de tuilage

On utilise un facteur de pondération W_{BL}, un seuil Th_{BL}, et une information sur le niveau de tuilage BL_{Ave} pour cette pondération (Voir le Tableau 8 pour les valeurs de W_{BL} et Th_{BL} .)

$$VQ \Leftarrow \begin{cases} VQ \times W_{BL}, & BL_{Ave} > Th_{BL} \\ VQ, & otherwise \end{cases}$$

On calcule BL_{Ave} en suivant les étapes ci-après:

Etape 1: on calcule les valeurs d'activité pour des blocs de 8 × 8 pixels dans une PVS . Comme indiqué sur la Fig. 26, on calcule la valeur moyenne (Act_{Ave}) des deux valeurs d'activité dans des blocs adjacents sur le plan horizontal ($ActBlock_1$, $ActBlock_2$) à l'aide de la formule:

$$Act_{Ave} = \frac{1}{2}\left(ActBlock_1 + ActBlock_2\right)$$

Etape 2: on calcule la différence absolue des valeurs de luminance le long de la limite entre deux blocs. Comme indiqué sur la Fig. 26, $Y_{1,0}$ et $Y_{2,0}$ représentent les valeurs de luminance dans les blocs de gauche et de droite le long de la limite. On exprime une valeur moyenne de la différence de luminance absolue à l'aide de la formule suivante:

$$DiffBound = \frac{1}{8}\sum_{i=0}^{7}\left|Y_{1,i} - Y_{2,i}\right|$$

Etape 3: on définit le niveau de tuilage ($BL_{i,j}$) par le rapport entre *DiffBound* et Act_{Ave}, c'est-à-dire:

$$BL_{i,j} = \frac{DiffBound}{Act_{Ave} + 1}$$

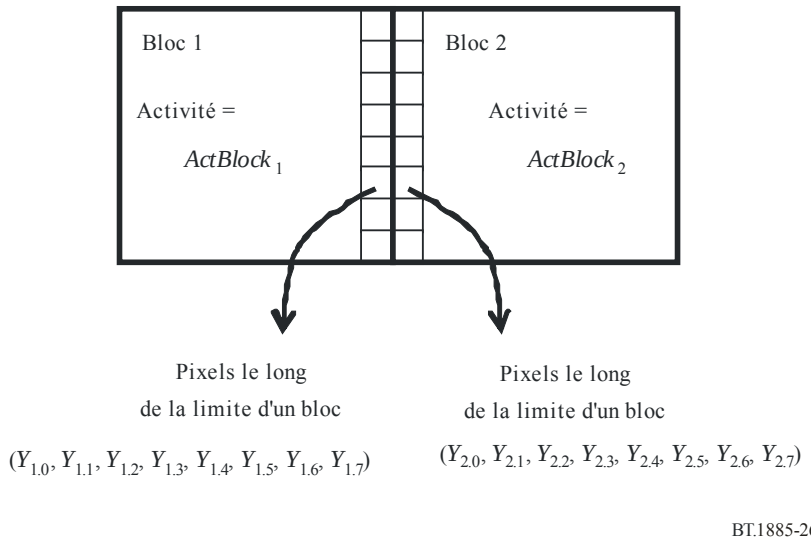
Etape 4: on calcule la valeur moyenne de *BL* à l'aide de la formule suivante:

$$BL_{Ave} = \frac{1}{N \times M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} BL_{i,j}$$

Pour les blocs situés le plus à droite dans les trames, la valeur de $BL_{i,j}$ est mise à zéro. Si BL_{Ave} est supérieur à un seuil déterminé au préalable, on considère que la séquence vidéo comprend un niveau important de tuilage et on applique une pondération à la valeur calculée de la qualité vidéo.

FIGURE 26

Valeurs des pixels et valeurs d'activité utilisées pour le calcul du niveau de tuilage



BT.1885-26

4.2.6 Pondération pour les défauts (artéfacts) dus à des dégradations locales

On utilise un facteur de pondération W_{LI}, un seuil Th_{LI}, et une dégradation locale *LI* pour cette pondération (Voir le Tableau 8 pour les valeurs de W_{LI} et Th_{LI} values.)

$$VQ \Leftarrow \begin{cases} VQ \times W_{LI}, & LI < Th_{LI} \\ VQ, & otherwise \end{cases}$$

On calcule *LI* en suivant les étapes ci-après. On utilise la différence de l'écart d'activité pour déceler une dégradation locale due à des erreurs de transmission.

1) Pour un bloc donné et les huit blocs qui l'entourent,, on calcule l'écart de l'activité pour le SRC ($ActVar_{SRC}$) et la PVS ($ActVar_{PVS}$) et la différence absolue entre les valeurs de ces écarts est calculée à l'aide de la formule:

$$\Delta ActVar = \left| ActVar_{SRC} - ActVar_{PVS} \right|$$

2) On calcule la valeur moyenne de ces valeurs de différence absolue pour chaque trame.

3) On calcule LI comme étant le rapport de la valeur maximale ($\Delta ActVar_{Max)}$ à la valeur minimale ($\Delta ActVar_{Min}$) de la moyenne

$$LI = \begin{cases} \Delta ActVar_{Min} / \Delta ActVar_{Max} & \Delta ActVar_{Max} \neq 0 \\ 1 & \Delta ActVar_{Max} = 0 \end{cases}$$

VQ représente la note de la qualité vidéo.

4.2.7 Paramètres pour les pondérations

Le Tableau 8 indique les valeurs des paramètres pour les pondérations. Ces valeurs sont déterminées par une expérience préliminaire avec un ensemble de données d'apprentissage.

TABLEAU 8

Paramètres pour les pondérations

Type de fonctionnement de la pondération	Valeur des paramètres	
Pondération fréquentielle spatiale	W_{SF}	0,36
	Th_{SF}	25
Pondération pour la couleur spécifique	W_{CR}	4,0
	Th_{CR}	175
Pondération pour la différence intertrames	W_{MAD1}	0,06
	Th_{MAD1}	17
	W_{MAD2}	25
	Th_{MAD2}	13
Détection de changement de scène	W_{SC}	0,0
	Th_{SC}	35
Pondération pour le tuilage	W_{BL}	0,870
	Th_{BL}	1,0
Pondération pour les dégradations locales	W_{LI}	0,870
	Th_{LI}	1,67

4.2.8 Alignement

1) Alignement spatial

Le modèle RR ne nécessite aucun alignement spatial, car on calcule l'erreur quadratique à partir de valeurs d'activité qui sont plus résistantes aux décalages spatiaux que celles reposant sur les valeurs des pixels.

2) Alignement du gain et du décalage

Le modèle RR ne nécessite aucun alignement du gain et du décalage. Les valeurs d'activité sont par nature exemptes de décalage (c'est-à-dire de composantes DC) et insensibles au gain.

3) Alignement temporel

La séquence PVS est subdivisée en sous-séquences de 1 seconde. Pour chaque sous-séquence, on calcule les erreurs quadratiques moyennes de l'activité avec cinq variations du retard de ±2 SRC au maximum. Enfin, on utilise la valeur minimale des erreurs quadratiques moyennes comme étant l'erreur quadratique moyenne dans cette sous-séquence. Le retard qui aboutit à ce niveau minimum de l'erreur quadratique moyenne est ajusté sous la forme d'un alignement temporel.

5 Codes types

Les codes types en langage C pour le modèle RR sont présentés ci-après.

5.1 Code commun côté serveur et coté client

```
// Calcule the activity value
unsigned int CalcActivitybyRect(unsigned char * lpBuffer, int nWidth, int iRectWidth, int iRectHeight)
{
  // lpBuffer: Luminance Frame Buffer
   // nWidth: Frame Buffer Width
   // iRectWidth: Width of the rectangle to calculate an activity value.
   // iHeightWidth: Height of the rectangle to calculate an activity value.
        unsigned int i, j, nTmp, nSum;
        unsigned char *pSrc;

        pSrc = lpBuffer;  nSum = 0;
        for (j = 0; j < iRectHeight; j++){
            for (i = 0; i <iRectWidth; i++){
                nSum += pSrc[i];
            }
            pSrc += nWidth;
        }
        nSum /= (iRectWidth*iRectHeight);

        pSrc = lpBuffer; nTmp = 0;
        for (j = 0; j < iRectHeight; j++){
            for (i = 0; i <iRectWidth; i++){
                nTmp += abs(pSrc[i] - nSum);
            }
            pSrc += nWidth;
        }
        return nTmp/iRectWidth/iRectHeight;
}
```

5.2 Côté serveur

```
// Server side
int nStart = 30; // the frame number to start transmission (30 or 25)
int nMaxFrame = 240; // the number of total video frames (240 or 200)
int nFrameIncrement = 1; // 1 for 256kbps, 4 for 80kbps
void ReadOneFrame(unsigned char, int, unsigned char *, int, int); // function to read one frame data
int nRim = 16 // 16 or 32 (use 32 to avoid the problem in HRC9)

// nWidth: Frame Buffer Width
// nHeight: Frame Buffer Height
// lpSrc: Frame Buffer
for(int nFrame = nStart; nFrame < nMaxFrame; nFrame+=nFrameIncrement){
        ReadOneFrame(SRC_file_name, nFrame, lpSrc, nWidth, nHeight);
        for (j= 16; j<nHeight-32; j+=16) {
            for (i= nRim; i<nWidth- nRim; i+=16) {
```

```
                    lpOrg = lpSrc + i + j * nWidth;
                    nActSrc = CalcActivitybyRect(lpOrg, nWidth, 16, 16);
        // OutputSRCInfo(nActSrc); // Output or transmission the SRC information
                }
            }
}
```

5.3 Côté client

```
// Client Side
int nStart = 30; // the frame number to start transmission (30 or 25)
int nMaxFrame = 240; // the number of total video frames (240 or 200)
int nFrameIncrement = 1; // 1 for 256kbps, 4 for 80kbps
int nFrameRate = 30; //30 or 25
void ReadOneFrame(unsigned char, int, unsigned char **, int, int); // function to read one frame data
void ReadRRData(unsigned char, int, unsigned char *); // function to read RR-data

// nWidth: Frame Buffer Width
// nHeight: Frame Buffer Height
// lpPvsByte[3]: Frame Buffer (0:Y, 1:Cb, 2:Cr)
// lpRRData: RR-data Buffer
// double ddActivityDifference[][]: Store the activity-difference
// double ddActivityVariance[][]:  Store the activity-variance
// double ddBlock[][]:      Store the blockiness level
// int nSceneChange:      Scene change detection

for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){ // Changing temporal alignment
            for(int nFrame = 0; nFrame < nMaxFrames; nFrame++){
                if(nFrame+nTemporalAlign >= nMaxFrames || nFrame+nTemporalAlign < 0){
                    continue;
                }
                ReadOneFrame(PVS_file_name, nFrame+nTemporalAlign, lpPvsByte, nWidth, nHeight);
                if(((nFrame-(nFrameRate+nStart)) % nFrameIncrement) == 0
                        && nFrame >= nStart ){
                    ReadRRData(RR_file_name, nFrame, lpRRData);
                    ddActivityDifference[nTemporalAlign+2][nFrame]
                                        = RRCalcObjectiveScore(lpPvsByte, lpRRData, nWidth, nHeight);
                    ddActivityVariance[nTemporalAlign+2][nFrame] = gnActVar;
                }else{
                    ddActivityDifference[nTemporalAlign+2][nFrame] = 0.0;
                    ddActivityVariance[nTemporalAlign+2][nFrame] = 0.0;
                }
                // Blockiness Level
                if(nTemporalAlign ==0){
                    ddBlock[nFrame] = BlockinessLevelEstimation(lpPvsByte[0], nWidth, nHeight);
                }
                // Pixel copy for inter-frame difference calculation
                memcpy(lpPrev, lpPvsByte[0], sizeof(char)*nWidth*nHeight);
                if(nSceneChange){
                    nSceneChange--;
                }
            }
}

double ddSum[8][5]; // Sum of the Activity-difference for each second
double ddActVarSum[8][5]; // Sum of the Activity-variance for each second
double ddActVarMax[8][5]; // Maximum of the Sum of the Activity-variance
double ddActVarMin[8][5]; // Minimum of the Sum of the Activity-variance
int nnMin[8];
int nnNumFrames[8][5];
#define LARGENUMBER 100000
for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){
            for(int j=0;j<8;j++){ // for each one second
                nnNumFrames[j][nTemporalAlign+2] = 0;
                ddActVarMax[j][nTemporalAlign+2] = 0.0;
                ddActVarMin[j][nTemporalAlign+2] = LARGENUMBER;
                ddActVarSum[j][nTemporalAlign+2] = 0.0;
                ddSum[j][nTemporalAlign+2] = 0.0;
```

```
                for(int i=nFrameRate*j;i< (j+1)*nFrameRate; i++){
                    if(ddActivityDifference[nTemporalAlign+2][i]){
                        ddSum[j][nTemporalAlign+2] += ddActivityDifference[nTemporalAlign+2][i];
                        nnNumFrames[j][nTemporalAlign+2]++;
                    }
                    ddActVarSum[j][nTemporalAlign+2] += ddActivityVariance[nTemporalAlign+2][i];
                    if(ddActivityVariance[nTemporalAlign+2][i]){
                                    if(ddActivityVariance[nTemporalAlign+2][i] >
                                    ddActVarMax[j][nTemporalAlign+2]){
                                            ddActVarMax[j][nTemporalAlign+2] =
                                            ddActivityVariance[nTemporalAlign+2][i];
                        }
                                    if(ddActivityVariance[nTemporalAlign+2][i] <
                                    ddActVarMin[j][nTemporalAlign+2]){
                                            ddActVarMin[j][nTemporalAlign+2] =
                                            ddActivityVariance[nTemporalAlign+2][i];
                        }
                    }
                }
            }
}

// Local Impariment Level Calculation
double dSum = 0.0;
double dActMax = 0.0;
double dActMin = LARGENUMBER;
int nNumFrames = 0;
for(int j=1; j<8; j++){
            double dMin = LARGENUMBER;
            double dMinSum = LARGENUMBER;
            for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){
                if(ddActVarSum[j][nTemporalAlign+2] < dMin){
                    dMin = ddActVarSum[j][nTemporalAlign+2];
                    dMinSum = ddSum[j][nTemporalAlign+2];
                    nnMin[j] = nTemporalAlign+2;
                }
            }
            dSum += dMinSum;
            nNumFrames += nnNumFrames[j][nnMin[j]];
            if(ddActVarMax[j][nnMin[j]] > dActMax){
                dActMax = ddActVarMax[j][nnMin[j]];
            }
            if(ddActVarMin[j][nnMin[j]] < dActMin){
                dActMin = ddActVarMin[j][nnMin[j]];
            }
}
double dTransError = dActMax/dActMin;

// Blockiness Level Calculation
double dBlockinessLevel = 0.0;
for(int i=0;i<nMaxFrames; i++){
            dBlockinessLevel += ddBlock[i];
}
dBlockinessLevel = dSumBlock / (double)(nMaxFrames-nFrameRate);

// Calculating the Video Quality Score
if(nNumFrames && nNumberOfBlocks && dSum){
            dSum = dSum / (double)(nNumFrames)/(double)nNumberOfBlocks;
            dSum = 10*log10(255.0*255.0/dSum); //PSNR based on the activity difference
            if(dBlockinessLevel > dBlokinessTh){
                dSum /= nBlockinessWeighting; // Weighting for blockiness level
            }
            if(dTransError > nErrorTh){
                dSum /=nErrorWeightin;  // Weighting for transmission errors
            }
}
return dSum;
-------------------------------------------------------------------------------------
// Calculating the MAD value
```

```
unsigned int CalcMAD(unsigned char *lpSrc, unsigned char *lpSrc2, int nWidth, int nHeight)
{
	// lpSrc:  Frame Buffer of the current frame
	 // lpSrc2: Frame Buffer of the previous frame
	 unsigned int nSum = 0;
	for (y = 0; y < nHeight; y++) {
		for (x = 0; x < nWidth; x++) {
			nSrc = lpSrc[x + y*nWidth];
			nSrc2 = lpSrc2[x + y*nWidth];
			nSum += abs(nSrc - nSrc2);
		}
	}
	return nSum/nWidth/nHeight;
}

// Calculating a mean squared error with weightings
double RRCalcObjectiveScore(unsigned char *lpBuffer[], unsigned char *lpRRData, int nWidth, int nHeight)
{
	int	i, j, nActSrc, nActSrc2, nY, nCb, nCr, nYMin, nYMax, nCbMin, nCbMax, nCrMin, nCrMax;
	int	nMBX, nMBY, nMB, nStart;
	unsigned int nMAD;
	double	e2, dMADFrame;
	unsigned char *lpRec, *lpRecCb, *lpRecCr, *lpPrev1;
	int nRim = 16 // 16 or 32 (use 32 to avoid the problem in HRC9)

	nYMin = 48; nYMax = 224; nCbMin = 104; nCbMax = 125; nCrMin = 135; nCrMax = 171;
	nMB = nMBY = nStart = 0;
	e2 = dMADFrame = 0.0;

	for (j=16+nStart; j<iImageHeight-32; j+=16) {
		nMBX = 0;
		for (i= nRim; i<nWidth- nRim; i+=16) {
			lpRec = lpBuffer[0] + i + j * nWidth;
			lpRecCb = lpBuffer[1] + i/2 + (j/2) * nWidth/2;
			lpRecCr = lpBuffer[2] + i/2 + (j/2) * nWidth/2;
			lpPrev1 = lpPrev + i + j * nWidth;

			nActSrc = lpRRData[nMB]; // SRC activity
			nActSrc2 = CalcActivitybyRect(lpRec, nWidth, 0, 16, 16); // PVS activity
			nActArray[nMB] = (double)nActSrc;
			nActArray2[nMB] = (double)nActSrc2;
			e2 += (double)(nActSrc - nActSrc2)*(nActSrc - nActSrc2); // Mean squared error

			nMAD   = CalcMAD(lpRec, lpPrev1, 16, 16, nWidth); // Inter-frame differnece
			dMADFrame += (double)nMAD;

			int nNumROIPixels=0;
			for(int jj=-16;jj<32; jj++){
				for(int ii=-16;ii<32; ii++){
				  nY = lpRec[ii];
				  nCb = lpRecCb[ii/2];
				  nCr = lpRecCr[ii/2];
				  if(nY >= nYMin && nY <= nYMax
				  && nCb >= nCbMin && nCb <= nCbMax
				  && nCr >= nCrMin && nCr <= nCrMax){
						nNumROIPixels++;
				  }
				}
				lpRec += nWidth;
				if((jj & 1) == 1){
				  lpRecCb += nWidth/2;
				  lpRecCr += nWidth/2;
				}
			}

			// Weighting for spatial frequency
			if(nActSrc2 > gdwActThHigh){
				e2 *= dActWeightingHigh;
			}else if(nActSrc2 > gdwActThLow){
```

```
                    e2 *= dActWeightingMiddle;
                }else {
                    e2 *= dActWeightingLow;
                }

                // Weighting for specific color region
                if(nNumROIPixels > dwROITh){
                    e2 *= dROIWeighting;
                }

                // Weighting for inter-frame difference
                if(nMAD > dwMADThHigh){
                    e2 *= dMADWeightingHigh;
                }else if(nMAD > dwMADThLow){
                    e2 *= dMADWeightingMiddle;
                }else {
                    e2 *= dMADWeightingLow;
                }
                nMB++;          nMBX++;
            }
            nMBY++;
        }

        // Calculating Activity-Variance for Surrounding Nine Blocks.
        double nSumActSrc, nSumActPvs, nActVar, nActVar2;
        nSumActSrc = nSumActPvs = nActVar = nActVar2 = 0.0;
        gnActVar = 0.0;
        for (j=1; j<nMBY-1; j++) {
            for (i=1; i<nMBX-1; i++) {
                nSumActSrc = 0.0;
                nSumActPvs = 0.0;
                for(int jj=-1; jj<2; jj++){
                    for(int ii=-1; ii<2; ii++){
                       nSumActSrc += nActArray[i+ii+nMBX*(j+jj)];
                       nSumActPvs += nActArray2[i+ii+nMBX*(j+jj)];
                    }
                }
                nSumActSrc /= 9.0;
                nSumActPvs /= 9.0;

                nActVar = 0.0;
                nActVar2 = 0.0;
                for(int jj=-1; jj<2; jj++){
                    for(int ii=-1; ii<2; ii++){
        nActVar += (nActArray[i+ii+nMBX*(j+jj)]-nSumActSrc)*(nActArray[i+ii+nMBX*(j+jj)]-nSumActSrc);
                                        nActVar2 += (nActArray2[i+ii+nMBX*(j+jj)]-
                                        nSumActPvs)*(nActArray2[i+ii+nMBX*
                                        (j+jj)]-nSumActPvs);
                    }
                }
                nActVar /= 9.0;
                nActVar2 /= 9.0;
                gnActVar += abs(nActVar- nActVar2);
            }
        }
        // Average of the Activity-Variance for the Frame
        gnActVar = gnActVar/(double)(nMBY-2)/(double)(nMBY-2);

        // Scene change detection
        if(dMADFrame/ nMB > 35){
            nSceneChange = 15;
        }
        if(nSceneChange){
            e2 = 0.0;
        }
        gnFrame++;

        return e2;
}
```

```
// Calculate Blockiness Level
double BlockinessLevelEstimation(unsigned char *lpBuffer, int nWidth, int nHeight)
{
	// lpBuffer: Frame Buffer
	int	i, j, nActSrc, nActSrc2, nDiff, nMB;
	unsigned char *lpRec = lpBuffer;
	double dBlock=0.0;

	nMB = 0;
	for (j=0; j<nHeight-16; j+=8) {
		for (i=0; i<nWidth-16; i+=8) {
			lpRec = lpBuffer + i + j * nWidth;
			nActSrc = CalcActivitybyRect(lpRec, nWidth, 0, 8, 8); // Activity of the left block
			nActSrc2 = CalcActivitybyRect(lpRec+8, nWidth, 0, 8, 8); // Activity of the right block
			nActSrc = (nActSrc + nActSrc2)/2; // Average of the activity values
			nDiff = 0;
			for(int jj=0;jj<8; jj++){
				nDiff += abs(lpRec[7+jj*nWidth] - lpRec[8+jj*nWidth]);
					// Difference of the luminance values at the boundary
			}
			nDiff/= 8;
			dBlock += (double)nDiff/(double)(nActSrc+1);
			nMB++;
		}
	}
	return (double)dBlock/(double)nMB;
}
```

6 Références pour information

[1] SMPTE 170M, «SMPTE Standard for Television – Composite Analog Video Signal – NTSC for Studio Applications», Society of Motion Picture and Television Engineers.

[2] Recommandation UIT-T J.143 – Prescriptions d'utilisateur relatives aux mesures objectives de la qualité vidéo perçue en télévision numérique par câble

Annexe C

Modèle C: Méthode de référence réduite de la NTIA

1 Généralités

Au cours de la période 2003-2004, la National Telecommunications and Information Administration (NTIA) des Etats-Unis a mis au point deux modèles de mesure de la qualité vidéo (VQM) avec une largeur de bande de référence réduite (RR) de l'ordre de 12 à 14 kbit/s pour la séquence vidéo échantillonnée selon la Recommandation UIT-R BT.601 [1]. On a appelé ces modèles le «modèle VQM de faible largeur de bande» et le «modèle VQM de faible largeur de bande à débit rapide». Ce dernier modèle constituait une version efficace, sur le plan du calcul, du modèle VQM de faible largeur de bande. Le modèle VQM de faible largeur de bande à débit rapide est près de quatre fois plus rapide, dans la mesure où il extrait les caractéristiques spatiales d'images vidéo pour lesquelles on établit au préalable une moyenne, au lieu d'extraire ces caractéristiques

spatiales directement à partir des images vidéo conformes à la Recommandation UIT-R BT.601. Des gains d'efficacité supplémentaires sur le plan des calculs ont également pu être réalisés, avec le modèle VQM de faible largeur de bande à débit rapide, en calculant les caractéristiques des informations temporelles (c'est-à-dire le mouvement) sur la base d'un sous-échantillonnage aléatoire des pixels dans le canal de luminance Y, et non pas en utilisant tous les pixels des trois canaux vidéo (Y, Cb et Cr). Ces deux modèles VQM sont disponibles dans nos outils logiciels VQM depuis plusieurs années et peuvent être librement utilisés pour des applications tant commerciales que non commerciales. Les exécutables binaires de ces outils VQM et leur code source associé peuvent également être téléchargés [2].

Etant donné que la NTIA voulait soumettre les deux modèles VQM de faible largeur de bande et de faible largeur de bande à débit rapide aux tests RRTV (télévision avec référence réduite), afin qu'ils fassent l'objet d'une évaluation indépendante par le Groupe d'experts en qualité vidéo (VQHD), elle a choisi de les soumettre à différentes catégories de débits binaires, même s'ils possèdent des caractéristiques de débit binaire RR identiques. La NTIA a choisi de soumettre le modèle VQM de faible largeur de bande à la catégorie 256 kbit/s et le modèle VQM de faible largeur de bande à débit rapide à la catégorie de débit 80 kbit/s, étant donné que l'on pensait que le modèle VQM de faible largeur de bande donnerait de meilleurs résultats que le modèle de faible largeur de bande à débit rapide. Les deux modèles VQM ont utilisé l'algorithme d'étalonnage RR de la NTIA, qui est décrit dans la Recommandation UIT-T J.244 [3]. Cet algorithme d'étalonnage nécessite environ 22 à 24 kbit/s de largeur de bande RR pour obtenir des estimations du décalage temporel, du décalage spatial, de l'agrandissement spatial, du décalage du gain et du décalage de niveau.

L'un des résultats intéressants des tests d'évaluation RRTV effectués par le Groupe VQEG [4] est que le modèle VQM de faible largeur de bande à débit rapide a surclassé le modèle VQM de faible largeur de bande, tant pour le test à 525 lignes que pour le test à 625 lignes. Cela signifie implicitement qu'il est plus intéressant d'extraire les caractéristiques spatiales à partir de trames moyennées qu'à partir de trames non moyennées. La question de savoir si ce résultat sera confirmé pour d'autres séries de données appelle des recherches complémentaires. Pour le moment, la NTIA ne voit pas l'intérêt de normaliser les deux modèles, de sorte que la présente Annexe ne décrit que le modèle VQM de faible largeur de bande à débit rapide.

2 Introduction

On trouvera dans la présente Annexe une description et un code de référence pour le modèle VQM de faible largeur de bande à débit rapide de la NTIA. Ce modèle utilise des techniques comparables à celles du modèle VQM général de la NTIA, dont on trouvera une description dans les Recommandations UIT-T J.144 [5] et UIT-R BT.1683 [6]. Le modèle VQM de faible largeur de bande à débit rapide utilise les caractéristiques RR avec beaucoup moins de largeur de bande que le modèle général VQM de la NTIA, encore que ces deux modèles utilisent un processus d'extraction et de comparaison des caractéristiques analogue. En ce qui concerne la séquence vidéo échantillonnée de la Recommandation UIT-R BT.601 [1], le modèle VQM de faible largeur de bande à débit rapide utilise des caractéristiques RR qui nécessitent de l'ordre de 12 à 14 kbit/s de largeur de bande de transmission. La présente Annexe ne traite que du modèle VQM de faible largeur de bande à débit rapide, étant donné que les algorithmes d'étalonnage complémentaires de faible largeur de bande sont présentés de manière détaillée dans la Recommandation UIT-T J.244 [3]. Cependant, dans un souci d'exhaustivité, le code de référence décrit dans la présente Annexe comprend à la fois le modèle VQM de faible largeur de bande à débit rapide et les algorithmes d'étalonnage de faible largeur de bande qui lui sont associés. En outre, ce code de référence contient des exemples de fonctions de quantification pour les caractéristiques utilisées par l'étalonnage de faible largeur de bande (ces fonctions de quantification ne font pas partie de la Recommandation UIT-T J.244).

3 Description du modèle VQM de faible largeur de bande à débit rapide

3.1 Aperçu du modèle VQM

La description du modèle VQM englobe les trois principaux éléments suivants:

1) Caractéristiques de faible largeur de bande extraites des flux vidéo d'origine et traités.
2) Paramètres résultant de la comparaison des flux de caractéristiques analogues aux flux d'origine et des flux traités.
3) Calcul du modèle VQM associant les différents paramètres, dont chacun mesure un aspect différent de la qualité vidéo.

Cette description utilise des références facilement accessibles pour les détails techniques.

3.2 Description des caractéristiques

3.2.1 Aperçu des caractéristiques

Le modèle VQM de faible largeur de bande à débit rapide utilise trois types de caractéristiques, à savoir: caractéristiques de couleurs, spatiales et temporelles. Chacun de ces types de caractéristiques quantifie des distorsions perceptuelles dans ses domaines respectifs. Le sous-programme du code de référence «model_fastlowbw_features» offre une description mathématique complète des caractéristiques utilisées par le modèle VQM de faible largeur de bande à débit rapide.

3.2.2 Caractéristiques de couleurs

Les caractéristiques de couleurs sont les mêmes caractéristiques $f_{\_COHER\_COLOR}$ que celles qui sont utilisées par le modèle général VQM de la NTIA. Ces caractéristiques sont décrites de manière détaillée dans l'Annexe D.7.3 de la Recommandation UIT-T J.144. Les caractéristiques $f_{\_COHER\_COLOR}$ permettent d'effectuer une mesure vectorielle à deux dimensions de la quantité d'informations sur la chrominance rouge et bleue (C_B, C_R) dans chaque région S-T. En conséquence, les caractéristiques $f_{\_COHER\_COLOR}$ sont sensibles aux distorsions de couleur. Les caractéristiques

$f_{\_COHER\_COLOR}$ du modèle VQM de faible largeur de bande à débit rapide de la NTIA sont extraites de dimensions des régions spatio-temporelles (S-T) de 30 lignes verticales × 30 pixels horizontaux × 1 s de temps (c'est-à-dire 30 × 30 × 1s), alors que le modèle général VQM de la NTIA utilisait des dimensions de régions S-T de 8 × 8 × 1 trames.

3.2.3 Caractéristiques spatiales

Les caractéristiques spatiales sont les mêmes caractéristiques f_{SI13} et f_{HV13} que celles qui sont utilisées par le modèle général VQM de la NTIA. Ces caractéristiques sont décrites de manière détaillée dans l'Annexe D.7.2.2 de la Recommandation UIT-T J.144. Les caractéristiques f_{SI13} et f_{HV13} mesurent la quantité et la distribution angulaire de gradients spatiaux dans les sous-régions spatio-temporelles (S-T) des images de luminance (Y). En conséquence, les caractéristiques f_{SI13} et f_{HV13} sont sensibles à des distorsions spatiales telles que le flou et la subdivision en blocs.

Les caractéristiques f_{SI13} et f_{HV13} du modèle VQM de faible largeur de bande à débit rapide de la NTIA sont extraites des régions spatio-temporelles (S-T) ayant une dimension de 30 lignes verticales × 30 pixels horizontaux × 1 seconde de temps (c'est-à-dire 30 × 30 × 1 s), alors que le modèle général VQM de la NTIA utilisait des dimensions de région S-T de 8 × 8 × 0,2 s. En outre, pour limiter le nombre de calculs à effectuer, on calcule au préalable la moyenne de la première seconde des images Y de luminance dans le temps, avant d'appliquer les filtres d'accentuation des contours 13 × 13 à deux dimensions qui sont décrits dans l'Annexe D.7.2.1 de la Recommandation UIT-T J.144.

On extrait une caractéristique spatiale additionnelle de la première seconde des images de luminance (Y) pré-moyennées. Cette caractéristique est le niveau *moyen* de luminance (Y) de chaque région S-T 30 × 30 × 1s (dénotée ici sous la forme f_{MEAN}). L'objectif de la caractéristique f_{MEAN} est de fournir une fonction de pondération perceptuelle au niveau de la luminance pour la perte d'informations spatiales (SI), telle que mesurée par les caractéristiques f_{SI13} et f_{HV13}. Cette notion sera décrite dans la partie consacrée à la description des paramètres.

3.2.4 Caractéristiques temporelles

On peut obtenir des estimations très fiables de la qualité vidéo perçue à partir de l'ensemble de caractéristiques de couleurs et spatiales décrites ci-dessus. Cependant, étant donné que les régions S-T dont ces caractéristiques sont extraites couvrent un grand nombre d'images vidéo (c'est-à-dire une seconde d'images vidéo), elles sont généralement insensibles aux perturbations temporelles de courte durée qui se produisent dans l'image. Ces perturbations peuvent être dues au bruit ou à des erreurs de transmission numérique. En outre, même si elles sont par nature de courte durée, elles peuvent influer de manière significative sur la qualité perçue de l'image. En conséquence, on utilise une caractéristique RR de type temporelle pour quantifier les effets perceptuels des perturbations temporelles. Cette caractéristique mesure l'information temporelle absolue (ATI), ou le mouvement, dans le plan de l'image Y de luminance et est calculée de la façon suivante:

$$f_{ATI} = rms\left\{rand\,5\%\left|Y(t) - Y(t - 0.2s)\right|\right\}$$

Pour plus d'efficacité du point de vue des calculs, on sous-échantillonne de manière aléatoire *Y*, afin que cette valeur contienne uniquement 5% des pixels de l'image (représentés ici par la fonction *rand5%*). L'image *Y* ainsi sous-échantillonnée à l'instant *t*-0,2s est soustraite de l'image *Y* sous-échantillonnée de manière identique à l'instant *t* et l'erreur quadratique moyenne (*rms*) du résultat est utilisée comme mesure de l'ATI. Suivant la convention donnée dans l'Annexe D.8 de la Recommandation UIT-T J.144, ce processus sera également représenté de la façon suivante:

$$f_{ATI} \cong \text{Y\_rand5\%\_ati0.2s\_rms}$$

La caractéristique f_{ATI} est sensible aux perturbations temporelles. Pour une séquence vidéo à 30 images par seconde, 0,2 s correspond à 6 images vidéo, tandis que pour des séquences vidéo de 25 images par seconde, 0,2 s correspond à 5 images vidéo. Si l'on soustrait les images de 0,2 s, la caractéristique devient insensible aux systèmes vidéo en temps réel à 30 images par seconde et 25 images par seconde, qui présentent une fréquence de mise à jour d'image d'au moins 5 images par seconde. Les aspects qualité de ces systèmes vidéo présentant une fréquence d'image peu élevée, ce qui est courant dans les applications multimédias, sont suffisamment pris en compte par les caractéristiques f_{SI13}, f_{HV13}, et f_{COHER_COLOR}. Par ailleurs, prévoir un espacement de 0,2 s correspond plus étroitement à la réponse temporelle de crête du système visuel humain que différencier deux images avec un espacement d'une trame dans le temps.

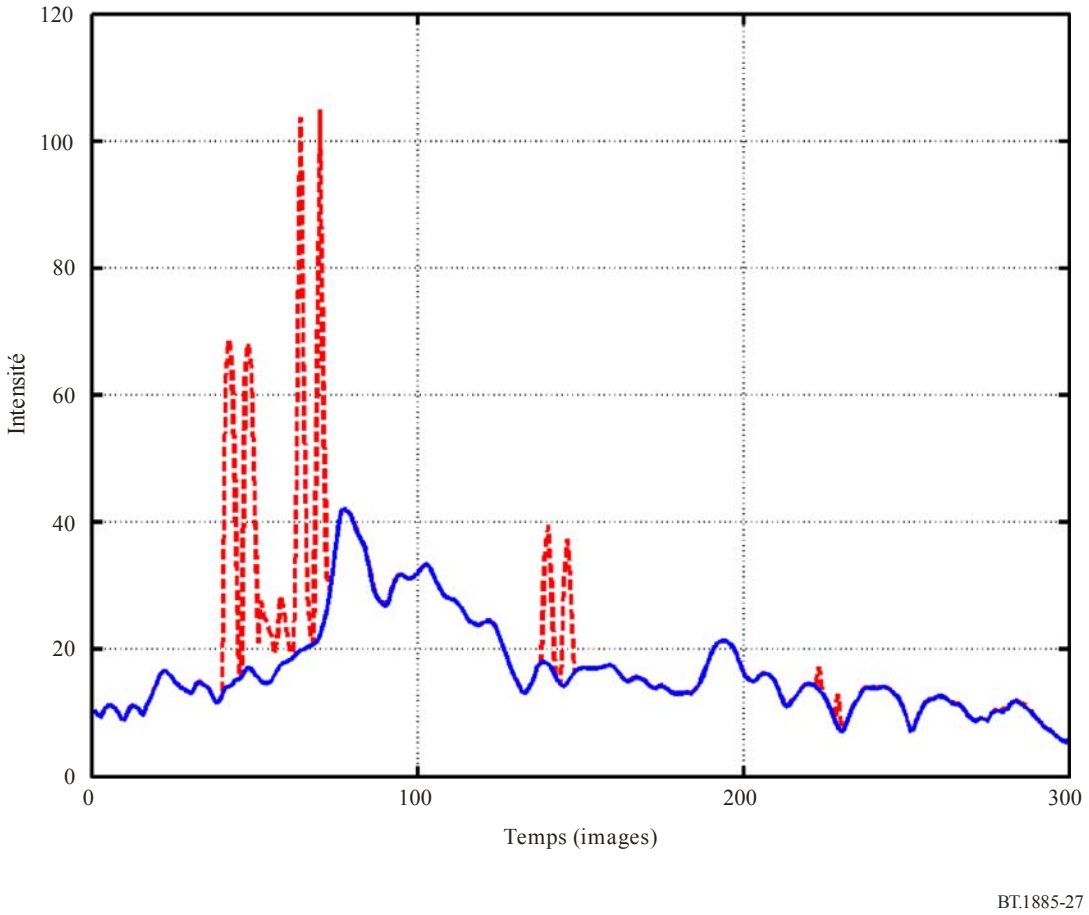
Le graphique de la Fig. 27 est un exemple de la caractéristique f_{ATI} d'une scène vidéo d'origine (traits pleins) et d'une scène vidéo traitée (pointillés rouges) provenant d'un système vidéo numérique avec des erreurs transitoires en salves dans le canal de transmission numérique. Les erreurs transitoires présentes dans l'image traitée créent des pics dans la caractéristique f_{ATI} . La largeur de bande nécessaire pour transmettre la caractéristique f_{ATI} est extrêmement faible, étant donné qu'elle ne nécessite que 30 échantillons par seconde pour des séquences vidéo de 30 images par seconde. D'autres types d'ajouts de bruit dans la séquence vidéo traitée, par exemple le bruit qui pourrait être généré par un système vidéo analogique apparaîtront sous la forme d'un décalage DC positif dans l'historique temporel du flux de caractéristiques traitées par rapport au flux de

caractéristiques d'origine. Les systèmes de codage vidéo qui éliminent le bruit entraineront un décalage DC négatif.

Avant d'extraire un paramètre d'erreurs transitoires des flux de caractéristiques f_{ATI} indiquées sur la Fig. 27, il est judicieux d'augmenter la largeur des pics de mouvement (pics rouges de la Fig. 27), car ces pics de mouvement de courte durée provenant d'erreurs transitoires ne représentent pas correctement les incidences perceptuelles d'erreurs de ce type. Pour augmenter la largeur des pics de mouvement, une méthode consiste à appliquer un filtre maximal aux flux de caractéristiques d'origine et aux flux de caractéristiques traitées avant de calculer la fonction du paramètre d'erreurs entre les deux formes d'onde. Pour le paramètre d'erreurs basé sur la caractéristique f_{ATI} , un filtre maximal de 7 points de largeur (qui sera exprimé ici sous la forme de la fonction *max7pt*) a été utilisé, pour obtenir un échantillon en sortie au niveau de chaque image qui représente le maximum de l'échantillon lui-même et les 3 échantillons voisins les plus proches de chaque côté (c'est-à-dire les échantillons temporels antérieurs et les échantillons ultérieurs).

FIGURE 27

Exemple d'historique temporel de la caractéristique f_{ATI}



BT.1885-27

3.2.5 Quantification des caractéristiques

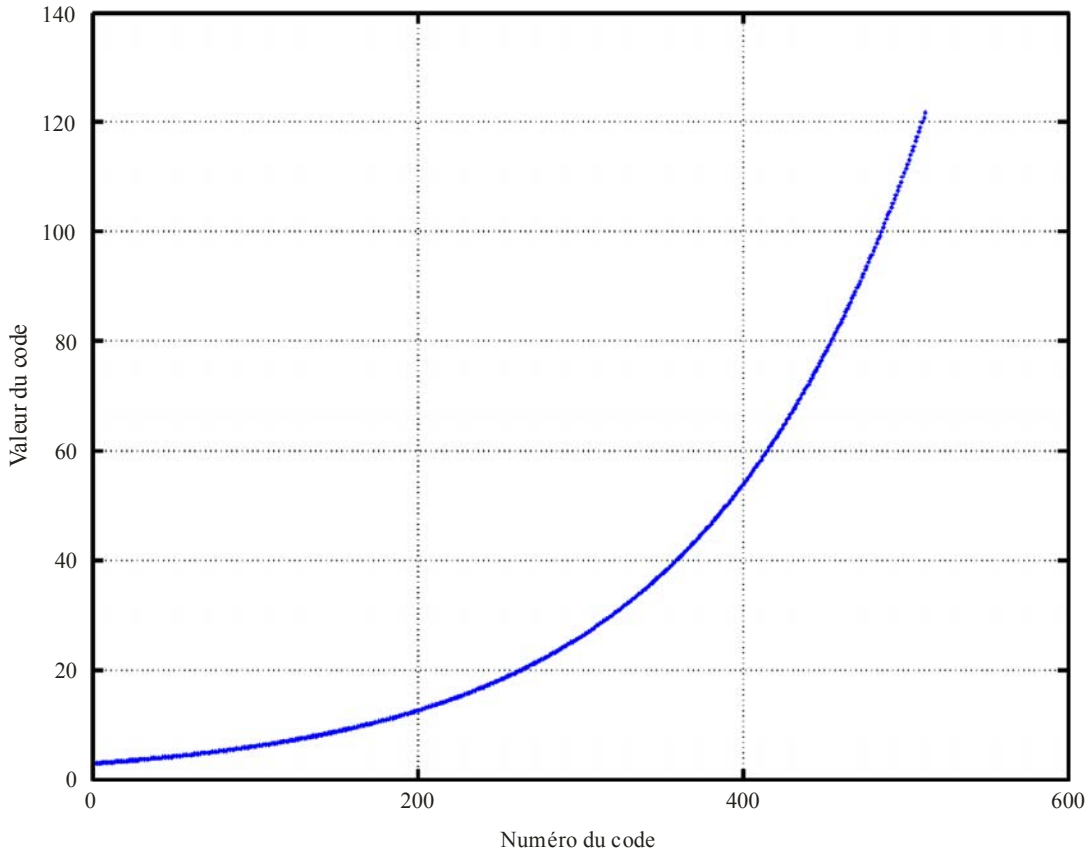
Une quantification avec une précision de 9 bits est suffisante pour les caractéristiques Y_{MEAN}, f_{SI13}, f_{HV13} et f_{COHER_COLOR}, tandis que la caractéristique f_{ATI} devrait être quantifiée à 10 bits. Pour limiter aux maximum les conséquences sur les calculs du paramètre de la qualité vidéo, il conviendra d'utiliser un quantificateur non linéaire dans les cas où l'erreur du quantificateur est proportionnelle à l'intensité du signal faisant l'objet de la quantification. Les valeurs très faibles sont quantifiées de manière uniforme jusqu'à une valeur de coupure, au-dessous de laquelle il n'y a aucune information utile pour l'évaluation de la qualité. Ce type de quantificateur réduit au minimum les erreurs dans

les calculs du paramètre correspondant, étant donné que ces calculs sont normalement effectués sur la base d'une fonction de rapport ou d'une fonction de logarithme des flux de caractéristiques traitées et d'origine (voir le paragraphe sur la description des paramètres ci-dessous).

On trouvera sur la Fig. 28 un graphique du quantificateur non linéaire à 9 bits utilisé pour la caractéristique d'origine f_{SI13}. Le sous-programme du code de référence «model_lowbw_compression» fournit une description mathématique complète des quantificateurs recommandés utilisés par le modèle VQM de faible largeur de bande à débit rapide. Si les caractéristiques ne se situent pas dans la gamme des qantificateurs recommandés à l'extrémité inférieure ou à l'extrémité supérieure (ce qui est fortement improbable), les paramètres S-T issus de ces caractéristiques sont mis à zéro, afin de ne pas influer sur le modèle VQM global.

FIGURE 28

Quantificateur non linéaire à 9 bits pour la caractéristique f_{SI13}



BT.1885-28

3.3 Description des paramètres

3.3.1 Aperçu des paramètres

Plusieurs étapes interviennent dans le calcul des paramètres qui suivent les différents aspects perceptuels de la qualité vidéo. Ces étapes peuvent consister:

– à appliquer un seuil perceptuel aux caractéristiques extraites de chaque sous-région S-T;

– à calculer une fonction d'erreur entre les caractéristiques traitées et les caractéristiques d'origine correspondantes;

– à rassembler l'erreur qui en résulte dans l'espace et dans le temps.

Voir l'Annexe D.8 de la Recommandation UIT-T J.144 pour obtenir une description détaillée de ces techniques et de la notation mathématique correspondante pour les noms des paramètres, qui seront également utilisés ici. Le sous-programme du code de référence «model_fastlowbw_parameters» fournit une description mathématique complète des paramètres utilisés par le modèle VQM de faible largeur de bande à débit rapide. Dans un souci de simplicité, la description des paramètres donnés dans le présent paragraphe ne tient pas compte des effets de la quantification des caractéristiques (par exemple le traitement des valeurs des caractéristiques susceptibles de se situer en dehors des gammes de quantification recommandées).

3.3.2 Nouvelles méthodes

On trouvera dans le présent paragraphe un résumé de nouvelles méthodes mises au point pour améliorer la corrélation objective/subjective des paramètres fondés sur les caractéristiques RR ayant de très faibles largeurs de bande de transmission, telles que celles qui sont utilisées pour le modèle VQM de faible largeur de bande à débit rapide de la NTIA (ces nouvelles méthodes n'apparaissent pas dans la Recommandation UIT-T J.144). Il convient de noter qu'aucune amélioration n'a été apportée pour la forme de base des fonctions d'erreur de paramètre indiquée dans l'Annexe D.8.2.1 de la Recommandation UIT-T J.144. Les deux fonctions d'erreur produisant systématiquement les meilleurs résultats de paramètres (pour les paramètres spatio-temporels) sont une fonction de logarithme $\{\log10\ [f_p(s,t) / f_o(s,t)]\}$ et une fonction de rapport $\{[f_p(s,t) - f_o(s,t)] / f_o(s,t)\}$, où $f_p(s,t)$ et $f_o(s,t)$ sont la caractéristique traitée et la caractéristique d'origine correspondante extraite de la région S-T avec les coordonnées spatiales s et les coordonnées temporelles t, respectivement. Il faut séparer les erreurs en gains et pertes, étant donné que l'être humain réagit différemment à des dégradations positives (par exemple une subdivision en blocs) et négatives (par exemple un flou). En appliquant un seuil perceptuel plus bas aux caractéristiques avant d'appliquer ces deux fonctions d'erreur, on empêche la division par zéro.

Après avoir calculé les paramètres S-T au moyen de l'une des fonctions d'erreur, on doit regrouper les paramètres S-T dans l'espace et dans le temps pour obtenir une valeur de paramètre pour le clip vidéo. Ce rassemblement d'erreurs peut s'effectuer en plusieurs étapes (par exemple dans l'espace, puis dans le temps). Le modèle VQM de faible largeur de bande à débit rapide utilise une nouvelle méthode de rassemblement d'erreurs appelée «rassemblement d'erreurs de macroblocs» (MB). Le rassemblement d'erreurs MB regroupe un nombre contigu de sous-régions S-T et applique une fonction de regroupement des erreurs à cet ensemble. Ainsi, la fonction exprimée sous la forme «MB(3,3,2)max» appliquera une fonction max sur les valeurs de paramètres issues de chaque groupe de 18 sous-régions S-T empilées sous la forme de trois verticales × 3 horizontales × 2 temporelles. Pour les sous-régions S-T 32 × 32 × 1s des caractéristique f_{SI13}, f_{HV13}, et f_{COHER_COLOR} décrites plus haut, chaque région MB(3,3,2) comprendra une partie du flux vidéo qui couvre 96 lignes verticales × 96 pixels horizontaux × 2 secondes. On a constaté que le rassemblement d'erreurs MB était utile pour suivre les conséquences perceptuelles des dégradations qui sont localisées dans l'espace et dans le temps. Ces dégradations localisées dominent souvent le processus de décision sur la qualité. On peut également mettre en oeuvre le rassemblement d'erreurs MB sous la forme d'un processus de filtrage au lieu de produire une seule valeur en sortie pour chaque macrobloc MB, chaque échantillon S-T étant remplacé par sa valeur filtrée MB, lorsque le macrobloc est centré sur l'échantillon S-T. Ce processus est appelé rassemblement d'erreurs MB avec recouvrement.

Une deuxième méthode de regroupement des erreurs est une sommation généralisée de Minkowski(P,R), définie sous la forme suivante:

$$Minkowski(P,R) = \sqrt[R]{\frac{1}{N}\sum_{i=1}^{N}|v_i|^P}$$

où v_i représente les valeurs de paramètre incluses dans la sommation. Cette sommation pourrait par exemple comporter toutes les valeurs de paramètre à un instant donné (regroupement spatial), ou peut être appliquée aux macroblocs décrits plus haut. La sommation de Minkowski, dans les cas où la puissance P est égale à la racine R, a été utilisée par de nombreux concepteurs de systèmes de mesure de la qualité vidéo aux fins du regroupement des erreurs. La sommation généralisée de Minkowski, dans les cas où $P \neq R$, offre davantage de souplesse pour linéariser la réponse des paramètres individuels à des changements de la qualité perçue. Il s'agit d'une étape nécessaire avant de combiner des paramètres multiples dans une seule estimation de la qualité vidéo perçue, effectuée avec un ajustement linéaire.

3.3.3 Paramètres de couleur

Deux paramètres sont extraits des caractéristiques f_{COHER_COLOR}. L'un de ces paramètres, appelé *color_extreme*, mesure les distorsions de couleur extrêmes susceptibles d'être causées par des blocs colorés dus à des erreurs de transmission. L'autre paramètre, appelé *color_spread*, donne une indication des variations ou de l'étendue des erreurs de couleur. Au lieu d'utiliser la mesure de la distance euclidienne pour quantifier les distorsions (conformément à l'Annexe D.8.2.2 de la Recommandation UIT-T J.144), ces deux paramètres utilisent la racine carrée de la distance de Manhattan. Conformément à la notation mathématique décrite dans l'Annexe D.8.2.2 de la Recommandation UIT-T J.144, où $f_p(s,t)$ et $f_o(s,t)$ représentent la caractéristique f_{COHER_COLOR} à deux dimensions extraite d'une région S-T des flux vidéo traités et d'origine, la fonction de comparaison de cette fonction est donnée par la formule:

$$sqrtmanhat(s,t) = \sqrt{\sum_{C_B,C_R} \left| \underline{f}_p(s,t) - \underline{f}_o(s,t) \right|}$$

Il semble que la mesure de la distance de Manhattan donne de meilleurs résultats que la mesure de la distance euclidienne et la fonction de racine carrée est nécessaire pour linéariser la réponse du paramètre aux changements de qualité. Conformément aux notations mathématiques décrites dans l'Annexe D.8 de la Recommandation UIT-T J.144, les paramètres de couleur sont donnés par les formules:

$$color\_extreme = \text{color\_coher\_color\_30x30\_1s\_mean\_sqrtmanhat\_OMB(3,3,2)above99\%\_Minkosski(0.5,1)}$$

$$color\_spread = \text{color\_coher\_color\_30x30\_1s\_mean\_sqrtmanhat\_OMB(3,3,2)Minkosski(2,4)\_90\%}$$

On calcule ensuite un paramètre de couleur combiné (*color_comb*) contenant la combinaison optimale des paramètres *color_extreme* et *color_spread*, de la façon suivante:

$$color\_comb = 0.691686 * color\_extreme - 0.617958 * color\_spread$$

Ce paramètre *color_comb* à valeur positive est ensuite coupé à l'extrémité inférieure, représentée mathématiquement par la formule (conformément à la notation décrite dans l'Annexe D.8.5 de la Recommandation UIT-T J.144):

$$color\_comb = color\_comb\_clip\_0.114$$

Ce paramètre *color_comb* est inclus dans la combinaison linéaire pour le calcul du modèle VQM.

3.3.4 Paramètres spatiaux

On calcule deux paramètres spatiaux à partir de la caractéristique f_{SI13}, l'un mesurant une perte d'informations spatiales (*si_loss*) et l'autre mesurant un gain d'informations spatiales (*si_gain*). Conformément à la notation mathématique donnée dans l'Annexe D.8 de la Recommandation UIT-T J.144, ces paramètres sont donnés par les formules:

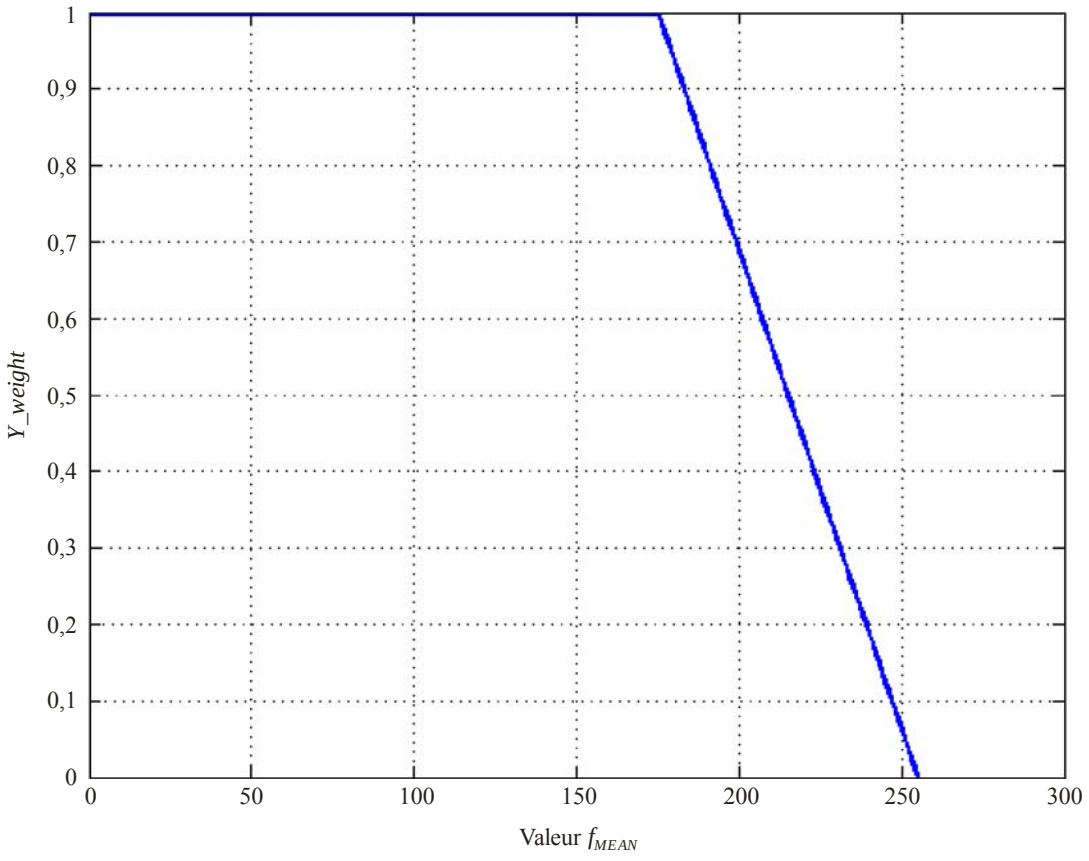
$$si\_loss = \text{avg1s\_Y\_sil3\_30x30\_std\_3\_ratio\_loss\_OMB(3,3,2)Minkosski(1,2)\_Minkosski(1.5,2.5)\_clip\_0.12}$$

$$si\_gain = \text{avg1s\_Y\_sil3\_30x30\_std\_3\_log\_gain\_clip\_0.1\_above95\%tail\_Minkosski(1.5,2)}$$

Plus le niveau moyen de luminance (Y) de la sous-région S-T augmente (c'est-à-dire tel qu'il est mesuré par la caractéristique f_{MEAN}), plus la capacité de percevoir des changements dans les détails spatiaux (par exemple le flou mesuré par *si_loss*) diminue. Pour y remédier, on peut introduire une fonction de pondération (*Y_weight*), comme indiqué sur la Fig. 29, aux valeurs *si_loss* issues de chaque sous-région (c'est-à-dire les valeurs *si_loss* après avoir appliqué la fonction de comparaison ratio_loss à chaque sous-région S-T, mais avant les fonctions de regroupement spatial et temporel). La fonction de pondération *Y_weight* est égal à un (pondération intégrale) jusqu'à ce que l'on atteigne un niveau de luminance moyen de 175, puis diminue de manière linéaire jusqu'à zéro, lorsque les valeurs de luminance passent de 175 à 255. On applique cette correction intermédiaire uniquement aux valeurs *si_loss*, mais non aux valeurs *si_gain*.

FIGURE 29

Fonction de pondération *Y_weight* pour la modification des paramètres *si_loss* S-T



BT.1885-29

On calcule deux paramètres spatiaux à l'aide de la caractéristique f_{HV13}, l'une mesurant une perte d'informations spatiales horizontales et verticales (HV) (*hv_loss*) et l'autre mesurant un gain (*hv_gain*).

Conformément à la notation mathématique donnée dans l'Annexe D.8 de la Recommandation UIT-T J.144, ces paramètres sont donnés par la formule:

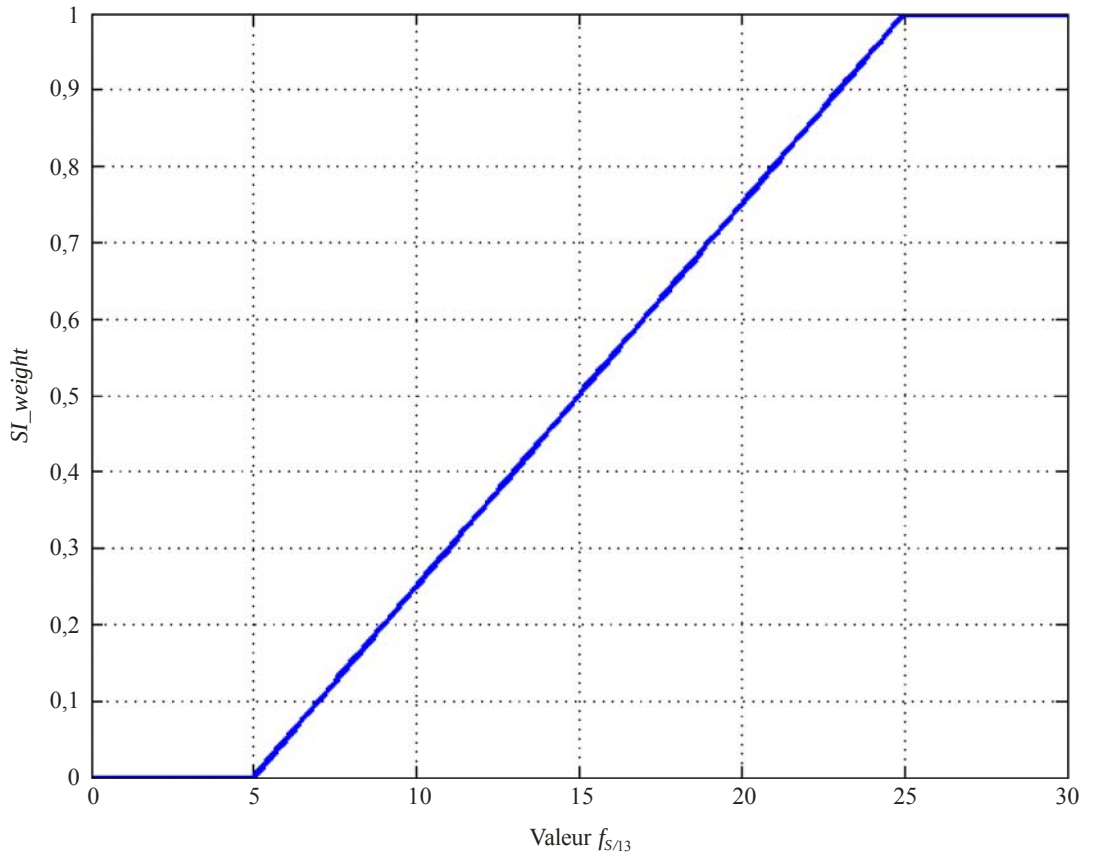
$$hv\_loss = \text{avg1s\_Y\_hv13\_angle0.225\_rmin20\_30x30\_mean\_4\_ratio\_loss\_...}$$
$$\text{OMB(3,3,2)below1\%\_Minkosski(1,1.5)\_clip\_0.08}$$

$$hv\_gain = \text{avg1s\_Y\_hv13\_angle0.225\_rmin20\_30x30\_mean\_4\_log\_gain\_...}$$
$$\text{clip\_0.06\_OMB(3,3,2)above99\%tail\_Minkosski(1.5,3)}$$

Les formules ci-dessus n'indiquent pas que la fonction *Y_weight* décrite sur la Fig. 29 est également appliquée aux valeurs *hv_loss* et *hv_gain* issues de chaque sous-région S-T avant les fonctions de regroupement spatial et temporel (après les calculs des fonctions ratio_loss et log_gain respectivement). On applique une autre fonction de pondération (*SI_weight* comme indiqué sur la Fig. 30) aux valeurs *hv_loss* issues de chaque sous-région S-T, ce qui est nécessaire pour réduire la sensibilité de *hv_loss* pour les régions S-T disposant de très peu d'informations spatiales (c'est-à-dire des valeurs de caractéristiques d'origine low f_{SI13} peu élevées).

FIGURE 30

Fonction de pondération *SI_weight* pour la modification des paramètres *hv_loss* S-T



BT.1885-30

Les paramètres de distorsion spatiale peuvent être écrasés (les excursions excessives autres que celles constatées dans les données d'apprentissage sont limitées ou compressées) au moyen de fonctions telles que la fonction d'écrasement VQM décrite dans le paragraphe sur les calculs du modèle VQM.

3.3.5 Paramètres temporels

On calcule deux paramètres temporels à partir de la caractéristique f_{ATI}, l'un mesurant le bruit aléatoire ajouté (*ati_noise*) et l'autre mesurant les perturbations de mouvement dues à des erreurs de transmission (*ati_error*). Conformément à la notation mathématique donnée dans l'Annexe D.8 de la Recommandation UIT-T J.144, ces paramètres sont donnés par les formules suivantes:

$$ati\_noise = \text{Y\_rand5\%\_ati0.2s\_rms\_5\_ratio\_gain\_between25\%50\%}$$

$$ati\_error = \text{Y\_rand5\%\_ati0.2s\_rms\_max7pt\_12\_ratio\_gain\_above90\%}$$

Pour que les paramètres *ati_noise* et *ati_error* soient plus résistants en cas de désalignement temporel, on calcule les paramètres pour tous les alignements temporels de la séquence vidéo traitée qui sont de l'ordre de ±0,4 s des meilleures estimations de l'alignement temporel par rapport à la séquence vidéo d'origine, puis on choisit la valeur minimale du paramètre.

3.4 Calcul du modèle VQM

Comme pour le modèle général VQM de la NTIA décrit dans l'Annexe D de la Recommandation UIT-T J.144, le calcul du modèle VQM de faible largeur de bande à débit rapide combine de façon linéaire deux paramètres à partir de la caractéristique f_{HV13} (*hv_loss* and *hv_gain*), deux paramètres à partir de la caractéristique f_{SI13} (*si_loss* and *si_gain*) et deux paramètres issus de la caractéristique f_{COHER_COLOR} (sauf que les deux paramètres ont été combinés en un seul paramètre de distorsion de couleur appelé *color_comb*). Le paramètre de bruit unique du modèle général VQM de la NTIA a été remplacé par deux paramètres fondés sur la caractéristique f_{ATI} de faible largeur de bande décrite ici (*ati_noise* and *ati_error*).

En conséquence, le modèle VQM_{FLB} (abréviation pour modèle VQM de faible largeur de bande à débit rapide) est une combinaison linéaire de huit paramètres. Ce modèle VQM_{FLB} est donné par la formule suivante:

$$\begin{aligned} VQM_{FLB} = \{ & 0.38317338378290 * hv\_loss + 0.37313218013131 * hv\_gain + \\ & 0{,}58033514546526 * si\_loss + 0.95845512360511 * si\_gain + \\ & 1{,}07581708014998 * color\_comb + \\ & 0{,}17693274495002 * ati\_noise + 0.02535903906351 * ati\_error \} \end{aligned}$$

Le VQM total (une fois additionnées les contributions de tous les paramètres) est coupé à un seuil inférieur de 0,0 pour éviter les nombres VQM négatifs. Enfin, on applique une fonction d'écrasement permettant une suroscillation maximale de 50% aux valeurs VQM supérieurs à 1,0 pour limiter les valeurs VQM dans les cas de séquences vidéo présentant des distorsions excessives et situées en dehors de la gamme des données d'apprentissage.

Si $VQM_{FLB} > 1.0$, then $VQM_{FLB} = (1 + c)*VQM_{FLB} / (c + VQM_{FLB})$, où $c = 0{,}5$.

En calculant VQM_{FLB} comme indiqué ci-dessus, on obtiendra des valeurs supérieures ou égales à zéro et une valeur maximale nominale de un. VQM_{FLB} peut de temps à autre dépasser un dans le cas de scènes vidéo faisant l'objet de distorsions extrêmes.

Pour que VQM_{FLB} soit plus résistant contre les désalignements spatiaux, on calcule cette valeur pour tous les alignements spatiaux de la séquence vidéo traitée qui sont de l'ordre de plus ou moins un pixel des meilleures estimations de l'alignement spatial par rapport à la séquence vidéo d'origine, puis on choisit la valeur minimale de VQM_{FLB}.

4 Références

[1] Recommandation UIT-R BT.601-6 (01/07) – Paramètres de codage en studio de la télévision numérique pour des formats standard d'image 4:3 (normalisé) et 16:9 (écran panoramique).

[2] Video Quality Model (VQM) Software Tools – Binary executables and source code, available from the National Telecommunications and Information Administration (NTIA) at: http://www.its.bldrdoc.gov/n3/video/VQM_software.php.

[3] Recommandation UIT-T J.244 (04/08) – Méthodes d'étalonnage avec référence complète et référence réduite pour les systèmes de transmission vidéo avec désalignement constant des domaines spatial et temporel avec un gain et un décalage constants.

[4] VQEG Final Report of MM Phase I Validation Test (2008), «Final report from the Video Quality Experts Group on the validation of objective models of multimedia quality assessment, phase I», Video Quality Experts Group (VQEG), http://www.its.bldrdoc.gov/vqeg/projects/multimedia, ITU-T Study Group 9 TD923, Study Period 2005-2008.

[5] Recommandation UIT-T J.144 (03/04) – Techniques de mesure objective de la qualité vidéo perçue pour la télévision numérique par câble en présence d'un signal de référence complet.

[6] Recommandation UIT-R BT.1683 (06/04) – Techniques de mesure objective de la qualité vidéo perceptuelle pour la télédiffusion numérique à définition normale en présence d'une image de référence complète.

5 Code de référence pour la mise en oeuvre du modèle VQM de faible largeur de bande à débit rapide

Le présent code de référence vise à aider les utilisateurs à mettre correctement en oeuvre le modèle VQM de faible largeur de bande à débit rapide. Bien que l'on utilise le code MATLAB® pour le code de référence, on peut faire appel à n'importe quel code logiciel reproduisant les résultats présentés ici. Chaque sous-paragraphe du § 5 contient le code MATLAB pour la fonction indiquée dans le titre de la section (par exemple sauvegarder le contenu du § 5.1 dans un fichier appelé «fastlowbw_ref.m»). Exécuter fastlowbw_ref sans arguments pour recevoir une assistance sur la manière d'appeler le programme. Ce code offre la souplesse nécessaire pour exécuter le modèle sur un clip vidéo de courte durée (c'est-à-dire de 5 à 15 s) dans une séquence vidéo plus longue (par exemple séquence de 1 minute). A cette fin, on décale le clip vidéo de courte durée d'une seconde et on recalcule le modèle pour chaque décalage temporel. Bien que cette fonctionnalité ne soit pas démontrée ci-dessous, les commentaires dans le code et les arguments restitués provenant de «model_fastlowbw_parameters.m» feront mention de cette fonctionnalité. Cette capacité cachée peut être utile pour mettre en oeuvre un système de surveillance de la qualité vidéo en service.

Lorsque les vecteurs d'essai types (c'est-à-dire les clips vidéo) sont traités à l'aide du code de référence du modèle VQM de faible largeur de bande à débit rapide (fonction «fastlowbw_ref.m»), des fichiers de texte sont produits qui contiennent les résultats du calibrage et du modèle. Pour les appels de la fonction MATLAB présentés ci-dessous à titre d'exemple, des fichiers en sortie analogues à ceux présentés ci-après devraient être obtenus (en raison des processus aléatoires utilisés par le modèle VQM de faible largeur de bande à débit rapide, les résultats peuvent être légèrement différents de ceux qui sont présentés ici):

5.Reference Code.doc

Appendice

Analyses des erreurs de transmission

Les tests de validation effectués par le VQEG pour le projet RRNR-TV comprenaient les formats 525 (NTSC) et 625 (PAL). Chaque expérience comprenait 12 séquences source et 156 séquences vidéo traitées (PVS). Sur ces 156 séquences PVS, 40 contenaient des erreurs de transmission et 116 ne contenaient que des erreurs de codage. Les Tableaux 9 et 10 indiquent le RMSE et l'OR pour les séquences PVS contenant des erreurs de transmission. Il convient de noter que le RMSE et l'OR ont été calculés au moyen des lignes de régression qui ont été obtenues à partir de la totalité des données. En d'autres termes, on a calculé les lignes de régression en utilisant toutes les données. En conséquence, le RMSE et l'OR pour les erreurs de transmission ont été calculés au moyen des séquences PVS avec erreurs de transmission.

TABLEAU 9

RMSE et OR pour le test de validation RRNR-TV (format 525).
TE: erreurs de transmission

Format à 525 lignes	Toutes		Avec erreurs de transmission		Sans erreurs de transmission	
	RMSE	OR	RMSE	OR	RMSE	OR
Model_A_15k	0,418	0,385	0,574	0,500	0,362	0,293
Model_A_80k	0,423	0,378	0,582	0,475	0,366	0,293
Model_A_256k	0,424	0,378	0,584	0,475	0,367	0,293
Model_B_80k	0,598	0,667	0,768	0,650	0,544	0,586
Model_B_256k	0,587	0,647	0,763	0,600	0,530	0,578
Model_C_80k	0,465	0,513	0,557	0,550	0,440	0,405
Model_C_256k	0,511	0,609	0,584	0,450	0,495	0,578
PSNR_NTIA	0,556	0,571	0,549	0,500	0,568	0,491

TABLEAU 10

RMSE et OR pour le test de validation RRNR-TV (format 625).
TE: erreurs de transmission

Format à 625 lignes	Toutes		Avec erreurs de transmission		Sans erreurs de transmission	
	RMSE	OR	RMSE	OR	RMSE	OR
Model_A_15k	0,524	0,468	0,597	0,450	0,508	0,414
Model_A_80k	0,513	0,462	0,594	0,500	0,494	0,379
Model_A_256k	0,516	0,468	0,593	0,500	0,499	0,379
Model_B_80k	0,887	0,724	0,545	0,500	0,986	0,716
Model_B_256k	0,864	0,744	0,523	0,600	0,962	0,716
Model_C_80k	0,585	0,583	0,282	0,200	0,663	0,647
Model_C_256k	0,657	0,590	0,292	0,175	0,747	0,638
PSNR_NTIA	0,605	0,564	0,338	0,250	0,678	0,517