

UIT-R

Secteur des Radiocommunications de l'UIT

Recommandation UIT-R BT.1907-0
(01/2012)

**Techniques de mesure objective de la
qualité vidéo perçue pour les applications de
radiodiffusion utilisant la télévision haute
définition en présence d'un signal
de référence complet**

Série BT
Service de radiodiffusion télévisuelle



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 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 œuvre 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, 2020

© UIT 2020

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.1907-0*

Techniques de mesure objective de la qualité vidéo perçue pour les applications de radiodiffusion utilisant la télévision haute définition en présence d'un signal de référence complet

(2012)

Domaine d'application

La présente Recommandation définit les méthodes à appliquer pour évaluer la qualité vidéo perçue des applications de radiodiffusion utilisant la télévision haute définition lorsqu'un signal de référence complet est disponible.

L'Assemblée des radiocommunications de l'UIT,

considérant

- a) qu'il n'est plus à démontrer depuis longtemps que la possibilité de mesurer automatiquement la qualité d'une séquence vidéo diffusée constitue un atout précieux pour le secteur d'activité concerné;
- b) que la Recommandation UIT-R BT.1683 décrit des techniques de mesure objective de la qualité vidéo perçue pour la télédiffusion numérique à définition normale en présence d'une image de référence complète;
- c) que la Recommandation UIT-R BT.709 donne la valeur de paramètres des normes de télévision haute définition pour la production et l'échange international de programmes et la Recommandation UIT-R BT.500 décrit des méthodes d'évaluation subjective de la qualité d'image y compris télévision haute définition;
- d) que la télévision haute définition (TVHD) est de plus en plus utilisée en radiodiffusion;
- e) que, sur la base des résultats figurant dans le rapport sur la TVHD envoyé par le Groupe VQEG, la Commission d'études 9 de l'UIT-T a élaboré la Recommandation UIT-T J.341, relative à la mesure objective de la qualité vidéo perçue pour la TVHD en présence d'une référence complète;
- f) qu'une mesure objective de la qualité vidéo perçue pour la TVHD peut compléter les méthodes d'évaluation subjective,

recommande

1 d'utiliser le modèle décrit dans les Annexes 1, 2 et 3 en ce qui concerne la mesure objective de la qualité vidéo perçue pour les applications de radiodiffusion utilisant la TVHD en présence d'un signal de référence complet.

* La Commission d'études 6 des radiocommunications a apporté des modifications rédactionnelles à la présente Recommandation en février 2020 conformément aux dispositions de la Résolution UIT-R 1.

Annexe 1

1 Introduction

La présente Recommandation décrit une méthode de mesure de la qualité vidéo perçue à utiliser pour les applications non interactives de télévision haute définition (TVHD) lorsqu'il est possible d'utiliser une méthode de mesure avec référence complète. Le modèle a été élaboré afin d'évaluer les notes de qualité subjectives.

Il est possible d'utiliser une méthode de mesure avec référence complète lorsque le signal vidéo de référence non dégradé est aisément disponible au point de mesure; tel peut être le cas des mesures effectuées en laboratoire ou dans un environnement fermé (station de télédiffusion par exemple) sur un seul équipement ou sur une chaîne de transmission. La méthode d'évaluation comporte un étalonnage et des évaluations objectives de la qualité vidéo.

Les séquences utilisées pour les tests de validation contenaient à la fois des dégradations dues au codage H.264 et MPEG-2 et diverses erreurs de transmission (par exemple erreurs sur les bits, perte de paquets). Le modèle décrit dans la présente Recommandation peut être utilisé pour contrôler la qualité de réseaux déployés et s'assurer que ces réseaux sont prêts à être exploités. Les effets visuels des dégradations peuvent comprendre des dégradations aussi bien spatiales que temporelles. Le modèle décrit dans la présente Recommandation peut également être utilisé pour des tests en laboratoire de systèmes vidéo. Lorsque ce modèle est utilisé pour comparer différents systèmes vidéo, il est recommandé d'utiliser une méthode quantitative (par exemple celle de la Recommandation UIT-T J.149) pour déterminer la précision du modèle dans ce contexte particulier.

La présente Recommandation est censée être applicable aux services de radiodiffusion fournis à un débit compris entre 1 Mbit/s et 30 Mbit/s. Les résolutions et fréquences d'image ci-après ont été utilisées dans les tests de validation:

- 1080/59,94/I
- 1080/25/P
- 1080/50/I
- 1080/29,97/P

Les tests de validation ont été effectués, pour chaque résolution, dans les conditions ci-après:

Facteurs de test
Résolution vidéo: 1920x1080 entrelacé et progressif
Fréquences d'image: 29,97 et 25 images par seconde
Débits binaires vidéo: 1 à 30 Mbit/s
Gel d'image temporel (pause avec saut) de 2 secondes au maximum
Erreurs de transmission avec perte de paquets
Conversion du circuit SRC de 1080 à 720/P, compression, transmission, décompression, puis nouvelle conversion à 1080.
Technologies de codage
H.264/AVC (MPEG-4 Partie 10)
MPEG-2

Il est à noter que la résolution 720/P a été utilisée dans le plan des tests de validation comme condition de test (HRC). Etant donné qu'à l'heure actuelle, la résolution 720/P fait généralement l'objet d'une conversion vers une résolution plus haute pour l'affichage, on a estimé que des circuits HRC 720/P seraient mieux adaptés pour ce format.

1.1 Application

Les applications du modèle d'évaluation décrit dans la présente Recommandation sont notamment les suivantes (la liste n'est pas exhaustive):

- 1) Contrôle de la qualité en service, éventuellement en temps réel, à la source.
- 2) Télécontrôle de la qualité au point de destination lorsqu'une copie de la source est disponible au point de mesure.
- 3) Mesure de la qualité en vue du contrôle d'un système de stockage ou de transmission qui utilise un seul jeu ou une concaténation de techniques de compression et de décompression vidéo.
- 4) Tests en laboratoire de systèmes vidéo.

1.2 Limitations

Le modèle d'évaluation de la qualité vidéo décrit dans la présente Recommandation ne peut pas être utilisé pour remplacer des tests subjectifs. Les valeurs de corrélation entre deux tests 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 des systèmes vidéo (comparaison de deux codecs par exemple), il est recommandé d'utiliser une méthode quantitative pour déterminer la précision du modèle dans ce contexte particulier.

Lorsqu'un gel d'image était présent dans les conditions de test, sa durée était généralement inférieure à 2 secondes. Le modèle décrit dans la présente Recommandation n'a pas été validé pour mesurer la qualité vidéo dans une situation de remise en tampon (séquence vidéo avec un délai augmentant constamment ou gel d'image sans saut). Le modèle n'a pas été testé pour des fréquences d'image autres que celles utilisées dans les systèmes de télévision (à savoir 29,97 images par seconde et 25 images par seconde, en mode entrelacé ou progressif).

Il convient de noter que, dans le cas de nouvelles technologies de codage et de transmission introduisant des artéfacts qui n'ont pas été pris en compte dans la présente évaluation, le modèle de mesure objective pourrait donner des résultats erronés. Dans ce cas, une évaluation subjective est indispensable.

2 Références

Aucune.

3 Définitions

3.1 Termes définis ailleurs

La présente Recommandation utilise les termes suivants définis ailleurs:

3.1.1 Evaluation subjective (image): détermination de la qualité ou de la dégradation d'images de type «programme de télévision» présentées à un groupe d'évaluateurs pendant des séances de visionnement.

3.1.2 Mesure objective de la qualité perçue (image): mesure de la qualité de fonctionnement d'une chaîne de programme par l'emploi d'images de type «programme de télévision» et de méthodes de mesure objective (au moyen d'instruments) pour obtenir une indication proche de la note qui serait obtenue au moyen d'une évaluation subjective.

3.1.3 Proposant: organisation ou entreprise qui propose de soumettre un modèle de mesure de la qualité vidéo à des tests de validation, en vue de son inclusion éventuelle dans une Recommandation de l'UIT.

3.2 Termes définis dans la présente Recommandation

La présente Recommandation définit les termes suivants:

3.2.1 Fréquence d'image: nombre d'images uniques (nombre total d'images – nombre d'images répétées) par seconde.

3.2.2 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 simulés et les erreurs sur les bits simulées. Les paramètres utilisés pour contrôler les erreurs de transmission simulées sont bien définis.

3.2.3 Erreurs de transmission: toute erreur imposée à la transmission vidéo. Comme exemples de type d'erreur, on citera les erreurs de transmission simulées et les conditions de réseau en exploitation.

4 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

ACR	évaluation par catégories absolues (<i>absolute category rating</i>) (voir UIT-R BT.500)
ACR-HR	évaluation par catégories absolues avec référence cachée (<i>absolute category rating with hidden reference</i>) (voir UIT-T P.910)
AVI	entrelacement audio vidéo (<i>audio video interleave</i>)
DMOS	note d'opinion moyenne différentielle (<i>difference mean opinion score</i>)
FR	référence complète (<i>full reference</i>)
FRTV	télévision avec référence complète (<i>full reference television</i>)
HRC	circuit de référence fictif (<i>hypothetical reference circuit</i>)
ILG	groupe de laboratoires indépendants (<i>independent laboratory group</i>) du VQEG
MOS	note d'opinion moyenne (<i>mean opinion score</i>)
MOSp	note d'opinion moyenne prédite (<i>mean opinion score, predicted</i>)
NR	pas de référence ou référence nulle (<i>no (or zero) reference</i>)
PSNR	rapport signal de crête sur bruit (<i>peak signal-to-noise ratio</i>)
PVS	séquence vidéo traitée (<i>processed video sequence</i>)
RMSE	erreur quadratique moyenne (<i>root mean square error</i>)
RR	référence réduite (<i>reduced reference</i>)
SFR	fréquence d'image de la source (<i>source frame rate</i>)
SRC	canal ou circuit de référence de la source (<i>source reference channel or circuit</i>)
VQEG	groupe d'experts en qualité vidéo (<i>video quality experts group</i>)
YUV	espace chromatique et format de fichier

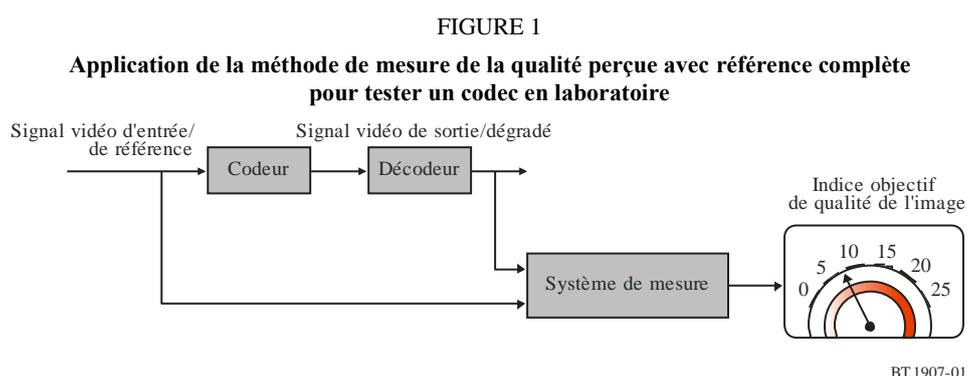
5 Conventions

Aucune.

6 Description de la méthode avec référence complète

La méthode de mesure aux deux extrémités avec référence complète, conçue pour mesurer de façon objective la qualité vidéo perçue, permet d'évaluer la performance de systèmes en comparant le signal vidéo de référence non distordu 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



La comparaison entre le signal d'entrée et le signal de sortie peut nécessiter un alignement temporel ou un alignement spatial, ce dernier pour compenser les éventuels décalages 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.

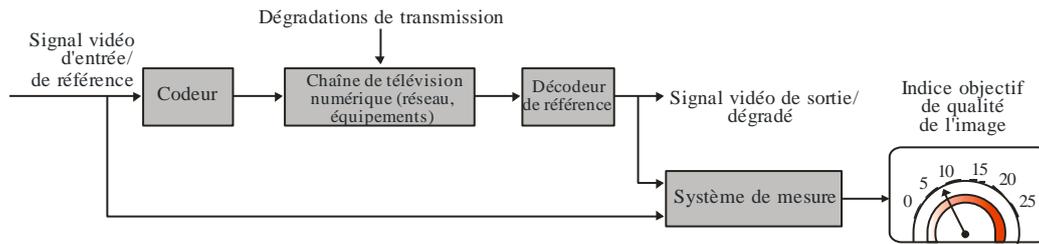
La normalisation désigne l'alignement et le réglage du gain. Cette opération est nécessaire puisque la plupart des méthodes avec référence complète comparent les images traitées et les images de référence effectivement pixel par pixel. Le modèle de mesure de la qualité vidéo décrit dans l'Annexe 2 utilise des méthodes de normalisation.

Comme les mesures de qualité vidéo sont généralement fondées sur des approximations des réponses visuelles humaines et non sur la mesure d'artéfacts de codage particuliers, elles sont en principe valables aussi bien pour les systèmes analogiques et pour les systèmes numériques. Elles sont aussi valables en principe pour les chaînes de transmission 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 complète pour tester une chaîne de transmission.

FIGURE 2

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



BT.1907-02

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 communément admis que la méthode avec référence complète 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 Recommandation UIT-T P.910.

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

Un groupe informel, le Groupe d'experts en qualité vidéo (VQEG), 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 de phase I sur la télévision haute définition récemment mené à bien par le Groupe VQEG a permis d'évaluer la performance des algorithmes proposés de mesure de la qualité vidéo perçue avec référence complète.

Les statistiques qui suivent sont extraites du rapport final sur la TVHD du Groupe VQEG. Il est à noter que ce rapport contient aussi d'autres informations: corrélation de Pearson et erreur RMSE calculées dans des expériences individuelles, intervalles de confiance, tests de significativité statistique dans des expériences individuelles, analyse de sous-ensembles des données qui incluent des dégradations spécifiques (par exemple codage H.264 uniquement), diagrammes de dispersion, et coefficients d'ajustement.

Analyse principale

La performance du modèle avec référence complète retenu est récapitulée dans le Tableau 1. Le rapport PSNR est calculé conformément à la Recommandation UIT-T J.340 et inclus dans cette analyse à des fins de comparaison. «Superset RMSE» désigne le paramètre principal (RMSE) calculé dans le superensemble agrégé (les six expériences reportées sur une même échelle). «Top performing group total» désigne le nombre d'expériences (0 à 6) pour lesquels ce modèle a été le modèle le plus performant ou statistiquement équivalent au modèle le plus performant. «Better than PSNR total» désigne le nombre d'expériences (0 à 6) pour lesquelles le modèle a été statistiquement meilleur que le PSNR. «Better Than Superset PSNR» indique si le modèle est statistiquement meilleur que le PSNR dans le superensemble agrégé. «Superset Correlation» désigne la corrélation de Pearson calculée dans le superensemble agrégé.

TABLEAU 1

Paramètre	PSNR	SwissQual
Superset RMSE	0,71	0,56
Top performing group total	1	5
Better than PSNR total	–	4
Better than superset PSNR	–	Oui
Superset correlation	0,78	0,87

Annexe 2

Description du modèle

Note de l'éditeur: Le code source à incorporer dans cette section, qui fait partie intégrante de la présente Recommandation, est disponible à l'adresse: <http://ifatemp.itu.int/t/2009/sg9/exchange/q2/>.

Aperçu du modèle

Le modèle prédit la qualité vidéo telle qu'elle est perçue par les sujets dans une expérience. Le modèle de prédiction utilise une modélisation psycho-visuelle et d'origine cognitive pour émuler la perception subjective.

En tant qu'approche avec référence complète, le modèle compare la séquence vidéo de référence de haute qualité et la séquence vidéo dégradée associée (voir la Fig. 3).

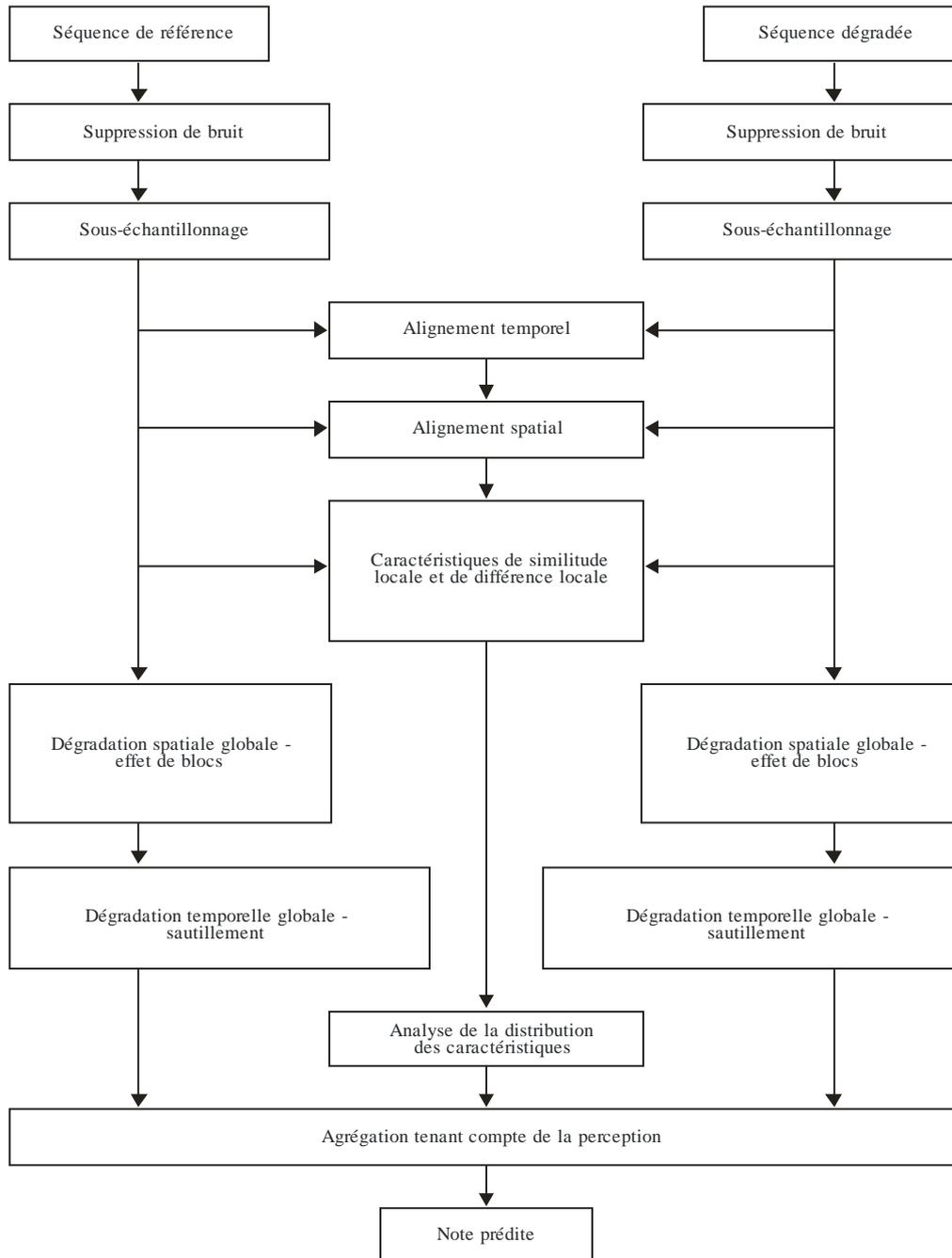
Les étapes menant à l'évaluation de la note sont les suivantes:

- 1) Les séquences vidéo sont d'abord prétraitées. En particulier, le bruit est supprimé par filtrage des images et les images sont sous-échantillonnées.
- 2) Les images de la séquence vidéo de référence et de la séquence vidéo traitée sont alignées temporellement.
- 3) Chaque image de la séquence vidéo traitée et l'image correspondante de la séquence vidéo de référence sont alignées spatialement.
- 4) Des caractéristiques de qualité spatiale locale sont calculées: similitude locale et différence locale, compte tenu de la perception visuelle.
- 5) La distribution des caractéristiques de similitude locale et de différence locale est analysée.
- 6) Une dégradation spatiale globale est mesurée au moyen d'une caractéristique d'*effet de blocs*.
- 7) Une dégradation temporelle globale est mesurée au moyen d'une caractéristique de *sautillement*. Le sautellement est déduit de l'évaluation de l'intensité des mouvements locaux et globaux et de la durée d'affichage des images.
- 8) La note de qualité est évaluée sur la base d'une agrégation non linéaire des caractéristiques ci-dessus.
- 9) Afin d'éviter toute prédiction erronée dans le cas d'un défaut d'alignement spatial relativement important entre la séquence vidéo de référence et la séquence vidéo traitée, les étapes ci-dessus sont suivies pour 3 alignements spatiaux horizontaux et verticaux différents de la séquence vidéo, et la note de qualité finale évaluée est donnée par la note maximale prédite parmi toutes les positions spatiales.

Les différentes étapes sont expliquées plus en détail aux § 2.1 à 2.9. Le § 2.10 renvoie au code source en C++ couvrant les parties et fonctions essentielles pour une mise en œuvre conforme à la description du modèle. Les noms de fonction en C++ mentionnés aux § 2.1 à 2.9 renvoient à ce code source de référence (par exemple le § 2.2 renvoie à `CFrameAnalysisFullRef::ContentTimeAlignment`).

FIGURE 3

Diagramme des étapes de traitement du modèle. En haut, se trouvent les signaux d'entrée, à savoir la séquence vidéo de référence et la séquence vidéo dégradée (ou traitée). En bas, se trouve le principal résultat du modèle obtenu après les différentes étapes de traitement, à savoir la note prédite



2.1 Prétraitement

Chaque image de la séquence vidéo de référence et de la séquence vidéo traitée passe par un filtre passe-bas spatial et est sous-échantillonnée pour donner les trois résolutions R1, R2 et R3:

	image d'origine	R1	R2	R3
hauteur x largeur	1080×1920	$\rightarrow 540 \times 960$	$\rightarrow 270 \times 480$	$\rightarrow 96 \times 128$

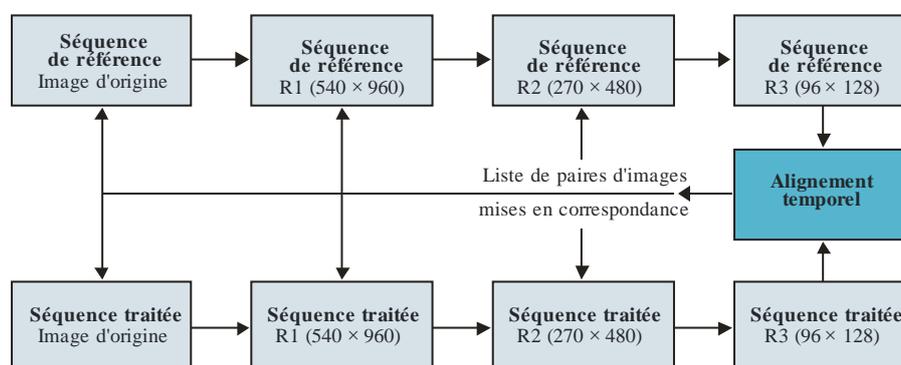
Voir les méthodes `CFrameAnalysisFullRef::ContentTimeAlignment` pour la production d'images de résolution R3 et `CFrameSeq::ReadFrame` pour la production d'images de résolution R1 ou R2.

Il est à noter que la mise en œuvre n'est pas immédiate en raison de contraintes liées à la mémoire.

La Fig. 4 illustre les trois résolutions issues du sous-échantillonnage.

FIGURE 4

Les images de la séquence vidéo de référence et de la séquence vidéo traitée passent par un filtre passe-bas et sont sous-échantillonnées pour donner trois résolutions différentes. La plus petite résolution (R3) sert à aligner temporellement les images. La liste résultante d'images mises en correspondance peut servir pour la mise en correspondance des images pour n'importe quelle autre résolution



BT.1907-04

2.2 Alignement temporel

L'alignement temporel est effectué à partir de la séquence vidéo de référence et de la séquence vidéo traitée de résolution R3 (faible résolution).

L'alignement temporel est effectué de manière récursive comme suit:

- 1) Trouver une image d'ancrage dans la séquence de référence (`Ref_anchor`).
- 2) Déterminer l'image qui correspond le mieux à cette image dans la séquence dégradée (`Deg_best_match`).

Prendre cette image correspondant le mieux dans la séquence dégradée (Deg_best_match) et la comparer aux images proches de l'image d'ancrage dans la séquence de référence (Ref_anchor). Essayer de trouver une meilleure correspondance, selon un critère de similitude, entre l'image Deg_best_match et les images proches de l'image Ref_anchor et stocker la paire d'images se correspondant le mieux ainsi obtenue. Comme critère de similitude entre le plan Y de l'image traitée x et de l'image de référence y , on utilise la fonction:

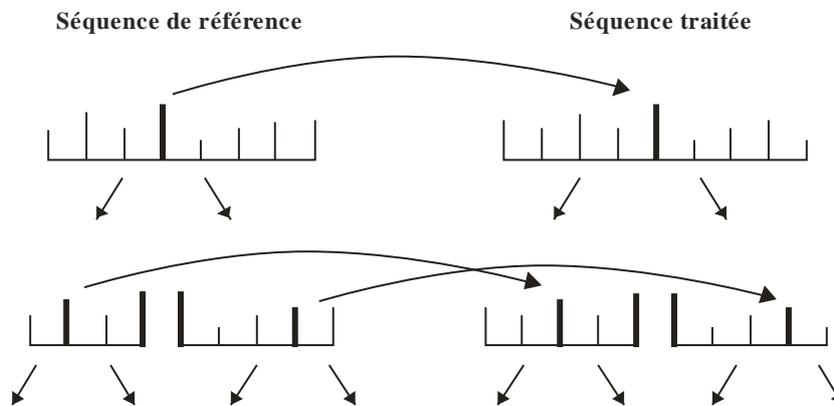
$$\text{sim} = \exp(-\text{mean_square_diff}(a*x+b, y)) \quad (2.1)$$

les paramètres a , b étant choisis de manière à réduire le plus possible la différence quadratique moyenne entre les valeurs du plan Y de l'image traitée x et de l'image de référence y ; voir la méthode `FrameSimilarity::similarity` dans la mise en œuvre de référence.

- 1) Si la paire d'images mises en correspondance est une bonne correspondance (le seuil d'acceptation pour le critère de similitude est respecté), subdiviser la séquence vidéo de référence et la séquence vidéo traitée au niveau de cette paire d'images, chacune en deux séquences vidéo avant et après l'image de la paire. Commencer à 1) avec les deux paires de sous-séquences de référence et dégradée.
- 2) Si la paire d'images mises en correspondance n'est pas une bonne correspondance, recommencer à 1) avec une image d'ancrage différente dans la séquence vidéo de référence. Etant donné qu'on ne connaît pas à l'avance la valeur attendue pour une bonne correspondance, le seuil de correspondance est abaissé de manière itérative. Les valeurs suivantes ont été déterminées à partir de nombreux échantillons de données d'apprentissage: le seuil de départ, en ce qui concerne le critère de similitude donné par la formule (2.1), est de 0,98. En l'absence de bonne correspondance pour 10 images d'ancrage, le seuil est abaissé d'un facteur égal à 0,98, et on recommence à 1). Au plus 10 autres images d'ancrage sont alors essayées et, en l'absence de bonne correspondance, la limite est de nouveau abaissée. Et ainsi de suite jusqu'à atteindre une valeur minimale de 0,1. Pour les détails complets sur la mise en œuvre, voir `SQ_TimeAlignment::findAnchorAndDescend`.

FIGURE 5

Illustration de l'approche récursive utilisée pour l'alignement temporel. Une image d'ancrage de la séquence de référence est mise en correspondance avec une image de la séquence traitée. Les deux séquences sont ensuite subdivisées et dans chaque sous-séquence, une image d'ancrage est choisie et mise en correspondance



Le résultat de l'alignement temporel est une séquence (en fait une liste de correspondances), dans laquelle chaque image de la séquence vidéo traitée est associée à une image de la séquence de référence ou à un indicateur indiquant qu'aucune correspondance suffisamment bonne n'a pu être trouvée. Ainsi, pour les étapes de traitement ultérieures, chaque image mise en correspondance de la séquence vidéo traitée a une image correspondante dans la séquence vidéo de référence. Les images de la séquence vidéo traitée ayant l'indicateur d'absence de mise en correspondance seront comparées aux deux images de référence correspondant aux images mises en correspondance précédente et suivante de la séquence vidéo traitée. Il est à noter que la valeur choisie pour la limite de correspondance est très faible, pour que seules les images très fortement dégradées aient l'indicateur d'absence de mise en correspondance.

Pour tous les détails sur la mise en œuvre, voir la méthode

CFrameAnalysisFullRef::sqVTA_ContentFrameTimeAlignement_M.

2.3 Alignement spatial des images

Procéder à une itération sur toutes les images de la séquence vidéo traitée et:

- 1) Si l'image considérée n'est pas mise en correspondance, utiliser l'alignement spatial précédent. Si l'image considérée est mise en correspondance, effectuer un alignement spatial entre l'image traitée et l'image de référence correspondante, d'après la liste de correspondances de l'alignement temporel:
 - a) Pour la première image, initialiser le décalage spatial à 0 (horizontalement et verticalement). Pour les images suivantes, utiliser l'alignement spatial de la précédente image mise en correspondance.
 - b) Procéder à une itération sur tous les décalages spatiaux possibles (horizontaux et verticaux), dans les limites indiquées au point 2) ci-dessous. Si un décalage spatial différent conduit à une différence nettement plus petite (donnée par une fonction de coût) entre l'image traitée et l'image de référence correspondante, le décalage spatial est ajusté. Comme fonction de coût, on utilise la fonction:

$$\text{rmse}(Y(dv, dh), Y_{\text{ref}}) + \text{abs}(dv) + \text{abs}(dh),$$
 où Y désigne le plan Y de l'image traitée de résolution R1, Y_ref celui de l'image de référence de résolution R1 et Y(dv,dh) celui de l'image traitée, décalée de dv et dh, où dv et dh sont les décalages vertical et horizontal. Le deuxième terme et le troisième terme sont inclus dans la fonction de coût pour favoriser les petits décalages spatiaux. Il est à noter qu'un faible pourtour des images est ignoré pour le calcul de l'erreur rmse, afin d'éviter un traitement plus complexe du pourtour.
 - c) Ainsi, les décalages spatiaux variables dans le temps peuvent être compensés. Les défauts d'alignement d'une image peuvent être corrigés par l'alignement des images suivantes.
- 2) Cette première étape de l'alignement spatial automatisé est limitée aux décalages de ± 4 pixels. Pour les décalages spatiaux plus grands, voir le § 2.9.
- 3) Après l'alignement spatial, chaque image de la séquence vidéo traitée a une image de référence correspondante (ou deux dans le cas d'une image non mise en correspondance) – d'après la liste de correspondances de l'alignement temporel – et une correction du décalage spatial bien définie. Ainsi, les images de la séquence vidéo traitée peuvent être comparées avec précision aux images de la séquence de référence. Ces alignements sont fondamentaux pour les extractions de caractéristiques à suivre.

Pour tous les détails sur la mise en œuvre, voir la méthode

`CFrameAnalysisFullRef::DetermineSpatialAlignment`. Le seuil constant indiqué à l'étape 2 (± 4 pixels) peut être augmenté afin de prendre en compte des décalages spatiaux plus grands.

2.4 Calcul des caractéristiques de similitude locale et de différence locale

Pour chaque paire d'images alignées, on calcule un ensemble de caractéristiques de qualité spatiale:

On calcule d'abord des caractéristiques de *similitude* locale et de *différence* locale par une itération sur des régions carrées contiguës de taille 13 x 13 réparties de la même façon dans l'image traitée et dans l'image de référence de résolution R2. Etant donné que la résolution R2 n'est pas divisible par 13, on ignore un faible pourtour.

Les régions locales étant appelées s_prc et s_ref , la similitude S et la différence D sont données par:

$$S = (\text{cor}(s_prc, s_ref) + 25) / (\text{var}(s_ref) + 25) \quad (4.1)$$

$$D = \text{sqrt}(\text{avg}((S * (s_prc - \text{mean}(s_prc)) - (s_ref - \text{mean}(s_ref)))^2)) \quad (4.2)$$

où cor est la corrélation et var la variance des valeurs de pixel dans la région carrée correspondante. La fonction avg calcule la moyenne sur tous les pixels de la région carrée, et sqrt désigne la racine carrée. La valeur de la qualité spatiale dépend essentiellement des valeurs D et S .

A ce stade, les caractéristiques de similitude et de différence S et D sont données sous forme de matrice de valeurs pour chaque image, une valeur correspondant à chaque région locale carrée. Ce qui est important pour la qualité perçue, c'est non seulement la valeur moyenne, mais aussi la forme de la distribution de S et D respectivement.

2.5 Analyse de la distribution des caractéristiques locales

Le présent paragraphe commence par la présentation de certaines notations:

Désignons par $\text{quantile}(X, c)$ le *quantile* d'ordre c de la distribution des valeurs d'un vecteur ou d'une matrice X . Plus précisément, pour un vecteur X et une constante c avec $0 \leq c \leq 1$, le quantile

$$q = \text{quantile}(X, c)$$

est la valeur q pour laquelle une fraction c de toutes les valeurs de X est inférieure ou égale à q .

La fonction *trimmed_mean* est définie ci-après. Cette notation sera utilisée ultérieurement. Pour une matrice x , la *moyenne tronquée*

$$\text{trimmedMean}(X, c)$$

est la moyenne de toutes les valeurs de x comprises entre les quantiles d'ordre c et $(1-c)$ de x .

Par exemple $\text{trimmedMean}(X, 0.1)$ est la moyenne de toutes les valeurs de x , exception faite des 10% des valeurs les plus petites et des 10% des valeurs les plus grandes de x .

La notation $x(X > c)$ désigne l'ensemble de toutes les valeurs de x supérieures à c . Par exemple

$$\text{trimmedMean}(X, c) = \text{mean}(X(X > \text{quantile}(X, c) \text{ and } X < \text{quantile}(X, 1-c)))$$

Compte tenu de ces notations, les valeurs des caractéristiques suivantes sont calculées sur la base de s donné par la formule (4.1) et de D donné par la formule (4.2):

$$s_m = \text{trimmedMean}(S, c) \quad (5.1)$$

$$d_m = \text{trimmedMean}(D, c) \quad (5.2)$$

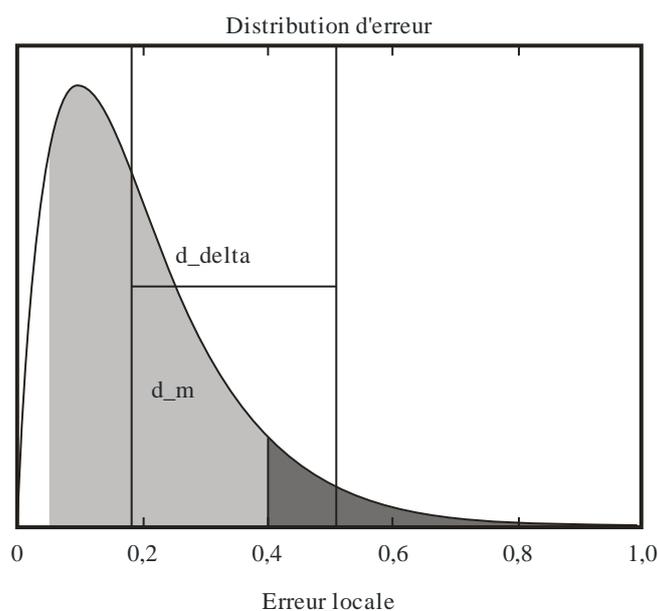
$$s_delta = s_m - \text{mean}(S(S < \text{quantile}(S, c))) \quad (5.3)$$

$$d_delta = \text{mean}(D(D > \text{quantile}(D, 1-c))) - d_m \quad (5.4)$$

avec $c=0,2$. Une représentation graphique est donnée sur la Fig. 6, avec d_m et d_delta .

FIGURE 6

Est représentée la distribution d'une caractéristique locale D . La moyenne tronquée d_m correspond à la moyenne sur la région en gris clair (ligne verticale noire). La valeur d_delta correspond à la différence entre la moyenne des valeurs de la région en gris foncé et celle des valeurs de la région en gris clair (ligne horizontale)



Voir la méthode `CFrameAnalysisFullRef::ComputeSimilarity` pour le calcul de s et de D .

2.6 Calcul de la caractéristique d'effet de blocs

On calcule une caractéristique d'*effet de blocs* sur les images de résolution $R1$. Cette caractéristique mesure la visibilité du pourtour de blocs liée au codage et/ou aux erreurs de transmission. Le calcul étant fait à la résolution $R1$, on se focalise automatiquement sur la visibilité perçue des contours. D'une manière générale, pour la caractéristique d'effet de blocs, on calcule:

- 1) Des dérivées pour les contours horizontaux et verticaux (images des contours). On obtient alors deux matrices, une pour les contours horizontaux et une pour les contours verticaux, pour chaque image de la séquence vidéo, appelées `verGrad_n` et `horGrad_n` dans le pseudocode ci-dessous.
- 2) Une somme logarithmique sur les lignes et sur les colonnes pour les contours horizontaux et les contours verticaux, donnant deux vecteurs, l'un correspondant à la somme pour les contours horizontaux et l'autre à la somme pour les contours verticaux, appelés `sumW` et `sumH` ci-dessous.
- 3) Une moyenne sur un sous-échantillon de `sumW` et `sumH`, respectivement, pour un pas n et un décalage m , calculée par la fonction `vq_AvgSubsample` ci-dessous.

L'idée est qu'en cas de forte structure de blocs de taille n , une différence importante delta_edge apparaîtra dans la valeur vq_AvgSubsample calculée pour le pas n et pour différents décalages.

Par exemple une structure de blocs de taille 4 dans l'image d'origine a une structure de blocs de taille 2 pour la résolution R1. Par conséquent, le calcul des valeurs $\text{vq_AvgSubsample}(x, 2, 0)$ et $\text{vq_AvgSubsample}(x, 2, 1)$ devrait montrer une différence importante si une forte structure de blocs est présente. Pour éviter toute dépendance en fonction du contenu, les expériences utilisant un vaste échantillon de séquences vidéo ont montré comment relier la différence calculée pour la séquence vidéo traitée aux valeurs de la séquence de référence.

Le pseudocode suivant permet de mieux expliquer les détails du calcul. Ici, horGrad et verGrad sont les dérivées spatiales horizontale et verticale d'une image, données par la différence entre pixels adjacents:

$$\begin{aligned} \text{verGrad}_n(i, j) &= Y_n(i+1, j) - Y_n(i, j), \text{ et} \\ \text{horGrad}_n(i, j) &= Y_n(i, j+1) - Y_n(i, j), \end{aligned}$$

où $Y_n(i, j)$ désigne la valeur du pixel à la position (i, j) du plan Y de l'image n . La fonction

$$\text{vq_AvgSubsample}(x, \text{step}, \text{offset})$$

calcule la valeur moyenne du vecteur x sur tous les échantillons pour un pas step en commençant par le décalage offset .

```
// loop over all frames and compute:
for( UINT i=0; i<horGrad.Height; i++ ){
    for( UINT j=0; j<horGrad.Width; j++ ){
        w = (double)verGrad(i, j);
        h = (double)horGrad(i, j);
        // sum edges (-2: small differences can be the result of integer
        // values used to store frames)
        sumW(i) += log(1.0+max(0.0, fabs(w)-2.0));
        sumH(j) += log(1.0+max(0.0, fabs(h)-2.0));
    }
}

double dH0 = vq_AvgSubsample( sumH, 2, 0 );
double dH1 = vq_AvgSubsample( sumH, 2, 1 );
double dW0 = vq_AvgSubsample( sumW, 2, 0 );
double dW1 = vq_AvgSubsample( sumW, 2, 1 );

edge_max = 0.5 * (vq_Max(dW0, dW1) + vq_Max(dH0, dH1) );
edge_min = 0.5 * (vq_Min(dW0, dW1) + vq_Min(dH0, dH1) );

// now: denote by edge_max(i) the value of edge_max above, corresponding to
// frame i of the processed video sequence, and by edge_max_ref(i) the values
// edge_max above corresponding to frame i of the reference video sequence,
// and analogously for edge_min(i), edge_min_ref(i). Then compute:

for( UINT i=0; i<nOfFramesInProcessedVideo; i++ ){
    // get frame nb of ref frame (according to match-list)
    UINT i_ref = (UINT)floor(ref_frameNb_all(i)+0.5f);

    float delta_edge = edge_max(i) - edge_min(i);
    float delta_edge_ref = edge_max_ref(i_ref) - edge_min_ref(i_ref);
}
```

```

x(i) = vq_Max(0.0f,delta_edge - delta_edge_ref) / (1.0f+edge_max(i));
}
// blockiness(i) is then a non-linear monotone transform of x(i) ...

```

Il est à noter que comme des images 720 suréchantillonnées sont possibles, le calcul est légèrement plus complexe:

Voir `vquad_hd::vq_BlockinessPhaseDiff` et `CQualityModelFullRef::Blockiness` pour tous les détails sur la mise en œuvre du calcul de la caractéristique d'effet de blocs.

2.7 Calcul de la caractéristique de sautellement (qualité temporelle)

On calcule une caractéristique de *sautellement* en calculant la moyenne du produit entre une durée d'affichage relative – transformée non linéaire de la durée d'affichage – et une transformée non linéaire de l'intensité de mouvement. L'intensité de mouvement est principalement déterminée à partir des différences entre images dans des régions particulières de l'image. La *durée d'affichage* est la durée, en millisecondes, pendant laquelle une image est affichée sur l'écran. Il est à noter que pour déterminer la durée d'affichage de chaque image, une analyse de l'intensité de mouvement locale est effectuée, étant donné que des images de la séquence vidéo traitée pourraient être des répétitions d'images précédentes.

La caractéristique de sautellement tient compte de la quantité d'informations sautées pendant la lecture de la séquence vidéo traitée. Elle est très faible dans le cas d'une séquence lue avec fluidité, tandis qu'elle augmente en cas de pauses ou de fréquences d'image réduites. Par ailleurs, pour une dégradation temporelle fixe, la caractéristique de sautellement prend des valeurs plus grandes pour les séquences vidéo avec une plus grande intensité de mouvement.

Le pseudocode ci-après montre les détails. Il est à noter qu'en entrée, on a le vecteur d'intensités de mouvement `motionInt`, le vecteur de probabilités de répétition d'image `repFrame` et un vecteur de durées d'affichage d'image `displayTime` et qu'en sortie, on a le vecteur `jerkiness`, donnant le sautellement pour chaque image de la séquence vidéo traitée. Plus précisément, le vecteur `motionInt` donne la différence quadratique moyenne entre images, mesurée sur le plan Y pour la résolution R2. Le vecteur `repFrame` donne la probabilité de répétition d'image: suivant l'intensité de mouvement, chaque image a une certaine probabilité d'être une répétition de l'image précédente. En cas de répétition parfaite de l'image précédente, l'image en cours a la probabilité 1 d'être une répétition. En cas d'intensité de mouvement élevée, l'image en cours a la probabilité 0 d'être une répétition de l'image précédente. On aura des valeurs intermédiaires de probabilité si l'intensité de mouvement a des valeurs très faibles mais non nulles. De manière explicite,

$$\text{repFrame}(i) = \begin{cases} 0 & \text{if } m(i) < p/2 \\ (m(i)-p/2)/p & \text{if } p/2 \leq m(i) < 3/2*p \\ 1 & \text{if } 3/2*p \leq m(i) \end{cases}$$

où $m(i)$ désigne l'intensité de mouvement de l'image i . On a choisi empiriquement le paramètre $p=0,01$.

```

int vq_CalcJerkiness( const CVector<float> & motionInt,
                    const CVector<float> & repFrame,
                    const CVector<float> & displayTime,
                    CVector<float> & jerkiness ){

    // -----
    // the 4 parameters of the jerkiness measure: determined using a
    // large sample of video sequences containing temporal degradations
    // only.
    float a = 0.9f;
    float b = 5.0f;

    float aT = 40.0f;
    float bT = 5.0f;
    // -----

    // get probability of new frame = 1 - prob of repeated frame:
    CVector<float> newFrame = repFrame*(-1.0f) + 1.0f;

    // count number of non-repeated frames
    float fNbRepeated = repFrame.Sum();
    float fNbNonRepeated = repFrame.Length() - fNbRepeated;

    // calculate jerkiness
    float fR = 0.0f;
    // look for frame repetition intervals (~ display time)
    // of length i
    for( UINT i=1; i<= iNbFrames; i++ )
    {
        // look for frame repetitions starting at position j
        for( UINT j=0; j<iNbFrames-i+1; j++ )
        {
            float fP = newFrame(j); // prob. : start of repetition block

            for( UINT k=1; k<i; k++ )
            {
                fP *= repFrame(j+k); // prob. : all repeated frames
            }
            if( i+j < iNbFrames )
            {
                fP *= newFrame(j+i); // prob. : end of repetition block
            }

            // calculate the display time (in s) of frame j,
            // if displayed from
            // time t_j until t_(j+i), which occurs with probability fP
            float fDispTime = displayTime.SumPart(j,j+i)/1000.0f;

            // -> measure jerking and add to result
            float fIFDiff = motionInt( j+i-1 );

            // normalisation values: such that at 0 jerkiness value is 0,
            // and saturates at 1
            float c = 1.0f / (1.0f+exp(b));
            float cT = 1.0f / (1.0f+exp(bT));

            float fJ = 1.0f / (1.0f + exp( -( a * fIFDiff - b ) ));
            float fJT = 1.0f / (1.0f + exp( -( aT * fDispTime - bT)));
            fJ = (fJ - c)/(1.0f - c);
        }
    }
}

```

```

        fJT = (fJT - cT)/(1.0f - cT);

        // jerkiness value: propability * interframeDiffFactor *
        // displayTimeFactor

        fR = fP * fJ * fJT * fDispTime;

        // add to jerkiness vector at position j+i (corresponding to the
        // end of display time)
        jerkiness( vq_Min(j+i,iNbFrames-1) ) += fR;
    }
}
return 0;
}

```

Pour les détails sur la mise en œuvre, voir les méthodes `vquad_hd::vq_ProbOfRepeatedFrame` et `vquad_hd::vq_CalcJerkiness`.

2.8 Agrégation pour l'obtention de la note MOS

Les caractéristiques décrites ci-dessus, la similitude donnée par s_m et s_{delta} , la différence donnée par d_m et d_{delta} , l'effet de blocs `blockiness` et le sautillerment `jerkiness`, servent de base à l'évaluation de la note, conjointement avec le vecteur de durées d'affichage d'image `displayTime`. Ce sont des vecteurs de longueur égale au nombre d'images de la séquence vidéo traitée.

Pour convertir ces caractéristiques sur une échelle de perception, on utilise une fonction paramétrée en forme de S:

$$S: \mathbb{R} \rightarrow \mathbb{R}, \quad y = S(x).$$

La fonction S est paramétrée par le triplet (p_x, p_y, q) . Les paramètres ont la signification suivante: (p_x, p_y) est l'emplacement dans $\mathbb{R} \times \mathbb{R}$ et q la pente du point d'inflexion. Plus précisément:

$$S(x) = a * x^b \quad \text{if } x \leq p_x \\ d / (1 + \exp(-c * (x - p_x))) + 1 - d \quad \text{else} \quad (8.1)$$

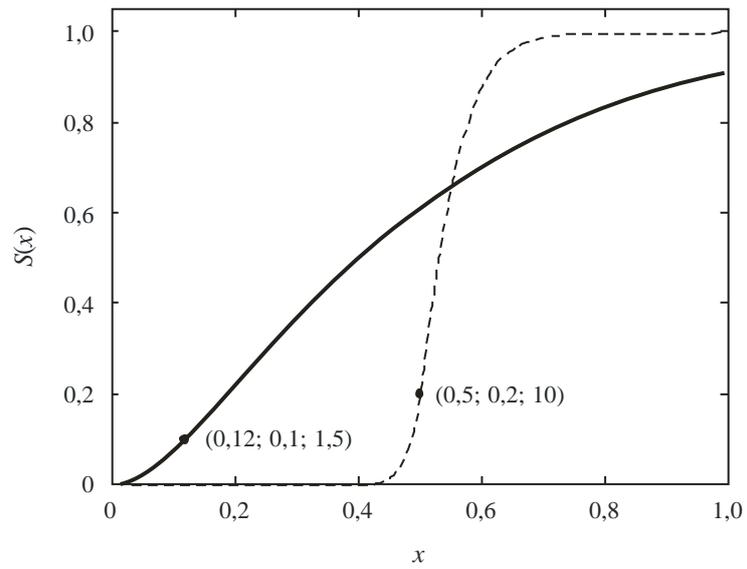
où:

$$\begin{aligned} a &= p_y / p_x^{(q * p_x / p_y)} \\ b &= q * p_x / p_y \\ c &= 4 * q / d \\ d &= 2 * (1 - p_y) \end{aligned}$$

Sur la Fig. 7, la fonction en forme de S est tracée pour différents paramètres. Elle commence à l'origine, croît de façon polynomiale jusqu'au point d'inflexion puis tend de façon exponentielle vers 1.

FIGURE 7

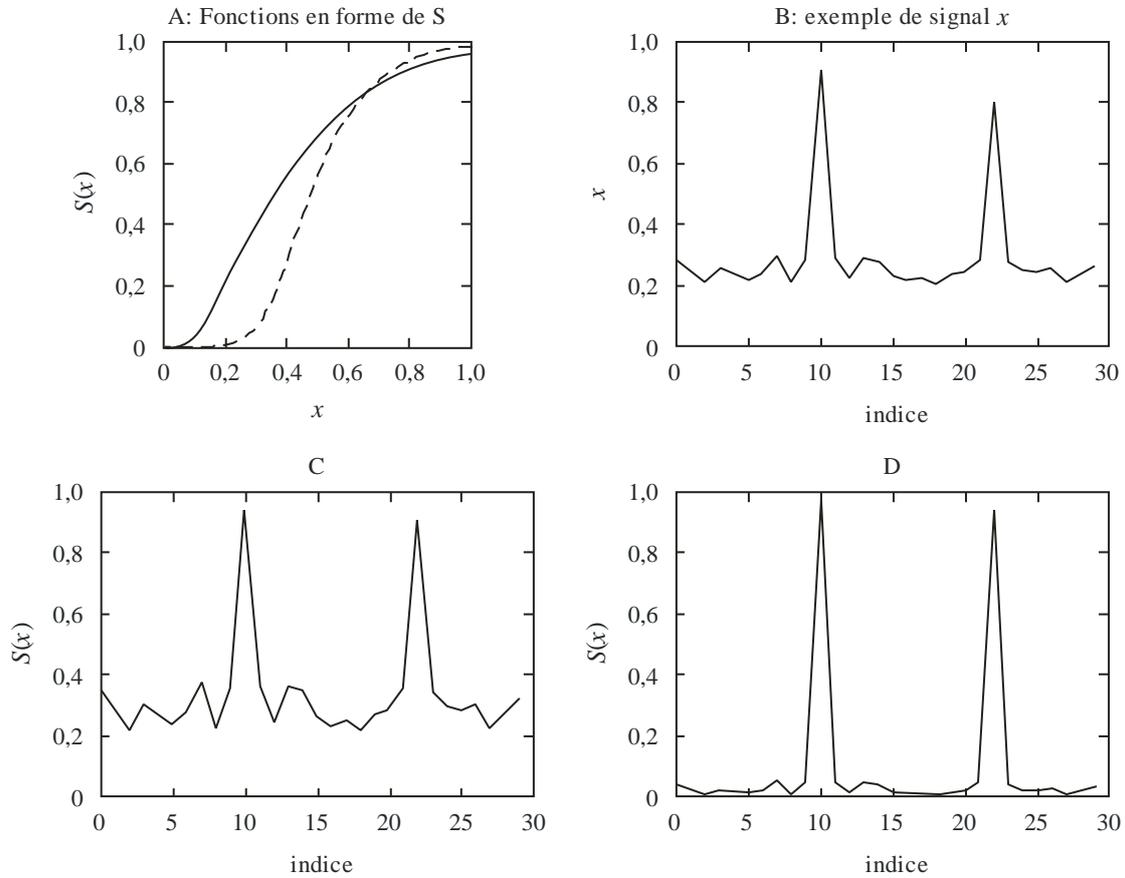
Fonctions en forme de S, paramétrées par la position et la pente du point d'inflexion.
Deux exemples de fonction sont représentés avec des paramètres différents



BT.1907-07

FIGURE 8

Sur la Fig. A) sont représentées les fonctions en forme de S pour 2 ensembles différents de paramètres. Sur la Fig. B) est représenté un exemple de vecteur de signal. Toutes les valeurs de x sont transformées au moyen des deux fonctions en forme de S de la Fig. A). Les résultats sont présentés sur les Fig. C) et D). L'utilisation d'une fonction en forme de S permet de convertir une caractéristique x sur une échelle de perception avec des paramètres ajustés aux échantillons de données (Fig. C)). L'utilisation d'une fonction en forme de S (courbe en pointillés sur la Fig. A)) avec des paramètres qui dépendent des données (pour le premier paramètre, on prend un multiple de la moyenne de x pour le signal) permet de concevoir un détecteur de dégradations transitoires (Fig. D))



BT.1907-08

L'utilisation d'une fonction en forme de S avec des paramètres constants permet de convertir la valeur d'une caractéristique sur une échelle de perception. Une fonction en forme de S avec des paramètres qui dépendent des données peut par exemple être utilisée pour compresser à une faible valeur toutes les valeurs situées au-dessous de la moyenne, et pour élargir toutes les valeurs situées au-dessus. Elle peut ainsi être utilisée pour calculer une caractéristique sensible aux dégradations transitoires (voir la Fig. 8).

Poursuivons la description du modèle et définissons d'abord une dégradation basée sur les caractéristiques de similitude ci-dessus, mais en pondérant davantage les dégradations les plus fortes:

$$d_s = 1 - s_m + 1,5 s_{\text{delta}}.$$

On utilise alors deux fonctions en forme de S. La première utilise un ensemble fixe de paramètres cod_par , pour transformer d_s en d_cod sur une échelle de perception relative à une dégradation de base, et rend compte des erreurs liées au codage vidéo.

La deuxième fonction en forme de S utilise des paramètres *qui dépendent des données*, pour transformer d_s en d_trans sur une échelle de perception relative aux dégradations transitoires, et rend compte des erreurs de transmission.

Plus précisément, par rapport au pseudocode ci-dessous, on appelle la fonction `SplitCodTrans` avec en entrée le vecteur `d_s` et le vecteur de durées d'affichage d'image `disp_time` et en sortie `d_cod`, `d_trans`,

```
SplitCodTrans(d_s, disp_time, d_cod, d_trans).
```

Le pseudocode ci-après présente la fonction en détail. Il est à noter que

```
stat.STransform(x,px,py,q)
```

désigne la fonction en forme de S avec en entrée la valeur réelle x qui sera transformée, et les trois paramètres désignés par (px, py, q) dans la formule (8.1).

```
SplitCodTrans( const CVec& v, const CVec& dispTime,
               CVec& v_cod, CVec& v_trans ){

    // these parameters are determined empirically
    float qPosSmall = 0.55f;
    float qPosLarge = 0.65f;

    // q is the mean of the values in v between the qPosSmall and qPosLarge
    // quantiles.
    float q = r.TrimmedMean( qPosSmall, qPosLarge, v, dispTime );

    for( UINT i=0; i<v.Length(); i++ ){

        // the parameters used here are the result of fitting to sample
        // data
        v_cod(i) = stat.STransform(v(i), 0.07f, 0.1f, 2.0f);

        // Note that the STTransform is not directly applied to v, but
        // to v(i)-q . This is preferred here for numerical reasons of the
        // resulting STTransform.

        // v_trans is part of v above quantile value q
        v_trans(i) = vq_Max(0.0f, v(i)-q);

        float px = 0.5f * (q+0.2f);
        // the parameters used here are the result of fitting to sample
        // data
        v_trans(i) = stat.STransform(v_trans(i),px, 0.1f,16.0f);
    }
}
```

Pour les détails complets sur la mise en œuvre, voir `CQualityModelFullRef::SplitCodTrans`.

Par analogie, `d_diff_cod` et `d_diff_trans` sont déterminés à partir de `d_m` et `d_delta`, en posant

$$d_diff = d_m + 1,5 d_delta,$$

et en appelant la fonction

```
SplitCodTrans(d_diff, disp_time, d_diff_cod, d_diff_trans),
```

qui utilise pour les deux transformées S les triplets de paramètres

$$\begin{aligned} \text{cod_par} &= (4,0f; 0,05f; 0,2f) \\ \text{trans_par} &= (0,5*(q+4,0f); 0,1f; 0,4f), \end{aligned}$$

où q désigne la moyenne interquantile, comme dans le pseudocode ci-dessus. La fonction donne en sortie d_diff_cod et d_diff_trans .

Ensuite, une valeur de caractéristique relative aux valeurs transitoires élevées de sautellement est calculée au moyen d'une transformée en forme de S du sautellement:

$$d_t_trans = S(\max(0, \text{jerkiness} - q))$$

les paramètres de la transformée en forme de S étant donnés par $(\max(0,048, q); 0,2; 40,0)$ et q désignant la moyenne interquantile entre les quantiles d'ordre 0,55 et d'ordre 0,65 du vecteur de sautellement. Les paramètres ont été déterminés par ajustement à un vaste ensemble d'échantillons de données.

Pour les détails complets sur la mise en œuvre, voir `CQualityModelFullRef:: SplitTempTrans`.

Ensuite, la qualité de base q_cod est définie par:

$$q_cod = (1 - d_cod) * (1 - d_diff_cod) * (1 - \text{blockiness}) \tag{8.2}$$

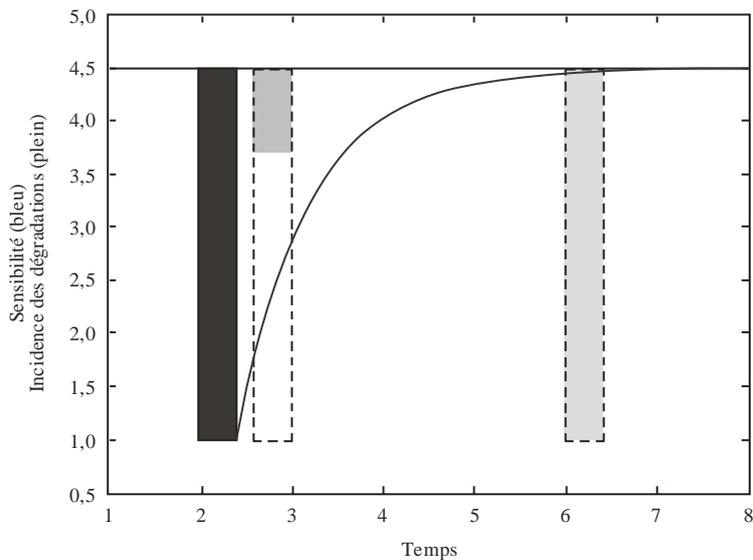
et la qualité transitoire est définie par:

$$q_trans = (1 - d_trans) * (1 - d_diff_trans) * (1 - d_t_trans).$$

Une dégradation ultérieure a une influence réduite si elle se produit peu après une première dégradation. Pour tenir compte de cet effet, q_trans est transformé en q_fq . Le principe est illustré sur la Fig. 9.

FIGURE 9

La barre noire indique qu'une dégradation se produit dans une séquence vidéo fictive. La courbe continue en noir indique la sensibilité aux dégradations ultérieures. Une dégradation ultérieure d'intensité donnée par la courbe en pointillés a une influence réduite (première barre grise) si elle se produit peu après une première dégradation. Son influence n'est pas modifiée si elle se produit longtemps après (barre grise de droite)



Le pseudocode qui suit permet de mieux décrire les détails du calcul, avec $1-q_{trans}$ comme vecteur d'entrée v , le vecteur $disp_time$ étant le vecteur des durées d'affichage d'image. La constante de temps de descente $dT=1000$ ms a été déterminée sur la base d'échantillons de données; on pose ensuite:

$$q_{fq} = 1 - w, \quad (8.3)$$

où w est le vecteur de sortie de la fonction:

```

DegFreq( const CVec& v, const CVec& disp_time, CVec& w, float dT ){

    // time constant for temporal integration of degradations
    float t_const = 80.0f;
    w(0) = v(0)*vq_Min(t_const,disp_time(0))/t_const;

    for( UINT i=1; i<v.Length(); i++ ){
        // integrate degradation over the last t_const milliseconds:
        // use an indicator function, which is 1 over the interval
        // [t_i-t_const, t_i] and 0 otherwise
        float dT_sum = 0.0f;
        UINT j=0;
        float v_sum = 0.0f;
        while( dT_sum < t_const && (int)i-(int)j>=0 ){
            // b is the overlap in [0,1] of the display time interval
            // of frame i-j with respect to the integration interval
            //      t_i-t_const   t_i
            //      |-----|-----|
            //      |-----|-----| integration window
            // -----> time
            // | | | | | frames
            //      ->| b*dT |<-
            //
            float b = vq_Min(t_const-dT_sum,disp_time(i-j)) / t_const;

            v_sum += v(i-j)*b;
            dT_sum += disp_time(i-j);
            j++;
        }
        // compute the fall-off factor a:
        float a = exp(-disp_time(i-1)/dT);
        // the degradation is now:
        // 1) a linear combination of the fall-off of a previous degradation
        // and the current (integrated) degradation, or
        // 2) the integrated current degradation (if it is stronger than 1)
    ).
        w(i) = vq_Max(v_sum, a * w(i-1) + (1.0f-a) * v_sum);

    }
}

```

Processus de moyennage

Désignons par $disp_time(i)$ la durée d'affichage de l'image i , et par $jerkiness(i)$ la valeur de sautillerment correspondant à l'image i . Rappelons q_{cod} donné par la formule (8.2) et q_{fq} donné par la formule (8.3). Posons

$$Q_t = 1 - 1/T \sum_i jerkiness(i)$$

$$Q_{fq} = 1/T \sum_i q_{fq}(i) * disp_time(i)$$

$$Q_{cod} = 1/T \sum_i q_{cod}(i) * disp_time(i)$$

où $T = \sum_{i=0}^{\text{number_of_frames}} \text{disp_time}(i)$, et $\sum_{i=0}^{\text{number_of_frames}}$ désigne la somme sur tous les indices $i=0, \dots, \text{number_of_frames}$.

La note prédite finale s'obtient par un produit et un recadrage sur l'intervalle des notes MOS [1,5]:

$$s = 4 * Q_t * Q_{\text{cod}} * Q_{\text{fq}} + 1$$

Pour plus de détails sur la prédiction de la note, voir la méthode `CQualityModelFullRef::PredictScoreCodTrans`.

2.9 Traitement des séquences vidéo présentant un important défaut d'alignement spatial

Afin d'éviter toute prédiction erronée dans le cas d'un défaut d'alignement spatial relativement important entre la séquence vidéo de référence et la séquence vidéo traitée, les étapes ci-dessus sont suivies pour trois valeurs différentes d'alignement spatial horizontal et vertical de la séquence vidéo, et la note de qualité finale évaluée est donnée par la note maximale prédite parmi toutes les positions spatiales.

On utilise un pas de quatre pixels dans chaque direction. Ainsi, la recherche est effectuée sur une plage de ± 8 pixels, ce qui permet de couvrir aisément le décalage spatial maximal utilisé dans l'ensemble test (± 5 pixels). Etant donné que l'élargissement est effectué dans une fonction de haut niveau du modèle, la plage d'alignement peut aisément être adaptée aux grands décalages ou peut être réduite (par exemple ± 4 pixels) pour utiliser moins de ressources informatiques.

Voir `vquad_hd::vq_vquad08`.

2.10 Mise en oeuvre du code source de référence

Le présent paragraphe renvoie au code source en C++ couvrant les parties et fonctions essentielles pour une mise en oeuvre conforme à la description du modèle. Tous les liens vers des mises en oeuvre effectives dont il est fait mention dans les paragraphes ci-dessus renvoient à ce code source de référence.

Annexe 3

Tests de conformité

La présente Annexe est complétée par une pièce jointe numérique contenant les informations et fichiers suivants:

- 1) Seize courtes séquences vidéo HD (huit séquences de référence et huit séquences dégradées associées). Ces séquences présentent des distorsions et des contenus différents. Elles sont destinées à être utilisées pour vérifier si une mise en oeuvre du modèle réalisée par un utilisateur de la présente Recommandation est conforme à la mise en oeuvre de référence du modèle. Les séquences vidéo comportent peu d'images afin de réduire la capacité de stockage. Elles ne sont pas destinées à être utilisées pour réaliser des tests visuels, mais uniquement pour vérifier la conformité de la mise en oeuvre du modèle HD.
- 2) Les notes MOS prédites pour les séquences HD mentionnées au point 1) (`HD_ConformanceReferenceResults.xls`). Ces notes ont été obtenues avec une mise en oeuvre de référence du modèle HD.

- 3) Les notes MOS prédites pour cinq bases de données HD publiques disponibles auprès du Groupe VQEG. Ces bases de données peuvent être utilisées pour réaliser un test étendu de conformité des mises en œuvre du modèle.

Critères applicables aux tests de conformité:

- i) Les huit notes de référence mentionnées au point 2) doivent être reproduites de façon exacte par une mise en œuvre du modèle. On entend par reproduction exacte l'obtention de notes identiques à la troisième décimale près.
- ii) Les notes MOS prédites pour les cinq bases de données publiques du Groupe VQEG doivent être reproduites avec un écart très limité. De légères variations sont autorisées, car l'expérience a montré que différentes optimisations en termes de vitesse et d'utilisation de mémoire peuvent entraîner des écarts légers et négligeables dans la note finale.

TABLEAU 2

Distribution permise de différences parmi toutes les données des tests de conformité

Différence absolue	Occurrence permise
> 0,0001	5,00%
> 0,001	1,00%
> 0,01	0,50%
> 0,1	0,05%
> 0,3	0,00%

Pour les bases de données autres que celles définies dans la présente Annexe 3, la même distribution d'erreur ne doit pas être dépassée. Dans le cas de données inconnues, il faut prendre, pour ces statistiques, un ensemble test d'au moins 500 paires de fichiers, issues de préférence d'expériences complètes d'évaluation subjective.

Bibliographie

- Rapport final du groupe VQEG concernant le test de validation de phase I de TVHD (2010), «*Video Quality Experts Group: report on the validation of video quality models for high definition video content*», Video Quality Experts Group (VQEG), <http://www.its.bldrdoc.gov/vqeg/projects/hdtv>.
- Recommandation UIT-R BT.601-7 (2011), *Paramètres de codage en studio de la télévision numérique pour des formats standards d'image 4:3 (normalisé) et 16:9 (écran panoramique)*.
- Recommandation UIT-T J.244 (2008), *Méthodes d'étalonnage du désalignement constant des domaines spatial et temporel avec un gain et un décalage constants*.
- Recommandation UIT-R BT.500-12 (2009), *Méthodologie d'évaluation subjective de la qualité des images de télévision*.
- Recommandation UIT-T J.149 (2004), *Méthode de spécification de la précision et du contre-étalonnage des mesures de la qualité vidéo*.
- Recommandation UIT-T J.247 (2008), *Mesure objective de la qualité vidéo multimédia perçue en présence d'une référence complète*.
- Recommandation UIT-T J.144 (2004), *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.264 (2012), *Codage vidéo évolué pour les services audiovisuels génériques*.
- Recommandation UIT-T J.340 (2010), *Algorithme de référence pour le calcul du rapport signal de crête sur bruit d'une séquence vidéo traitée avec des décalages spatiaux constants et un délai constant*.
- 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 d'évaluation subjective 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*.
-