Reconstruction robuste de vecteurs mouvements appliquée au codeur H.264

A. Ouled Zaid M. Kieffer C. M. Lee P. Duhamel

LSS (Laboratoire Signaux et Systèmes)– CNRS

SUPELEC–Université Paris-Sud
Plateau de Moulon, 91192 Gif sur Yvette, France

{Azza.OULED-ZAID, kieffer, lee, pierre.duhamel}@lss.supelec.fr

Résumé

Cet article présente une première étape d'un travail visant à l'obtention de codecs vidéo robustes aux erreurs de transmission. En effet, il est bien connu (i) que l'efficacité des codeurs est essentiellement due à la compensation de mouvement (ii) que les vecteurs mouvements générés pendant cette étape présentent la plus grande sensibilité aux erreurs de transmission. C'est pourquoi l'objet de ce travail est de proposer un codeur vidéo basé sur H264 qui ne nécessite pas de transmission de vecteurs mouvement (ils sont réestimés au décodeur). Le reste du train binaire sera robustifié ultérieurement par des techniques de décodage souple utilisant la sémantique du flux vidéo [1].

Mots clefs

Codeur H.264, codes BCH, compensation de mouvement, réestimation de mouvement.

1 Introduction

Pour assurer la protection de flux vidéo, des schémas de codage robustes à l'égard des erreurs de transmission ont été développés [2] [3]. Bien qu'elles soient efficaces, ces méthodes restent sensibles aux erreurs de transmission résiduelles qui peuvent affecter les vecteurs mouvements (VMs).
Récemment, une nouvelle technique de robustification des VMs a été développée [4] dans le but de se passer de la transmission des VMs générés par le codeur H.263+. Cette approche consiste à introduire une redondance structurée dans le flux vidéo à compresser en utilisant une expansion sur des frames BCH. Cette redondance est introduite en imposant une propriété dans chaque image du flux vidéo. Cette propriété doit se retrouver au moins en partie au niveau des images reconstruites. La transmission des VMs n'est alors plus nécessaire car ceux-ci peuvent être estimés au niveau du décodeur en recherchant ceux qui permettent d'assurer la propriété imposée au niveau des images du flux vidéo initial. Cette technique permet de se passer de la transmission des VMs, mais au prix d'un taux de compression légèrement diminué.
Dans ce travail, nous avons tenté de pallier au problème de compromis robustesse/performance de compression. Ceci est réalisé en couplant au codeur H.264 la même approche d'estimation des VMs par l'utilisation d'une expansion sur des frames BCH (à base de transformée FFT et DCT) . En effet, H.264 permet d'obtenir un gain en compression par rapport aux codeurs précédents tel que H.263. Ce gain est dû particulièrement à la structure du codeur qui est caractérisée par l'adaptabilité de la partition et le choix du mode de codage de chaque macrobloc (MB) à compresser. Cette adaptabilité nous permet d'augmenter l'efficacité de notre procédure de réestimation des VMs au récepteur.
Afin d'évaluer les performances de notre méthode, nous avons d'abord comparé nos résultats, en terme de courbes débit/PSNR, en utilisant des frames BCH à base de DFT et de DCT. Nous avons pu conclure que les meilleurs résultats sont obtenus en utilisant une expansion sur frames BCH à base de la DCT avec un accord localisé dans les moyennes/hautes fréquences. Nous avons aussi comparé nos résultats par rapport à ceux obtenus par le même codeur H.264 sans robustification, le codeur H.263+ et H.263+ robuste. D'après les courbes comparatives, nous avons pu constater que les performances en compression sont inférieurs à celles obtenues par le codeur H.264 original. Toutefois, notre codeur a des performances comparables à celles obtenues par H.263+ sans robustification.

2 Expansion sur frames BCH

La famille des codes BCH(k, n) sur le corps des réels [5] constitue une classe particulière de frame dans $\mathbb{R}^k$. Le principe du codage BCH consiste à imposer que le spectre des mots de code soit nuls sur un ensemble $\mathcal{A}$, appelé *accord*, de $n-k$ fréquences entières comprises entre 0 et $n-1$.
Le codage BCH d'un *mot d'information* $\mathbf{x}_{(k)} \in \mathbb{R}^k$ donne un *mot de code* $\mathbf{c}_{(n)} \in \mathbb{R}^n$ défini par

$$\mathbf{c}_{(n)} = \mathbf{W}_{(n)}\mathbf{P}_{(n,k)}\mathbf{W}_{(k)}^{-1}\mathbf{x}_{(k)}, = \mathbf{F}_{(n,k)}^{\mathrm{BCH}}\mathbf{x}_{(k)}, \quad (1)$$

où $\mathbf{W}_{(k)}$ et $\mathbf{W}_{(n)}$ sont des matrices unitaires (par exemple de Fourier, de cosinus discrète, ...) et $\mathbf{P}_{(n,k)}$ est une matrice de bourrage de zéros. Le codage BCH correspond donc à une expansion sur une *frame BCH*.
Les zéros imposés dans le spectre des mots de codes per-

mettent de définir un syndrome à partir d'un mot reçu quelconque $\mathbf{r}_{(n)} \in \mathbb{R}^n$

$$\mathbf{s}_{(n-k)}\left(\mathbf{r}_{(n)}\right) = \mathbf{R}_{(n-k,n)}\left(\mathbf{W}_{(n)}\right)^{-1}\mathbf{r}_{(n)} = \mathbf{H}_{(n-k,n)}\mathbf{r}_{(n)},$$

avec $\mathbf{H}_{(n-k,n)}$ *matrice de détection de parité*, et $\mathbf{s}_{(n-k)}$ le vecteur du syndrome. Une valeur non nulle du syndrome implique la présence d'erreurs de transmission. Dans le cas des images d'une séquence vidéo, l'expansion produit sur un frame des blocs $\mathbf{I}_{(k,k)}$ des images constituant la séquence initiale en blocs $\mathbf{M}_{(n,n)}$ de taille $n \times n$. La matrice de syndrome est alors définie de la manière suivante

$$\begin{aligned}\mathbf{S}_{(n,n)}(\mathbf{M}_{(n,n)}) = \mathbf{W}_{(n)}^{-1}\mathbf{M}_{(n,n)}(\mathbf{W}_{(n)}^{-1})^{\mathbf{T}} \\ -\mathbf{F}_{(n,k)}^{\text{BCH}}\mathbf{M}_{(n,n)}(\mathbf{F}_{(n,k)}^{\text{BCH}})^{\mathrm{T}}\end{aligned} \quad (2)$$

Elle contient $n \times n - k \times k$ éléments non nuls lorsque $\mathbf{M}_{(n,n)}$ a été corrompu par une erreur de transmission.
Dans ce qui va suivre, nous allons rappeler le principe de fonctionnement du codeur H.264. Afin de comprendre notre approche de réestimation des VMs, nous nous proposons d'expliquer brièvement la méthode d'estimation des VMs adoptée dans le standard H.264.

3 Codeur H.264 et estimation des VMs

3.1 Phase de codage directe

Le concept du codage H.264 est semblable à celui des autres normes telle que H.263. Chaque image d'une vidéo qui peut être une image ou une tranche d'image, est partitionnée en MBs $\mathbf{M}_{(n,n)}$ de dimension fixe qui couvrent une zone rectangulaire de l'image. Tous les échantillons de luminance et de chrominance d'un MB font l'objet d'un codage en Intra (prédiction spatiale) ou en Inter (prédiction temporelle). Dans le cas d'un codage en Intra, le MB prédit $\mathbf{X}_{(n,n)}$ est déterminé à partir des MBs précédemment codés dans la même tranche. Dans le cas d'un codage Inter $\mathbf{X}_{(n,n)}$ est obtenu avec compensation de mouvement à partir d'une ou plusieurs images de référence.
Une fois déterminés, les échantillons de $\mathbf{X}_{(n,n)}$ sont soustraits de ceux de $\mathbf{M}_{(n,n)}$ produisant un MB de texture $\mathbf{T}_{(n,n)}$. Cette texture est alors subdivisée en blocs. une transformée entière est ensuite appliquée à chaque bloc. Les coefficients de la transformée quantifiés sont transmis après codage entropique.

3.2 Décodage local

A cette étape du décodage local, l'estimée du MB de texture $\tilde{\mathbf{T}}_{(n,n)}$ est obtenue après déquantification et transformée inverse. Les échantillons du MB prédit $\mathbf{X}_{(n,n)}$ sont ajoutés à ceux de $\tilde{\mathbf{T}}_{(n,n)}$ pour obtenir le MB reconstruit $\tilde{\mathbf{M}}_{(n,n)}$. L'ensemble des MBs $\tilde{\mathbf{M}}_{(n,n)}$ constituent l'image de référence à utiliser durant la phase de compensation de mouvement suivante.

3.3 Recherche du meilleur VM

Le codage Inter utilise une prédiction (compensation de mouvement) sur la base d'images de référence. Chaque MB correspond à un partitionnement spécifique en blocs de taille fixe (s'étendant de 16×16 à 4×4) utilisé pour décrire le mouvement. L'estimation d'un VM pour chaque bloc est alors nécessaire pour déterminer le déplacement appliqué à tous les échantillons de ce bloc. Dans notre travail, afin de simplifier le processus d'estimation des VMs, nous nous sommes limités à des partitions de taille 16×16 et à une prédiction par rapport à une seule image de référence.
Les composants du VM $\mathbf{m}$ font l'objet d'un codage différentiel basé sur la prédiction directionnelle ou médiane des blocs voisins pour donner un VM de prédiction $\mathbf{p}$ [6]. Du fait que la précision de la compensation de mouvement est à un quart de pixel, le VM est d'abord calculé en utilisant une précision au pixel, puis des positions au demi pixel et des positions au quart de pixel. Le VM $\mathbf{m} = (m_x, m_y)^T$ avec une précision au pixel qui minimise

$$J(\mathbf{m}, \lambda) = SAD(\mathbf{M}_{(n,n)}, \tilde{\mathbf{M}}_{(n,n)}(\mathbf{m})) + \lambda \cdot R(\mathbf{m}\text{-}\mathbf{p}) \quad (3)$$

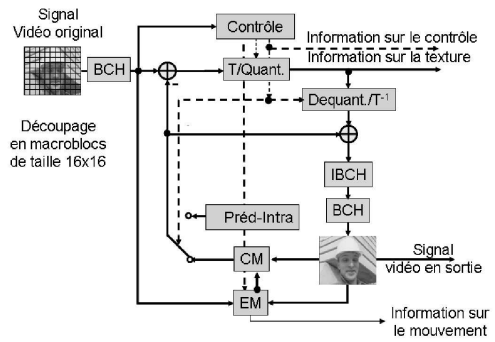
est le VM recherché, avec $\mathbf{p} = (\mathbf{p}_x, \mathbf{p}_y)^{\mathrm{T}}$ le VM de prédiction correspondant et λ le multiplicateur de Lagrange. R(m-p) correspond aux nombre de bits alloués à l'information sur le mouvement.

4 Codeur H.264 modifié

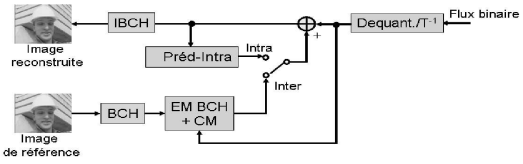
Nous allons rappeler la technique de robustification des VMs qui a été appliquée au codeur H.263+ et que nous avons intégré dans le codeur H.264.
Le principe de l'approche est d'utiliser une stratégie de sélection des VMs sans avoir besoin de transmettre l'information sur le mouvement.
Les figures 1a et 1b présentent le schéma fonctionnel du codeur/décodeur H.264 incluant une expansion sur frame BCH. Les blocs BCH et IBCH correspondent respectivement à une expansion et à une synthèse produit sur un frame BCH. Dans ce qui va suivre, les erreurs de transmission affectant la texture seront supposées corrigées. De plus, aucun VM n'est transmis au décodeur. Une réestimation des VMs doit donc être réalisée.
Afin de s'assurer de la faisabilité du système, nous nous limitons à une réestimation du VM avec une précision de 1/2 pixel.
Considérons un MB $\mathbf{M}_{(n,n)}$ de la i ^{ème} image du flux vidéo. Au niveau du codeur vidéo, la réestimation du VM consiste à trouver le VM $\hat{\mathbf{m}} = (m_x, m_y)^{\mathbf{T}}$ qui minimise la norme de Frobenius $||\mathbf{S}_{(n,n)}||_{\mathrm{F}}$ du syndrome correspondant au MB reconstruit $\tilde{\mathbf{M}}_{(n,n)}$ qui est une version estimée du MB $\mathbf{M}_{(n,n)}$.

$$\tilde{\mathbf{M}}_{(n,n)}(\mathbf{m}) = \tilde{\mathbf{T}}_{(n,n)} + \mathbf{X}_{(n,n)}(\mathbf{m})$$

$\mathbf{X}_{(n,n)}(\mathbf{m})$ est un MB extrait d'une matrice de recherche $\mathrm{N}_{(l,l)}$ dans une image de référence précedemment reconstruite. Le VM $\hat{\mathbf{m}}$ peut être alors réestimé en utilisant la pro-



(a) Codeur.



(b) Décodeur.

Figure 1 – *Schéma bloc du système de codage/décodage H.264 sans transmission des VMs.*

prieté imposée sur chaque bloc de l'image lors de l'expansion sur frame BCH. En effet, en absence de bruit de quantification, l'estimée $\hat{\mathbf{m}}_{\text{BCH}}$ du VM $\hat{\mathbf{m}}$ peut être obtenue en minimisant la fonction coût $J_{\text{BCH}}(\cdot,\cdot)$:

$$J_{\text{BCH}}(\mathbf{m}, \tilde{\mathbf{T}}_{(n,n)}(\hat{\mathbf{m}})) = \quad (4)$$
$$||\mathbf{S}_{(n,n)}(\tilde{\mathbf{T}}_{(n,n)}(\hat{\mathbf{m}}) + \mathbf{X}_{(n,n)}(\mathbf{m}))||_{\text{F}}$$

Cependant, à cause du bruit de quantification, rien ne garantit que $\hat{\mathbf{m}}$ est l'argument du minimum global de $J_{\text{BCH}}(\mathbf{m}, \tilde{\mathbf{T}}_{(n,n)})$. Une perte d'accord peut surgir entre $\hat{\mathbf{m}}$ et le VM réestimé au niveau du codeur. Afin d'améliorer la robustesse de la réestimation des VMs, l'estimation classique doit être remplacée par une estimation robuste qui permet de garantir au niveau du codeur vidéo que le VM non transmis pourra être réestimé sans erreur. Soit $\ell = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\}$ la liste des VMs générée par le processus d'estimation de mouvement classique utilisé dans H.264. Chaque élément de cette liste satisfait

$$J(\mathbf{m}_1, \lambda) \leq J(\mathbf{m}_2, \lambda) \leq \ldots \leq J(\mathbf{m}_N, \lambda).$$

$\tilde{\mathbf{T}}_{(n,n)}(\mathbf{m}_1)$ et $\mathbf{m}_1$ sont ensuite transmis. La procedure d'estimation robuste doit alors calculer $\mathbf{T}_{(n,n)}(\mathbf{m}_{\underline{k}})$ et $\mathbf{m}_{\underline{k}}$ où

$$\underline{k} = \min\{k \mid \mathbf{m}_k = \arg\min_{\mathbf{m}} J_{\text{BCH}}(\mathbf{m}, \tilde{\mathbf{T}}_{(n,n)}(\mathbf{m}_k))\},$$

lorsqu'aucun $\underline{k}$ ne peut être obtenu, le MB considéré sera codé en Intra. En outre même lorsqu'un tel $\underline{k}$ existe, il est possible que le coût de transmission excède celui du MB codé en Intra. Dans ce cas, il n'y a aucun intérêt de réaliser la compensation de mouvement et le mode de codage en Intra est appliqué directement.

5 Résultats expérimentaux

5.1 Conditions expérimentales

Dans nos simulations le schéma de codage H.264 modifié a été mis en œuvre sur les 101 premières composantes en luminance de la vidéo *foreman*. Nous avons utilisé la version JM 7.0 de la norme H.264 avec l'ensemble des paramètres de compression suivants : zone de recherche de taille 8x8 ; compensation de mouvement avec précision au 1/2 pixel ; codage binaire à longueur variable (UVLC). L'évaluation des performances de notre technique de robustification intégrée dans la chaîne de codage H.264 est assurée en comparant les résultats de nos tests (PSNR en fonction du débit) à ceux obtenus par l'utilisation du codeur H.263+ (avec et sans robustification). Trois frames BCH ont été utilisées : un frame BCH reposant sur une DFT : $\mathbf{F}^{\text{BCH-F}}_{(16,15)}$ avec $\mathcal{A} = \{8\}$ et trois frames BCH reposant sur une DCT $\mathbf{F}^{\text{BCH-C}}_{(16,15)}$ avec $\mathcal{A} = \{4\}$, $\mathcal{A} = \{10\}$ et $\mathcal{A} = \{12\}$.

5.2 Réglage de notre méthode

Tout d'abord, nous nous sommes interessés à la comparaison des résultats obtenus avec différents codes BCH. La figure 2 montre la variation du PSNR en fonction du débit (en bpp) en utilisant les codeurs cités ci-dessus. Parmi

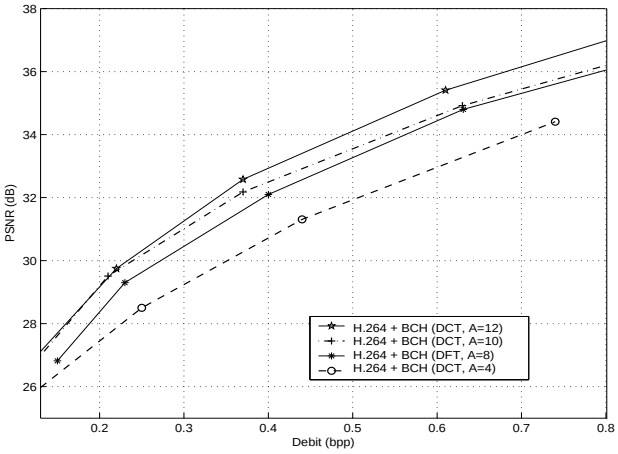


Figure 2 – *Courbes du PSNR en fonction du Débit en utilisant un codeur H.264 avec précision 1/2 pixel en utilisant une expansion sur frame BCH à base de la DCT avec un accord ($\mathcal{A}$=4, $\mathcal{A}$=10 et $\mathcal{A}$=12) et à base de DFT avec un accord $\mathcal{A}$=8.*

toutes les expansions sur frame BCH testées, celle reposant sur la DCT avec le quatrième coefficients mis à zéro est la moins bonne. Ceci est dû au fait que $\mathcal{A}$ choisi dans les basses fréquences génére des effets de bloc sur l'image aprés expansion. Dans le cas de la DFT, la redondance est introduite au huitième coefficient car il est le seul permettant d'assurer une expansion réelle. Ce frame BCH est meilleur que celui basé sur la DCT avec $\mathcal{A} = \{4\}$, toutefois il est moins performant que le frame BCH reposant sur

une DCT $\mathbf{F}^{\text{BCH-C}}_{(16,15)}$ avec $\mathcal{A} = \{12\}$ qui est le meilleur choix. Ceci peut être expliqué comme suit : il y a un compromis dans le choix de la localisation de la redondance à ajouter : les amplitudes de la texture dans les hautes fréquences sont faibles, ainsi, le critère basé sur la norme du syndrome n'est pas très discriminant. Cependant, les amplitudes de la texture deviennent de plus en plus importantes en allant vers les basses fréquences, ce qui peut causer un manque d'efficacité pour l'évaluation des VMs. C'est pourquoi, dans le cas de la DCT, le meilleur choix est dans les mi-hautes fréquences. Ces interprétations nous ont permis de régler notre méthode en utilisant une expansion sur frame BCH reposant sur la DCT avec un accord $\mathcal{A} = \{12\}$. Dans ce qui va suivre, nous comparons notre méthode à un codeur H.263+ robuste utilisant un frame BCH reposant sur une DCT $\mathbf{F}^{\text{BCH-C}}_{(16,15)}$ avec un accord $\mathcal{A} = \{8\}$.

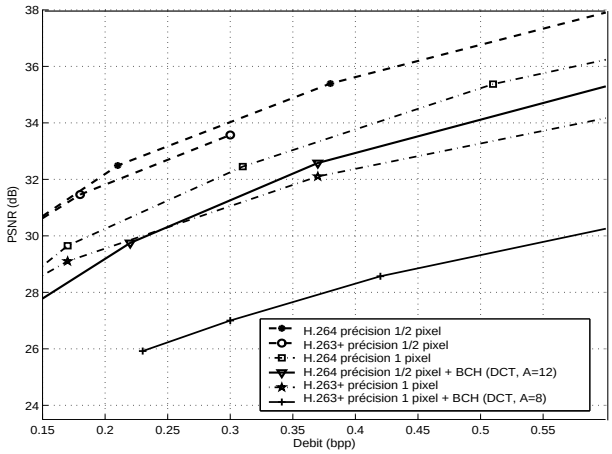


Figure 3 – *Courbes du PSNR en fonction du débit en utilisant un codeur H.264 avec expansion sur frame BCH (DCT, $\mathcal{A}$=12, précision 1/2 pixel), un codeur H.263+ avec expansion sur frame BCH (DCT, $\mathcal{A}$=8, précision 1 pixel) et les codeurs H.264 et H.263+ avec précision 1 pixel et 1/2 pixel.*

5.3 Comparaison avec d'autres méthodes

Dans la figure 3 nous avons comparé nos résultats avec ceux obtenus par les codeurs H.264 et H.263+ originaux avec une précision de 1 et 1/2 pixel et H.263+ modifié avec une reconstruction robute des VMs.

Notre codeur réalise un gain considérable de l'ordre de 3 à 4 dB par rapport à H.263+ robuste. Cela est dû au fait que le mode de codage en Intra est activé s'il a un coût de compression plus faible que celui obtenu en utilisant la prédiction de mouvement. Dans notre méthode nous avons égalemment utilisé une précision de 1/2 pixel ce qui a contribué à l'amélioration de la qualité de reconstruction. A partir de la même figure, nous avons pu contater que les performances en compression de notre méthode sont inférieures à celles obtenues par les codeurs H.264 et H.263+ originaux avec précision de 1/2 pixel c'est le prix à payer pour l'amélioration de la robustesse. Toutefois, les résultats de notre codeur sont comparables à ceux obtenus par H.263+ avec une précision de 1 pixel dans le cas de bas débits et sont meilleurs aux moyens et hauts débits. Nous pouvons ainsi réaliser un codeur vidéo qui a des performances supérieures à celles fournies par H.263+ (avec précision de 1 pixel) sans avoir besoin de transmettre les VMs. Par conséquent, cet algorithme semble être une bonne base pour développer des codeurs utilisant d'autres techniques de robustification telles que le décodage souple tenant compte de la sémantique du train binaire généré [7], sachant qu'il ne peut plus y avoir d'erreur sur les VMs qui ne sont pas transmis.

6 Conclusion

Ce travail a montré qu'à l'aide d'une expansion sur frame BCH des images composant une séquence vidéo, il est possible de se passer de la transmission des vecteurs mouvement générés par le codeur H.264 en ayant des performances en compression comparables voire meilleures que celles obtenues par le codeur H.263+ avec une précision au pixel. Les VMs sont réestimés en utilisant le fait que les images reconstruites doivent satisfaire certaines contraintes introduites par l'expansion sur frame au niveau du codeur vidéo. Actuellement, cette méthode est en cours d'amélioration en utilisant une compensation de mouvement avec une précision au quart de pixel.

Références

[1] Hang Nguyen et P. Duhamel. Optimal VLC sequence decoding based on compressed image and video stream properties. Dans *ICASSP'04*, 2004.

[2] U. Horn, B. Girod, et B. Belzer. Scalable video coding for multimedia applications and robust transmission over wireless channels. Dans 7^{th} *Int. Workshop on Packet Video*, March 1996.

[3] L. P. Kondi, F. Ishtiaq, et A. K. Katsaggelos. Joint source-channel coding for motion-compensated DCT-based SNR scalable video. *IEEE transactions on Image Processing*, 11(9) :1043–1052, 2002.

[4] C. M. Lee, M. Kieffer, et P. Duhamel. Robust Motion Vectors and Texture Transmission for the H263 Video encoder family. Dans *Proc. of PCS*, Saint–Malo, 2003.

[5] A. Gabay, P. Duhamel, et O. Rioul. Real BCH codes as joint source channel code for satellite images coding. Dans *Proceedings of Globecom*, San Francisco, USA, November 2000.

[6] ITU-T Recommendation. Advanced Video Coding. FINAL Committee Draft, Document JVT-E022 11496-10, H.264/ISO/IEC, September 2002.

[7] H. Nguyen et P. Duhamel. Compressed image and video redundency for joint source-channel decoding. *Golobecom*, 2003.