

Décodage robuste de vidéos H.264 AVC vs SVC : réduction de l'overhead et comparaison de performance

¹LEONARDO H. FONTELES, ¹PIERRE DUHAMEL, ²ANISSA MOKRAOUI

¹Laboratoire de Signaux et Systèmes (SUPELEC/CNRS/UPSud)
3 rue Joliot Curie 91190 Gif sur Yvette

²L2TI, Université Paris 13 Sorbonne Paris Cité, Institut Galilée
99 avenue Jean-Baptiste Clément 93410 - Villetaneuse

¹fonteles@lss.supelec.fr, pierre.duhamel@lss.supelec.fr,
²anissa.mokraoui@univ-paris13.fr

Résumé - Lors de travaux précédents [2], nous avons proposé des méthodes de décodage robuste de flux vidéo encodés suivant la norme H264 AVC, qui amélioreraient très nettement la qualité des vidéos reçues. Cette amélioration basée sur des algorithmes de décodage séquentiels vérifiant la syntaxe du flux binaire était obtenue au prix de l'insertion d'informations supplémentaires (la longueur des séquences CAVLC) dans le flux émis, qui augmentaient d'environ 30% la taille du flux transmis. Dans cet article, nous appliquons tout d'abord la même stratégie aux flux vidéo encodés suivant la norme SVC, qui permet d'obtenir des flux scalables, et réduisons cet inconvénient. Après application aux flux SVC de la méthode [2], nous expliquons comment réduire la taille des informations supplémentaires, qui peut alors tomber à environ 10% du flux total pour des performances encore excellentes. Enfin, nous démontrons la supériorité des flux encodés SVC par rapport aux flux AVC en présence de décodage robuste, et pour des tailles de séquences codées comparables.

Abstract - In previous work [2], we proposed robust methods for decoding video streams according to the H264 AVC standard, which significantly improved the received video quality. This improvement, based on sequential decoding algorithms checking the bitstream syntax, was achieved at the cost of inserting additional information (the length of the CAVLC sequences) in the transmitted stream, which increased by about 30% the size of the transmitted stream. In this paper, we first apply the same strategy to the encoded video stream using the SVC standard, which has the advantage of providing scalable streams and provide a method to reduce this problem: After applying the SVC stream method [2], we explain how to reduce the size of the additional information, which may then fall to about 10% of the total stream while keeping excellent performance. Finally, SVC encoded streams are shown to be more error resilient (vs. AVC streams), in the presence of robust decoding for comparable size coded sequences.

1 Introduction

La transmission de flux vidéo sur des canaux bruités peut aboutir à des résultats visuels catastrophiques, car un seul bit erroné peut désynchroniser le décodeur. La raison essentielle est l'utilisation de codes à longueur variable, éventuellement dépendants du contexte (CAVLC) qui sont par définition très sensibles aux erreurs.

En conséquence, le standard H.264 prévoit des outils de résistance aux erreurs pour en réduire l'impact visuel. Parmi ceux-ci, on trouve des outils de partition de données (DP), de la protection inégale (UEP) qui permet de mieux protéger les parties les plus sensibles du flux. Pendant la phase essentielle de paquetsisation des flux (qui utilise des « network abstraction layer units » – NALU), cet outil encapsule les éléments importants de syntaxe (entêtes et vecteurs mouvement) dans des NALUs de haute priorité (data partition A – DPA) et les éléments moins importants (DCT/Hadamard transform coefficients), encapsulés dans des NALUs moins prioritaires (data partition B – DPB – pour les macroblocs intra or data partition C – DPC – pour les macroblocs inter). L'utilisation de cette

partition des données permet de supposer que les NALUs de type DPA sont reçues sans erreur au récepteur. Cependant, les éléments qui se trouvent dans les NALUs DPB et DPC peuvent être lourdement impactés par des erreurs, et notre but est d'améliorer la qualité de la vidéo reçue en réduisant l'impact visuel de ces erreurs.

Dans ce contexte, de nombreuses techniques ont été proposées, comme la retransmission des paquets DPB et DPC corrompus, l'utilisation de codes correcteurs, et l'utilisation de techniques de « concealment » spatio-temporelles. Cependant, dans une situation de diffusion, une retransmission n'est pas possible, un codage correcteur pourrait résulter en une mauvaise utilisation du débit disponible sur des canaux variables, et l'« error concealment » peut s'avérer très coûteux en calcul dans le récepteur. C'est pourquoi nous avons préféré proposer des décodeurs robustes qui bénéficient de la redondance inhérente des flux vidéo paquetsés.

Combiné avec les méthodes de décodage protocole/canal conjoint que nous avons proposées par ailleurs [1], nous avons maintenant un ensemble de méthodes tout à fait réalistes permettant des réceptions

correctes des flux vidéo avec des rapports signaux à bruit dégradés de plus de 2 dB par rapport aux situations habituelles.

2 Adaptation du flux vidéo SVC pour un décodage robuste

La technique de décodage robuste proposée requiert l'utilisation d'un mécanisme de protection d'erreur inégale (Unequal Error Protection ou UEP) de façon à mieux protéger les éléments les plus importants contenus dans le flux vidéo (en-tête et vecteurs de mouvement) par rapport aux éléments moins importants (coefficients de la transformée Hadamard).

2.1 Le partitionnement des données

La norme H.264 AVC prévoit un certain nombre d'outils de protection de données parmi lesquels on trouve le partitionnement des données (data partition ou DP). Cet outil permet le partage de l'information des slices de la vidéo encodée dans des sections de données séparément décodables selon leur importance vis-à-vis de la qualité de la séquence vidéo décodée. Pour des raisons de compatibilité avec la norme AVC, le partitionnement des données a été également défini par la norme H.264 SVC pour la couche de base.

Cependant, l'outil de partitionnement de donnée pour la couche de base n'est pas implémenté dans l'encodeur SVC de référence de la norme, appelé JSVM software.

De plus, actuellement, la norme SVC ne spécifie pas l'utilisation du DP pour les couches d'amélioration. De ce fait, avant l'utilisation du décodeur robuste pour un flux SVC il est indispensable de modifier le flux SVC standard en intégrant le partitionnement des données pour la couche de base et la couche d'amélioration.

Puisque la norme H.264 SVC ne définit pas l'outil de DP pour les couches d'amélioration, nous avons donc créé un nouveau type de NALU tout en restant compatible avec la norme.

2.2 NALU de synchronisation

En plus de l'information présente dans la couche VCL (NALUs des slices encodés) la méthode de décodage robuste proposée se base sur la connaissance de la longueur des séquences binaires générées à l'encodage de l'information résiduelle stockée dans les NALUs DPB et/ou DPC. Dans le profil de base de la norme H.264, l'encodeur utilisé est un encodeur à longueur variable et à contexte adaptatif dénommé CAVLC (Context Adaptive Variable Length Coding). Ainsi, les séquences binaires générées par cet encodeur sont appelées séquences CAVLC. Chaque séquence représente l'information d'un bloc de 4 par 4 coefficients DCT (voir Section 4 pour plus de détail) issue d'un traitement de prédiction intra ou inter selon le

type de macrobloc. Cependant, l'information sur la taille des séquences CAVLC n'est pas prévue par la norme H.264 puisqu'une séquence CAVLC non corrompue peut être décodée uniquement à partir de mots binaires prédéfinis et connus au décodeur.

Afin de transmettre la longueur des séquences CAVLC au décodeur robuste et maintenir la compatibilité avec le décodeur SVC standard, nous utilisons deux types additionnels de NALU dénommés NALUs de synchronisation B (Synchro_B) et C (Synchro_C) en leur attribuant respectivement les numéros de code 26 et 27 (pour rappel, les numéros de code de 24 à 31 ne sont pas réservés par la norme). Les longueurs binaires des séquences CAVLC sont encodées à l'aide d'un codage Exponentiel-Golomb (ou Exp-Golomb) et sont regroupées dans les NALUs Synchro_B ou Synchro_C selon le type de NALUs (DPB ou DPC) auxquels les séquences appartiennent.

3 Algorithme de décodage robuste

Le principe du décodage robuste que nous proposons repose sur les mêmes bases que [2] pour le décodeur séquentiel H.264 AVC combiné à une vérification de syntaxe et utilisation de toutes les redondances.

La norme H.264 spécifie que pour chaque bloc de 4 par 4 coefficients DCT, l'encodeur CAVLC décompose l'ensemble des coefficients préalablement quantifiés dans différents éléments syntaxiques. Les séquences sont formées en utilisant des tables VLC spécifiques à chacun des différents éléments syntaxiques, à savoir :

- i) Le nombre total des coefficients non nuls et le nombre des *trailing ones*, à savoir le nombre de coefficients ayant une valeur absolue égale à un (*coeff_token*) ;
- ii) La signalisation de chacun des *trailing ones* (*trailing_ones_sign_flag*) ;
- iii) Les valeurs des coefficients non nuls et non *trailing ones* (*level_prefix* and *level_suffix*) ;
- iv) Le nombre total de zéros qui précèdent le dernier coefficient non nul dans l'ordre du balayage en zig-zag (*total_zeros*) ;
- v) Le nombre de coefficients zéro précédant les coefficients non nuls (*run_before*).

Dans ce contexte, le décodeur robuste utilise les contraintes présentes dans la syntaxe des codes à longueur variable et les combine avec les propriétés sémantiques (ex. le nombre maximum de coefficient) du codeur source afin d'extraire de l'information utile inhérente aux séquences CALVC. Les flux décodés sont donc parfaitement compatibles avec le standard, ce qui n'est habituellement pas garanti en cas d'erreur de transmission. Par contre, il est nécessaire de transmettre les marqueurs de position des séquences CAVLC dans les paquets DPB et DPC qui sont envoyés comme

information additionnelle. Ces marqueurs, qui représentent donc la longueur des séquences CAVLC, sont transmis dans des paquets NALU de synchronisation (Synchro_B et Synchro_C.).

Ensuite, le décodeur robuste divise chaque séquence CAVLC dans des segments plus petits composés de n bits. Etant donné qu'un ou plusieurs bits parmi les n peuvent être erronés, le décodeur calcule la probabilité *a posteriori* de chacun des 2^n différents segments de n bits basés sur la syntaxe et la sémantique de l'encodeur CAVLC. Ainsi, les M segments les plus probables sont retenus et concaténés avec les prochains segments les plus probables jusqu'à ce que la longueur totale de la séquence CAVLC soit atteinte. La séquence la plus probable est finalement retenue par le décodeur.

4 Réduction des coûts de synchronisation des NALU

La bonne performance du décodeur robuste est essentiellement due à l'information des longueurs. Nous avons réussi à réduire la taille de cette information supplémentaire en introduisant la longueur d'un groupe de séquence CAVLC, ce qui diminue l'overhead en proportion. Il a fallu modifier le M-algorithme en conséquence, comme écrit ci dessous.

L'algorithme robuste utilisé jusqu'ici suppose que la fin de la séquence CAVLC se situe à la fin de la zone indiquée par les marqueurs. Par conséquent, il a besoin uniquement de vérifier si la syntaxe des 2^s nouvelles séquences possibles (pour chaque sous-séquence de s bits) est compatible avec celle des tables CAVLC et, si c'est le cas, il calcule leur probabilité. Cependant, l'utilisation d'une information de longueur pour un groupe de NALU, plutôt que pour chaque NALU introduit naturellement une incertitude sur la longueur de chaque séquence individuelle.

Par conséquent, il est nécessaire de vérifier si plusieurs sous-séquences de syntaxe correcte forment réellement une séquence CAVLC complète. Ensuite, les M séquences terminées les plus probables (longueur pouvant varier de 1 au nombre de séquences restantes dans le groupe) sont stockées. Pour chacune d'elle, nous ré-exécutons la même procédure pour les bits restants. Ce processus doit être exécuté de manière séquentielle, puisque le choix de la table VLC pour la séquence suivante dépend du nombre total de coefficients non nuls dans la séquence actuelle.

Enfin, en introduisant un codeur entropique sur ces données, l'overhead est réduit à environ 10% du flux total, comme observé dans la Table 1.

5 Performance

Les courbes des figures 1 et 2 donnent les performances en terme de PSNR vs SNR canal pour la

séquence QCIF « Bus » encodée à 420 kbps, partitionnée et mise en paquets suivant la norme, modulés BPSK, et transmis à travers un canal Gaussien.

La figure 3 montre l'image numéro 60 de cette séquence au format QCIF, transmise sur un canal de 5 dB de SNR. La séquence SVC comporte deux couches, et le code canal utilisé est celui de la norme IEEE 802.11a.

La figure 3.a montre l'image telle qu'elle serait reçue en absence d'erreurs. La figure 3.b montre ce que fournirait un décodeur standard (muni d'error concealment) dans ces circonstances, alors que la figure 3.c montre l'image fournie par le décodeur robuste qui dispose de toutes ses informations additionnelles, et la figure 3.d le décodeur proposé ici, qui dispose de seulement $\frac{1}{4}$ des informations supplémentaires. L'amélioration de qualité fournie par le décodeur robuste est évidente, et il est difficile de voir la différence entre les images des figures 3.c et 3.d.

6 Conclusion

Les résultats de simulations démontrent que pour des conditions de transmission raisonnables et des débits pas trop élevés, le codeur SVC, combiné avec le décodeur robuste fournit de meilleures vidéos que le codeur AVC sous les mêmes conditions (longueur des flux). De plus, lorsque l'on regroupe les séquences CAVLC par paquets de 4, l'overhead est réduit d'un facteur jusque 2.5 (si l'on utilise des codes ExpGolomb), voire 4 avec un codeur entropique plus efficace, les résultats visuels sont toujours excellents, ce qui aboutit à un très bon compromis entre la qualité visuelle et le taux de transmission. Nous avons également montré que, malgré le coût binaire des marqueurs de synchronisation, le décodage robuste (*souple* ou *hard*) garantit une sensible amélioration de la qualité de la vidéo décodée par rapport au décodeur standard. Ceci peut se traduire en terme d'économie de puissance à l'émetteur ou d'augmentation du rayon de couverture.

Références

- [1] P. Duhamel, M. Kieffer, "Joint Source-Channel Decoding. A Cross-Layer Perspective with Applications in Video Broadcasting over Mobile and Wireless Networks," Dec., 2009, ISBN-13: 978-0-12-374449-4, 334 pages, Elsevier, EURASIP and Academic Press Series in Signal and Image Processing.
- [2] C. Marin, K. Bouchireb, M. Kieffer, P. Duhamel, "Joint Exploitation of Residual Source Information and MAC Layer CRC Redundancy for Robust Video Decoding," *IEEE Trans Wireless Com*, vol. 9 p.2165 - 2175 (2010).

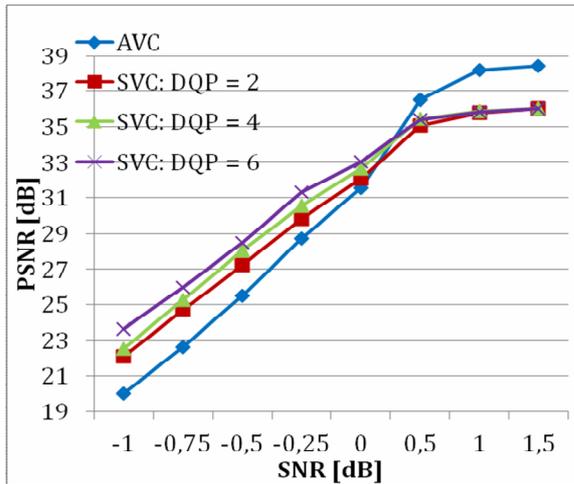


Figure 1 : PSNR vs SNR canal pour des vidéos décodées par AVC et SVC

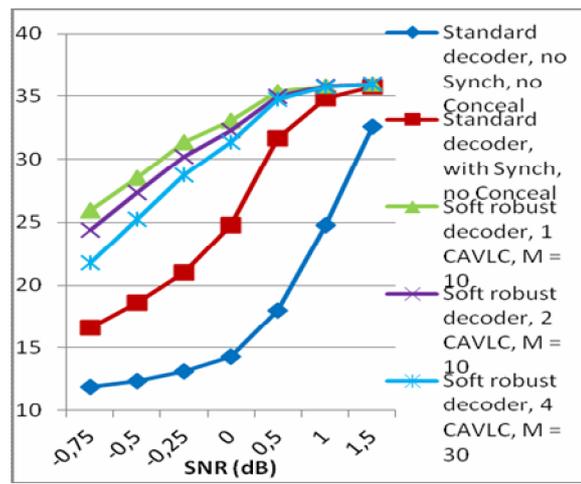
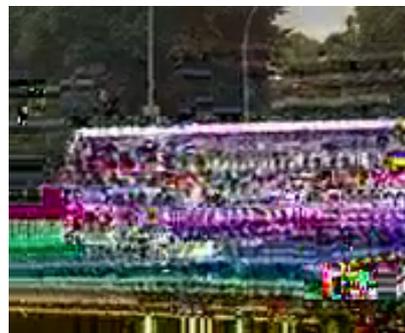


Figure 2 : PSNR vs SNR canal pour des vidéos décodées par AVC et SVC



(a)



(b)



(c)



(d)

Figure 3 : (a) Image 60 décodée sans erreur ; Flux SVC corrompu (SNR=5dB) : (b) décodeur standard avec error concealment ; (c) Décodeur robuste ; (d) Décodeur robuste proposé avec 1/4 d'informations supplémentaires

Tab 1 : Réduction de l'overhead : pourcentages d'overhead dans le flux vidéo

Séquence	Sync NALUs ExpGolomb (%)	Sync NALUs Entropy Based 2 CAVLC (%)	Sync NALUs Entropy Based 4 CAVLC (%)
AVC:			
High rate	26,22	10,98	6,22
Medium rate	34,99	16,00	9,19
Low rate	41,40	20,56	11,91
SVC: 2 Layers			
High rate, delta QP = 2	36,95	16,72	9,64
High rate, delta QP = 4	37,01	16,44	9,46
High rate, delta QP = 6	36,45	15,97	9,17
Medium rate, delta QP = 2	45,58	21,47	12,34
Medium rate, delta QP = 4	45,78	21,23	12,16
Medium rate, delta QP = 6	45,09	20,81	11,93