

Nouvelle technique de génération de l'information adjacente en codage vidéo distribué basée sur les algorithmes génétiques

JOUMANA FARAH¹, CHARLES YAACOUB^{1,2}, BEATRICE PESQUET-POPESCU²

¹Faculté des Sciences et de Génie Informatique, Université St-Esprit de Kaslik, Liban

²Département Traitement du Signal et des Images, Télécom ParisTech, Paris, France

¹joumanafarah@usek.edu.lb, charlesyaacoub@usek.edu.lb,

²beatrice.pesquet@telecom-paristech.fr

Résumé – Dans ce travail, nous nous proposons de fusionner plusieurs sources d'information adjacente (en anglais «Side Information» ou SI), en codage vidéo distribué. Ces SI, générées par plusieurs techniques connues, sont plus ou moins performantes suivant le type de la vidéo codée et le niveau de mouvement dans les trames de la séquence. La fusion des différentes sources se fait à l'aide d'un algorithme génétique (GA), ce qui constitue une nouveauté importante par rapport aux techniques antérieures de génération de la SI. Les résultats de simulation montrent une amélioration considérable de la qualité de la SI générée par le GA. Le gain en PSNR (Peak Signal to Noise Ratio), par rapport aux techniques précédentes, est la plupart du temps supérieur à 1 dB et peut même atteindre 20 dB pour certaines trames.

Abstract – The aim of this work is to propose a novel technique for improving the quality of the side information in distributed video coding. Based on genetic algorithms, our proposed technique combines several frames, interpolated using previously developed methods, in a fusion-based approach. Simulation results show a significant improvement in the side information quality. The gain in PSNR exceeds 1dB most of the time, and can even reach 20 dB for some frames, compared to classical interpolation techniques available in the literature.

1 Introduction

Dans les techniques actuellement utilisées pour la compression vidéo, le codeur exécute un grand nombre d'opérations à cause de l'étape d'estimation de mouvement dans le codage des trames prédictives [1]. Ceci conduit à un codeur très complexe, alors que le décodeur est très simple, du fait qu'il ne comprend principalement qu'une compensation de mouvement.

Le codage vidéo distribué (DVC) permet de résoudre le problème de la complexité de codage en déplaçant une grande partie du processus de codage vers le décodeur. Dans un codeur DVC [2-7], les trames vidéo sont divisées en deux groupes: les trames clés et les trames Wyner-Ziv (WZ). Ces dernières sont comprimées à l'aide d'un codeur de canal poinçonné. Au niveau du récepteur, les trames clés et certaines trames WZ précédemment décodées sont utilisées pour la génération de la SI nécessaire au décodage d'une trame WZ. Par suite, la technique utilisée pour la génération de la SI influera énormément sur la qualité de décodage, ainsi que sur le débit de transmission nécessaire des trames WZ. Plusieurs techniques ont été proposées dans la littérature pour améliorer la qualité de la SI générée au décodeur. Parmi elles, la transmission d'informations d'empreinte (en anglais «hash») de la trame WZ au codeur a été proposée dans [4-5]. Dans [8], deux SI différentes sont utilisées pour le décodage de chaque trame WZ. Toutes ces techniques présentent des avantages et des inconvénients suivant le type et la vitesse de mouvement dans la vidéo codée. Le but de ce travail est de proposer une technique, basée sur le GA,

permettant le fusionnement de plusieurs SI générées par les techniques d'interpolation précédemment développées. Le but du GA est de déterminer la meilleure candidate SI dans différentes régions de la trame vidéo. A notre connaissance, une telle stratégie n'a jamais été utilisée en DVC. A cette fin, nous commencerons, dans la partie 2, par présenter l'algorithme proposé, suivi par les résultats de simulation dans la partie 3.

2 Description de l'algorithme de fusion génétique

Le GA [9] est un outil d'optimisation basé sur la théorie de l'évolution en génétique. Son application nécessite tout d'abord une représentation génétique de l'espace des solutions, ainsi qu'une fonction d'évaluation (en anglais «fitness function») de chaque solution possible. Un *chromosome* (solution candidate) sera défini par un ensemble de pixels d'une trame, arrangés sous la forme d'un bloc. L'ensemble des chromosomes dans l'espace des solutions sera appelé *population*. A l'origine, pour un certain bloc d'une trame WZ, chacun des blocs à la même position dans les trames SI, générées par les différentes méthodes d'interpolation, constitue une solution possible. La similarité entre un chromosome et le bloc WZ correspondant est déterminé par la fonction d'évaluation. Puisqu'en DVC seules les trames clés sont disponibles au décodeur, il est indispensable de transmettre certaines informations sur les trames WZ de façon à permettre l'évaluation d'une certaine solution. Etant donné que la complexité de codage est une contrainte incontournable en DVC,

l'extraction de ces informations supplémentaires ne doit pas constituer une charge significative pour le codeur, de même que le débit nécessaire pour leur transmission. Pour cette raison, nous avons opté pour l'utilisation d'un hash-DVC tel que celui décrit dans [4-5]. Dans ce dernier, un sous-ensemble des coefficients DCT (Discrete Cosine Transform) de chaque trame WZ est transmis sous forme de mot d'empreinte au décodeur. Dans notre système, nous nous proposons d'utiliser ces empreintes pour le calcul de la fonction d'évaluation. Ainsi, par rapport au hash-DVC, aucune charge calculatoire n'est rajoutée par notre système au codeur. Par suite, nous définissons la fonction d'évaluation d'un chromosome par $F = 1 / D$, où D est l'erreur quadratique moyenne (EQM) entre l'empreinte reçue pour le bloc WZ et l'empreinte générée dans le décodeur pour le chromosome en cours d'évaluation. Pour chaque bloc d'une trame WZ, l'algorithme consiste en la procédure itérative représentée par la figure 1, où i est le numéro de l'itération et I_{max} le nombre maximal d'itérations possible. Les principales étapes du GA sont les suivantes:

1^{ère} étape: Initialisation

La population initiale est créée en considérant, tout d'abord, les trames SI générées par la méthode d'interpolation moyenne (AVI) [2], par l'interpolation par compensation de mouvement à vecteurs symétriques (MCI) [2] et par l'interpolation par compensation de mouvement basée sur les empreintes (HMCI) [5]. En HMCI, l'estimation de mouvement se fait par minimisation de l'EQM entre l'empreinte reçue et celle du bloc SI candidat. A ces trois candidats, on ajoute les deux trames clés utilisées dans les processus d'interpolation, d'où un total de cinq candidats initiaux qui seront dupliqués de façon proportionnelle à leur niveau d'adaptation (estimé par la fonction d'évaluation), jusqu'à ce que la population atteigne une taille désirée S_p .

2^{ème} étape: Croisement

Les chromosomes de la population courante sont aléatoirement entrelacés et arrangés en paires. Une paire est combinée par échange de gènes (pixels), avec une probabilité P_c , pour produire deux chromosomes « enfants » comme illustré sur la figure 2. Autrement dit, les parents demeurent inchangés avec la probabilité $1-P_c$. Quant à la position du croisement, elle est choisie de façon aléatoire.

3^{ème} étape: Mutation

Elle consiste à inverser un bit d'un gène d'un chromosome choisi au hasard. La mutation a lieu avec une faible probabilité P_m [10] et permet d'étendre l'espace des solutions, de façon à diminuer le risque de converger vers un optimum local.

4^{ème} étape: Evaluation et sélection

La fonction d'évaluation (fitness) est tout d'abord estimée pour chaque chromosome de la population courante. Ensuite, parmi les S_p chromosomes de la population, seuls les S_f chromosomes les plus

« adaptés » (c.à.d. ayant la plus grande évaluation) sont sélectionnés, tandis que les $S_p - S_f$ chromosomes restants sont abandonnés pour permettre l'introduction de nouveaux chromosomes. A noter que S_f ne doit pas être très proche de S_p , afin de ne pas permettre aux chromosomes les moins adaptés de se reproduire, et ne doit pas être non plus trop faible afin de permettre à une grande variété de chromosomes d'échanger leurs gènes et de se reproduire.

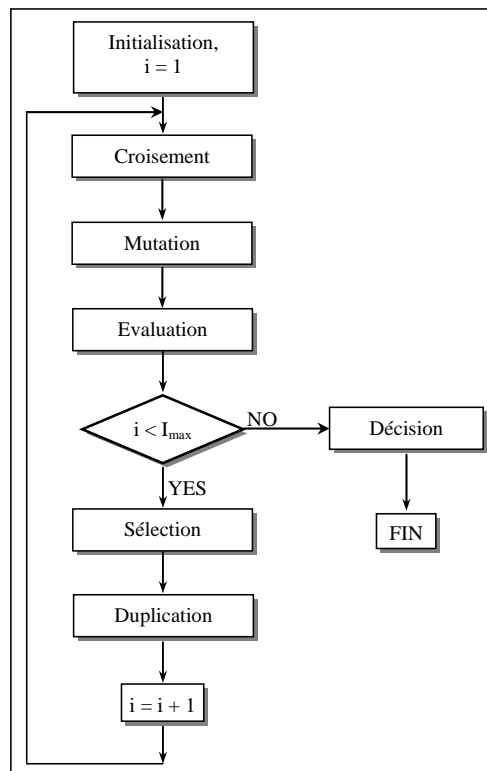


Figure 1 : Organigramme de l'algorithme génétique proposé.

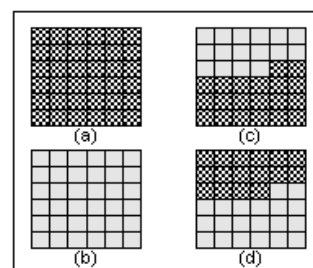


Figure 2 : Exemple de croisement de deux parents (a) et (b) pour produire deux enfants (c) et (d).

5^{ème} étape: Duplication

Chacun des chromosomes restants est dupliqué un nombre de fois proportionnel à son évaluation, de façon à atteindre la taille de population S_p .

Les étapes 2 à 5 sont répétées jusqu'à ce que le nombre maximal d'itérations soit atteint. Finalement, le chromosome le plus adapté est retenu pour être utilisé comme SI dans le décodage du bloc correspondant de la trame WZ.

Cependant, nous avons remarqué que lorsque le niveau de mouvement est très important dans certaines parties de la vidéo, les techniques d'interpolation

utilisées pour initialiser le GA échouent dans ces parties, ce qui se répercute sur les performances du GA. Dans ces conditions, il peut s'avérer plus avantageux d'adopter tout simplement la DCT inverse (IDCT) des mots d'empreinte reçus. Cependant, le bloc IDCT ne peut être introduit comme candidat dans le GA puisqu'avec la fonction d'évaluation adoptée, le GA convergerait constamment vers ce bloc. Pour cette raison, nous avons choisi d'incorporer la IDCT de la façon suivante dans la technique de fusion: la fonction d'évaluation F_{GA} du meilleur chromosome retenu comme sortie du GA est comparée à un seuil F_T : si $F_{GA} > F_T$, le système adopte la sortie du GA comme SI, sinon, il adopte le bloc correspondant de la trame IDCT.

3 Résultats pratiques

Trois séquences vidéo ont été testées dans nos simulations : Carphone, Foreman et News de longueur 300 trames chacune. Elles sont échantillonnées au débit de 30 trames/s et codées avec une taille de GOP (Group of Pictures) de 2, c.à.d. qu'une trame WZ sépare chaque deux trames clés. Les mots d'empreinte transmis par le codeur constituent 1/8 des coefficients DCT de chaque trame WZ. La taille de bloc considérée est de 16×16 pixels. Quant aux paramètres du GA, les suivants ont été choisis après un certain nombre de tests : $S_p = 60$, $S_f = 40$, $I_{max} = 10$, $P_c = 0.8$, $P_m = 0.01$ et $F_T = 0.02$.

Nous estimons tout d'abord les performances du système en fonction du PSNR entre la trame SI générée par le GA et la trame WZ correspondante. La figure 3 montre les variations du PSNR de la séquence Carphone obtenu par le GA, d'un côté, et par les différentes techniques d'interpolation, de l'autre, pour les trames WZ de 60 à 110 (le graphe a été limité à cet intervalle pour une meilleure clarté). On remarque la supériorité du GA sur les autres techniques pour presque la totalité des trames, sauf pour quelques-unes où le PSNR du GA converge vers celui de la meilleure parmi les autres courbes.

Dans la table 1, nous montrons les gains moyen et maximal obtenus par notre GA pour les différentes séquences vidéo. Nous remarquons que le plus petit gain est obtenu par rapport à la technique HMCI puisque cette dernière présente, en moyenne, les meilleures performances par rapport aux méthodes AVI

et MCI. Nous notons aussi que les performances du GA sont considérables pour la séquence News dans laquelle le gain maximal dépasse 19 dB. Ceci est dû à la nature particulière (complexe) du mouvement dans cette séquence : la scène représente deux présentateurs quasi-statiques (en mouvement très lent) en premier plan, avec un couple de danseurs à l'arrière-plan alternant les mouvements rapides et lents.

Tab 1 : Gain moyen et maximal du PSNR (en dB) pour les différentes séquences vidéo.

	AVI	MCI	HMCI
Carphone	(3.8, 9.0)	(3.4, 9.2)	(3.2, 7.6)
Foreman	(4.6, 15.5)	(2.4, 14.0)	(1.6, 11.3)
News	(7.0, 19.6)	(7.1, 21.9)	(6.2, 19.7)

En fait, il faut noter que ces gains en PSNR de la SI ne sont pas nécessairement les mêmes en termes de performances Débit-Distorsion (Rate-Distortion ou RD) du codec vidéo distribué (c.à.d. après turbo-décodage et reconstruction). Dans le but d'évaluer notre algorithme en termes des performances RD, nous avons codé les séquences vidéo avec notre codec WZ présenté dans [6], mais sans la suppression du canal de retour. Les figures 4 à 6 montrent le PSNR moyen des trames WZ décodées et reconstruites en fonction du débit WZ moyen. Puisque le hash-DVC présente généralement de meilleures performances par rapport aux codecs WZ classiques, nous comparons les résultats du GA à ceux d'un codec dans lequel la SI est générée par la méthode HMCI seule dans le décodeur. Notons que puisque les mêmes informations d'empreinte sont transmises dans les deux cas, le débit supplémentaire nécessaire pour la transmission des empreintes n'a pas été pris en considération. Autrement, les courbes (GA et HMCI) devraient être décalées horizontalement de la même valeur. Comme prévu, le GA présente une amélioration notable par rapport à la technique HMCI. Le gain moyen dépasse 3 dB avec la séquence Carphone, 1 dB avec Foreman et 6 dB avec News.

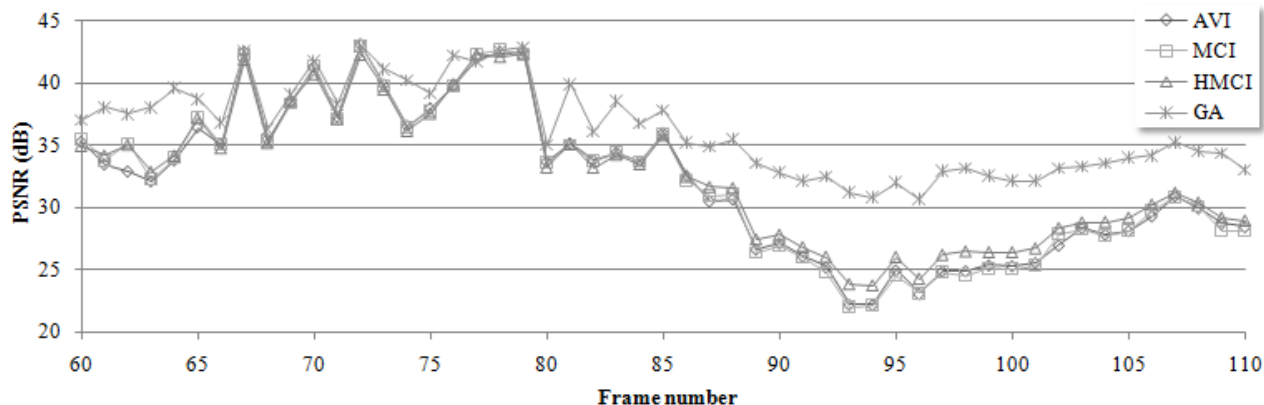


Figure 3 : PSNR en fonction du numéro de la trame WZ dans la séquence Carphone.

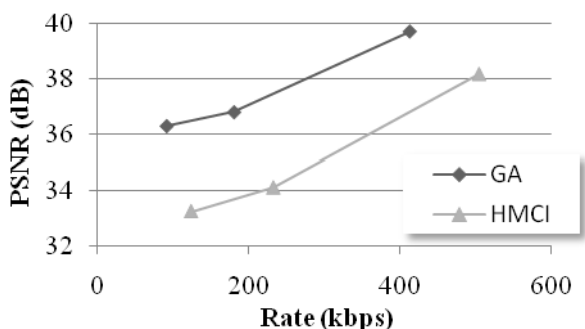


Figure 4 : Courbe de débit-distorsion de la séquence Carphone

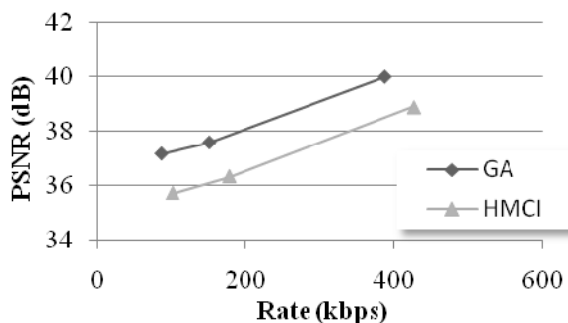


Figure 5 : Courbe de débit-distorsion de la séquence Foreman

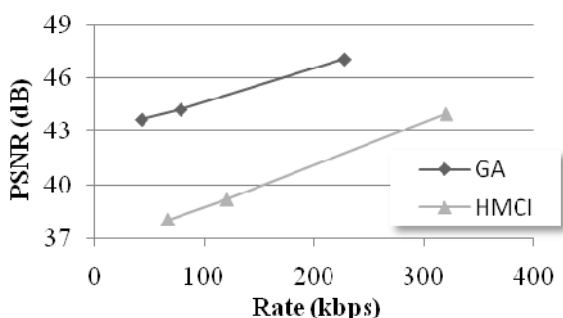


Figure 6 : Courbe de débit-distorsion de la séquence News

4 Conclusion et perspectives

Le but de cet article est d'introduire une nouvelle approche permettant l'amélioration de l'information adjacente en codage vidéo distribué. Basé sur le principe d'évolution en génétique, l'algorithme proposé permet de fusionner plusieurs trames interpolées par des méthodes précédemment connues. Le GA n'ajoute aucune complexité d'encodage et aucun débit de transmission supplémentaire par rapport au codeur hash-DVC. Les résultats de simulation montrent que le GA permet, la plupart du temps, de réaliser un gain significatif dans la qualité de la SI.

Du point de vue complexité, le GA présente une charge calculatoire importante, du fait que le mot d'empreinte doit être estimé, dans le décodeur, pour chaque chromosome et dans chaque itération. Cependant, avec les derniers avancements technologiques, une implémentation exploitant la parallélisation des processus peut permettre son utilisation dans les applications temps-réel. D'un autre côté, les paramètres du GA (en particulier S_p et I_{max})

peuvent être réglés afin de limiter la charge calculatoire de l'algorithme. De plus, un critère d'arrêt supplémentaire (par exemple le niveau de variabilité de la fonction d'évaluation d'une itération à l'autre) peut être appliqué pour limiter le nombre d'itérations réalisées par le GA à celui nécessaire à la convergence de l'algorithme.

5 Remerciements

Ce travail a été en partie financé par le CNRS Libanais et a eu lieu dans le cadre du projet franco-libanais CEDRE.

6 Bibliographie

- [1] ITU-T and ISO/IEC JTC1, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264 – ISO/IEC 14496-10 AVC, 2003.
- [2] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv Coding of Motion Video", 36th Asilomar Conf. Signals, Systems and Computers, pp. 240-244, Nov. 2002.
- [3] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding", 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, Jun. 2005.
- [4] A. Aaron, S. Rane and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver", IEEE International Conference on Image Processing, ICIP-2004, Singapore, Oct. 2004
- [5] J. Ascenso, F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding", International Conference on Image Processing (ICIP), USA, Sept. 2007.
- [6] C. Yaacoub, J. Farah, B. Pesquet-Popescu, "Feedback Channel Suppression in Distributed Video Coding with Adaptive Rate Allocation and Quantization for Multiuser Applications", EURASIP Journal on Wireless Communications and Networking (WCN), 2008.
- [7] C. Yaacoub, J. Farah, B. Pesquet-Popescu, "New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding", International Journal of Digital Multimedia Broadcasting, special issue on Advances in Video Coding for Broadcast Applications, 2009.
- [8] D. Kubasov, J. Nayak and C. Guillemot, "Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information", IEEE International Workshop on Multimedia Signal Processing, Crete, Greece, Oct. 2007.
- [9] D.E. Goldberg, "Genetic Algorithms: Search, Optimization, and Machine Learning", Addison-Wesley, Reading, MA, 1989.
- [10] P. H. Chang, J. J. Leou, and H. C. Hsieh "A genetic algorithm approach to image sequence interpolation", EURASIP Journal on Signal Processing: Image Communication, Vol. 16, No. 6, pp. 507-520, 2001.