Codage vidéo multi-vue pour une vision interactive au récepteur

Thomas MAUGEY, Pascal FROSSARD

École Polytechnique Fédérale de Lausanne (EPFL)
Laboratoire de Traitement du Signal 4 (LTS4)
Lausanne, Suisse
thomas.maugey@epfl.ch, pascal.frossard@epfl.ch

Résumé – Dans cet article, nous proposons un nouveau système de codage de vidéo multi-vues, dont la spécificité est de proposer à l'utilisateur une interactivité lui permettant de changer de vue en temps réel. Contrairement aux schémas de codages existant dans la littérature, notre approche tire son originalité du fait qu'elle permet un décodage à faible charge de calcul, au prix d'un ajout en débit qui s'avère être raisonnable dans les expériences présentées. Afin de réduire encore plus ce coût induit par la possibilité d'interactivité, nous proposons également d'optimiser les performances de transmission en se fondant sur une modélisation du comportement de l'utilisateur.

Abstract – In this paper, we propose a new coding scheme for multi-view video sequences, that offers to the user an interactivity which allows him to navigate between the views with no delay. In opposition to the schemes existing in the literature, our approach's originality lies on a low decoding complexity, compensated by a reasonable additive rate as presented in the section dedicated to experiments. In order to reduce this cost brought by this interactivity, we propose to optimize the transmission performance by modeling the user behavior and using this model for a smart rate allocation.

1 Introduction

Un des objectifs principaux de la recherche en description de contenu multimédia est aujourd'hui de transmettre les données de manière à ce que l'utilisateur ait une sensation d'immersion augmentée. Le codage vidéo multi-vue en est un des exemples les plus caractéristiques car il vise à ce que l'utilisateur ait la sensation d'observer une scène en trois dimensions. Pour cela, plusieurs approches sont possibles. Parmi elles, le codage vidéo stéréoscopique [1] consiste à transmettre deux images correspondant à l'oeil droit et gauche afin de donner une profondeur aux objets de la scène. Une autre approche intéressante consiste à transmettre un nombre plus important de vues afin que l'utilisateur puisse naviguer entre elles. Ce sont les systèmes, dits de "télévision à point de vue libre" [2], qui suscitent de nos jours de nombreuses recherches. La sensation d'immersion est alors apportée par la possibilité de se déplacer dans la scène, et donc donc l'impression de se déplacer dans une scène en trois dimensions [3].

Les schémas permettant aujourd'hui d'offrir à l'utilisateur une interactivité lui donnant la liberté de choix de la vue affichée, sont multiples. Ils varient principalement selon les données brutes traitées. En effet, certains schémas [4] travaillent avec un ensemble de vues (texture) et estiment la profondeur au décodeur afin d'y permettre la génération de vues synthétiques. Dans d'autres approches [5], la profondeur est capturée en même temps que la texture, et est transmise comme une séquence vidéo classique. Pour ces deux types de schémas, la génération de vues synthétiques au décodeur nécessite, en plus, des algorithmes de remplissage d'image afin d'estimer les parties manquantes dues aux occlusions. Ces techniques sont complexes et donc nécessitent un décodeur spécifique, ce qui limite les capacités d'utilisation d'un tel système.

Quelques approches, cependant, proposent d'anticiper dès l'encodeur la possibilité d'interactivité donnée à l'utilisateur. Certains auteurs [6] proposent un schéma complet de transmission sur réseau de flux vidéos destinés à une vision interactive. Bien que celui-ci se fonde sur une approche permettant d'optimiser l'utilisation de la bande passante, les algorithmes au décodeur restent complexes et le nombre de vues disponibles limité. Dans [7, 8], les trames prédites sont encodées avec un codeur Wyner-Ziv (*i.e.* codeur *distribué* [9]) afin que le décodage ne dépende pas de la trame utilisée pour la prédiction. Autrement dit le résidu transmis peut s'adapter à tout type de navigation. Les trois facteurs limitants d'une telle approche sont d'une part, l'utilisation d'une boucle de retour très fréquente (pour le décodage de chacune des trames), d'autre part un choix des vues disponible limité (car la synthèse de vue n'est pas considérée), et enfin une forte complexité au décodeur due aux algorithmes itératifs des techniques de codage distribué. Notons également que ce type de décodage n'est pour le moment pas standardisé ce qui peut également constituer un problème.

Dans cet article, nous proposons une généralisation du schéma de codage vidéo multi-vue avec possibilité d'interactivité au décodeur proposé dans [10]. La particularité de celui-ci est qu'il fonctionne avec des décodeurs standards non complexes et qu'il permet une navigation sur un large ensemble de vues (capturées et virtuelles). L'idée est de transférer la plupart des algorithmes complexes (synthèse d'image notamment) du décodeur à l'encodeur. La baisse de la complexité au décodeur implique une

augmentation de l'information à transmettre afin de corriger le manque de précision des méthodes utilisées. Les résultats expérimentaux obtenus montre que le débit ajouté par la transmission de cette information est raisonnable comparativement au gain en complexité dont le décodeur bénéficie et à la qualité de l'interactivité offerte. De plus, nous proposons d'utiliser des modèles de comportement d'utilisateurs lors de la navigation afin d'optimiser au mieux les performances débit-distorsion de notre schéma. Une fois de plus, les résultats expérimentaux valident l'efficacité de notre approche.
Dans la Sec. 2, nous décrivons la structure générale du système proposé, ensuite (Sec. 3), nous présentons les résultats de notre schéma en terme de débit et de complexité de calcul. Les conclusions et travaux futures sont donnés en Sec. 4.

2 Schéma proposé

2.1 Structure de base

Supposons qu'une scène dynamique soit capturée par N caméras (couleur+profondeur) dont on connait les paramètres de calibrage (intrinsèques et extrinsèques). Ces vues, dites de références, sont transmises et stockées sur un serveur. Sur ce serveur (supposé supporter une lourde charge de calcul) sont générées et encodées M vues intermédiaires synthétiques, placées de manière à effectuer une transition lisse entre les vues de référence. Contrairement aux schémas existants qui ne considèrent ces M vues qu'au décodeur, l'approche proposée dans cet article tient compte dès le serveur de l'ensemble des $N+M$ vues visibles au décodeur. Les trames sont toutefois divisées en deux types, traités différemment : les trames de référence appartenant aux N vues dont on a fait l'acquisition, et les trames e appartenant aux vues virtuelles (voir la Fig. 1).

Sur le serveur les vues de référence (couleur + profondeur) sont encodées indépendamment entre elles en utilisant JSVM (avec prédiction entre images de la même vue) [11] de manière à ce que l'on puisse accéder à plusieurs niveaux de qualité. Les trames "e" estimées grâce aux images de profondeur et de texture des vues de références, comme dans [12]. Les zones d'occlusion sont estimées avec une méthode de remplissage classique [13]. Ces zones reconstruites constituent donc les résidus encodés et transmis au décodeur. Les trames "e" sont codées indépendamment entre elles à l'aide également d'un codage scalable en qualité (JSVM utilisé en mode intra). Afin de diminuer encore plus la complexité du décodage, nous proposons également de calculer la vue synthétique par bloc (et non par pixel comme dans les algorithmes originaux). Naturellement les résidus (trames "e" stockées et transmises) sont plus lourds. Ce compromis efficacité de codage/complexité est étudié dans la Sec. 3.2.

Du côté du décodeur, supposons qu'à un instant t l'utilisateur observe la vue numéro n. Supposons de plus que le récepteur ne puisse, que tous les N_T trames, envoyer au serveur l'information de la position de l'utilisateur. Le serveur doit pouvoir anticiper la navigation de l'utilisateur pour les N_T prochaines

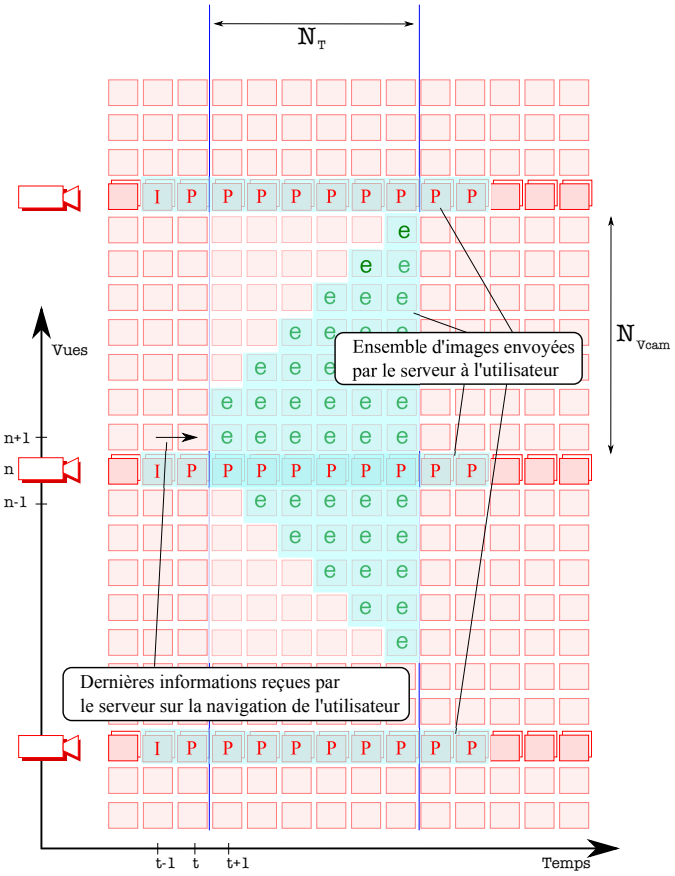

FIGURE 1 – Exemple d'un ensemble d'images à envoyer du serveur à l'utilisateur pour une position initiale (donnée par la flèche) et un N_T donnés. Les trames notées "e" sont les trames virtuelles, les trames notées "IPPPP" sont les trames des vues de référence dont la périodicité du GI n'est pas forcément synchronisée avec N_T.

trames. Etant donné que l'utilisateur ne peut naviguer que de gauche à droite, le serveur ne lui envoie que les trames "visibles" (en bleu sur la Fig. 1) à partir de la position initiale. Le serveur envoie tout d'abord les trames des caméras de référence utilisées. Il est contraint d'envoyer tout le groupe d'image (GI), car la taille de celui-ci n'est pas forcément alignée avec N_T. Puis le serveur envoie les résidus, ou trames "e" que l'utilisateur peut atteindre. Le décodeur peut alors reconstruire les vues virtuelles visionnées par l'utilisateur. Avec ce système l'utilisateur a un niveau d'interactivité optimal dans le sens où il peut à tout moment changer de vue. Dans la section suivante, nous proposons d'utiliser un modèle de navigation afin d'optimiser le débit utilisé entre le serveur et l'utilisateur.

2.2 Allocation de débit fondée sur une modélisation du comportement de l'utilisateur

Le schéma présenté dans la section précédente est une structure de base dont certains blocs peuvent être optimisés. En

particulier, la transmission des trames "e" est sous-optimal si tous les résidus sont transmis avec la même précision. En effet, à une situation donnée (positions précédentes de l'utilisateur) certaines trames ont plus de chances que d'autres d'être visionnées et la précision de leur quantification doit être supérieure à celle d'autres images moins probablement utilisées. En premier lieu, nous adoptons un modèle de transition schématisé dans la Fig. 2. Celui-ci estime la probabilité d'apparition de la trame suivante en fonction des deux trames passées. A partir de ce modèle, nous en déduisons pour chaque trame une probabilité d'apparition dont un exemple est donné dans la Fig. 3. En se fondant sur cette probabilité le serveur résout un problème du type :

$$\min_{\mathbf{r}} \sum_{n} \sum_{t} D(\mathbf{r}(t,n))P(t,n) + \lambda||\mathbf{r}||_1$$

où $\mathbf{r}$ est le vecteur de distribution du débit, $D(\mathbf{r}(t,n))$ est la distorsion de la trame à l'instant t appartenant à la vue n, et $P(t,n)$ la probabilité de la même trame.

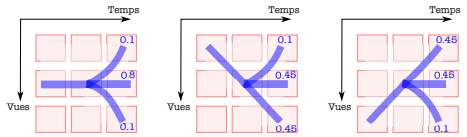

FIGURE 2 – Modèle de probabilité employé

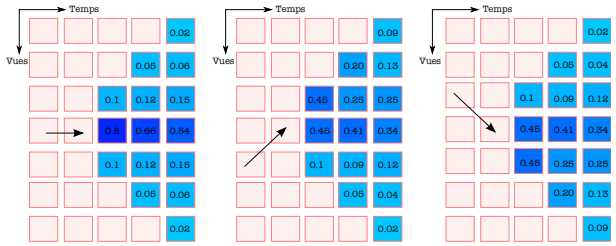

FIGURE 3 – Exemple de probabilités des trames d'un GI

3 Résultats

L'évaluation des performances de notre schéma de codage s'est effectuée en trois temps par l'intermédiaire de nombreux tests. Premièrement, nous montrons les performances générales du codeur, et l'influence des différents paramètres (taille du GI des trames intra, quantification des différents types d'images, etc.). Puis dans une deuxième partie nous étudions le compromis entre efficacité de calcul et efficacité de transmission. Ensuite, nous prouvons l'efficacité de la solution consistant à utiliser le modèle de comportement de l'utilisateur. Enfin nous proposons une comparaison de notre méthode avec d'autres approches simples pour le codage de contenu multi-vues.

	Débit total	Débit de la texture	Débit de la profondeur	Débit des trames "e"
N_T = 4 GI = 4	22.59	16.49 (73 %)	3.21 (14 %)	2.89 (13 %)
	6.99	4.26 (61 %)	1.68 (24 %)	1.05 (15 %)
	2.83	1.51(53 %)	0.80 (28 %)	0.52 (19 %)
	1.57	0.78 (49 %)	0.37 (24 %)	0.42 (27 %)
N_T = 8 GI = 8	16.39	11.61 (71 %)	2.47 (15 %)	2.30 (14 %)
	5.14	2.84 (55 %)	1.28 (25 %)	1.01 (20 %)
	2.37	1.00 (42 %)	0.59 (25 %)	0.78 (33 %)
	1.53	0.50 (33 %)	0.26 (17 %)	0.76 (50 %)

TABLE 1 – Débit de codage pour les différents types d'images, *breakdancer* [in Mbs] lorsque la taille des groupes d'images est alignée sur N_T.

3.1 Comportement du schéma

Dans le Tab. 1, nous présentons un exemple de répartition des débits pour différents points de quantification, différentes valeurs de N_T et différentes tailles des GI pour les vues de références. Le débit des trames "e", comparativement à leur grand nombre, reste assez limité, surtout si on considère que leur codage n'est pas encore optimisé. Dans ce tableau, la taille du GI du codage des vues de référence est calé sur N_T. Or lorsque la taille du GI est trop faible, les performances de codage des vues de référence deviennent sous-optimales. On se retrouve alors dans le cas où un N_T (et donc une taille de GI) plus faible obtient de moins bonnes performances qu'un N_T plus grand sensé devoir transmettre plus de trames (entre $N_T = 4$ et $N_T = 8$ on peut parfois gagner entre 1 et 2 dB). C'est pourquoi nous avons envisagé de ne pas caler la taille des GI sur la valeur de N_T. Pour un GI de taille fixe (par exemple 16), si N_T varie de 2 à 4 on observe une augmentation du débit total de 23%, et s'il augmente de 2 à 8, l'augmentation du débit est de 77%. Cette évolution plus logique est due au plus grand nombre de trames à transmettre dans le cas d'un grand N_T. Si ce débit supplémentaire est trop fort, on peut facilement essayer de baisser le nombre de trames "e" transmises en diminuant le niveau d'interactivité.

3.2 Complexité

Réduire la complexité du décodeur implique nécessairement une compensation en terme de débit ou de qualité. Dans le schéma que nous proposons, l'estimation des vues synthétique peut s'effectuer à différentes tailles de bloc (1 pour une précision maximale, 4, 8 ou 16). Lorsque la taille des blocs augmente, le nombre d'opérations lors de la synthèse de vue diminue, et donc la complexité également. Cependant, avec des tailles de bloc supérieures à 1 deux pixels voisins pourront avoir la même valeur de projection (disparité) alors que ce ne serait pas le cas si la compensation se faisait de manière dense. Ainsi on perd en qualité, et la taille des résidus (trames "e") à stocker sur le serveur devient plus importante. Le Tab. 2 montre qu'en augmentant la taille des blocs, on peut réduire le temps de calcul considérablement (le temps de décodage peut baisser jusqu'à 4.47% du temps pris lorsque l'estimation est dense). Le stockage supplémentaire nécessaire reste dans ce cas très

taille du bloc	4	8	16
Temps de calcul relatif par rapport au cas dense	12.34 %	5.72 %	4.47 %
Surplus de stockage des trames "e" sur le serveur par rapport au cas dense	3.32 %	4.32 %	5.37 %

TABLE 2 – Etude du compromis complexité/erreur d'estimation pour différentes tailles de bloc. Les valeurs présentées sont toutes données en % par rapport au cas où l'estimation est dense, c'est à dire faite pixel par pixel.

raisonnable (5.37 % de surplus de stockage par rapport au cas dense).

3.3 Utilisation d'un modèle de comportement

Lors de l'accès aux trames "e" par le décodeur, l'idée initiale consistait à effectuer la transmission de chacune d'entre elles avec le même débit, sans prendre en compte de leur probabilité d'apparence. Comme nous l'avons expliqué précédemment, nous avons considéré un modèle de comportement de l'utilisateur et nous l'avons pris en compte dans le choix des débits alloués. Les résultats sont montrés dans le cas où une vue de référence est utilisée lors de la synthèse de vue. On voit que l'allocation de débit augmente les performances débit-distorsion (jusqu'à 0.3 dB).

Il est important de noter que ces tests ont été obtenus avec un modèle de comportement particulier, utilisé pour l'allocation, mais aussi pour la génération des chemins aléatoires adoptés pour les tests expérimentaux. Cette approche revient simplement à supposer que le modèle est connu mais n'enferme pas le système proposé dans un modèle précis. En effet, l'allocation de débit se fonde seulement sur une probabilité d'apparition des trames, et donc peut fonctionner avec n'importe quel modèle de comportement du moment que celui-ci permet de fournir ces probabilités.

3.4 Comparaison

Nous nous proposons dans cette section de situer les performances de notre codeur dans un contexte plus général. Vu qu'il n'existe pas vraiment d'autres schémas similaires dans la littérature (avec une faible complexité au décodeur et des vues synthétiques), nous avons envisagé une autre solution plus intuitive. Nous avons comparé les performances débit-distorsion de notre schéma avec celui qui consisterait à ne pas coder les trames "e" ce qui permet de mesurer l'intérêt de transmettre ces trames "e". Dans la Fig. 4, nous pouvons voir que nous obtenons de bien meilleures performances que lorsqu'elles sont transmises.

4 Conclusion

Le schéma proposé présente une nouvelle approche de codage permettant d'atteindre des niveau d'interactivité très satisfaisants dans le sens où il permet à l'utilisateur de changer

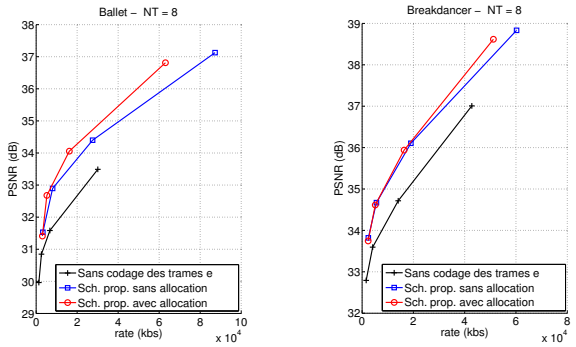

FIGURE 4 – Résultats débit-distorsion pour $N_T = 8$ prouvant l'efficacité de l'approche avec allocation.

de point de vues sur toutes les trames sans aucun délai. Le débit permettant cette interactivité reste raisonnable. De plus le schéma proposé fonctionne avec une complexité de décodage très faible comparativement aux autres méthodes existantes de la littérature.

Références

[1] P. Merkle, H. Brust, K. Müller, and T. Wiegand, "Stereo video compression for mobile 3d services," in *3D TV Conference*, Cancun, Mexico, Jun. 2009.

[2] M. Tanimoto, MP Tehrani, T. Fujii, and T Yendo, "Free-viewpoint TV," *IEEE Signal Processing Magazine*, vol. 11, pp. 67–76, 2011.

[3] K. Müller, P. Merkle, and T. Wiegand, "3d video representation using depth maps," *Proc. IEEE*, vol. 99, pp. 643–656, 2011.

[4] W. Li, J. Zhou, B. Li, and MI Sezan, "Virtual view specification and synthesis for free viewpoint television," *IEEE Trans. on Circ. and Syst. for Video Technology*, vol. 19, pp. 533–546, 2009.

[5] KJ Oh, S. Yea, and YS Ho, "Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3-D video," in *Picture Coding Symposium (PCS)*, Chicago, IL, USA, May 2009.

[6] Z. Huang, W. Wu, K. Nahrstedt, R. Rivas, and A. Arefin, "Synccast : synchronized dissemination in multi-site interactive 3d tele-immersion," in *Proc. ACM Int. Conf. on Multimedia*, San Jose, California, USA, Feb. 2011.

[7] G. Cheung, A. Ortega, and NM Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. on Image Proc.*, vol. 3, pp. 744–761, 2011.

[8] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Using distributed source coding and depth image based rendering to improve interactive multiview video access," in *Proc. Int. Conf. on Image Processing*, Bruxelles, Belgium, 2011, vol. 1, pp. 5–8.

[9] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the receiver," *IEEE Trans. on Inform. Theory*, vol. 22, pp. 1–11, Jan. 1976.

[10] T. Maugey and P. Frossard, "Interactive multiview video system with low decoding complexity," in *Proc. Int. Conf. on Image Processing*, Bruxelles, Belgium, Sep. 2011.

[11] ITU-T and ISO/IEC JTC1, "Joint scalable video model jsvm-8.6," Tech. Rep., 2007.

[12] D. Tian, P. Lai, P. Lopez, and C Gomila, "View synthesis techniques for 3d video," *Proc. of SPIE, the Int. Soc. for Optical Engineering*, vol. 7443, 2009.

[13] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Proc.*, vol. 13, no. 9, pp. 1200–1212, 2004.