Codage d'Objets de type VOP Par Représentation en Régions Auto-Extractibles

O. DEFORGES, M. BABEL, J. RONSIN

Laboratoire CNRS UMR 6164 IETR / INSA Rennes

20, av des Buttes de Coëmes, CS 14315, 35043 Rennes Cedex, France

Contact : {odeforge, mbabel, ronsin}@insa-rennes.fr, Tél : 02 23 23 82 86

Résumé – Le nouveau standard multimédia Mpeg-4 traite aussi bien des aspects compression que de fonctionnalités de haut niveau telles que la manipulation d'objets. Notre approche propose un cadre unique permettant une compression efficace tout en offrant une segmentation à coût « quasi-nul » à travers le principe de représentation en régions auto-extractibles. Aussi montrons-nous ici que notre méthode peut constituer une alternative au mode classique de codage intra de Mpeg-4. En particulier, l'approche réussi à unifier le codage de la forme et de la texture d'un VOP.

Abstract *–The new multimedia standard Mpeg-4 addresses compression aspects and high level functionalities such as object manipulation. The LAR method provides an unique framework achieving an efficient compression while allowing a nearly free reliable segmentation map of the image through the principle of self-extracting region representation. Thus, we demonstrate that our method can be a profitable alternative to the traditional intra mode coding of natural video in Mpeg-4. In particular, the approach succeeds to unify the coding of the shape and the texture of a VOP.*

1. Introduction

Mpeg-4 est le nouveau standard multimédia destiné aussi bien au codage de l'audio et de la vidéo, qu'à la synchronisation de ces flux et de leurs combinaisons [1]. L'introduction de fonctionnalités au niveau objet telles que la composition de scènes, l'interaction avec l'utilisateur, a nécessité la définition de descripteurs de forme et de codage de contenu associé. Un VOP (Virtual Object Plan) dans Mpeg-4 peut représenter toute une image ou seulement un objet dans celle-ci. La description du VOP concerne le codage de sa forme. Pour compresser son contenu (texture), les techniques classiques de transformée par bloc ont dû être adaptées aux formes quelconques.

La méthode LAR (Locally Adpative Resolution) a été introduite initialement pour le codage d'image en multi-niveaux de gris. Des extensions ont ensuite été apportées pour les images couleur et le codage par région d'intérêt. Des travaux récents ont permis d'adapter avantageusement l'approche à du codage sans perte [2], ainsi que d'améliorer les performances du codeur basse résolution. Cet article s'attache à montrer plus généralement le schéma de codage complet homogène, depuis le codage très bas débit, jusqu'à la représentation en régions pouvant être utilisée à la fois pour la compression des images de chrominance, et la description de région d'intérêt. En particulier, nous montrons ici que notre approche peut globalement constituer une alternative au mode standard de Mpeg-4, pour le codage d'image intra, mais aussi intégrer de manière plus naturelle la notion de VOP.

Récemment, les résultats obtenus en compression par H.264 ont montré leur supériorité sur un codage basé bloc classique, et même sur des approches ondelettes [3]. L'atout essentiel en codage intra a été apporté par la possibilité de disposer de tailles de blocs différentes en fonction du contenu de l'image. Notre approche suit cette philosophie mais la décline beaucoup plus en avant. Des tests comparatifs avec H.264 pourront être trouvés dans [4].

Ce papier est organisé comme suit : le principe du codeur LAR de base est rappelé section 2. La partie suivante présente l'approche dite de représentation en région auto-extractible, et son utilisation pour la compression. L'intégration de VOP est introduite section 4.

2. Codeur LAR : principes généraux

Le schéma complet du codeur apparaît figure 1. Le codeur LAR pour la compression d'image en multi-niveaux de gris est composé de deux étages : un codeur dit spatial pour les forts de taux de compression, et un codeur spectral pour coder l'image d'erreurs [4].

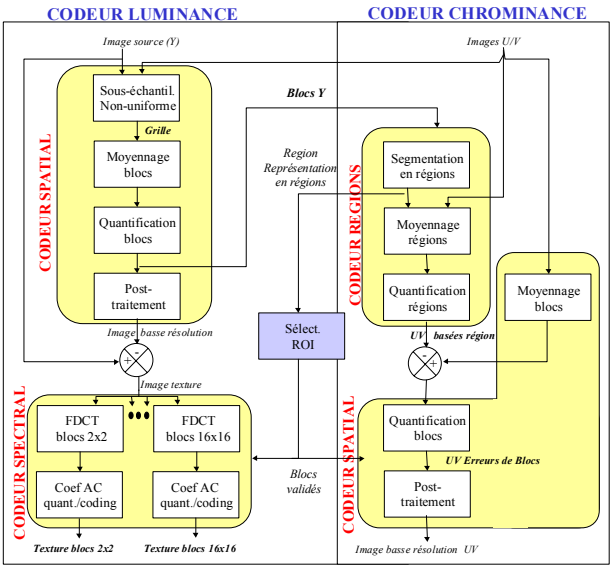

FIG. 1 : Schéma de principe général

2.1 Codeur Spatial

Le principe repose sur la notion de résolution locale (taille de pixel) variable. L'image est découpée en macro-blocs 16x16, eux-mêmes divisés sur une structure de type quad-tree (taille minimale 2x2) selon l'activité locale. En choisissant comme critère une mesure de gradient dans le bloc, une image basse résolution est obtenue en remplissant chaque bloc par sa valeur moyenne.

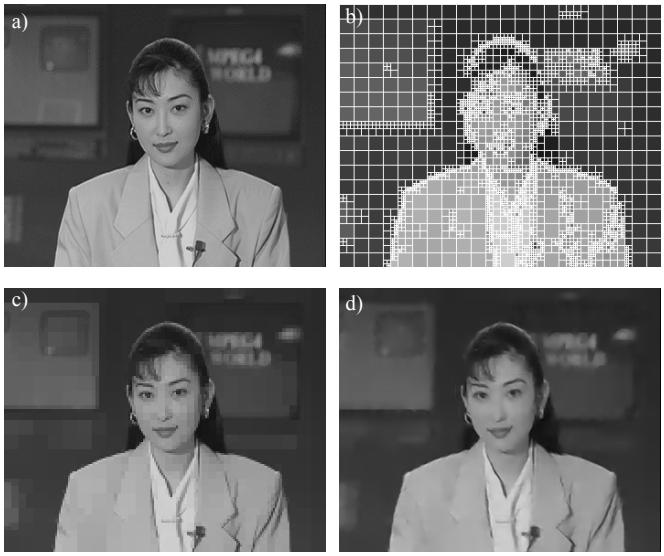

FIG. 2: codeur spatial , a) image source,
b) grille à taille variable (0.003 bpp),
c) Image des blocs (0.13 bpp), d) après postraitement.

Cette image respecte les contours et la représentation à taille de bloc variable fournit une carte de segmentation grossière de l'image. Une "quantification psychovisuelle" simple peut ainsi être mise en œuvre en appliquant une quantification forte pour les petits blocs situés sur les contours, et une plus fine pour les grands blocs localisés dans les zones uniformes. Initialement codés par un schéma prédictif de type MICD à simple balayage raster, nous avons développé une nouvelle forme de décomposition pyramidale de type quad-tree à redondance minimale et basée sur des contextes enrichis de prédiction à 360° [2]. La méthode offre ainsi un schéma de codage "scalable" basé sur un sous-échantillonnage non uniforme spatial, où l'image est transmise progressivement par raffinement local de la résolution. L'image finale de la décoposition est alors sous-échantillonnée d'un facteur 2 (bloc 2x2 pleine résolution -> bloc 1x1). Un post-traitement simple est appliqué pour lisser les zones homogènes (blocs supérieurs à 1), puis un filtre d'interpolation est utilisé pour étendre l'image à la pleine résolution.

2.2 Codeur Spectral

Le codage de l'image d'erreur correspond réellement ici à la transmission de la texture locale. Nous appliquons une approche de type DCT à taille de bloc variable, où à la fois la taille et la composante DC sont déjà fournies par le codeur spatial. La nature des blocs permet ici un codage progressif dépendant du contenu : restauration de la texture des zones uniformes à travers les grands blocs et/ou amélioration des contours à travers les blocs 2x2.

3. Représentation en Régions Auto-extractibles

La particularité majeure du codeur spatial est de ne pas engendrer de fortes distorsions et de respecter les contours. Aussi est-il possible de réaliser une segmentation directement sur cette image. La représentation en régions auto-extractibles consiste donc à transmettre l'image basse résolution, puis à effectuer au codeur et au décodeur la phase de segmentation. Si le découpage en blocs peut être vu comme une phase de pré-segmentation (« split »), la phase suivante est donc une phase de fusion effectuée sur les blocs. Notre approche offre ainsi une solution unique de représentation en régions, avec :

- un coût nul (pas d'information à transmettre pour la description des contours),
- la forme des régions et leur contenu partagent une unique grille de représentation : les blocs.

Le but n'est pas ici d'obtenir le meilleur algorithme de segmentation, mais bien de permettre des fonctionalités au niveau région pour les images bas débit.

Notre méthode de segmentation est basée sur les hypergraphes. Elle est brièvement présentée dans le paragraphe suivant.

3.1 Segmentation

Soit $S=\{ (x,y) : 1 \le x \le N_c, 1 \le y \le N_r \}$ les coordonnées spatiales d'un pixel dans une image de N_r lignes et N_c colonnes. La segmentation d'une image en K régions R consiste à trouver la partition Δ_K de S telle que :

$$S = \bigcup_{k=1}^{K} R_k \ , R_i \cap R_j = \varnothing, \forall i,j \in \{1, \ldots, K\}, \text{ pour } i \neq j,$$

Partant d'une partition initiale Δ_{K0} ($K_0 \leq N_r \times N_c$), le but du processus de fusion est de transformer Δ_{K0} en une partition Δ_K ($K < K_0$) selon un critère d'homogénéité, et à travers des séquences de fusion de régions deux à deux.

Dans notre méthode, l'image des blocs de luminance Y issue du codeur spatial constitue la partition initiale. Les régions (blocs) sont alors bien caractérisées soit par une homogénéité de luminance, soit par une position spatiale recouvrant les frontières (blocs 2x2).

La structure de données classique pour représenter des partitions est le « Region Adjacency Graph » (RAG) [5]. Le RAG d'une K-partition est défini comme un graphe non orienté, $G = (V, E)$, où $V = \{1, 2, \ldots, K\}$ est l'ensemble des nœuds et $E \subset V \times V$ est l'ensemble des arcs. Chaque région est représentée par un nœud du graphe, et entre deux nœuds (régions) $i, j \in V$ il existe un arc (i,j) si les régions sont adjacentes. Un unique coût est généralement attribué à chaque arc. Le processus de fusion optimal va alors consister à trouver les deux régions adjacentes les plus similaires, puis les fusionner. Si cet algorithme donne de bons résultats, en revanche il est très coûteux en termes de temps de calcul.

Afin de faire converger l'algorithme plus rapidemment, la fusion entre deux régions est ici simplement effectuée si leur fonction de coût est inférieure à un seuil. Le processus est réitéré jusqu'à ce qu'aucune nouvelle région ne soit créée.

Nous avons introduit une fonction de coût $COST(R_i, R_j)$ entre deux régions qui intègre un calcul de distance à la fois sur les luminances moyennes et sur le gradient. Cette fonction est également pondérée par un paramètre lié à la taille de la partition ($DIM(R_i)$), de sorte que les petites régions soient plus rapidement absorbées

$$COST''(R_i, R_j) = COST(R_i, R_j) \times \log_{10}(DIM(R_i))$$

$$\Rightarrow COST''(R_i, R_j) < COST''(R_j, R_i) \text{ si } DIM(R_i) < DIM(R_j)$$

Le RAG est alors un graphe orienté.

Le nombre total de fusions est borné à $(K_0 - K_F)$, si K_F est la partition finale. En faisant croître le seuil de fusion afin d'obtenir successivement des partitions finales décroissantes, la représentation en régions devient hiérarchique (voir fig. 3).

3.2 Codage de Région d'Intérêt (ROI)

La représentation en régions permet une sélection semi-automatique d'une ROI qui sera déterminée comme un ensemble de régions élémentaires, autrement dit décrite par uniquement les étiquettes des régions concernées. Les régions étant issues des blocs élémentaires, l'amélioration de la qualité de la ROI est immédiate, puisqu'il suffit d'activer le codeur spectral uniquement pour les blocs concernés, c'est à dire appartenant à la ROI.

3.3 Codage basé région des composantes U/V

Dans les méthodes classiques, le passage au codage des images couleur consiste essentiellement à dupliquer le schéma général pour les trois composantes. Dans notre approche, nous tirons à la fois profit de l'information couleur pour la phase de segmentation, et utilisons la représentation en régions pour coder à très bas débit l'information de chrominance.

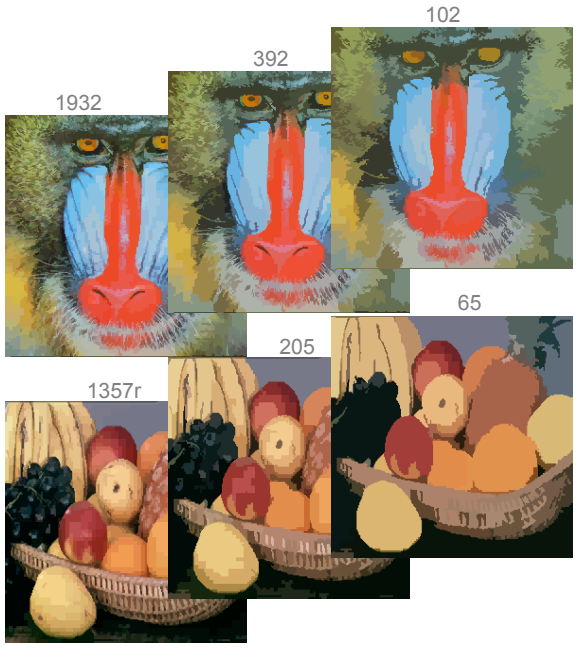

FIG. 3 : Exemples de représentation en région obtenues par contrôle de chrominance

Les images sont traitées dans un espace YUV, et les tailles de bloc sont estimées à partir des trois composantes (grille unique). Les résultats ont montré que le codeur spatial seul appliqué aux composantes UV permet d'obtenir une très bonne qualité d'image d'un point de vue couleur. Il est toutefois possible de compresser plus fortement l'information en fournissant uniquement les valeurs de chrominance moyenne de chaque région issue de la segmentation. Afin de garantir des régions cohérentes d'un point de vue chrominance, nous avons introduit le principe de "contrôle de la chrominance" dans le processus de segmentation : la segmentation reste basée sur la seule luminance pour la recherche des plus proches voisins, mais la fusion finale est validée au codeur par le calcul d'une distance chromatique entre les deux candidats. Un bit est donc transmis pour chaque tentative de fusion, mais les composantes YUV étant fortement corrélées, seuls quelques rejets seront effectifs, engendrant une entropie de l'information très faible. Le surcoût apporté par ce contrôle est typiquement compris entre 0.01 et 0.04 bpp, mais permet d'obtenir au décodeur des régions globalement cohérentes d'un point de vue couleur.

Le réhaussement des composantes UV consiste éventuellement à utiliser un codeur spatial pour la compression de l'erreur de chrominance par bloc, de manière globale sur toute l'image ou localement dans une ROI. Le codage basé région des images UV et la sélection d'une ROI peuvent s'appuyer sur des niveaux hiérarchiques différents de la représentation en régions. Des résultats de segmentation sont donnés figure 3. Des tests comparatifs uniquement en termes de compression entre Jpeg2000, Mpeg-4 mode intra (issue de MoMuSys) et LAR , apparaissent figure 4.

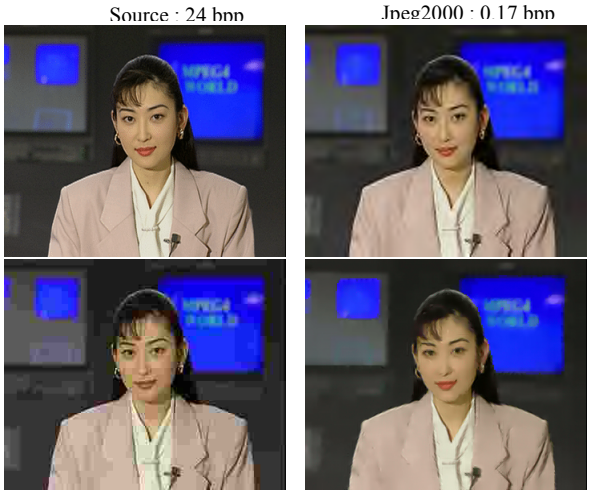

FIG. 4: Tests comparatifs (zoomer sur l'image à 200% pour visualiser en pleine résolution)

4. Codage de VOP

Pour un VOP de forme quelconque, il est nécessaire de transmettre en général ses contours, puis de coder sa texture interne. Pour les blocs contours, il faut recourir à des transformées spécifiques de type SA-DCT.

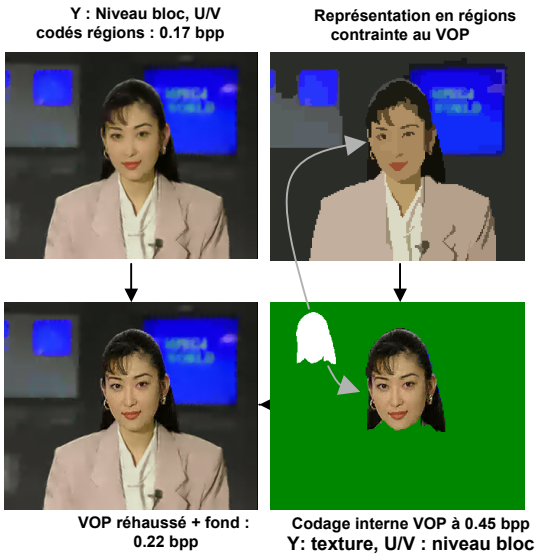

FIG. 5 : Principe de reconstruction de la forme du VOP et réhaussement de sa texture

Considérant une image intra (I) codée en LAR, nous proposons plusieurs alternatives pour la description et le codage de la texture du VOP.

- *Transmission préalable de l'image basse résolution, VOP résultant de la segmentation*

Le VOP est décrit uniquement pour des étiquettes de régions contenues dans celui-ci (= ROI). La représentation hiérarchique en régions permet localement un choix fin de résolution. Le choix du VOP peut être effectué aussi bien au codeur qu'au décodeur. Ce mode est le moins coûteux d'un point de vue description du VOP.

Les deux modes suivants considèrent un choix de VOP indépendant de notre segmentation. Afin que la représentation en blocs soit adaptée au VOP, le masque binaire du VOP est introduit dans l'estimation des tailles du codeur spatial.

- *Transmission préalable de la forme du VOP*

Schéma classique engendrant un coût de description. L'utilisation du codeur LAR permet néanmoins une dissociation spatiale des blocs entièrement extérieurs ou intérieurs au VOP.

- *Transmission préalable de l'image basse résolution, VOP issu d'une phase de segmentation quelconque*

Notre processus de segmentation en régions peut alors être effectué en incorporant un mécanisme de contrôle par la forme du VOP, identique à celui de la chrominance. Cela garantit ainsi des régions entièrement externes ou internes au VOP. La figure 5 illustre un tel schéma de codage, avec Y compressée par le codeur spatial (0.127 bpp), U/V codés régions (119 régions, contrôle chromatique : 0.034 bpp + valeur par région : 0.0067 bpp). Le coût additionnel du contrôle du VOP est de 302 bits (0.003 bpp), alors que le coup de codage de cette même forme est de 452 bits (source MoMuSys). Le VOP global est ici reconstruit par 12 régions élémentaires, et réhaussé par correction de la luminance et chrominance.

5. Conclusion et perspectives

Nous montrons ici un schéma de codage général alliant tout à la fois progressivité, efficacité dans la compression, et fonctionnalités au niveau région disponibles de manière automatique et à très faible coût. L'approche à plusieurs couches de codage conserve la même structure d'échantillonnage, permettant ainsi une totale concordance entre forme des objets et codage de leur texture. Nous montrons également que le principe de VOP introduit dans Mpeg-4 peut s'intégrer de manière plus naturelle dans notre schéma.

Références

[1] Signal Processing : Image communication, *"Special Mpeg-4"*, Published by Elsevier Science B.V., January 2000.

[2] M. Babel, O. Déforges, J. Ronsin, *"Décomposition Pyramidale à Redondance Minimale pour Compression d'Image sans Perte"*, Gretsi 2003, Paris, septembre 2003.

[3] JVT of ISO/IEC MPEG & ITU-T VCEG, Document: JVT-D039, *" Performance comparison: H.26L intra coding vs. JPEG2000"*, July 2002, pp 1-9.

[4] M. Babel, O. Déforges, J. Ronsin, *"Adaptive Multiresolution Scheme for Efficient Image Compression"*, PCS'2003, Saint-Malo, av. 2003.

[5] K. Haris and Co, *"Hybrid Image Segmentation Using Watersheds and Fast Region Merging"*, IEEE Trans. on Image Processing, vol 7, no 12, pp 1684-1698, Dec. 1998.