Couleur et ROI: Deux Options de JPEG2000. Investigations et Simulateur MATLAB

Franck Luthon, Philippe Arnould, C. Baillot, X. Navarro, J.M. Coutellier

Couleur et R.O.I. : deux options de JPEG2000. Investigations et simulateur Matlab

F. Luthon[1,2] P. Arnould[1] C. Baillot[1] X. Navarro[1] J.M. Coutellier[3]

[1] LIUPPA, EA3000 (Laboratoire d'Informatique de l'Université de Pau & Pays de l'Adour)

[2] IUT Informatique, Château Neuf, Place Paul Bert, 64100 Bayonne, FRANCE

[3] MAGSYS S.A., Technopole Izarbel Estia, 64210 Bidart, FRANCE

Franck.Luthon@univ-pau.fr

Résumé

Pour une application de télésurveillance reposant sur la transmission à très bas débit d'images acquises par une caméra fixe, nous présentons deux innovations concernant deux fonctionnalités de JPEG2000 : le choix d'une transformée couleur non-linéaire pour améliorer le rendu couleur à très fort taux de compression, et l'extraction automatique de la ROI par détection de mouvement.

Mots clefs

compression, transformée couleur, région d'intérêt, détection de mouvement, JasPer, Matlab.

1 Introduction

JPEG2000, plus qu'un simple standard de compression d'images fixes destiné à remplacer JPEG, est un système flexible de représentation d'images [1]. Ce nouveau standard utilise une transformée en ondelettes discrète (DWT) au lieu de la DCT, et un codage arithmétique par plans de bits au lieu du codage de Huffman [2].

Les améliorations induites sont [3] : performance supérieure en terme de compromis compression/qualité (typ. $1:60$ au lieu de $1:30$ à qualité identique) ; modes de compression avec ou sans perte ; *scalabilité* en résolution, en qualité et spatiale ; transmission et reconstruction progressives d'image ; robustesse aux erreurs pour applications mobiles à très bas débit ; gestion de régions d'intérêt (ROI) ; architecture ouverte (choix de la transformée couleur multi-composantes MCT, choix de la DWT) ; format de fichier flexible.

Pour une application de vidéo-surveillance routière utilisant un module vidéo portable avec carte d'acquisition à 8 img/s et transmission par GSM (9600 bauds), un très fort taux de compression est nécessaire pour atteindre un débit de transmission souhaité d'environ 1 img/s. Les modules actuels utilisant la norme JPEG permettent un taux de compression d'environ $1:30$, ce qui conduit à une cadence de transmission insuffisante (de l'ordre de 8s/img). MPEG n'étant pas utilisable vu la cadence vidéo trop faible pour permettre une estimation de mouvement satisfaisante au décodeur (reconstruction impossible des images codées B et P), notre choix s'est porté vers JPEG2000, avec un double objectif :

– obtenir une qualité acceptable (rendu couleur) avec un très fort taux de compression de $1:90$,
– gagner encore un facteur 2 sur le taux de compression par une gestion intelligente de la ROI.

Pour cela, nous proposons d'utiliser une transformée couleur non-linéaire originale, et de gérer la ROI par détection de mouvement, puisque l'application se place dans le cas d'une caméra fixe (Fig. 1).

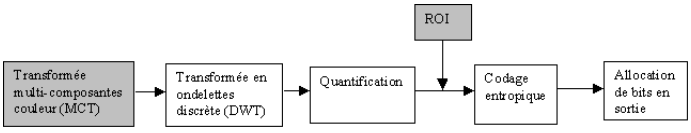

Figure 1 – *Schéma de principe d'un codeur JPEG2000 (en grisé : notre contribution).*

2 Transformée couleur non-linéaire

JPEG2000 permet de traiter des images multi-composantes (MCT). A partir des 3 composantes RGB, la MCT effectue un changement de l'espace des couleurs, pour satisfaire deux critères : une décorrélation des composantes afin d'obtenir une compression efficace, une adéquation au système visuel humain afin de minimiser les pertes d'information liées à la quantification. Deux transformations linéaires (YUV et YCrCb) sont implantées en standard dans le paquetage *JasPer* [4]. A titre d'exemple, l'espace YCrCb classique se définit par la relation matricelle :

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.41869 & -0.08131 \\ -0.16875 & -0.33126 & 0.5 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Nous présentons ici une transformée non-linéaire LUX, très performante en segmentation couleur [5], que nous

adaptons au cas de la compression (dynamique bornée et inversibilité). Cette tranformée couleur a pour origine l'idée d'introduire une non-linéarité de type logarithmique (en adéquation avec le système visuel humain) permettant une description efficace des teintes et s'affranchissant des problèmes de bruit et de variation d'éclairage : elle permet en effet d'amplifier le contraste sur les composantes chrominances, et donc d'obtenir un niveau de détail plus élevé. Pour une image RGB codée sur 24 bits, elle est définie par les formules suivantes :

$$L = (R+1)^{t_{11}}(G+1)^{t_{12}}(B+1)^{t_{13}} \quad (2)$$

$$U = \alpha(R+1)^{t_{21}}(G+1)^{t_{22}}(B+1)^{t_{23}} \quad (3)$$

$$X = \beta(R+1)^{t_{31}}(G+1)^{t_{32}}(B+1)^{t_{33}} \quad (4)$$

où la dynamique est réglée ici par $\alpha = \beta = 128$ et où t_{ij} sont les coefficients de la matrice T inspirée de $YCrCb$:

$$T = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.5 & -0.4 & -0.1 \\ -0.2 & -0.3 & 0.5 \end{bmatrix} \quad (5)$$

Son inverse est donnée par :

$$R = L^{a_{11}}\left(\frac{U}{\alpha}\right)^{a_{12}}\left(\frac{X}{\beta}\right)^{a_{13}} - 1 \quad (6)$$

$$G = L^{a_{21}}\left(\frac{U}{\alpha}\right)^{a_{22}}\left(\frac{X}{\beta}\right)^{a_{23}} - 1 \quad (7)$$

$$B = L^{a_{31}}\left(\frac{U}{\alpha}\right)^{a_{32}}\left(\frac{X}{\beta}\right)^{a_{33}} - 1 \quad (8)$$

où a_{ij} sont les coefficients de la matrice A :

$$A = \begin{bmatrix} 1 & 1.4348 & 0.0870 \\ 1 & -0.7391 & -0.3478 \\ 1 & 0.1304 & 1.8261 \end{bmatrix} \quad (9)$$

La Fig. 2 montre, sur une image typique de surveillance autoroutière, l'amélioration potentielle du rendu couleur à très fort taux de compression ($1:90$). L'image originale au format BMP est présentée sur la Fig. 2f. Si JPEG fournit une qualité acceptable à un taux $1:30$ (Fig. 2a), la dégradation est insupportable pour $1:90$ (Fig. 2b). JPEG2000 en revanche donne un résultat acceptable avec la transformée YUV (Fig. 2c et d correspondant aux 2 filtres d'ondelettes de Daubechies implantés par défaut dans le standard). Néanmoins, on remarque que le feu arrière gauche du camion est devenu jaune suite à la forte compression. La transformée non-linéaire LUX proposée ici (Fig. 2e) permet de mieux restituer les couleurs d'origine, notamment la couleur orange du feu arrière gauche du camion (comparer les Fig. 2c et e).

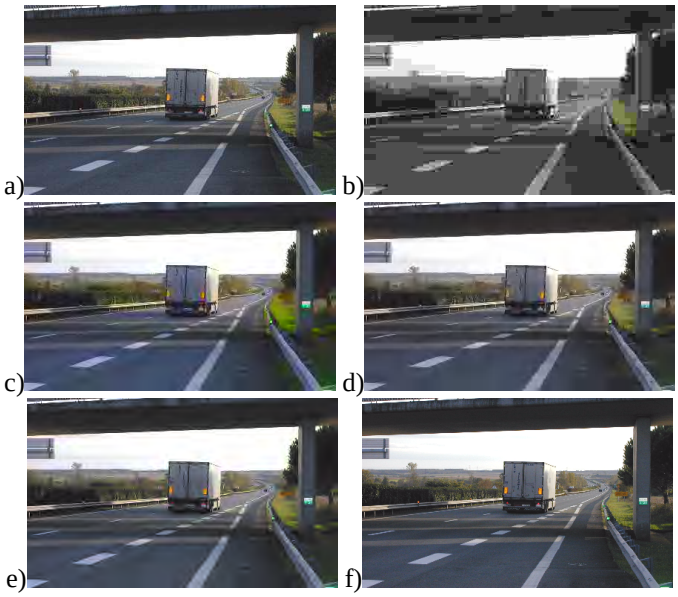

Figure 2 – *Variantes de compression : a) JPEG (1 :30) ; b) JPEG (1 :90) ; c) JPEG2000 (1 :90 avec YUV & WT9/7) ; d) JPEG2000 (1 :90 avec YUV & WT5/3) ; e) JPEG2000 (1 :90 avec LUX & WT9/7) ; f) Image BMP originale sans compression (*i.e., *taux 1 :1).*

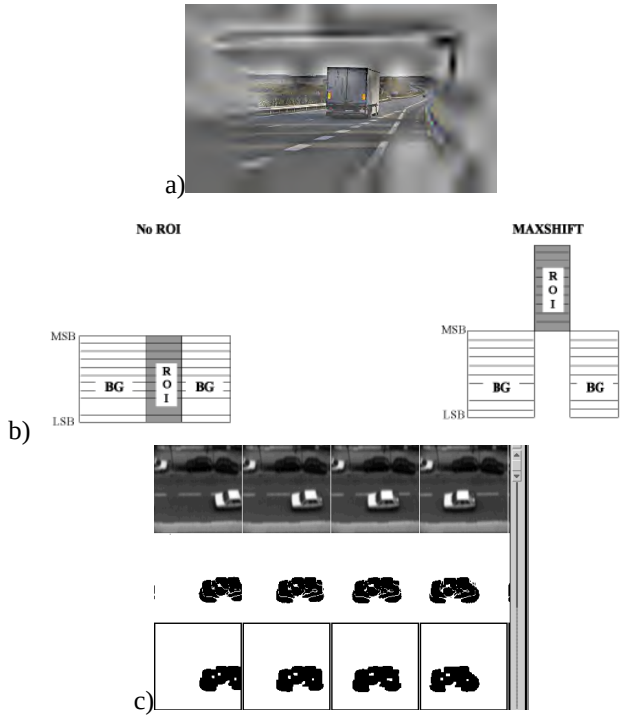

Figure 3 – *a) Illustration de compression avec une ROI définie manuellement (quart de l'image) ; b) Technique du décalage de bits MAXSHIFT ; c) Exemple de détection de mouvement sur une séquence Auto.*

3 ROI par détection de mouvement

Une autre fonctionnalité inniovante de JPEG2000 est la gestion de ROI [6], autorisant des taux de compression différents selon les parties de l'image, les régions

d'intérêt définies par l'utilisateur étant encodées avec plus de précision que les autres parties de l'image. La Fig. 3a illustre cette fonctionnalité.
La gestion de ROI se fait par la technique du Maxshift, implantée entre la quantification et le codage entropique. Elle consiste à appliquer un décalage binaire vers la droite aux coefficients associés au fond de l'image (*background* BG), les coefficients de ROI étant codés sur les bits de poids fort (Fig. 3b).
L'intérêt de cette technique est de ne pas avoir à transmettre au décodeur d'information spatiale explicite (coordonnées p.ex.) sur la ROI. Lors de la phase de codage entropique, les différents coefficients sont codés avec les plans de bits de poids fort en premier.
Cette méthode permet de visualiser en premier la ROI dans le cadre d'un transfert progressif avec reconstruction progressive de l'image. Mais elle peut aussi être utilisée pour augmenter le taux de compression sur des images dont seules certaines régions sont jugées pertinentes par l'utilisateur. A cette fin, nous proposons ici de gérer une ROI extraite automatiquement par détection de mouvement. Dans le cas d'une caméra fixe, une détection de mouvement peu coûteuse en calculs (différence temporelle de luminance, seuillage, érosion, dilatation) peut être mise en œuvre.
La détection de mouvement consiste à étiqueter chaque pixel s de l'image à l'instant t pour obtenir une carte binaire des changements temporels :

$$e_s = e(x,y,t) = \begin{cases} \text{“1”} & \text{si le pixel} \in \text{une zone mobile,} \\ \text{“0”} & \text{si le pixel} \in \text{fond fixe.} \end{cases} \tag{10}$$

Pour cela, on utilise comme observation la valeur absolue de la différence temporelle d'intensité lumineuse entre deux instants d'acquisition :

$$o_s = |I_t(s) - I_{t-1}(s)| \tag{11}$$

Evidemment, cette information est bruitée (bruit d'acquisition de la caméra, bruit de quantification) et nécessite un seuillage adéquat (seuil θ). A l'issue du seuillage, tous les pixels étiquetés mobiles ($o_s > \theta$) sont des candidats à la ROI. Un post-traitement de type morphologie mathématique sur l'image binaire (érosion-dilatation par un élément structurant, ouverture-fermeture) permet de combler les trous et d'éliminer les points isolés pour obtenir des domaines connexes dont l'union formera la ROI.
La ROI ainsi obtenue n'est pas une fenêtre de forme simple appliquée sur l'image, mais un ensemble de régions de forme quelconque correspondant aux masques des différents objets en mouvement dans la séquence (Fig. 3c). Les régions mobiles de l'image sont alors codées prioritairement et un gain supplémentaire de 2 sur le taux de compression est *a priori* envisageable.
Cette approche ouvre comme perspective la possibilité de proposer une norme Motion-JPEG2000 applicable au cas restreint de séquences acquises avec une caméra fixe. Elle permet de gérer une information de mouvement codée en intra dans JPEG2000 (via la ROI), contrairement à MPEG (images B et P). Elle s'applique donc même si la cadence vidéo restituée au décodeur est très faible, puisque le décodage ne nécessite pas de recourir à l'image précédente.

4 Simulateur Matlab du codec

JPEG2000 étant un standard ouvert et flexible, l'utilisateur peut être amené à programmer et tester des fonctionnalités spécifiques pour développer ses applications de compression. Il est alors intéressant de diposer d'une plate-forme de tests conviviale. Matlab constitue, grâce à l'environnement qu'il propose, une des plates-formes les plus complètes et les plus utilisées dans le domaine de l'ingénierie mathématique appliquée au traitement du signal. Il propose en particulier des fonctions de base dans le domaine du traitement des images telles que la décomposition en ondelettes ou la création des blocs. Il offre aussi une grande souplesse d'utilisation dans la manipulation des fichiers d'images normalisés. Enfin, il permet l'intégration de primitives écrites en langage C et l'interfaçage de circuits spécialisés comme les DSP. Il est donc pertinent de créer sous cet environnement de développement un codec d'images respectant la norme JPEG2000.
Le simulateur que nous développons reprend la structure normalisée du format JPEG2000 : il met à disposition du concepteur un ensemble de fonctions respectant la norme. Ces fonctions peuvent être utilisées sans modification ou faire l'objet d'améliorations dans le cadre d'un développement d'application spécifique. La Fig. 4 représente les différents paramètres d'entrée/sortie de chacune des fonctions développées.

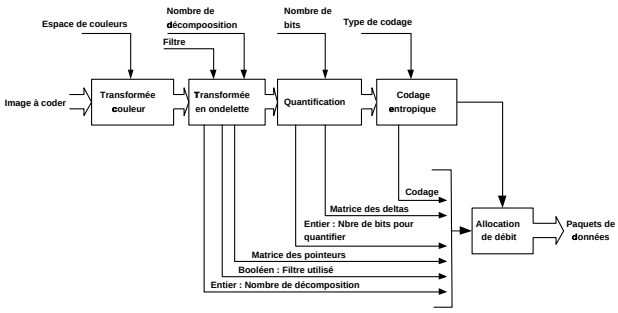

Figure 4 – *Simulateur Matlab du codeur JPEG2000.*

Les deux transformées (MCT et DWT) sont écrites en langage Matlab (M-File). Les autres fonctions (quantification, codage et allocation) sont écrites en langage C interfacé avec Matlab (Mex-File). Nous détaillons ci-dessous notre implantation.

– **Image à coder** : l'image à coder est entrée dans Matlab à l'aide de la fonction intégrée «imread» qui permet de lire différents types de fichiers images (bmp, tiff …). Bien sûr, il est aussi possible de lire un fichier brut («raw»).

– **Transformée couleur** : cette fonction est implantée sous la forme d'une fonction écrite dans le langage des fichiers de commandes Matlab (M-File). Comme cette fonction est écrite en langage Matlab, il est très facile de changer d'espace couleur afin d'améliorer les performances du codage. *Entrée* : nom du fichier et transformée couleur. *Sortie* : l'image codée suivant l'espace des couleurs choisi.
– **Transformée en ondelettes** : ce bloc est implanté sous la forme d'une fonction qui utilise la transformée en ondelettes de la toolbox Wavelet de Matlab («wavedec2»). Nous n'avons pas intégré les coefficients du filtre de Le-Gall. *Entrée* : les vecteurs de sortie du bloc de la transformée couleur, le type d'ondelette. *Sortie* : les coefficients de la décomposition en ondelettes
– **Quantification** : nous avons implanté en C des fonctions de quantification sur 8, 16, 32 ou 64 bits. Elles utilisent un pas de quantification $\Delta = \frac{max}{2^b}$ où b est le nombre de bits utilisé pour quantifier et max est la valeur maximum dans la sous-bande à quantifier. Elles quantifient séparément chaque sous-bande obtenue grâce à la transformée en ondelette, avec un Δ spécifique à chacune. *Entrée* : vecteur à quantifier. *Sorties* : vecteur quantifié et Δ.
– **Codage entropique** : actuellement, nous n'avons pas encore implanté le codage arithmétique EBCOT de la norme Jpeg2000 ; nous avons écrit une fonction de codage entropique en utilisant le codage classique de Huffman qui est celui de la norme JPEG. *Entrée :* vecteur à compacter. *Sortie* : vecteur compacté.
  Une solution rapide pour finir d'intégrer la norme Jpeg2000 dans notre simulateur consiste à faire des appels depuis Matlab aux fonctions en C de la bibliothèque JasPer, ceci pour les blocs qui ne sont pas encore implantés. La principale difficulté vient de la complexité du code source JasPer (20k lignes de code) et de la limitation de sa licence d'utilisation.

Voici un exemple de fichier de commandes Matlab qui permet de réaliser la compression/décompression au format Jpeg2000.

```
% CODAGE JPEG2000:
fichier='image.bmp'; %choix Nom Fichier a compresser
coul=0;              %choix Transfo Couleur
[luminance,bleu,rouge]=transcouleur(fichier,coul);
%->Sortie Bloc MCT

filtre='Db1';        %choix Filtre Ondelettes
n=7;                 %choix Nb Niveaux resolution
[data1,ptr1,data2,ptr2,data3,ptr3]=
ondelette(luminance,bleu,rouge,n,filtre);
%->Sortie Bloc DWT (coeff. ondelettes)

bit=8;               %choix Nb Bits de quantification
[datQ1,delta1]=quantification(ptr1,data1,n,bit);
[datQ2,delta2]=quantification(ptr2,data2,n,bit);
[datQ3,delta3]=quantification(ptr3,data3,n,bit);
%->Sortie Bloc Quantification

[datCod1]=codage_entropique(ptr1,datQ1,n,bit);
[datCod2]=codage_entropique(ptr2,datQ2,n,bit);
[datCod3]=codage_entropique(ptr3,datQ3,n,bit);
%->Sortie Bloc Codage entropique

% DECODAGE JPEG2000:
[datDec1]=decodage_entropique(datCod1,n,bit);
[datDec2]=decodage_entropique(datCod2,n,bit);
[datDec3]=decodage_entropique(datCod3,n,bit);
%->Sortie Bloc Décodage entropique

[datDQ1]=dequantification(ptr1,datDec1,delta1,n,bit);
[datDQ2]=dequantification(ptr2,datDec2,delta2,n,bit);
[datDQ3]=dequantification(ptr3,datDec3,delta3,n,bit);
%->Sortie Bloc Dé-Quantification

[L,B,R]=
recondelette(datDQ1,ptr1,datDQ2,ptr2,datDQ3,ptr3,filtre);
%->Sortie Bloc Reconstruction Ondelette

[sortie]=transcouleurinv(L,B,R,coul);
%->Sortie Bloc Transfo.Couleur Invers:image décompressée
```

5 Discussion

Malgré l'implantation encore partielle du codec JPEG2000 sous Matlab, on dispose d'ores et déjà de ce simulateur pour tester la norme avec divers choix de transformées couleur MCT, de DWT, ainsi que de quantification. La boîte à outils développée est accessible sur le site :

`http://www.univ-pau.fr/~arnould`

Il reste également à implanter dans le simulateur la gestion de ROI associée à la détection de mouvement.
Si la transformée LUX (que nous avons adaptée au cas de la compresssion JPEG2000) semble performante sur les images-tests de cette application, il reste à évaluer et valider cette approche sur une plus large base d'images. De plus, une étude du couplage entre le choix de ce type de transformée couleur non-linéaire et le choix de la quantification est en cours.

Références

[1] D.S. Taubman et M.W. Marcellin. *JPEG 2000 Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, The Netherlands, 2001.

[2] M. Rabbani et R. Joshi. An overview of the JPEG 2000 still image compression standard. *Signal Processing : Image Communication*, 17 :3–48, 2002. free download : http ://www.jpeg.org/JPEG2000.htm.

[3] A.N. Skodras, C.A. Christopoulos, et T. Ebrahimi. JPEG2000 : The upcoming still image compression standard. Dans *Proc. 11th Portuguese Conf. on Pattern Recognition*, pages 359–366, Porto, Portugal, May 11th-12th 2000.

[4] M.D. Adams et F. Kossentini. JasPer : A software-based JPEG-2000 codec implementation. Dans *Proc. ICIP*, Vancouver, Canada, Septembre 2000.

[5] M. Lievin. *Analyse entropico-logarithmique de séquences vidéo couleur. Application à la segmentation markovienne et au suivi de visages parlants*. Thèse de doctorat, National Polytechnic Institute, Grenoble, France, Septembre 2000.

[6] D. Nister et C. Christopoulos. Lossless region of interest coding. *Signal Processing*, 78(1) :1–17, 1999.