

Codec LAR couleur : Schéma de représentation et de compression d'images couleur fondé sur l'adaptation de la résolution locale et sur la représentation en régions auto-extractibles

Color LAR codec :
a color image representation and compression scheme
based on local resolution adjustment and on self-extracting
region representation

**Olivier Déforges, Marie Babel, Laurent Bédard,
Joseph Ronsin**

Institut d'Électronique et des Télécommunications de Rennes, UMR CNRS 6164, Groupe Image et Télédétection
INSA, 20 av. des Buttes de Coësmes, CS 14315, 35043 Rennes cedex
{odeforge, mbabel, lbedat, ronsin}@insa-rennes.fr

Manuscrit reçu le 15 juin 2004

Résumé et mots clés

Cet article présente un schéma original de codage progressif d'images couleur apportant à la fois une efficacité en termes de compression (meilleure qualité subjective que Jpeg2000) et des fonctionnalités au niveau région à bas débits pour le codeur et le décodeur. À partir de l'image des luminances codée à bas débit par le codec LAR (Locally Adaptive Resolution), une description en régions, sans codage des contours, est obtenue à travers un procédé de segmentation effectué au codeur et au décodeur. Cette segmentation peut être contrôlée par les composantes chromatiques pour une meilleure cohérence du résultat d'un point de vue couleur. Un codage basé régions appliqué sur les images de chrominance produit alors une compression de ces composantes à très bas débit. Comme les régions et le codage de leur contenu partagent une même grille de représentation, l'amélioration de la qualité de l'image peut être globale, ou limitée à une zone d'intérêt.

Codage progressif, segmentation d'images en niveaux de gris et couleur, codage par représentation en régions, codage par Région d'Intérêt.

Abstract and key words

This LAR (Locally Adaptive Resolution) color image coding scheme yields to an efficient progressive compression with a better subjective quality than Jpeg2000. Additionally, it offers region functionalities for low bit rate coding and decoding. From highly compressed luminance, a region description, without contours encoding, can be obtained through a segmentation process performed at both coder and decoder. Considering color results, controlled chrominance components segmentation provides a better data consistency simultaneously with a low bit rate compression. As regions and their encoding are based on a same representation grid, enhancement of image quality can be global, or only restricted to a Region Of Interest.

Scalable coding, gray-level and color images segmentation, regions representation based coding, region of Interest Coding.

1. Introduction

La première génération de codeurs d'images et de vidéos s'appuie essentiellement sur la théorie de l'information. La principale limitation de ses performances en termes de compression est due au caractère non-stationnaire des images naturelles, aboutissant à une distribution variable de l'information dans l'espace et dans le temps. De plus, les structures de représentation utilisées, indépendantes du contenu, comme la grille cartésienne conventionnelle ou les pavages en blocs carrés réguliers (JPEG ou MPEG), ne permettent pas de prendre en compte ces non-stationnarités, et ne peuvent par conséquent servir comme structure de données efficace pour les images et la vidéo. En introduisant une modularité dans la partition de l'image, le nouveau standard H.264 ou MPEG4-AVC démontre que les performances de codage peuvent alors être fortement accrues [1]. Des améliorations ont pu être apportées par rapport aux schémas classiques, grâce à la représentation des données visuelles sous forme de régions, définies par leurs contours et leurs textures, et correspondant potentiellement à des objets ou parties d'objets dans l'image. Les méthodes de codage s'appuyant sur ce concept sont couramment appelées de seconde génération [2, 3].

Le principe d'une représentation en objets fait partie intégrante de MPEG-4 [4] pour la compression et la manipulation de flux vidéo, et est un pré-requis nécessaire dans MPEG-7 pour l'indexation [5]. Les régions sont définies comme des parties convexes d'une image partageant une propriété commune. Les objets sont quant à eux définis comme des ensembles de régions qui représentent une entité sémantiquement cohérente d'une image [6]. Pour des représentations en régions, deux types d'information sont nécessaires : **la forme** (les contours) et **le contenu** (la texture). Un troisième élément peut être ajouté dans le cas de séquences vidéo : **le mouvement**.

Une approche région tend par ailleurs à combler le fossé séparant les systèmes numériques et le système visuel humain (SVH) pour le traitement d'une image et sa perception. Une telle approche fournit également de nouvelles fonctionnalités telles que l'interaction entre objets ou régions, la composition de scènes, etc. Un autre atout important réside dans la possibilité pour un schéma de compression de favoriser des zones visuellement sensibles de l'image tout en négligeant les parties moins significatives. Ce concept porte le nom de compression par Région D'Intérêt ou ROI (Region Of Interest). Une telle approche s'adapte bien au développement de nouvelles applications à travers les réseaux (consultation de grandes bases de données d'images, vidéosurveillance et vidéoconférence) [7]. Les bandes passantes limitées des canaux actuels comparées au volume de données nécessaire pour la représentation de ces images amènent à établir un compromis entre le débit et la distorsion (qualité) de l'image. Lorsqu'une ROI est identifiée et définie, ce compromis peut être réalisé localement de telle sorte que la compression à l'intérieur de la ROI n'introduise que peu de distorsion pour une bonne qualité visuelle, alors que le reste de l'image peut être représenté avec une moindre qualité.

Malgré tous les attraits que présentent les approches régions, il faut reconnaître que les standards actuels sont toujours et sans doute encore pour longtemps basés sur des techniques traditionnelles de transformation de l'information. Les raisons se déclinent en quatre points.

1. La description de la forme (le plus souvent un polygone) constitue un surcoût qui peut rapidement devenir significatif à très bas débit. Pour limiter ce coût, il est donc aussi nécessaire de limiter le nombre de régions, et par conséquent de se contenter d'une description rudimentaire de la scène.
2. Les méthodes basées régions s'attachent en général essentiellement à la composante « forme », et négligent le plus souvent la composante « contenu ». Dès lors, cette forme est indépendante de la représentation utilisée pour coder le contenu.
3. Une représentation en régions induit forcément une phase de segmentation de l'image. Or cette étape constitue en général un verrou important pour la réalisation de systèmes de traitement temps-réel.
4. Le schéma de codage classique impose que seul le codeur soit à même de définir la représentation en régions, le décodeur ne pouvant avoir un rôle de décideur. Cela exclut alors des applications du type consultation de base de données d'images avec définition, par l'utilisateur, d'une région d'intérêt au décodeur.

Sur un autre plan, la littérature ne fait que très peu cas du codage des images couleurs. D'abord parce que les composantes de chrominance sont plus faciles à coder que celle de luminance. Ensuite parce que le schéma de codage développé pour la luminance est généralement simplement dupliqué avec quelques adaptations mineures aux deux composantes de chrominance. Les techniques de codage sont ainsi présentées et appréciées sur des images en niveaux de gris. Or, en pratique, les images à compresser sont essentiellement en couleur : la qualité d'une méthode de codage devrait par conséquent s'apprécier uniquement sur ce type d'images.

L'objectif des travaux présentés est la recherche d'une nouvelle orientation dans les méthodes de codage, qui puisse notamment relier les approches dites classiques aux approches régions. Il s'agit également d'aborder le problème de représentation en régions dans sa globalité, c'est à dire en tentant d'unifier les notions de forme et de contenu. Notre approche repose en grande partie sur l'exploitation de l'information couleur pour l'obtention d'une représentation en régions de bonne qualité, tout en exploitant cette même représentation en régions pour un codage très bas débit des images de chrominance. Une grande originalité de ce schéma de codage est de ne pas recourir à l'envoi d'une carte de segmentation pour la représentation en régions, mais de la déduire à partir d'une image de luminance codée à bas débit. Ce procédé a été appelé « **Représentation en Régions Auto-Extractibles** ».

L'article est organisé en quatre parties. La section suivante présente les principes du codeur LAR (Locally Adaptive Resolution) pour le codage des images en niveaux de gris et en

couleur. Il se fonde sur un partitionnement de l'image à taille de blocs variable. Le principe de la représentation en régions auto-extractibles, appliqué uniquement sur les images en niveaux de gris, est ensuite détaillé dans la section 3. L'intégration de l'information couleur, ainsi que le codage en régions des composantes de chrominances sont présentés section 4. L'article s'achève sur des conclusions et des perspectives.

2. Présentation du CODEC LAR simple

À la base de la méthode LAR se trouve l'idée suivante : la résolution locale peut s'adapter en fonction de l'activité dans l'image. Ainsi, lorsque la luminance s'avère localement uniforme, la résolution peut être abaissée. À l'inverse, si l'activité reste localement élevée, alors la résolution doit être plus importante. D'autre part, une image I peut être considérée comme la superposition de deux composantes

$$I = \bar{I} + \underbrace{(I - \bar{I})}_E \quad (1)$$

où \bar{I} représente une information globale de l'image (type moyenne locale par exemple) estimée sur un support donné, et E la variation locale autour de celle-ci (texture locale). La dynamique de E est ainsi dépendante de deux facteurs essentiels :

1. l'activité locale dans l'image,
2. la dimension du support de \bar{I} .

Si l'on admet de plus qu'une image peut grossièrement être considérée comme formée de zones relativement homogènes et de contours, alors E aura une faible dynamique dans les zones uniformes par la seule adaptation du support. En revanche, E sera de forte dynamique sur les contours dès lors que le support de \bar{I} sera supérieur à un pixel.

Les bases de la méthode LAR reposent sur un codeur à deux couches : couche spatiale pour le codage de \bar{I} , et spectrale pour

l'image d'erreurs E (texture). Par construction, le codec offre donc au moins deux niveaux de progressivité. La figure 1 montre le schéma de principe global.

Dans les sections suivantes, nous décrivons plus en avant le contenu des différentes couches de codage. L'espace de couleur choisi reste celui le plus classique en codage avec pertes, à savoir Y:Cr:Cb. Ce choix d'espace de représentation pour le codage des images couleur est motivé par la décorrélation de l'information contenue dans les composantes Y:Cr:Cb résultantes, par la répartition uniforme de l'entropie sur les composantes couleurs [8], par la simplicité de la transformation, et enfin par la simplicité d'utilisation de cet espace de représentation (transformation linéaire, espace à valeurs entières).

2.1. Le codeur spatial

Le terme « spatial » indique que le procédé de représentation et de compression est réalisé directement dans le domaine spatial de l'image. En s'attachant à représenter et à compresser uniquement l'information globale dans l'image, ce codeur vise clairement les forts taux de compression. Son rôle est d'une part de distinguer les contours du reste de l'image, d'autre part d'adapter le support de \bar{I} de telle sorte que l'image reconstruite soit visuellement acceptable avec une erreur E réduite dans les zones uniformes. La forme des supports considérés est ici celle de blocs carrés.

Le schéma bloc du codeur spatial est donné par la figure 2. Il s'appuie sur une représentation de l'image avec un partitionnement à taille de blocs variable, où chaque bloc prend pour valeur sa luminance moyenne. Si cette technique est *a priori* connue, nous montrons dans la suite les éléments distinctifs importants dans notre approche. L'ensemble de symboles générés (partitionnement, erreurs de prédiction, image d'erreurs...) par ces différentes phases de représentation et de codage de l'information est compressé par un codage arithmétique adaptatif propre.

Les sections suivantes vont expliciter les différentes techniques mises en œuvre dans chacune de ces étapes.

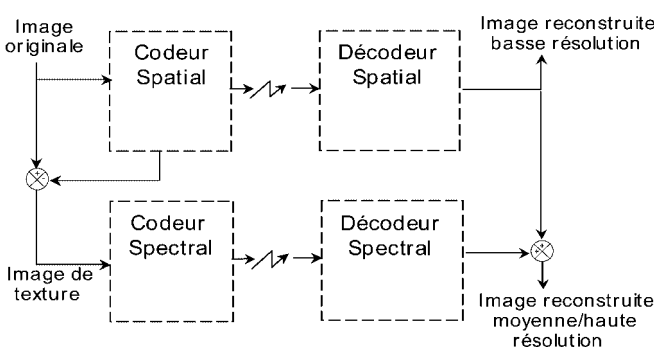


Figure 1. Schéma global LAR à deux couches : codeurs spatial + spectral.

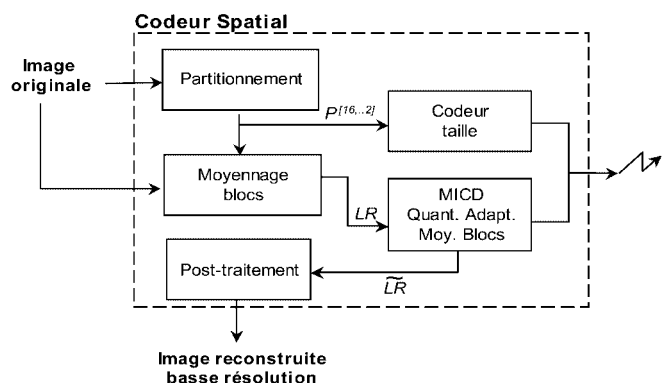


Figure 2. Schéma de principe du codeur spatial.

2.1.1. Partitionnement

Tout système fondé sur une représentation à taille de blocs variable s'appuie sur un critère d'homogénéité et sur une topologie particulière de partition. Par la suite, on considère la partition Quadtree $P^{[N_{max} \dots N_{min}]}$, où N_{max} et N_{min} représentent les tailles respectivement maximale et minimale autorisées des blocs. $I(x,y)$ désigne un point de coordonnées (x,y) dans I , et $I(b^N(i,j))$ le bloc $b^N(i,j)$ dans l'image I tel que

$$b^N(i,j) = \{(x,y) \in N_x \times N_y \mid N \times i \leq x < N \times (i + 1), \text{ et } N \times j \leq y < N \times (j + 1)\}. \tag{2}$$

Parmi les nombreux exemples existants, citons celui de H.264 en mode intra, autorisant une partition Quadtree $P^{[16,4]}$ (partitionnement de l'image en blocs de taille 4x4 à 16x16), où la taille choisie est celle fournissant le meilleur rapport débit/distorsion d'un point de vue PSNR [1]. Les méthodes fondées sur des structures en arbres fonctionnent depuis le plus haut niveau (ou taille maximale) par découpage des nœuds en fils, lorsque un seuil d'homogénéité est dépassé. Si plusieurs tests d'homogénéité ont pu être proposés dans la littérature [9, 10], dans la majorité des cas ils s'appuient sur un calcul de distance de norme L_1 ou L_2 , entre la valeur d'un bloc et celle de ses quatre fils.

Nous proposons ici un critère différent fondé sur une détection de contours dont la justification sera donnée ultérieurement. Parmi les nombreux types de filtre possibles, nous avons opté pour un gradient morphologique (différence entre valeurs de luminance maximale et minimale sur un support donné), à la fois pour sa mise en œuvre rapide et récursive, et pour la propriété qui en découle de limitation de la valeur absolue de l'image de texture E (voir § 2.1.2).

On considère une partition Quadtree $P^{[N_{max} \dots N_{min}]}$. Soient $\min[I(b^N(i,j))]$ et $\max[I(b^N(i,j))]$ respectivement les valeurs minimum et maximum dans le bloc $I(b^N(i,j))$.

L'image des tailles en tout point est donnée par

$$Siz(x,y) = \begin{cases} N \in [N_{max} \dots N_{min}] & \text{si } |\max[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))] - \min[I(b^N(\lfloor \frac{x}{N} \rfloor, \lfloor \frac{y}{N} \rfloor))]| \leq Th \\ \text{et si } \exists(k,m) \in \{0,1\}^2 / & |\max[I(b^N(\lfloor \frac{x+k}{N/2} \rfloor, \lfloor \frac{y+m}{N/2} \rfloor))] - \min[I(b^N(\lfloor \frac{x+k}{N/2} \rfloor, \lfloor \frac{y+m}{N/2} \rfloor))]| > Th \\ N_{min} \text{ sinon.} & \end{cases} \tag{3}$$

où Th est le seuil d'homogénéité.

L'image des tailles donne alors directement une carte de segmentation grossière de l'image, où les blocs de taille N_{min} se retrouvent essentiellement sur les frontières et zones très texturées de l'image. Nous verrons dans la suite de cette étude que cette caractéristique est le fondement des différentes phases du procédé de codage.

Dans le cas des images couleur, la solution adoptée consiste à définir une seule partition régulière pour l'ensemble des trois composantes Y:Cr:Cb pilotée par la taille minimale.

L'image des tailles en tout point $p(x,y) \in I$ s'obtient alors par :

$$Siz(x,y) = \min [Siz_Y(x,y), Siz_{Cr}(x,y), Siz_{Cb}(x,y)]. \tag{4}$$

Les seuils Th pour la composante de luminance et celles de chrominance peuvent être fixés indépendamment dans notre schéma. Pour un même seuil Th , le minimum est majoritairement fourni par la composante Y. Dans la suite de cet article, nous nous sommes placés dans cette configuration.

2.1.2. Estimation des valeurs moyennes des blocs

Une image couleur basse résolution ($LR_Y : LR_{Cr} : LR_{Cb}$) sera obtenue par le codeur spatial en représentant, pour chaque composante, chaque bloc par sa moyenne. En chaque point $p(x,y)$, l'image LR de chaque composante est définie par :

$$LR(x,y) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} I(\lfloor \frac{x}{N} \rfloor \times N + k, \lfloor \frac{y}{N} \rfloor \times N + m), \tag{5}$$

avec $N = Siz(x,y)$

Comme la valeur moyenne du bloc est naturellement comprise entre les valeurs minimales et maximales, une propriété intrinsèque de la décomposition est que l'erreur de représentation $E(x,y)$ se trouve bornée pour les blocs de taille supérieure à N_{min} :

$$E(x,y) = |I(x,y) - LR(x,y)| \leq Th, \tag{6}$$

pour tout $p(x,y) \in P^{[N_{max} \dots N_{min}]}$

Pour chaque composante de l'image, l'entropie de l'erreur, l'erreur quadratique moyenne, ainsi que le PSNR admettent ainsi une limite

$$\begin{aligned} H(E) &\leq \log_2(Th) \text{ bits,} \\ MSE &\leq Th^2, \\ PSNR &\geq 10 \log \frac{255^2}{Th^2} \text{ dB.} \end{aligned} \tag{7}$$

2.1.3. Encodage des moyennes par prédiction de type MICD

En plus du facteur de compression apporté par le sous-échantillonnage de l'image, le coût global du codage est réduit à travers une quantification des valeurs des blocs, et une phase prédictive de ces valeurs.

Quantification des moyennes des blocs. Les techniques de compression fondées sur l'optimisation débit/distorsion tentent de trouver le meilleur compromis entre coût de codage et erreur globale dans l'image d'un point de vue $PSNR$ ou MSE , sans tenir compte de la perception visuelle humaine. Or des expérimentations ont démontré que l'œil s'avère beaucoup moins sensible à des variations de luminance et de chrominances dans des

zones de type frontière (hautes fréquences visuelles [11, 12]) que dans des zones uniformes (basses fréquences visuelles). La loi de Ricco indique également que le seuil de détection d'un saut de luminance d'une zone est inversement proportionnel à la dimension de cette zone. Autrement dit, les dégradations visuelles engendrées sur un bloc par une quantification linéaire [13], sont inversement proportionnelles à la taille de celui-ci. Ce principe va simplement s'appliquer dans notre schéma de codage en opérant une quantification adaptée à la taille des blocs. Si q_N représente le pas de quantification pour les blocs de taille N , une relation du type $q_N = \frac{q_{N/2}}{2}$ entre les pas des blocs de taille N et $N/2$ aboutit à une qualité visuelle quasiment constante sur toute l'image.

Prédiction des valeurs moyennes des luminances. Le codage des luminances moyennes des blocs est ici directement réalisé dans le domaine spatial, par une approche de type MICD (Modulation d'Impulsions à Code Différentiel). Ce choix a été essentiellement motivé, d'une part par la simplicité du codage qui ne nécessite qu'un balayage régulier de l'image, d'autre part parce que la représentation en blocs obtenue fournit un *a priori* sur les zones d'activité, pouvant être utilisé pour adapter la prédiction. Nous nous sommes inspirés des techniques existantes dans le domaine de la compression sans perte, où beaucoup de méthodes sont fondées sur ce type de prédicteur, afin de trouver le meilleur rapport efficacité/simplicité. Nous avons notamment implanté les prédicteurs MED (Median Edge Detector) de LOCO-I [14], le prédicteur DARC (Differential Adaptive Run Coding) proposé dans [15]. Ce meilleur compromis a finalement été obtenu grâce à un simple prédicteur de Graham [16] adapté à notre contexte. Cette adaptation consiste dans le principe à réaliser une prédiction linéaire dans les zones homogènes, et non linéaire sur les contours. La prédiction est pilotée par le gradient local et permet d'optimiser le prédicteur en fonction du contexte. Le prédicteur est donné par :

À tout point sommet $p(x,y)$ du bloc $b^N(x,y)$, on définit la valeur estimée du bloc $L\check{R}_Y(x,y)$ à partir des valeurs reconstruites $L\check{R}_Y(x-k,y-m), (k,m) \in \{0,1\}$. Le prédicteur est donné par la relation

$$L\check{R}_Y(x,y) = \begin{cases} L\check{R}_Y(x-1,y) & \text{si } |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x,y-1)| < |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x-1,y)| \\ \text{et si } A_N < |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x-1,y)| & (8) \\ L\check{R}_Y(x,y-1) & \text{si } |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x-1,y)| < |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x,y-1)| \\ \text{et si } A_N < |\check{L}R_Y(x-1,y-1) - \check{L}R_Y(x,y-1)| & \\ (\check{L}R_Y(x-1,y) + \check{L}R_Y(x,y-1))/2 & \text{sinon.} \end{cases}$$

A_N est un paramètre croissant de N , avec $A_1 = 0, A_2 = 10, A_4 = 20, A_8 = 40, A_{16} = 80$.

Les quantifications des erreurs de prédictions suivent le principe énoncé à la section précédente, avec une quantification adaptée à la taille du bloc. En appelant $E_{LR_Y}(x,y)$ l'erreur de prédiction, $\hat{E}_{LR_Y}(x,y)$ et $\check{E}_{LR_Y}(x,y)$ respectivement les erreurs quantifiées et déquantifiées, q_N le pas de quantification appliqué aux blocs de taille N , les relations suivantes sont définies :

$$\begin{cases} E_{LR_Y}(x,y) = LR_Y(x,y) - L\check{R}_Y(x,y), \\ \hat{E}_{LR_Y}(x,y) = Q(E_{LR_Y}(x,y)) = \text{round} \left[\frac{E_{LR_Y}(x,y)}{q_N} \right], \\ \check{E}_{LR_Y}(x,y) = Q^{-1}(\hat{E}_{LR_Y}(x,y)) = q_N \cdot \hat{E}_{LR_Y}(x,y), \\ L\check{R}_Y(x,y) = L\check{R}_Y(x,y) + \check{E}_{LR_Y}(x,y). \end{cases} \quad (9)$$

Les pas de quantification q_N fournis dans le tableau 1 correspondent aux valeurs possibles utilisables sans trop de dégradations de l'image. Nous avons empiriquement observé le fait suivant : les valeurs retenues sont telles qu'en les divisant par deux, la distorsion due à leur quantification devient imperceptible pour un affichage écran.

Tableau 1. Valeurs de quantification selon la taille.

Taille	q_N
16 × 16	2
8 × 8	4
4 × 4	8
2 × 2	16
1 × 1	32

Prédiction des valeurs moyennes des chrominances. Un grand avantage du codage opérant directement dans le domaine spatial est de pouvoir se servir de la forte corrélation de structure existant entre les trois composantes. La composante LR_Y étant transmise en première, le codage des valeurs moyennes des blocs de chrominance tire profit de cette information pour optimiser sa prédiction. L'estimation des valeurs prédites des chrominances en fonction des valeurs reconstruites des luminances est formalisée ci-contre.

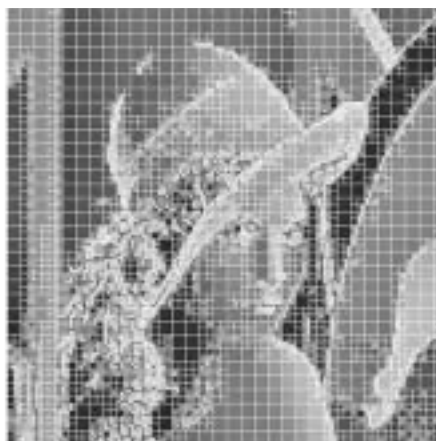
$$\text{Soit } GradMin_Y(x,y) = \min \left[\left| \check{L}R_Y(x,y) - \check{L}R_Y(x,y-1) \right|, \left| \check{L}R_Y(x,y) - \check{L}R_Y(x-1,y) \right|, \left| \check{L}R_Y(x,y) - \frac{\check{L}R_Y(x,y-1) + \check{L}R_Y(x-1,y)}{2} \right| \right], \quad (10)$$

alors

$$L\check{R}_{Cr/b}(x,y) = \begin{cases} \check{L}R_{Cr/b}(x-1,y) & \text{si } |\check{L}R_Y(x,y) - \check{L}R_Y(x-1,y)| = GradMin_Y(x,y) \\ \check{L}R_{Cr/b}(x,y-1) & \text{si } |\check{L}R_Y(x,y) - \check{L}R_Y(x,y-1)| = GradMin_Y(x,y) \\ \frac{\check{L}R_{Cr/b}(x-1,y) + \check{L}R_{Cr/b}(x,y-1)}{2} & \text{sinon.} \end{cases}$$



Image source Lena 512×512 - 8 bpp



Partition Quadtree : 13 888 blocs
dont 8 472 b^2 - 4 034 b^4 - 990 b^8 - 392 b^{16}



Image reconstruite basse résolution :
0.2 bits/pixel
taux compression = 40, PSNR 29.2 db



Image reconstruite après post-traitement
sur $P^{[16...2]}$, PSNR 29.7 db

Figure 3. Résultats pour une partition $P^{[16...2]}$, $Th = 30$.

Cette optimisation de la prédiction apporte un gain très significatif de l'ordre de 20 % par rapport à un codage direct.

2.1.4. Post-traitement

Par construction, une image *LR* obtenue présente des «effets de blocs», surtout perceptibles sur la composante de luminance. Ceux-ci ne sont toutefois pas comparables aux artefacts des techniques fondées sur la décomposition de l'image en blocs de taille fixe comme JPEG, MPEG-2 ou MPEG-4. Ils apparaissent en effet ici essentiellement dans les zones uniformes représentées par valeurs de plateau, et sur les contours par un manque de résolution si $N_{min} > 1$. La possibilité d'un post-traitement peut s'envisager avec comme but de lisser les zones homogènes tout en conservant les contours. Une méthode d'interpolation linéaire adaptée au partitionnement de l'image a été élaborée. Les images obtenues sont alors de très bonne qualité visuelle sur les zones uniformes (blocs de taille 8×8 et 16×16) ainsi que sur

les zones de contours (blocs de taille 2×2 et 4×4). Cependant, de meilleurs rendus visuels ont été obtenus par l'utilisation de l'algorithme d'interpolation directionnelle décrit par D. Muresan [17], fondé sur la théorie de la récupération optimale adaptative des valeurs manquantes. Cette technique élaborée par Golomb [18], a été appliquée initialement à l'interpolation par Shenoy et Parks [19]. Sur la figure 3, on constate que le post-traitement permet d'obtenir un lissage des zones homogènes tout en préservant les contours, par l'application de l'interpolation le long de ceux-ci [20].

2.2. Le codeur spectral

L'image d'erreur *E* issue de la représentation du codeur spatial est compressée dans un espace transformé fréquentiel, par une seconde couche, appelée codeur spectral. Le support de *E* considéré est le même que celui de \bar{I} , ce qui permet une caractérisa-

tion *a priori* de E , et une adaptation du schéma de compression. La technique de codage définie se base sur une approche DCT à taille de blocs variable, où la taille est définie par la partition $P^{[N_{max} \dots N_{min}]}$ de l'étage précédent (voir fig. 4). Seuls les coefficients AC nécessitent une transmission puisque le premier codeur fournit déjà la luminance moyenne (coefficient DC).

Les étapes majeures du procédé sont :

- l'application d'une transformation DCT de taille adaptée au bloc,
- le codage des coefficients: un balayage zig-zag intra-bloc, suivi d'un codage des valeurs non nulles par «longueur de plage» (LDP), incluant les indicateurs spécifiques de fin de bloc et de «plage maximale»,
- la quantification: la table de quantification est adaptée à chaque taille de bloc.

Par construction, le schéma de codage est progressif, autorisant des transmissions séparées des flux en fonction des tailles des blocs. Ainsi, est-il possible de n'améliorer que les contours par l'envoi de la texture des blocs de taille N_{min} .

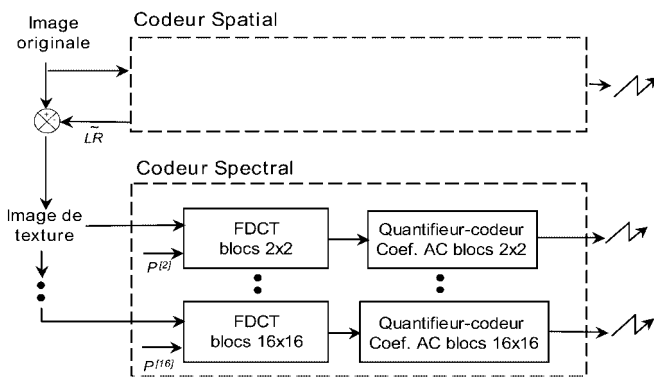


Figure 4. Schéma de principe du codeur spectral pour une partition $P^{[16...2]}$.

2.2.1. Énergie des blocs à taille fixe / variable

Par construction, les blocs contenant la texture locale présentent une erreur bornée, l'essentiel des erreurs se concentrant dans les blocs de petite taille. Par voie de conséquence, l'énergie moyenne des coefficients AC pour la partition $P^{[N_{max} \dots N_{min}]}$, reste très inférieure à celle trouvée dans des approches classiques à taille de blocs fixe (voir fig. 5). De plus, les coefficients se regroupent très majoritairement près de la composante continue. Après quantification, les coefficients à coder sont à la fois moins nombreux et de dynamique moindre.

2.3. Quantification des coefficients AC

Nous avons opté dans un premier temps pour les tables de quantification issues de JPEG, en les tronquant ou extrapolant suivant la dimension des blocs. Cette solution a montré ses limites, particulièrement pour des compressions à hauts débits.

La raison réside dans le fait que les caractéristiques de quantification pour du JPEG ont été établies pour prendre en compte tout aussi bien des blocs à forte ou à faible activité. Par construction, l'énergie des blocs traités ici est limitée, ce qui remet en cause les hypothèses sur lesquelles s'appuie JPEG. Nous donc opté pour un autre schéma très simple de quantification linéaire à base de deux paramètres Q_N et Δ_{Q_N} tels que pour un coefficient AC sur la k^{eme} diagonale d'un bloc $N \times N$, son pas de quantification soit donné par

$$Q = Q_N + k \cdot \Delta_{Q_N} \quad (11)$$

2.4. Résultats codec LAR simple

Dans une optique de codage d'image couleur, s'est avéré le fait que le codeur spatial seul est à même de coder et représenter les images de chrominances avec une bonne qualité. En particulier, dans les zones uniformes (blocs de taille supérieure à N_{min}), le sous-échantillonnage plus important réalisé n'est pratiquement pas perceptible. Par ailleurs, pour une partition dans laquelle $N_{min} = 2$, coder les chrominances par le seul codeur spatial revient «au mieux» à une compression dans le format 4 : 2 : 0, où Cr et Cb sont sous-échantillonnées d'un facteur 2 dans les deux directions. Enfin d'un point de vue complexité, l'intérêt principal est le suivant : l'extension du schéma de codage à trois composantes ne multiplie pas par trois la complexité du système. Pour illustrer cette bonne qualité de codage des chrominances, la figure 6 présente des images où seules les deux composantes chromatiques ont été compressées en utilisant uniquement le codeur spatial, l'image reconstruite incluant ici la composante Y d'origine.

Il s'avère de fait que le critère de débit/distorsion n'est pas réellement un indicateur rendant compte de la qualité des images reconstruites. Pour leur évaluation subjective, des images issues de notre schéma de compression ont été intégrées à une série de tests comparatifs pour juger de la qualité d'images compressées par différents schémas de codage. Ces tests ont été réalisés par P. Le Callet de l'IRCCYN : l'objectif était d'en extraire des lois pour un système automatique de mesures de qualité.

Cette étude a porté sur trois méthodes JPEG, JPEG2000 (codec *ImagXpress*, simple profile) et le LAR, et a été effectuée sur huit images standard (*Lena*, *Baboon*, *Boats*, *House*, *Pepper*, *Fruits*, *Airplane*, *Barbara*) compressées à différents débits. L'environnement était parfaitement standardisé en termes de distance à l'écran, de luminosité, de calibrage du moniteur, d'éclairage ambiant et de température de couleur pour une évaluation rigoureuse.

Le protocole élémentaire d'évaluation d'une image se déroulait ainsi :

- image originale durant six secondes,
- gris uniforme durant deux secondes,
- image à évaluer pendant six secondes,
- gris uniforme durant deux secondes.

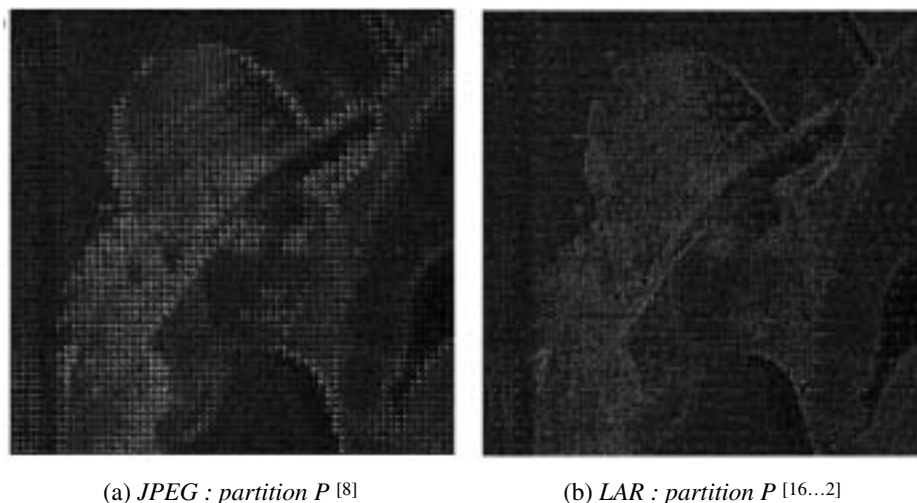


Figure 5. Répartition d'énergie entre partitions à taille fixe/variable.

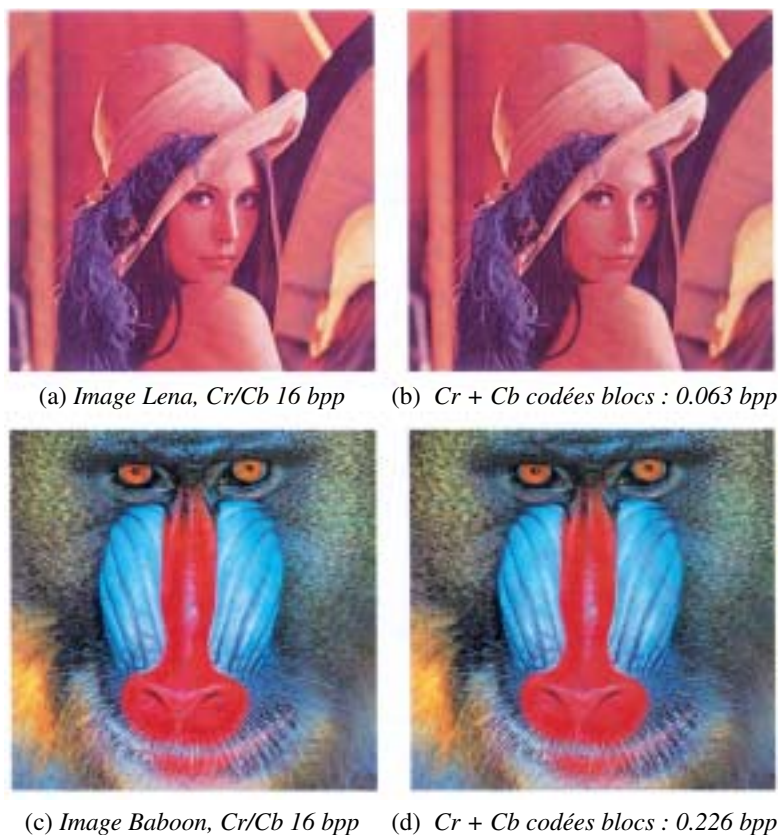


Figure 6. Exemples d'images reconstruites avec codage des chrominances par le codeur spatial (*Y* source non modifiée).

Chaque observateur (quatorze au total) devait ensuite donner une note située sur une échelle de qualité entre un (très mauvaise qualité) et cinq (très bonne qualité). Sur huit séries d'images différentes, le LAR s'est montré supérieur pour sept d'entre elles. Seule l'image baboon très texturée a fait exception et cela pour les bas débits. Nous montrons (figure 7) les résultats obtenus pour trois de ces séries. Notons toutefois que les trois

codeurs n'ont pas les mêmes caractéristiques d'un point de vue progressivité : le mode de codage JPEG choisi est celui du non progressif. Les résultats débits/qualité correspondants ont été obtenus en faisant varier le paramètre de quantification, pour des codages distincts. À l'opposé, le codeur JPEG2000 est entièrement progressif et les courbes associées reflètent l'évolution continue de la qualité possible. Le codage LAR se situe ici entre

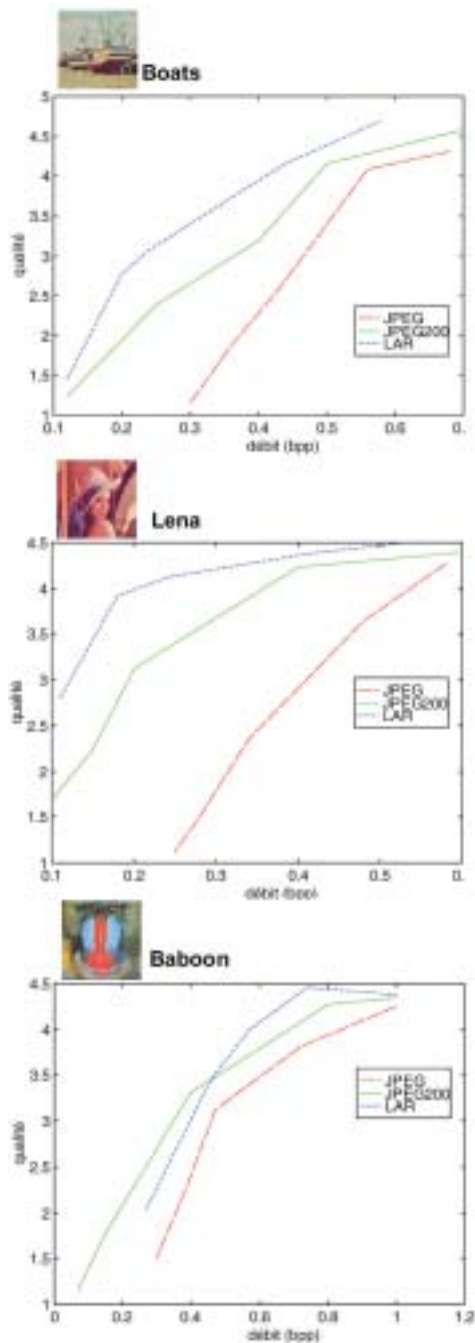


Figure 7. Résultats tests comparatifs de perception visuelle.

les deux, avec une progressivité du codage par paliers. Par ailleurs, les codeurs JPEG et LAR admettent une limite de compression à bas débits qui, pour le LAR, est fixée par la première étape de codage spatial. Cette limite se révèle d'autant plus basse que l'image est faiblement texturée. La figure 8 illustre à titre d'exemple, la qualité visuelle de notre schéma de codage. Le codec LAR présenté ici constitue la version la moins élaborée et donc bien évidemment la moins complexe. Une alternative, à la fois au codage MICD de la couche spatiale et au codeur spectral dans son ensemble, a été développée. Il s'agit d'une décomposition pyramidale originale dite à « contexte enrichi ».

3. Représentation en régions auto-extractibles et à coût nul

Elle permet d'introduire de la progressivité dans le codeur spatial [21], mais aussi de coder l'image d'erreur cette fois sans perte [22]. Cette forme de codage permet en grande partie de lever les limitations du LAR dans sa forme simple, à savoir une progressivité par paliers beaucoup plus fine, et un codage possible à très bas débits par diminution de la résolution spatiale. Toutefois les principes de base restent les mêmes (partition suivant l'activité), et la technique de représentation en régions présentée dans la section suivante, est applicable dans les deux versions du codec.

Pour obtenir une représentation en régions à coût de description nul, la seule solution réside dans une transmission d'une image compressée, suivie d'une segmentation identique au codeur et au décodeur. N'importe quel schéma de codage peut alors convenir lorsque l'image est codée à une bonne qualité (moyen/haut débit). Toutefois les méthodes standard engendrent à très bas débit des dégradations telles que le procédé de segmentation n'est plus envisageable. Pour sa part notre approche, fondée sur une représentation de l'image cohérente en termes de contours et de zones uniformes, se prémunit de ces dégradations pénalisantes. En rapprochant notre méthode de la technique de segmentation fondée Quadtree exposée dans [23], nous avons développé une technique de segmentation en régions s'appuyant sur la représentation du codeur spatial LR_{γ} . L'opération de découpage en blocs (« split ») ayant déjà été effectuée à travers le codeur spatial, il suffit de réaliser l'opération de fusion (« merge ») sur les blocs au codeur et au décodeur pour disposer d'une carte de segmentation identique. La reconstruction directe des régions à partir de la représentation en blocs apporte une compatibilité entre forme et contenu des régions, et exploitée par la suite pour du codage par ROI de la texture et le codage par régions de la chrominance. Notons qu'il n'est pas question ici de vouloir proposer la meilleure technique de segmentation qui soit, mais bien de développer une méthode globale de représentation à partir d'images compressées bas débit. Nous avons également cherché à modifier les approches existantes dans le but d'une simplification du procédé de segmentation pour, à terme, envisager une implantation temps réel de notre schéma de codage.

Nous présentons donc ici les principes fondamentaux, en ne considérant que la composante de luminance, de la représentation en régions auto-extractibles et son application pour le codage par ROI de la texture. Le chapitre 4 montrera son extrapolation et son utilisation pour le codage des images couleur.

3.1. Méthodes de segmentation en régions par graphes d'adjacence

3.1.1. Problématique

Soient $S = \{(x,y) | 1 \leq x \leq N_x, 1 \leq y \leq N_y\}$ les coordonnées spatiales des pixels dans une image de N_x lignes et N_y colonnes. La segmentation d'une image en K régions R_k^K consiste à trouver la partition Δ^K de S telle que

$$S = \bigcup_{k=1}^K R_k^K, \quad (12)$$

avec $R_i^K \cap R_j^K = \emptyset, \forall (i,j) \in \{1 \dots K\}^2$, pour $i \neq j$.

Notons S^K l'ensemble des régions dans la partition Δ^K . Partant d'une partition initiale Δ^{K_0} ($K_0 \leq N_x \times N_y$), le but du procédé de segmentation est de transformer Δ^{K_0} en une nouvelle partition Δ^K ($K < K_0$) selon un critère d'homogénéité, et à travers des séquences de fusions de régions. Dans notre cas, Δ^{K_0} vaut $P^{[N_{max} \dots N_{min}]}$, et S^{K_0} correspond aux blocs de luminance issus du codeur spatial.

Partitionner l'ensemble des régions élémentaires S^{K_0} en sous-ensembles peut se traduire par la recherche d'une relation d'équivalence \mathfrak{R} sur S^{K_0} . Ces sous-ensembles formés constituent alors des classes d'équivalence. Dans la suite, nous noterons $[R_i^{K_0}]_{\mathfrak{R}^K}$ une région de S^K dans la partition Δ^K , initialement associée à la région R_i de S^{K_0} .



Figure 8. Résultats comparatifs de qualité visuelle.

3.1.2. Graphe d'adjacence

De manière naturelle, les régions reconstruites doivent former des ensembles spatialement connexes. Aussi, la relation d'adjacence est-elle au cœur des principes de segmentation. Nous appellerons par la suite A_i^K l'ensemble des régions connexes à $\left[R_i^{K_0} \right]_{\mathfrak{R}^K}$ dans la partition Δ^K .

La structure de données classique pour représenter des partitions est le « Region Adjacency Graph » (RAG) [24]. Le RAG^K d'une K-partition est défini comme un graphe non orienté, $G^K = (V, E)$, où $V = \{1, \dots, K\}$ est l'ensemble des sommets et $E \subset V \times V$ est l'ensemble des arêtes. Chaque région est représentée par un sommet du graphe, et entre deux sommets (régions) $R_i^K, R_j^K \in V^2$ il existe une arête (i, j) si les régions sont adjacentes.

3.1.3. Classification hiérarchique et métrique

Fusionner les régions selon un critère d'homogénéité se ramène en général à un problème de classification hiérarchique, consistant à chercher les éléments les plus proches au sens d'une distance D , puis à mesurer les agrégations entre les classes suivant un critère $Crit$ donné. La hiérarchie est dite *indicée* si pour toute partie H de la hiérarchie, la relation d'inclusion $H \subset H'$ implique que $D(H) \leq D(H')$. Un niveau hiérarchique donné correspond alors à la fusion entre un sommet et un ensemble de sommets connexes.

Les méthodes de segmentation utilisant ce schéma fonctionnent en général sur un même modèle, où les fusions sont effectuées deux à deux suivant un critère de distance minimale [25, 26]. Celui-ci consiste simplement à fusionner les deux régions, au niveau courant de hiérarchie, dont la distance est minimale. La hiérarchie est par conséquent indicée pour tous les niveaux.

La distance retenue est en général fondée sur la différence des luminances moyennes des régions, mais certaines méthodes proposent des métriques moins rudimentaires comme par exemple des caractéristiques de texture [27].

L'intérêt d'une approche par fusion selon la distance minimale réside dans le fait que l'on puisse exactement contrôler le nombre final de régions. Par construction, les régions fusionnées sont également les plus proches du point de vue de la distance choisie. Cet avantage n'est toutefois que relatif, car le nombre nécessaire de régions pour décrire « correctement » une image reste bien sûr dépendant de la complexité de celle-ci.

Le point faible majeur de cette approche tient surtout dans sa complexité. Même si des algorithmes rapides, fondés sur les listes d'attente, existent pour classer les distances, la méthode n'en demeure pas moins très coûteuse en temps [25]. Ceci est d'autant plus vrai lorsque l'ultramétrique (nouvelle distance après fusion de deux régions) est fondée sur une nouvelle mesure de distance en fonction des régions fusionnées, nécessitant alors aussi un reclassement.

D'autre part, les méthodes classiques considèrent une symétrie des distances entre deux régions. Or pour éviter la sur-segmentation sur les contours tout en préservant l'intégrité des zones homo-

gènes, un critère prenant en compte la taille des régions peut, pour une même valeur de distance, permettre de favoriser davantage la fusion de petites régions par rapport à des plus grandes.

3.2. Méthode de segmentation proposée

Les principales modifications apportées ici aux schémas classiques de segmentation construits sur des RAG concernent d'une part la notion de distance pondérée, d'autre part la distance conjointe moyenne/gradient, et enfin le procédé de classification lui-même.

3.2.1. Distance pondérée

Pour rendre le pouvoir de fusion dépendant de la taille des régions, nous avons introduit une pondération des distances en fonction de la surface des régions. Si $Cost\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right)$ définit la distance entre deux classes, alors la distance pondérée est donnée par

$$Cost'\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right) = \tag{13}$$

$$Cost\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right) \log_{10}\left(Surf\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}\right)\right)$$

où $Surf\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}\right)$ désigne la surface de la région $\left[R_i^{K_0} \right]_{\mathfrak{R}^K}$

Cela entraîne la relation suivante :

$$Surf\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}\right) > Surf\left(\left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right) \Leftrightarrow \tag{14}$$

$$Cost'\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right) > Cost'\left(\left[R_j^{K_0} \right]_{\mathfrak{R}^K}, \left[R_i^{K_0} \right]_{\mathfrak{R}^K}\right)$$

Un effet immédiat de cette non symétrie des distances est que le RAG n'est plus un graphe non orienté : entre deux sommets connexes se trouvent deux arêtes, pondérées différemment suivant le sens de la relation d'adjacence.

3.2.2. Distances pondérées par la moyenne et le gradient

La distance couramment utilisée est celle de la différence des valeurs moyennes de niveau de gris (ou couleur) des régions. Elle permet de juger du degré de similarité des régions d'un point de vue représentation par valeur plateau. Ce coût a également été retenu ici sous la notation $Cost_M\left(\left[R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[R_j^{K_0} \right]_{\mathfrak{R}^K}\right)$. La gestion de l'ultramétrique au sein de la hiérarchie est particulièrement simple à implanter, puisqu'il suffit de conserver les deux caractéristiques suivantes : la surface et la valeur moyenne d'une région.

Il est bien connu qu'un critère de fusion des régions, fondé uniquement sur les valeurs moyennes, fonctionne mal dans des zones uniformes comportant un gradient local. La segmentation

génère alors des phénomènes dits de faux contours. Pour y palier, nous avons donc ajouté une distance $Cost_{Gr}$ à cette distance $Cost_M$.

Cette distance $Cost_{Gr}$ est fondée sur une mesure de gradient au niveau des frontières entre les régions. Estimer le gradient implique de considérer la relation d'adjacence non pas uniquement au niveau courant de la partition, mais bien au niveau local de la partition initiale Δ^{K_0} . Ceci est particulièrement simple pour une partition Quadtree où la longueur de frontière commune entre deux blocs correspond alors simplement à la taille minimale entre les deux blocs. La fonction de coût fondée gradient est évidemment plus complexe à mettre en œuvre que celle exploitant la moyenne. Malgré tout, cette topologie particulière permet d'envisager une solution algorithmique qui ne nécessite pas de traitement directement au niveau de l'image, mais uniquement des opérations dans la structure de représentation associée aux graphes d'adjacence.

Nous avons exprimé dans un premier temps la distance totale comme une somme pondérée des deux distances $Cost_M$ et $Cost_{Gr}$, de manière à évaluer la qualité de la segmentation en fonction du poids de chacune. De cette étude est ressorti le fait que les meilleurs résultats s'obtenaient lorsque les contributions des deux distances étaient sensiblement égales. La distance totale a donc simplement été fixée comme la moyenne entre $Cost_M$ et $Cost_{Gr}$.

3.2.3. Critère de fusion

K_0 : partition initiale (blocs)

$Nb_{fusions} = 0$; $K = K_0$;

Faire

```

     $Nb_{fusions\_prec} = Nb_{fusions}$ ;  $i = 1$ ;
    Faire
        Si  $\left[ R_i^{K_0} \right]_{\mathfrak{R}^K} \in RAG^K$ 
            Trouver  $\left[ R_j^{K_0} \right]_{\mathfrak{R}^K} \in A_i^K$ 
                tel que  $Cost \left( \left[ R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[ R_j^{K_0} \right]_{\mathfrak{R}^K} \right) \leq$ 
                     $Cost \left( \left[ R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[ R_l^{K_0} \right]_{\mathfrak{R}^K} \right), \forall \left[ R_l^{K_0} \right]_{\mathfrak{R}^K} \in A_i^K$ 
            Incrémenter  $i$ ;
        Tant que  $i \leq K_0$ ;
         $i = 1$ ;
        Faire
            Si  $\left[ R_i^{K_0} \right]_{\mathfrak{R}^K} \in RAG^K$ 
                Si  $Cost' \left( \left[ R_i^{K_0} \right]_{\mathfrak{R}^K}, \left[ R_j^{K_0} \right]_{\mathfrak{R}^K} \right) < Th_{Cost}$ 
                    Fusionner  $\left[ R_i^{K_0} \right]_{\mathfrak{R}^K}$  et  $\left[ R_j^{K_0} \right]_{\mathfrak{R}^K}$ ;
                     $K = K - 1$ ; Incrémenter  $Nb_{fusions}$ ;
                Fin Si;
            Fin Si;
            Incrémenter  $i$ ;
        Tant que  $i \leq K_0$ ;
    Tant que  $Nb_{fusions\_prec} < Nb_{fusions}$ ;
```

Nous avons déjà souligné que le critère de distance minimale couramment utilisé, présente une complexité importante. Plus encore, son utilisation conjointement à des distances asymétriques conduit dans notre contexte à s'intéresser initialement aux petits blocs, pour une construction des régions depuis les contours jusqu'aux zones homogènes : une telle démarche n'est pas cohérente. Nous avons donc mis en place une méthode plus rapide, et surtout mieux adaptée aux distances asymétriques. Pour chaque balayage du graphe, on détermine simplement pour chaque région celle la plus proche d'un point de vue distance, puis la fusion est effectuée si la distance est inférieure à un seuil. Le procédé est réitéré jusqu'à ce qu'aucune fusion ne soit obtenue. Nous donnons ici l'algorithme de manière schématique (voir ci-dessus).

Th_{Cost} constitue le paramètre de réglage du degré de simplification de l'image.

3.2.4. Hiérarchie indicée

La méthode proposée n'engendre pas de hiérarchie indicée, puisque les niveaux dans la hiérarchie vont dépendre essentiellement du nombre de fusions réalisées pour une région. Toutefois, en réitérant le procédé pour des valeurs croissantes de seuils de segmentation, une hiérarchie indicée s'obtient, avec autant de niveaux que de seuils. *La représentation en régions reconstruite s'avère ainsi multirésolution.* Typiquement, nous utilisons trois seuils, mais le choix du nombre de niveaux de résolution reste complètement libre.

3.2.5. Élimination des petites composantes

Compte tenu de la pondération par la taille des composantes, la segmentation proposée n'engendre pas de sur-segmentation sur les zones contours. On peut même observer une relative stabilité du nombre de régions obtenues pour un seuil de segmentation identique appliqué sur des images de complexité très différente. L'explication réside simplement dans le fait que plus une image est complexe, plus elle va engendrer dans sa partition initiale de petites régions qui ultérieurement auront une plus forte tendance à fusionner. Néanmoins, de petites régions peuvent persister si elles présentent une forte différence avec leurs régions voisines. En conséquence, nous avons également mis en place un procédé classique de fusion des petites régions. Son seul paramètre tient en une valeur de surface, en dessous de laquelle toute région fusionnera avec la région la plus proche d'un point de vue distance.

3.2.6. Complexité de la segmentation

La prochaine section fournira plusieurs exemples d'images segmentées, permettant d'apprécier les performances qualitatives de la méthode. Du point de vue de la complexité, l'algorithme de fusion (cf. expression 15) converge rapidement (en général de 5 à 8 itérations). Le calcul de la distance $Cost_{Gr}$ représente à lui seul pratiquement la moitié du temps total de calcul.

L'implantation des algorithmes n'a pas pour l'instant été optimisée. Les performances de segmentation d'une image 512×512 avec 20 000 blocs dans la partition initiale, en intégrant la notion de distance gradient, sont d'environ une seconde sur un PC à 2 GHz.

3.2.7. Codage par ROI de la texture locale

Une application directe de la représentation en régions auto-extractibles est la conception d'un schéma de codage de régions d'intérêt. À partir de la carte de segmentation disponible à la fois au codeur et au décodeur, l'un ou l'autre peut définir sa propre ROI comme ensemble de régions de la description. La méthode procure alors tout à la fois *un outil semi-automatique de sélection des ROI, et probablement le moyen le plus efficace de la représenter*. En effet une ROI va alors être simplement décrite par les étiquettes des régions qui la composent.

D'autre part, la description d'une région se fonde sur la représentation en blocs. *Le rehaussement d'une ROI va donc simplement consister en l'utilisation du codeur spectral pour les blocs contenus dans la ROI*. La compatibilité entre la forme de la ROI et codage de son contenu est ainsi immédiate et totale, puisque la ROI agit directement comme une validation de type « on/off » sur les blocs à traiter.

L'ordonnement du procédé est le suivant : à partir du codage spatial de la composante de luminance Y, la représentation en régions est obtenue au codeur et décodeur à travers un procédé commun de segmentation. Il suffit ensuite de transmettre les valeurs chromatiques moyennes contenues dans ces régions. En ajoutant une phase de codage différentiel de ces valeurs puis une phase de quantification, le coût final pour chaque région s'avère de l'ordre de 4 bits. Ainsi, le coût de codage en termes d'octets, pour les deux composantes chromatiques, correspond approximativement au nombre de régions.

Nous donnons figure 9 plusieurs exemples d'images reconstruites avec Y source (non codée) et Cr/Cb codées régions. Afin de mieux rapprocher ces résultats de l'image source, la représentation en régions est donnée dans l'espace couleur (moyenne dans les régions pour chaque composante de couleur).

Il faut surtout souligner ici l'étonnant rapport qualité / taux de compression sur les composantes de chrominance : le coût de leur codage se situe tout au plus en centième de bit par point et devient insignifiant. On peut par ailleurs remarquer que les défauts visibles et gênants restent dus à des problèmes initiaux de segmentation, issus d'une partie d'objet affectée à une autre région, ou plus globalement d'une fusion de plusieurs régions (fig. 9).

Le codage de l'erreur commise sur les composantes chromatiques s'avère alors très simple : il consiste en effet uniquement à coder l'erreur au niveau des blocs, à travers le codeur spatial, et ce sur toute l'image ou bien simplement limité à une ROI.

4. Codage et représentation des images couleur

Nous avons déjà souligné que les composantes chromatiques pouvaient être codées très efficacement avec le codeur spatial du LAR. Par ailleurs, il est clair que l'information couleur permet d'améliorer très sensiblement la segmentation des images, même si cette dernière s'opère le plus souvent dans des espaces couleur R:V:B ou L:a:b [28]. Le schéma global de codage le plus naturel est donc de coder les images Y : Cr : Cb à travers le codeur spatial, puis de les transmettre au décodeur pour réaliser la segmentation sur les trois composantes. Notre but n'est pas ici d'établir une comparaison de notre segmentation avec d'autres approches dédiées. Nous avons simplement vérifié que ce mode opératoire est valide, et fournit de très bonnes descriptions en régions.

4.1. Codage des images de chrominances par régions

Un autre travail a consisté en l'utilisation de la représentation en régions issue de la seule composante LR_Y , afin de développer un codage par régions des deux composantes de chrominance.

4.2. Segmentation supervisée par le contrôle de la chrominance

Pour combiner à la fois compression par régions de Cr/Cb et amélioration de la qualité de segmentation par l'information couleur, nous avons défini un procédé de codage avancé, par le principe du « contrôle de la chrominance ». L'idée générale est la suivante : partant toujours d'une transmission préalable de l'image Y basse résolution des luminances, le procédé de segmentation de fusion continue d'être globalement dirigé suivant un critère de luminance, mais supervisé au codeur par un critère complémentaire de chrominance validant chaque tentative de fusion. Le résultat de ce contrôle de supervision doit bien sûr être transmis au décodeur. La forte corrélation existant entre la composante de luminance et de chrominance, d'un point de vue structuration des objets dans l'image, induit une faible quantité d'information supplémentaire à transmettre. Le schéma bloc du codeur est donné figure 10.

Description de l'algorithme. L'algorithme de fusion, à savoir la recherche de la région la plus proche uniquement pour l'image Y, n'est pas modifié. Soient $ControlChrom \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K} \right)$ une information binaire à transmettre pour chaque essai de fusion, et $CoefChrom$ un coefficient multiplicatif appliqué sur Th_{Cost} . Le critère de fusion devient :

$$\begin{array}{l}
 \text{Si } Cost' \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K}, \left[R_j^{K_0} \right]_{\mathcal{R}^K} \right) < Th_{Cost} \\
 \left| \begin{array}{l}
 \text{Si } CostCr'_M \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K}, \left[R_j^{K_0} \right]_{\mathcal{R}^K} \right) < \\
 \qquad \qquad \qquad CoefChrom.Th_{Cost} \\
 \text{Et } CostCb'_M \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K}, \left[R_j^{K_0} \right]_{\mathcal{R}^K} \right) < \\
 \qquad \qquad \qquad CoefChrom.Th_{Cost} \\
 \left. \begin{array}{l}
 ControlChrom \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K}, \left[R_j^{K_0} \right]_{\mathcal{R}^K} \right) = 1; \\
 \text{Fusionner } \left[R_i^{K_0} \right]_{\mathcal{R}^K} \text{ et } \left[R_j^{K_0} \right]_{\mathcal{R}^K};
 \end{array} \right. & (16) \\
 \text{Sinon} \\
 \left| \begin{array}{l}
 ControlChrom \left(\left[R_i^{K_0} \right]_{\mathcal{R}^K}, \left[R_j^{K_0} \right]_{\mathcal{R}^K} \right) = 0; \\
 \text{Fin Si}
 \end{array} \right. \\
 \text{Fin Si}
 \end{array}$$

Notons dans la formule précédente que nous avons uniquement considéré un calcul s'appuyant sur la moyenne des régions pour les deux composantes chromatiques. Ceci se justifie en premier lieu par l'objectif visé, à savoir une segmentation efficace pour coder Cr/Cb par régions, en second lieu par le souci de ne pas trop compliquer l'algorithme.

Par ailleurs, *ControlChrom*, en raison de sa très faible entropie, peut être fortement compressé par un codeur arithmétique. D'autre part, comme un coefficient multiplicatif *CoefChrom* est appliqué sur Th_{Cost} , Th_{Cost} garde alors son rôle de paramètre global de simplification de l'image. *CoefChrom* agit donc véritablement comme le contrôle adapté des chrominances. Il peut être ainsi vu comme un curseur de réglage entre coût du contrôle et qualité de la segmentation d'un point de vue couleur :

- si $CoefChrom \rightarrow \infty$, alors toutes les valeurs de contrôle valent 1, le coût entropique associé est donc nul, et la segmentation s'opère uniquement sur la luminance,
- si $CoefChrom = 1$, alors le seuil est le même pour les trois composantes, et le contrôle par la chrominance intervient avec une meilleure pondération,
- si $CoefChrom < 1$, alors la segmentation devient fortement dépendante des chrominances, mais le coût de contrôle croît avec l'entropie des valeurs binaires. En pratique, une valeur de 0.5 assure une très bonne qualité de segmentation couleur. Même lorsque le contrôle est important, le coût résultant demeure très faible, comme le montrent les résultats donnés des tableaux 2, 3 et 4. La décision de fusion doit évidemment être accompagnée d'un marquage de la région rejetée, de manière à ce que celle-ci ne

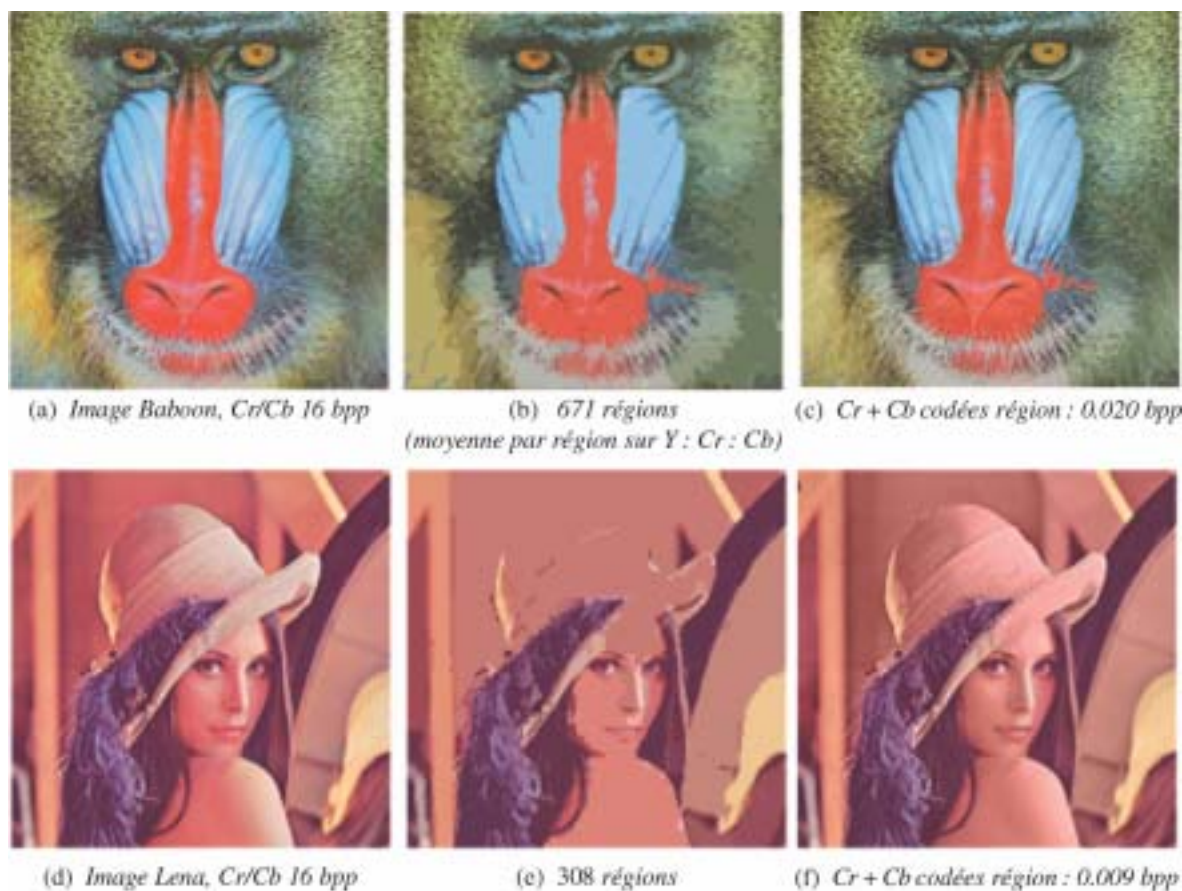


Figure 9. Exemples d'images reconstruites avec codage des chrominances par une représentation en régions, Seuil de distance $Th_{Cost} = 50$.

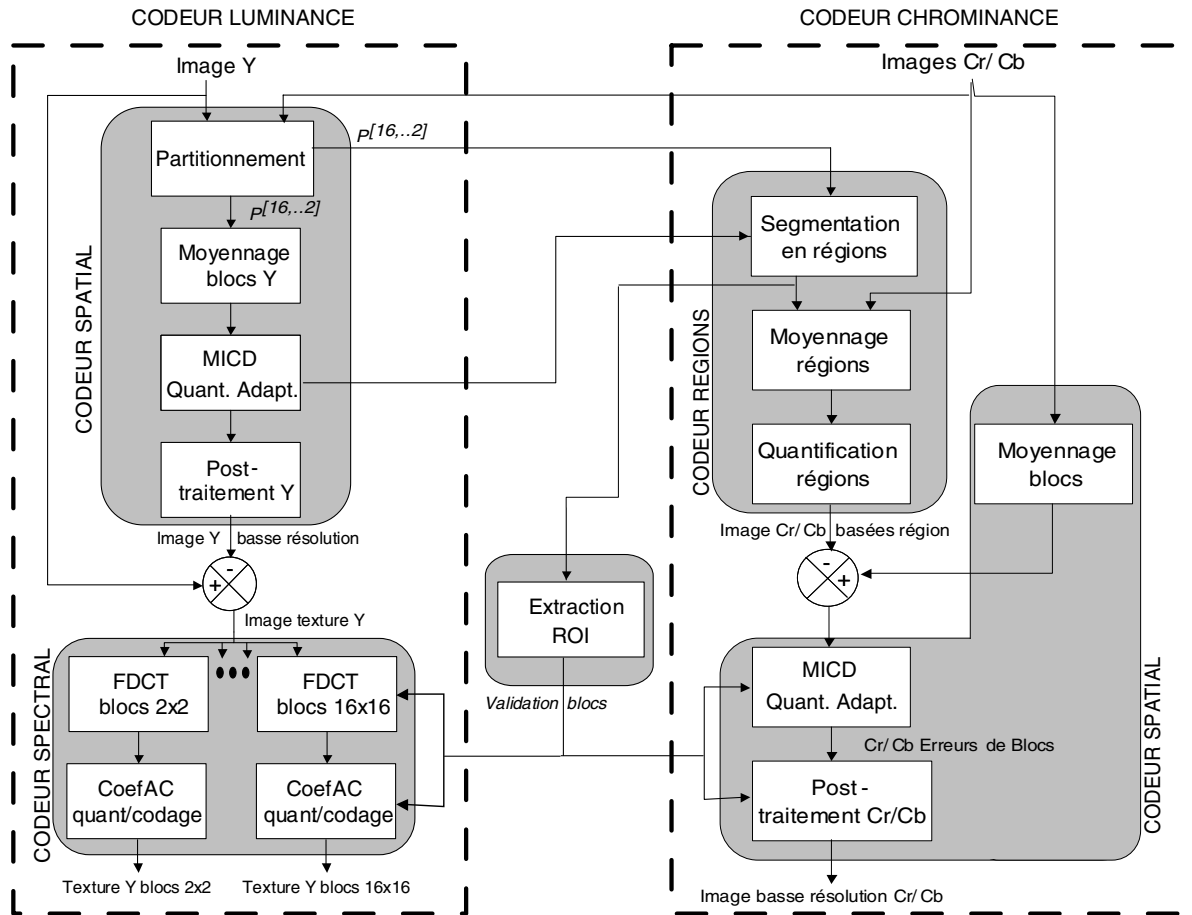


Figure 10. Schéma bloc du codeur LAR couleur fondé sur une segmentation supervisée.

reste pas la meilleure candidate pour la fusion d'un point de vue luminance. Nous avons donc introduit un graphe de dissymétrie des régions ou RDG (Regions Dissymetric Graph), sous-graphe du RAG. L'ensemble D_i^K associé fournit par conséquent l'ensemble des classes rejetées pour une classe donnée. Sa gestion lors de la fusion de deux régions est analogue à celle du RAG.

Résultats et influence des paramètres de segmentation. Pour juger de l'influence des paramètres Th_{Cost} d'une part, et $CoefChrom$ d'autre part, à la fois en termes de qualité de segmentation et de coût de codage des images de chrominance, nous avons fait varier ces deux paramètres indépendamment. La taille de l'image «Barbara» utilisée est de 512×512 pixels codés sur 24 bits, le nombre de blocs initiaux dans la partition est de 20 153 pour une image Y compressée à 0.19 bpp, et le coût codage direct des composantes Cr et Cb par le codeur spatial est de 0.12 bpp avec un PSNR de 40.1 dB. Les résultats quantitatifs sont résumés dans les tableaux 2, 3 et 4, et les images de segmentation ainsi reconstruites sont présentées figure 11. D'autres exemples de segmentation sont également proposés figure 12.

Plusieurs remarques peuvent être faites sur les résultats.

- Le codage des images de chrominance par régions permet d'atteindre des forts taux de compression.

- L'influence du paramètre $CoefChrom$ sur le nombre de régions générées est plus forte pour les partitions contenant peu de régions. Le coût du contrôle s'accroît. En revanche avec beaucoup de régions, son coût reste faible comparé à celui du codage des régions chromatiques, mais il apporte toujours un gain significatif du point de vue distorsion.

Tableau 2. Influence des paramètres de segmentation sur «Barbara», $Th_{cost} = 100$

Th_{Cost}	Niveau 3: $Th_{Cost} = 100$		
$CoefChrom$	2.0	1.0	0.5
Nb régions	50	81	189
Contrôle (bpp)	0.0037	0.0090	0.0310
Régions Cr/Cb (bpp)	0.0030	0.0043	0.0089
Coût total contr+reg (bpp)	0.0067	0.0133	0.0399
PSNR Cr/Cb (dB)	32.15	34.3	35.92
Err. Cr/Cb cod. spatial (bpp)	0.08	0.072	0.061
Coût total Cr/Cb niv. blocs (bpp)	0.0867	0.0853	0.1000

Tableau 3. Influence des paramètres de segmentation sur « Barbara », $Th_{cost} = 50$

Th_{Cost}	Niveau 2: $Th_{Cost} = 50$		
CoefChrom	2.0	1.0	0.5
Nb régions	178	201	304
Contrôle (bpp)	0.0026	0.006	0.0220
Régions Cr/Cb (bpp)	0.0070	0.0075	0.0130
Coût total contr+reg (bpp)	0.0096	0.0135	0.0350
PSNR Cr/Cb (dB)	33.42	34.8	36.2
Err. Cr/Cb cod. spatial (bpp)	0.072	0.070	0.067
Coût total Cr/Cb niv. blocs (bpp)	0.0816	0.0835	0.1020

Tableau 4. Influence des paramètres de segmentation sur « Barbara », $Th_{cost} = 25$

Th_{Cost}	Niveau 1: $Th_{Cost} = 25$		
CoefChrom	2.0	1.0	0.5
Nb régions	692	713	800
Contrôle (bpp)	0.0016	0.0045	0.0150
Régions Cr/Cb (bpp)	0.0240	0.0260	0.0290
Coût total contr+reg (bpp)	0.0256	0.0305	0.0440
PSNR Cr/Cb (dB)	34.76	35.3	36.86
Err. Cr/Cb cod. spatial (bpp)	0.063	0.070	0.056
Coût total Cr/Cb niv. blocs (bpp)	0.0886	0.1005	0.1000

- Lorsque l'on code ensuite l'erreur des images de chrominance au niveau des blocs dans la partition, c'est-à-dire par le codeur spatial, les résultats sont relativement stables, et assez proches d'un codage direct (0.12 bpp).

La figure 13 fournit des exemples d'images reconstruites avec la composante Y originale, et les composantes Cr/Cb codées régions, pour différents taux de compression. On peut constater qu'un codage chromatique avec un nombre suffisant de régions, permet d'obtenir une qualité acceptable de reconstruction dans un contexte bas débit, évitant ainsi le codage de l'erreur par le codeur spatial. On notera également la relative stabilité du nombre de régions résultantes pour différentes images.

4.3. Codage par régions contraintes par masques

Cette technique de contrôle par la chrominance a été étendue très simplement pour contraindre la segmentation à toute forme binaire d'objet donné. Cette solution s'avère par exemple néces-

saire lorsque la ROI est déjà définie au codeur de manière manuelle ou automatique (principe du VOP de forme quelconque dans MPEG-4). Le partitionnement initial de l'image inclut alors le masque binaire de l'objet. La segmentation s'effectue alors sous le contrôle de la forme, afin de garantir des régions soit entièrement externes, soit entièrement internes à l'objet. Celui-ci est alors désigné par l'ensemble des régions qui le composent. Le procédé est illustré figure 14, avec un rehaussement de qualité appliqué uniquement dans l'objet.

5. Conclusion et perspectives

Le schéma de codage « simple » du LAR présenté initialement dans cet article s'appuie sur une représentation à taille de blocs variable. Ce codec à deux couches de faible complexité autorise notamment un codage progressif sémantique : zones homogènes ou contours.

Nous avons ensuite développé une approche unique de représentation en régions dans un schéma de codage. Grâce aux bonnes propriétés de l'image produite par le codeur spatial, notamment l'intégrité des contours, nous avons mis en place un procédé de segmentation efficace à partir de cette seule image bas débit. Le résultat fournit une description hiérarchique de l'image en régions de bonne qualité d'un point de vue représentation de la scène, et surtout sans coût supplémentaire de description des formes.

Un important atout de notre schéma de représentation tient dans sa partition initiale en blocs, à la base des régions formées. Il s'en suit une adéquation parfaite entre forme et contenu des régions. Une application immédiate réside dans le codage par régions d'intérêt dans l'image : la description de celles-ci entraîne alors un coût quasi-nul, et le contenu d'une ROI donnée peut être directement et simplement rehaussé à l'aide du codeur spectral.

L'introduction d'un contrôle de la segmentation par l'information chromatique permet d'obtenir un schéma de codage très efficace tant du point de vue taux de compression que de la qualité des images reconstruites. En effet, l'introduction du codage par régions des images de chrominance autorise des taux de compression inégalés, par comparaison aux techniques classiques.

En définitive, l'ensemble de l'approche fournit une technique globale et cohérente pour un codage progressif des images couleur. Cet outil permet également la manipulation et la compression au niveau régions.

Actuellement, un nouveau champ de développement porte sur le remplacement de l'espace couleur actuel par un espace psychovisuel de représentation des couleurs [29, 30]. Celui-ci se particularise par une intégration des dépendances perceptuelles entre la composante achromatique (A) et les composantes chromatiques (C1C2) [31].

D'autres travaux en cours sur le codec LAR concernent son adaptation à la vidéo et au codage multirésolution avec et sans pertes, son intégration pour de l'électronique embarquée (cibles

DSP et FPGA), et enfin son imbrication dans des chaînes de transmissions numériques troisième (UMTS) et quatrième (MC-CDMA) générations.

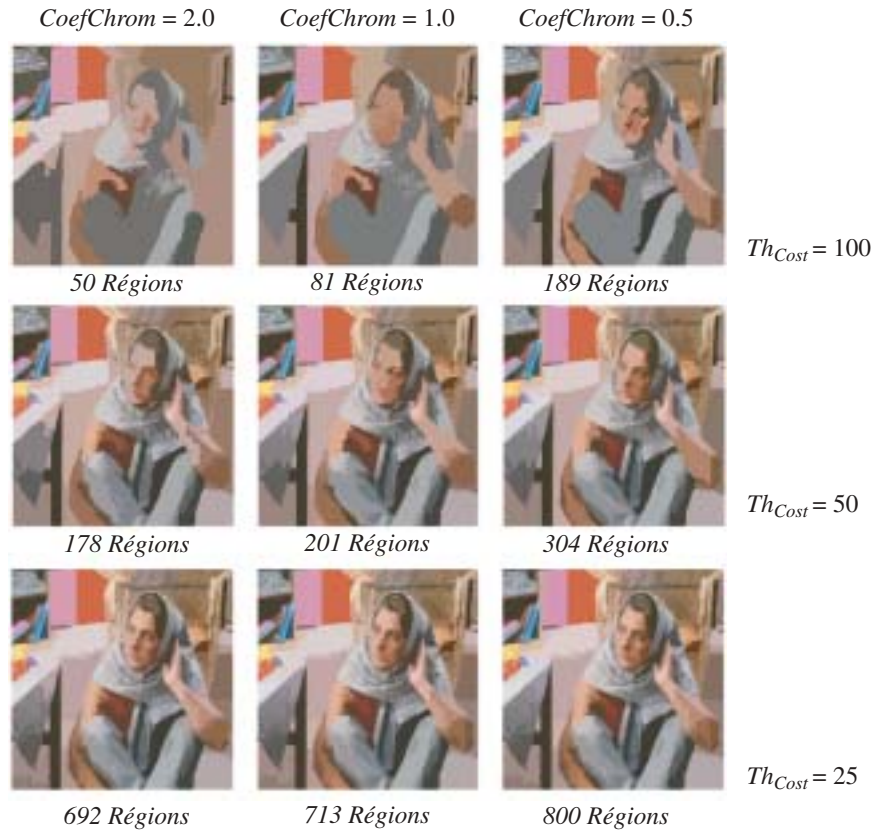


Figure 11. Résultats de segmentation en fonction des paramètres Th_{Cost} et $CoefChrom$ sur image « Barbara ».

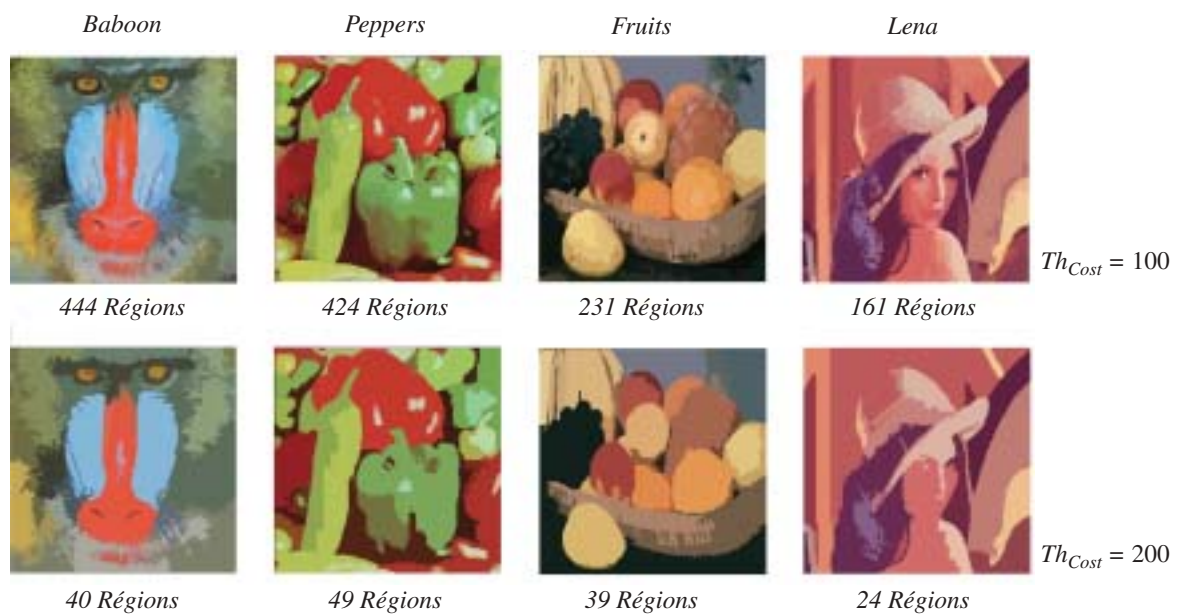


Figure 12. Résultats de segmentation sur « Baboon », « Fruits » et « Lena » ($CoefChrom = 0.5$).



Figure 13. Codage des chrominances par régions.

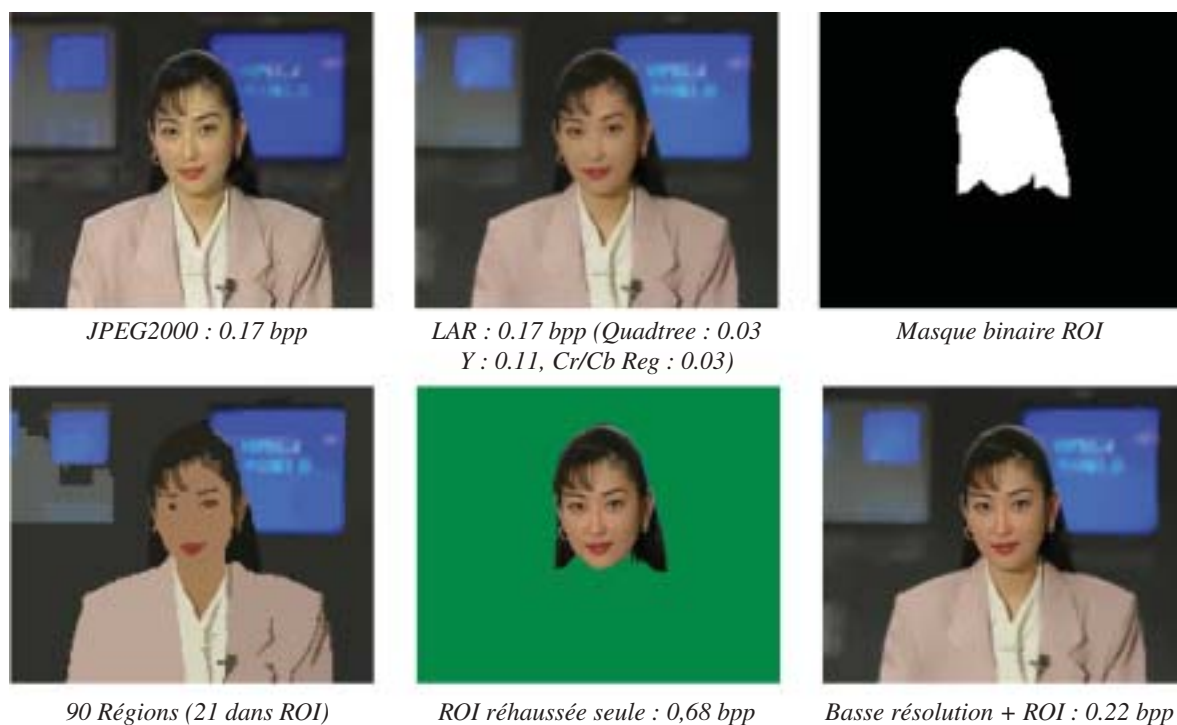


Figure 14. Codage par ROI sur image « Akiyo » avec contraintes de chrominance et masque binaire.

Références

- [1] H264 MPEG-4 10 AVC, "Joint Committee Draft (CD)", Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEGn 3rd Meeting: Fairfax, Virginia, USA, May 2002.
- [2] M. KUNT, A. IKONOMOPOULOS, M. KOCHER, "Second Generation Image Coding Techniques", *Proceedings of the IEEE*, 73(4): 549-575, April 1985.
- [3] L. TORRES, M. KUNT, "Video Coding: The Second Generation Approach", Kluwer Academic, 1996.
- [4] MPEG-4 Committee, "Generic Coding of Moving Pictures and Associated Audio Information: Video". ISO/IEC, 2000, May 2000.
- [5] MPEG-7 Requirements Group. "MPEG-7 : Context, Objectives and Technical Roadmap, V.12". ISO/IEC JTC 1/SC29/WG11 MPEG99/N2861, D Lib Magazine, Vancouver, July 1999.
- [6] T. EBRAHIMI, M. KUNT, "Visual Data Compression for Multimedia Applications", *Proceedings of the IEEE*, 86(6): 1109-1125, June 1998.
- [7] N. PETERFREUND, Y.Y. ZEEVI, "Nonuniform Image Representation in Area-Of-Interest Systems", *Trans. on Image Processing*, 4(9): 1202-1212, September 1995.
- [8] B. DEKNUYDT, J. SMOLDERS, L. VAN EYCKEN, A. OOSTERLINCK, "Color Space Choice for Nearly Reversible Image Compression", *SPIE, Visual Communications and Image Processing '92*, volume 1818, pages 1300-1311, Boston, November 1992.
- [9] C.A. SHAFFER, H. SAMET, "Optimal Quadtree Construction Algorithms", *Computer Vision, Graphics, Image processing*, 37(3): 402-419, March 1987.
- [10] P. STROBACH, "Tree-Structured Scene Adaptive Coder", *IEEE Trans. on Communication*, 38(4): 477-486, April 1990.
- [11] M. A. LOSADA, K.T. MULLEN, "The Spatial Tuning of Chromatic Mechanisms Identified by Simultaneous Masking", *Vision Research*, 34(3): 331-341, Februar 1994.
- [12] K.K. DE VALOIS, M.A. WEBSTER, E. SWITKES, "Orientation and Spatial-Frequency Discrimination for Luminance and Chromatic Gratings", *Journal of the Optical Society of America*, 7(6): 1034-1049, June 1990.
- [13] L. BÉDAT, A. SAADANE, D. BARBA, "Masking Effects of Perceptual Colour Components on Achromatic Grating", *ECVP 97*, Helsinki, August 1997.
- [14] M.J. WEINBERGER, G. SEROUSSI, G. SAPIRO, "LOCO-I: a low complexity, context-based, lossless image compression algorithm", *IEEE Data Compression Conference*, pages 141-150, March 1996.
- [15] B. GANDHI *et al.* "Dierential Adaptive Run Coding". Rapport Technique, ISO Working Document ISO/IEC JTC1/SC29/WG1 N204, 1995.
- [16] R.E. GRAHAM. "Predictive Quantizing of Television Signals", *IRE-WESCON Convention Record*, 22(4) :147-157, August 1958.
- [17] D. MURESAN, T. PARKS, "Optimal Recovery Approach to Image Interpolation", *ICIP'01*, volume 3, pages 848-851, 2001.
- [18] M. GOLOMB, H.F. WEINBERGER, "On Numerical Approximation", *Chapitre Optimal Approximation and Error Bounds*, pages 117-190, R. E. Langer ed., The University of Wisconsin Press, 1959.
- [19] R.G. SHENOY, T.W. PARKS, "An Optimal Recovery Approach to Interpolation", *IEEE Trans. Signal Processing*, 40(8): 1987-1996, August 1992.
- [20] D. MURESAN, "Review of Optimal Recovery", *Rapport Technique TR-2002-10, DMMD*, 2002.
- [21] M. BABEL, O. DÉFORGES, J. RONSIN, "Adaptive Multi-Resolution Scheme for Efficient Image Compression", *Picture Coding Symposium, PCS'03*, Saint-Malo, France, April 23-25 2003.
- [22] M. BABEL, O. DÉFORGES, J. RONSIN, "Lossless and Lossy Minimal Redundancy Pyramidal Decomposition for Scalable Image Compression Technique", *4th IEEE International Conference on Multimedia and Expos, ICME'03*, volume 3, pages 161-164, Baltimore, USA, July 6-9 2003.
- [23] P. WILLEMIN, T. REED, M. KUNT, "Image Sequence Coding by Split and Merge", *IEEE Trans. on Communication*, 39(12): 1845-1855, December 1991.
- [24] M. SONKA, V. HLAVAC, R. BOYLE, "Image Processing, Analysis, and Machine Vision", *Chapman & Hall*, 1993.
- [25] K. HARIS, S.N. EFSTRATIADIS, N. MAGLAVERAS, A.K. KATSAGGELOS, "Hybrid Image Segmentation Using Watersheds And Fast Region Merging", *IEEE Trans. on Image Processing*, 7(12): 1684-1699, December 1998.
- [26] J.-M. BEAULIEU, M. GOLDBERG, "Hierarchy in Picture Segmentation : a Stepwise Optimization Approach", *IEEE Trans. on Patern Analysis and Machine Intelligence*, 11(2) :150-163, February 1989.
- [27] J. MAEDA, V.V. ANH, T. ISHIZAKA, Y. SUZUKI, "Integration of Local Fractal Dimension and Boundary Edge in Segmenting Natural Images", *IEEE International Conference on Image Processing, ICIP'96*, volume 1, pages 845-848, 1996.
- [28] A. TREMEAU, "Analyse d'Images Couleurs : du Pixel à la Scène", H.D.R., Université de Saint-Etienne, May 1998.
- [29] J. KRAUSKOPF, D.R. WILLIAMS, D.W. HEELEY, "The Cardinal Directions of Color Space", *Vision Research*, 22(9): 1123-1131, 1982.
- [30] K.T. MULLEN, "The Contrast Sensitivity of Human Colour Vision to Red-Green and Blue-Yellow Chromatic Gratings", *Journal on Physiology*, 359: 381-400, 1985.
- [31] L. BÉDAT, A. SAADANE, D. BARBA, "Optimal Quantisation of Colour Image", *ECVP 99*, August 1999.



Olivier **Déforges**

Olivier Déforges est né le 02 octobre 1969 (100 ans après Gandhi) à Nantes. Il a obtenu son doctorat (École Polytechnique de l'Université de Nantes) en septembre 1995. Maître de Conférences depuis 1996 à l'INSA Rennes/ UMR CNRS 6164 IETR, il a soutenu son H.D.R. (Université Rennes) en novembre 2004. Ses travaux de recherche concernent le codage et la représentation des images, les méthodologies de prototypage rapide pour systèmes embarqués.



Laurent **Bédât**

Laurent Bédât est né le 17 octobre 1969, Maître de Conférences depuis 2000 à l'INSA Rennes/IETR. Travaux de recherche : compression d'images, psychophysique, perception des couleurs. Doctorat soutenu en octobre 1998.



Marie **Babel**

Marie Babel, née le 22 Janvier 1979, a été diplômée en 2002 du DEA STIR de l'Université de Rennes 1 et est ingénier de l'INSA de Rennes (Spécialité Électronique et Informatique Industrielle). Marie Babel effectue depuis lors sa thèse au sein du laboratoire IETR à Rennes, et dispense des enseignements à l'INSA. Ses recherches concernent la compression scalable d'images et de vidéos, et son application dans la télémédecine.



Joseph **Ronsin**

Joseph Ronsin est né en 1948. Il a obtenu son Habilitation à Diriger des Recherches en 1989 et est professeur à l'INSA de Rennes dans le département Électronique et Informatique Industrielle, et co-responsable du groupe Image et Télédétection au sein de l'Institut d'Électronique et Télécommunications de Rennes (IETR- UMR CNRS 6164). Ses activités de recherche sont orientées sur l'analyse, la représentation et la compression des images.