Protection de données à coût nul dans un codeur d'images multirésolution

Jean Motsch, Olivier Déforges, Marie Babel

Protection de données à coût nul dans un codeur d'images multirésolution

Jean MOTSCH[1] Olivier DÉFORGES[2] Marie BABEL[2]

[1] CREC Saint-Cyr/LESTP
IETR UMR CNRS 6164
Groupe Image et Télédétection

[2] INSA RENNES
IETR UMR CNRS 6164
Groupe Image et Télédétection

jean.motsch@st-cyr.terre.defense.gouv.fr, {olivier.deforges, marie.babel}@insa-rennes.fr

Résumé

Les performances des codeurs d'images fixes ne sont plus uniquement évaluées à l'aide des courbes débit-distorsion. Les services fournis sont également un critère de choix. Dans cet article, nous proposons d'intégrer dans un codeur d'images multirésolution offrant des performances supérieures à l'état de l'art avec et sans perte, un schéma de protection de contenu. L'utilisation judicieuse du flux binaire généré par le codeur permet d'obtenir ce service à coût nul. Des éléments tant théoriques que pratiques sont avancés pour justifier l'emploi de ce schéma de protection.

Mots clefs

Protection de données, codage d'image avec et sans perte, multirésolution.

1 Introduction

L'évolution récente des codeurs d'images fixes montre pleinement leur efficacité. Qu'ils soient normalisés (JPEG 2000) ou issus de laboratoires de recherche, leurs performances en terme de courbe débit–distorsion sont meilleures que celles des codeurs de la génération précédentes (JPEG, par exemple). Ceci est également vrai pour les codeurs sans perte.
Said avait résumé en 1999 [1] les caractéristiques qu'il considérait comme souhaitables pour un schéma de codage d'images fixes, à savoir :

– scalabilité ;
– flexibilité et adaptabilité ;
– régulation automatique du débit et contrôle de la qualité ;
– unicité de l'algorithme ;
– décodage de régions d'intérêt ;
– faible complexité ;
– compression efficace ;
– résistance aux erreurs ;
– bonne qualité visuelle.

Les propriétés énoncées concernent pricipalement les qualités du codage de l'image, l'adéquation à la bande passante disponible et la facilité de mise en œuvre.

Depuis, l'Internet haut débit, le faible coût des supports de reproduction, la prise en compte des droits de la propriété intellectuelle et la gestion de la sécurité introduisent de nouvelles demandes pour les schémas de codage. Un codeur ne doit plus seulement être bon en terme de courbe débit-distorsion, il doit également fournir des services supplémentaires.

Ces services concernent, entre autres, la protection du contenu et l'insertion de données cachées. D'une part, la protection du contenu consiste principalement à garantir l'intégrité des données et à masquer le contenu. Les méthodes habituelles utilisées à ces effets sont respectivement le hachage [2] et le chiffrement [3]. D'autre part, l'insertion de données cachées vise soit à protéger les droits de l'image, soit à enrichir le document avec des métadonnées. Les techniques respectives peuvent être le tatouage [4] et la stéganographie.

Masquer le contenu d'une image est le plus souvent effectué via le chiffrement de tout ou partie de l'image. Ce chiffrement peut se faire dans le domaine direct ou le domaine transformé [5]. L'idée est donc d'interdire la visualisation de tout ou partie de l'image en l'absence d'autorisation (la clef de chiffrement).

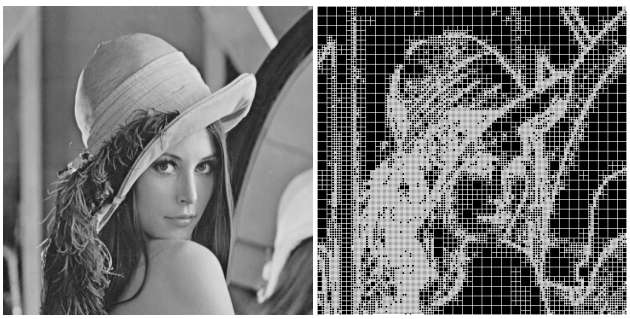
Cet article propose d'utiliser directement une partie du flux binaire généré par le codeur LAR pour garantir la protection du contenu de l'image. Multirésolution, scalable en débit et en distorsion, avec et sans perte, en niveaux de gris ou en couleurs, ce codeur présente des performances au-delà de l'état de l'art.

Le présent document s'articule comme suit : une présentation succinte de la famille de codeurs LAR (section 2) et une analyse détaillée du flot binaire associée (section 3) permettent de définir une méthode de protection du contenu (section 4). Le raisonnement est suivi d'éléments de justification théorique permettant d'évaluer la pertinence de la méthode (section 5). Enfin, quelques résultats expérimentaux valident l'ensemble (section 6).

2 La famille des codeurs LAR

La famille des codeurs LAR (Locally Adaptive Resolution) adapte la résolution locale en fonction de l'activité de l'image. À une luminance localement uniforme correspond une résolution réduite. À l'opposé, une activité locale importante implique une résolution élevée. En outre, le schéma global de codage considère une image I comme étant la superposition de deux composantes : $I = \bar{I} + \tilde{I}$. Ainsi, $\bar{I}$ est une information globale tandis que $\tilde{I}$ représente la texture.

L'image globale est obtenue à partir d'un partitionnement quadtree par blocs carrés en utilisant un gradient morphologique comme critère d'activité. Un simple seuil d'activité sert de principal paramètre. On appellera ce partitionnement partition LAR. À chaque bloc de cette partition est associé la valeur moyenne du bloc. La composition des deux fournit l'image $\bar{I}$. Cette image constituée de plateaux constants de taille variable est nommée image plate (flat LAR).



(a) image originale (b) partition LAR

Figure 1 – *Exemple de partition LAR*

Les différents codeurs LAR utilisant cette idée sont soit plats soit hiérarchiques. Ils diffèrent également par la manière d'encoder la texture, avec des approches prédictives, dans le domaine direct ou transformé. Un panorama complet est disponible dans [6] et [7]. Les résultats en terme de qualité de codage sont probants comme le montrent [8], [9] et [10].

Dans la suite de l'article, nous prendrons comme outil le codeur hiérarchique *Interleaved* S+P[10] comme base de travail. Par simplicité, nous l'appellerons LAR-H. La figure 2 donne un aperçu graphique de son fonctionnement.

La superposition de l'image plate et de la texture se retrouve au niveau du codeur. Une première passe, à gauche sur la figure 2, assure la construction de l'image plate pour un niveau de résolution donnée. La seconde passe consiste à ajouter la texture par niveau, jusqu'au niveau souhaité. Ce procédé permet d'aller graduellement d'un codage avec pertes vers un codage sans perte.

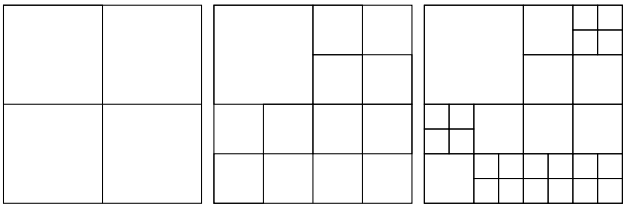
Figure 2 – *Fonctionnement du codeur LAR-H*

3 Description d'un flux binaire LAR-H

Cette section décrit succintement un flux binaire LAR-H. Ce flux comporte trois composantes entrelacées :
– le codage de la partition LAR ;
– le codage de l'image plate ;
– l'ajout de la texture.
La composante nous intéressant particulièrement est la partition LAR. Nous précisons donc la façon de la représenter.

3.1 Codage de la partition LAR

Le codage de la partition quadtree LAR se fait classiquement à partir d'une partition uniforme au plus haut niveau. À chaque passage au niveau inférieur, un bit transmis indique si le bloc est à découper. Ce codage est donc de type conditionnel. Le balayage se fait selon les lignes. La figure 3 donne un exemple pour lequel le flux binaire est le suivant : 0111 pour le passage entre le premier et le deuxième niveau, 0100 1000 0111 pour le passage entre le deuxième et le troisième niveau.



(a) Premier niveau (b) Deuxième niveau (c) Troisième niveau

Figure 3 – *Codage de la partition LAR*

3.2 Structure du flux binaire LAR-H

Le flux binaire LAR-H est scindé en deux. D'une part la partie permettant de reconstituer l'image plate au niveau de résolution souhaité (passe 1) et d'autre part, la partie ajoutant la texture à l'image plate (passe 2). La figure 4 présente le détail du flux. Dans chaque partie, le flux est décomposé en sous-flux, un par niveau. Chaque sous-flux de la passe 1 comprend le flux de la partition LAR (cf. 3.1) et le flux de construction de l'image plate.

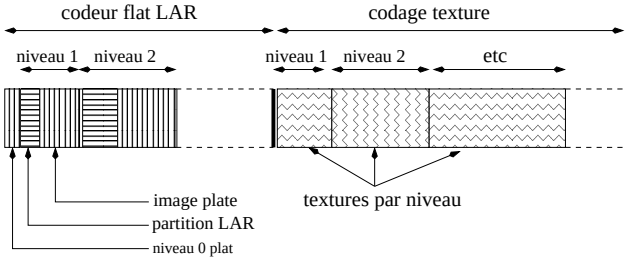
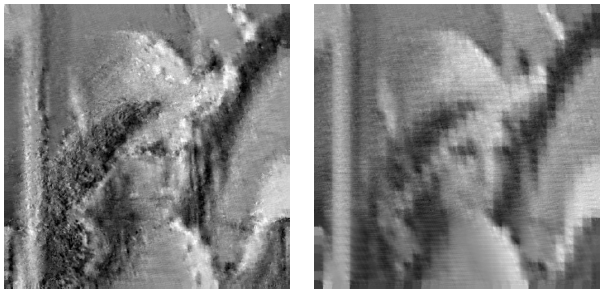


Figure 4 – *Flux binaire LAR-H*

3.3 Influence des erreurs du quadtree sur les images reconstruites

Le flux binaire LAR-H est sensible aux erreurs faites dans le décodage de la partition LAR. La figure 5 présente les effets de deux types d'erreurs différentes. La première est une erreur faite sur la taille d'un bloc au plus haut niveau de la partition LAR. L'effet de l'erreur se propage à toute l'image. La deuxième consiste, en l'absence de la partition LAR, à utiliser comme a priori une partition uniforme.

Deux remarques peuvent être formulées. D'abord, sans une parfaite connaissance de la partition LAR, une seule erreur est interdite, l'image reconstruite est très éloignée de l'image originale. Un schéma de protection par redondance du flux binaire LAR est proposé dans [11]. Ensuite, faire l'hypothèse d'une partition LAR uniforme au niveau le plus haut ne permet pas d'obtenir une image reconstruite de bonne qualité.



(a) erreur sur le premier bloc du niveau le plus haut (16x16)

(b) en considérant une grille uniforme (16x16)

Figure 5 – *Influence des erreurs du quadtree*

4 Un schéma de masquage du contenu

Les sections 2 et 3 présentent un codeur qui décorrèle deux informations : une image plate dérivant de la partition LAR et la texture associée. La section 3.3 montre l'effet d'une connaissance incomplète de la partition LAR sur les images reconstruites.

La partition LAR peut être utilisée pour masquer le contenu de l'image codée, éventuellement par niveau. Le schéma 6 détaille cette méthode. L'idée est d'ôter du flux binaire tout ou partie du flux correspondant à la partition LAR.

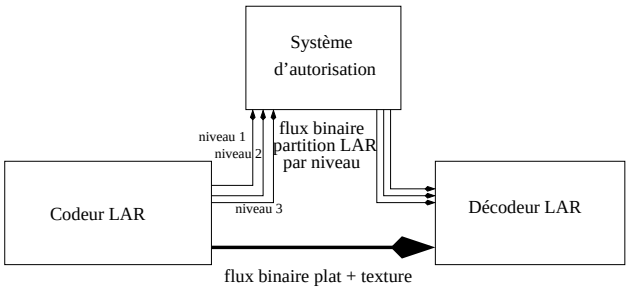


Figure 6 – *Proposition de schéma de masquage d'image par niveaux*

Cette méthode diffère des techniques de chiffrement habituellement proposées. Le schéma de masquage intégré au codeur LAR permet d'autoriser la récupération d'une image à différents niveaux de résolution.

Comment démontrer que la reconstruction de l'image sans connaitre la partition LAR est difficile ? Cette question amène quelques pistes :

- montrer que le nombre de partitions est grand ;
- montrer que, connaissant la partition à un niveau de résolution donné, il est difficile de trouver celle du niveau suivant ;
- montrer que la corrélation restant entre le flux binaire de la partition et les flux binaires de l'image plate et de textures est faible.

La section 5 présente quelques résultats théoriques sur le nombre de partitions LAR (section 5.1) et sur le passage d'un niveau à l'autre de la partition (section 5.2).

5 Étude théorique

5.1 Nombre de partitions LAR

Soit une image I de M lignes et N colonnes avec $S = M \times N$ pixels. Chaque pixel est représenté par q bits et peut donc prendre $Q = 2^q$ valeurs différentes.

$\mathrm{QP}^{[L_T..L_B]}(I)$ est une partition quadtree de l'image I avec des blocs de tailles allant de $L_T \times L_T$ pour le niveau le plus haut (premier niveau) à $L_B \times L_B$ pour le niveau le plus bas (dernier niveau). $\left|\mathrm{QP}^{[L_T..L_B]}(I)\right|$ est le nombre de partitions quadtree par blocs possibles pour une image I.

Soient $l_T = \log_2 L_T$, $l_B = \log_2 L_B$ et $l = l_T - l_B + 1$ le nombre de niveaux dans la partition quadtree, ou partition LAR. On note $\Omega^{[L_T..L_B]}$ le nombre de partitions LAR possibles sur un bloc de taille $L_T \times L_T$. $\omega^{[L_T..L_B]} = \log_2 \Omega^{[L_T..L_B]}$ est donc le nombre de bits nécessaires pour coder cette partition quadtree.
Pour illustrer notre propos, nous considérons une image 256×256 sur 8 bits avec une partition LAR $\text{QP}^{[32..2]}$ sur 5 niveaux.
Le nombre de partitions quadtree $\Omega^{[L_T..L_B]}$ est lié à la fonction récursive Φ comme suit :

$$\begin{aligned} \Phi(0) &= 1 \\ \Phi(n) &= 1 + \Phi^4(n-1) \end{aligned} \quad (1)$$

En fait, $\Omega^{[L_T..L_B]} = \Phi(l-1)$. Le tableau 1 donne les premières valeurs de $\Phi(n)$. Il est à remarquer que $\log_2 \Phi(n)$ est le nombre de bits nécessaires pour représenter la valeur de $\Phi(n)$. Pour $n > 0$, $\Phi(n)$ peut être approché grossièrement par $\Phi_a(n) = 2^{(4^{n-1})}$. Ceci établit un coût de codage du quadtree $C(\text{QP}^{[L_T..L_B]})$ à 4^{l-2} bits par bloc $L_T \times L_T$.

n	$\Phi(n)$	$\log_2 \Phi(n)$	$\Phi_a(n)$
0	1	0	1
1	2	1	2
2	17	4.09	2^4
3	83522	16.35	2^{16}
4	4.86×10^{19}	65.40	2^{64}

Tableau 1 – $\Phi(n)$ *et* $\Phi_a(n)$ *pour les premières valeurs de* n

Pour notre exemple, $\Omega^{[32..2]} = 4.86 \times 10^{19}$ et il faut jusqu'à 65.40 bits pour coder la partition quadtree sur un bloc 32×32.
Le nombre de partitions LAR pour l'image entière I est simplement calculé en considérant que chaque bloc de taille $L_T \times L_T$ contient une partition quadtree $\text{QP}^{[L_T..L_B]}$. $\left|\text{QP}^{[L_T..L_B]}(I)\right|$ est alors donné par :

$$\left|\text{QP}^{[L_T..L_B]}(I)\right| = \Phi(l-1)^{\frac{MN}{L_T^2}} \quad (2)$$

Le coût de codage est souvent exprimé en bit par pixel (bpp). Ce coût est donné par :

$$\frac{\log_2 \Phi(l-1)}{L_T^2} \quad (3)$$

En continuant avec notre exemple, $\left|\text{QP}^{[32..2]}(I)\right| \approx 2^{4186}$. Cela indique que le coût de codage sera au maximum de 4186 bits soit 0.064 bpp.

5.2 Passage d'un niveau à un autre dans la partition LAR

Il s'agit de répondre à la question suivante : connaissant la partition LAR à un niveau donné, sachant le nombre de blocs à décomposer, combien y-a-t'il de partitions LAR de niveau suivant ?
La partition initiale, en blocs de taille maximale $L_T \times L_T$ est toujours la même. Il y a $b = \frac{MN}{L_T^2}$ blocs différents. Parmi ces blocs, d seront découpés en blocs de taille inférieure. Le nombre de partitions LAR de passage, P_{L_T} est donc une combinaison :

$$P_{L_T} = C_b^d \approx 2^{bH_2(\frac{d}{b})}$$

avec $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ l'entropie binaire. La figure 7 présente le tracé de $H_2(p)$. En tenant compte de $0 \leq H_2(x) \leq 1$, de $0 \leq \frac{d}{b} \leq 1$ et $H_2(x)$ maximale pour $x = 0.5$, il vient :

$$0 \leq P_{L_T} \leq 2^b$$

et P_{L_T} maximale pour $d = \frac{b}{2}$. En fait, si zéro ou tous les blocs sont à découper, il n'y a aucune difficulté à retrouver la partition suivante. En revanche, si le nombre de blocs à découper est proche de la moitié des blocs, le nombre de partitions possibles est de l'ordre de 2^b.
En reprenant notre exemple canonique, $b = 512^2/32^2 = 256$ et $0 \leq P_{32} \leq 2^{256}$. Un calcul identique peut également être fait entre deux niveaux quelconques de la partition LAR.

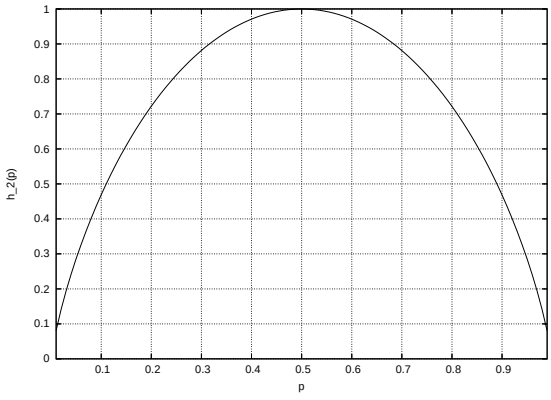


Figure 7 – *Fonction entropie binaire* $H_2(p)$

5.3 Remarques

La section 5.1 montre le grand nombre de partitions LAR possibles sur une image. Deux remarques viennent nuancer ce résultat.
La première concerne le résultat de l'équation (2). Ce résultat repose sur l'hypothèse que les blocs d'une image sont indépendants. Ce qui implique que le coût de codage fourni par (3) est un coût maximal. La présence d'une corrélation entre blocs fera baisser l'entropie du flux binaire généré et donc le coût de codage. L'expérimentation

présentée à la section 6 montre que l'entropie restante est toujours importante.
La deuxième remarque indique que l'espace des images est toujours beaucoup plus grand que l'ensemble des partitions LAR. Dans l'exemple utilisé, chaque pixel est représenté par 8 bits, mais le coût maximum de codage de la partition LAR est de 0.064 bit. Ceci explique que la partition LAR ne peut pas servir d'identifiant unique pour une image, comme clef de hachage par exemple. En effet, nombre d'images différentes partagent la même partition LAR.
La section 5.2 indique que le nombre de partitions d'un niveau au suivant dépend fortement de la proportion de blocs à découper. Si cette proportion est proche de $1/2$, le nombre de partitions est immensément grand. Par contre, pour une proportion éloignée de $1/2$, ce nombre peut être réduit à 1. Valider ce résultat revient à vérifier expérimentalement la proportion de blocs à découper.

6 Experimentations

6.1 Présentation des résultats

Les images utilisées pour l'expérimentation sont `lena`, `airplane`, `baboon`, `goldhill`, `man`, `pepper` et `woman`, toutes de taille 512×512 et codées sur 8 bits. Les partitions LAR sont de type $\text{QP}^{[64..2]}$ soit au moins 6 niveaux dans la pyramide. Dans ce cas, l'entropie maximale pour la partition LAR est de 0.06387 bpp.
La figure 8 montre l'influence du seuil d'activité sur l'entropie de la partition LAR. En effet, le seuil pilote le processus de découpage en quadtree. Plus précisement, lorsque le seuil augmente, le nombre de régions diminue, avec des blocs de taille plus grande. La valeur du seuil est fixée par défaut à 30 (sur une échelle de 256 niveaux de gris).

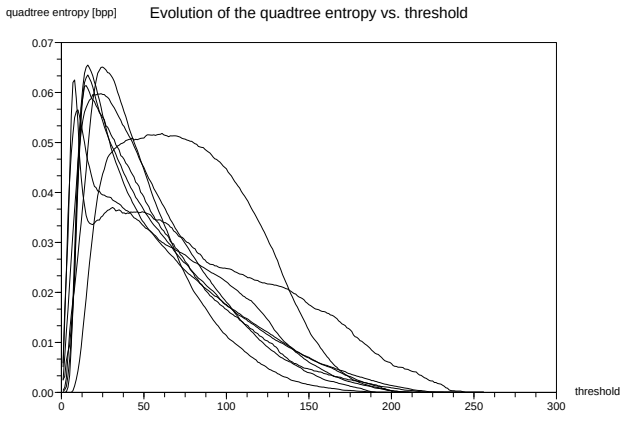


Figure 8 – *Entropie de la partition LAR en fonction du seuil d'activité*

Pour les images de l'expérimentation, l'entropie de la partition LAR reste au-dessus de 0.03 bpp (environ la moitié du coût de codage) pour une large plage de seuil. Avec une telle entropie pour la partition LAR et en prenant une image de taille 512×512, l'entropie de la partition LAR est d'environ 7864 bits pour toute l'image. Ainsi, le nombre de partitions LAR possible est de l'ordre de $2^{7864} \approx 10^{2367}$. Même si décoder un flux binaire ne prenait que $1\mu s$, une attaque de type *brute force* ne serait pas envisageable.
À l'inverse, si la protection de l'image devait durer 100 ans, c'est-à-dire $3.156 \times 10^9 s$, l'entropie que devrait avoir la partition LAR ne devrait être que de 51.5 bits, soit 0.0002 bpp, toujours pour une image 512×512.
La figure 9 montre l'influence du nombre de blocs sur l'entropie de la partition LAR. Le graphe a l'allure de la fonction entropie binaire $H_2(p)$ comme sur la figure 7. Cela démontre bien que l'entropie de la partition LAR est maximale lorsque le nombre de blocs est égal à la moitié du nombre maximal de blocs possible.

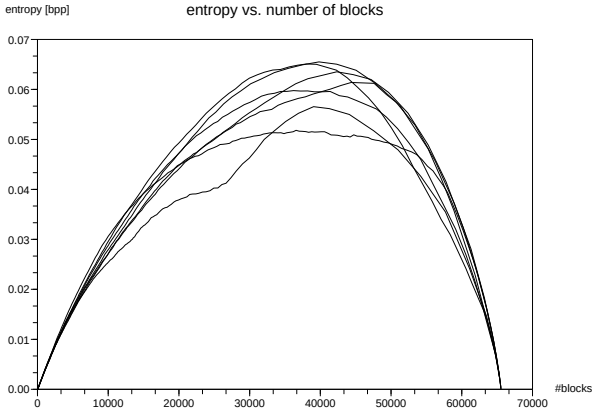


Figure 9 – *Entropie de la partition LAR en fonction du nombre de blocs*

Comme la partition LAR est transmise progressivement (cf. section 3.1), l'entropie de chaque niveau de la partition est calculée et le tableau 2 présente les résultats obtenus lorsque le seuil vaut 30. L'entropie du premier niveau est nulle en l'absence de blocs de taille 64×64 dans la partition LAR. Le second niveau possède une entropie d'environ 50 bits qui peut être suffisante pour fournir une protection raisonnable de l'image. De plus, l'entropie augmente avec la diminution de la taille des blocs, rendant la reconstruction de l'image sans la partition LAR plus difficile. L'image `airplane` fait exception, les 17 bits d'entropie étant insufisants. Dans tous les cas, le niveau le plus haut de la partition LAR peut être décodé, ne fournissant qu'une image de très basse résolution avec une grande distorsion.
Un dernier résultat expérimental concerne le nombre de partitions LAR possibles en passant d'un niveau à l'autre (cf. section 5.2). Le tableau 3 montre le nombre P_{32} de partitions LAR possibles à partir du décodage du niveau de taille 32×32. Mis à part l'image `baboon` qui présente beaucoup de petits blocs (cf. tableau 2), P_{32} est assez élevé

image	64×64	32×32	16×16	8×8	4×4
lena	0	61	656	2916	8886
airplane	17	175	601	1922	7243
baboon	0	0	21	1647	10356
goldhill	0	65	218	2659	12218
man	0	35	466	2824	11014
pepper	0	41	667	3148	8245
woman	0	86	560	2810	9483

Tableau 2 – *Entropie des niveaux de la partition LAR en [bits] pour des images de* 512×512

pour toutes les autres images.

image	P_{32}
lena	2.79×10^{17}
airplane	2.23×10^{51}
baboon	1
goldhill	6.24×10^{18}
man	8.81×10^{9}
pepper	3.69×10^{11}
woman	1.01×10^{25}

Tableau 3 – *Nombre de partitions P_{32} du passage du niveau du haut au suivant*

6.2 Discussion des résultats

La section 5.3 proposait quelques pistes d'expérimentations. Celles-ci mettent en évidence deux choses. La première est que le nombre de partitions LAR possibles pour des images naturelles est très grand. La seconde est que, même en connaissant la partition LAR pour un niveau donné, il reste très difficile, pour ne pas dire impossible, de déterminer la partition LAR pour le niveau suivant. Ainsi, la partition LAR apparaît comme un outil potentiellement utilisable pour effectuer une protection du contenu de l'image, avec un fonctionnement par niveaux. De plus, l'adaptation du seuil d'activité semble être le moyen de choisir le niveau de protection souhaité.

7 Conclusion et perspectives

Cet article présente quelques résultats sur l'intégration dans un codeur d'images fixes avec et sans perte d'un mécanisme de protection de contenu. Ce mécanisme utilise une partie du flux binaire généré par le codeur, sans coût additionnel. Il autorise des niveaux différenciés d'accès au contenu d'images codées avec un outil dont les performances, tant en sans perte que avec pertes, sont au-dela de l'état de l'art.

Le partitionnement LAR autorise à valider d'un point de vue théorique les bonnes propriétés de protection de contenu de la méthode. Les premiers résultats expérimentaux indiquent la faisabilité de l'ensemble. D'autres travaux doivent suivre, notamment la recherche de vulnérabilité de la méthode et la vérification de la faible corrélation entre le flux de la partition LAR et les flux codant l'image plate et de la texture. L'essentiel de la théorie dans ce domaine est inexistant, et l'expérimentation qui lui correspond est lourde.

Une application de ce travail est la mise en œuvre d'un système sécurisé d'archivage pour des images d'art haute résolution. Cette banque numérique d'images fournira un accès à ses fonds de manière sélective, avec différentes qualités d'image. Le projet TSAR[1] (Transfert Sécurisé d'image d'Art haute-Résolution) a pour objectif l'implantation d'un tel système d'archivage.

Références

[1] A. Said. Wavelet based image compression. Rapport technique, Imaging Technology Dept., Hewlett-Packard Labs, 1999.

[2] C. De Roover, C. De Vleeshouwer, F. Lefebvre, et B. Macq. Robust image hashing based on radial variance projections of key-frames. Dans *International Conference on Image Processing*, volume 3, pages II–77–80, 2005.

[3] M. Yang, N. Bourbakis, et S. Li. Data, image and video encryption. *IEEE Potentials*, 23(3) :28–34, Aug.-Sept. 2004.

[4] F. Davoine et S. Pateux, éditeurs. *Tatouage de documents audiovisuels numériques*. Hermés Science Publications, 2004.

[5] S. Lian, J. Sun, et Z. Wang. A novel image encryption scheme based on jpeg encoding. Dans *International Conference on Information Visualisation*, pages 217–220, 2004.

[6] O. Déforges. *Codage d'images par la méthode LAR et méthodologie Adéquation Algorithme Architecture. De la définition des algorithmes de compression au prototypage rapide sur architectures parallèles hétérogènes*. Habilitation à diriger des recherches de l'université de Rennes 1, 2004.

[7] M. Babel. *Compression d'images avec et sans perte par la méthode LAR (Locally Adaptive Resolution)*. Thèse de doctorat, INSA Rennes, 2005.

[8] O. Déforges et J. Ronsin. Region of interest coding for low bit-rate image transmission. Dans *ICME*, volume 1, pages 107–110, july 2000.

[9] M. Babel, O. Déforges, et J. Ronsin. Lossless and lossy minimal redundancy pyramidal decomposition for scalable image compression technique. Dans *ICME*, volume 3, pages 249–252, 2003.

[10] M. Babel, O. Déforges, et J. Ronsin. Interleaved S+P pyramidal decomposition with refined prediction model. Dans *ICIP*, volume 2, pages 750–753, 2005.

[11] M. Babel, B. Parrein, O. Déforges, et N. Normand. Secured and progressive transmission of compressed images on the internet : application to telemedicine. Dans *SPIE 17th annual symposium/ Electronic Imaging Internet - Internet Imaging*, volume 5670, pages 126–136, 2005.

[1] `http ://www.lirmm.fr/tsar`