



CODEBOOK OPTIMAL ET NOUVELLE STRATEGIE DE QUANTIFICATION VECTORIELLE D'IMAGE

M. ANTONINI, M. BARLAUD, P. MATHIEU

Laboratoire de Signaux et Systèmes - CNRS UA 814
41, Bd Napoléon III 06041 Nice Cédex FRANCE

RESUME

Dans cet article, nous proposons une nouvelle méthode d'optimisation d'un codebook combinant l'algorithme LBG (Linde-Buzo-Gray) et la technique de SPLITTING, avec son application en codage d'image par quantification vectorielle.

Un des principaux problèmes rencontrés en quantification vectorielle est de trouver un codebook "globalement optimal", en effet, un codebook sous-optimal engendrera toujours un codage sous-optimal des données. Or, lorsque l'on utilise l'algorithme LBG, l'obtention d'un tel codebook est conditionnée par le choix du codebook initial utilisé; ce choix est prépondérant si l'on ne veut pas converger vers un minimum local de distorsion "pauvre".

Notre méthode permet de modifier les conditions initiales du LBG dans le but de sortir d'un minimum local de distorsion, en effectuant le splitting des vecteurs du codebook qui ont une grande probabilité d'apparition. L'entropie du codebook est alors améliorée (augmentée): une recherche du meilleur compromis entre la distorsion et l'entropie est donc effectuée.

Nous présentons aussi un nouveau schéma de codage hiérarchique d'image qui utilise la quantification vectorielle en considérant une image résiduelle (image originale après soustraction de moyennes locales) dans le domaine transformé.

SUMMARY

In this paper, we propose a new method of codebook design optimization combining the LBG algorithm (Linde-Buzo-Gray) and the SPLITTING technics, with its application in image coding using vector quantization.

One of the most important problems encountered in vector quantization is to find a "globally optimal" codebook. In fact, a suboptimal codebook design will always result in a suboptimal coding of the data. Now, using the LBG algorithm, the resulting codebook is a function of the initial codebook choice; this choice is very important if we don't want to get trapped in a "poor" local minimum of the distorsion.

Our method modify the initial condition of the LBG algorithm in order to go out from a local minimum of the distorsion, by splitting the more frequently used codewords. So, the codebook entropy is improved (increased): a research of the best trade off between distorsion and entropy is then realized.

We present also, a new hierarchical image coding scheme using vector quantization of a residual image (original image minus the space-variant mean) in the transform domain.

I. INTRODUCTION

La compression des images numériques s'avère être une opération essentielle pour des applications de transmission et de stockage du fait de la quantité importante d'information qui décrit une image. Les techniques de compression consistent à coder les images sous une forme plus dense que la simple description numérique et doivent altérer le moins possible l'information contenue dans le message visuel. Parmi toutes les méthodes actuelles de compression, la Quantification Vectorielle (VQ), développée récemment par LINDE, BUZO et GRAY (1980) [1], s'est avérée particulièrement performante en traitement des images. Son principe est de coder une séquence d'échantillons au lieu de coder chacun de ces échantillons individuellement. Le codage s'effectue alors en approximant la séquence à coder par une forme appartenant à un catalogue de formes usuellement désigné dans la littérature par le terme anglais codebook.

Le codebook est l'élément essentiel d'un quantificateur vectoriel. Sa construction est une opération effectuée une fois pour toute, et en dehors de toute opération de codage; la complexité de celle-ci n'affecte donc pas la performance d'un codeur temps réel. Cependant, les méthodes utilisées actuellement pour créer et optimiser le codebook sont basées sur l'algorithme de LINDE, BUZO et GRAY (LBG) qui converge rapidement vers un minimum local de distorsion; en effet, ce minimum dépend des conditions initiales utilisées par l'algorithme. Le codebook produit est alors sous-optimal (ou "localement optimal").

Dans le but d'obtenir un codebook "optimal", Gersho a récemment proposé une méthode combinant le Recuit Simulé ("Simulated Annealing") avec l'algorithme LBG, appelée le SA-GLA [2]. Cette méthode de recuit simulé est basée sur un changement aléatoire de la répartition des vecteurs de la séquence d'apprentissage lors de la classification et s'avère lourde de mise en oeuvre.



Ces observations nous ont motivés pour chercher un moyen d'améliorer le codebook dans le but de s'approcher du minimum global de distorsion. Nous avons retenu une méthode combinant l'algorithme LBG avec la technique de SPLITTING. Cette méthode est présentée en première partie de l'article.

La deuxième partie est consacrée à une nouvelle stratégie de compression d'image qui utilise la quantification vectorielle en considérant une image résiduelle. Elle permet d'effectuer une compression sur les moyennes à coder tout en supprimant les effets de blocs qui apparaissent lors du codage.

II. LE SPLITTING ET L'ALGORITHME LBG

II-1. Introduction

Le codebook est le coeur d'un quantificateur vectoriel. Pour une séquence d'apprentissage donnée et une taille de codebook fixée, la tâche la plus difficile est de construire le codebook contenant la meilleure collection de codewords (mots de codage) qui représente efficacement la grande variété de vecteurs de la source à coder. En effet, un codebook sous-optimal engendra toujours un codage sous-optimal des données.

L'utilisation d'une séquence d'apprentissage comme point de départ (initialisation) d'un algorithme de construction d'un codebook est nécessaire dans la plupart des applications en quantification vectorielle, parce que l'on a pas toujours de connaissance a priori sur les vecteurs de la source à coder. C'est pourquoi, l'algorithme LBG basé sur une répartition inconnue des vecteurs, est très utilisé en quantification vectorielle et a été retenu pour nos applications.

II-2. L'algorithme LBG

Le LBG [1] est un algorithme qui a pour but d'effectuer une opération de classification sur une séquence d'apprentissage pour un codebook initial donné. Deux étapes peuvent étre distinguées:

-Une première étape de classification qui permet de regrouper chaque vecteur de la séquence d'apprentissage autour du vecteur du codebook initial qui lui est le plus représentatif: création d'une partition.

-Une deuxième étape d'optimisation qui permet, en calculant le centroid (centre de gravité) de chaque classe créée lors de la première étape, de modifier au mieux le vecteur du codebook qui la représente.

L'algorithme est itéré avec ce nouveau codebook de façon à obtenir une nouvelle partition. Il converge alors vers une position stable en améliorant à chaque itération, un critère mathématique [1]. Le choix du codebook initial peu bien sûr faire une grande différence dans la détermination du minimum local que l'algorithme atteindra (le minimum global correspondant à un codebook initial possible). La méthode d'optimisation que nous proposons paragraphe II-5 est basée sur la modification de ce codebook initial.

II-3. Performance d'un codebook

Nous présentons ici quelques grandeurs qui servent à caractériser l'information apportée par les vecteurs du codebook.

Considérons une séquence d'apprentissage contenant M vecteurs et un codebook contenant V vecteurs équiprobables (partition optimale), c'est-à-dire ayant chacun une probabilité d'apparition P telle que $P = 1/V$. Nous pouvons alors définir la quantité suivante:

$$C_a = -\log_2 P \quad \text{bits/vecteur} \quad (1)$$

qui représente la capacité du codebook, c'est-à-dire la quantité d'information apportée par un vecteur de ce codebook.

Cependant, ce cas correspond à un codebook bien construit, c'est-à-dire à redondance nulle mais pas à distorsion minimale (la redondance correspondant à la répétition d'information dans le codebook). Or, dans la réalité, les V vecteurs ne sont pas équiprobables mais ont une probabilité d'utilisation égale à P_i ($i=1, \dots, V$) décrivant alors la partition $\{S_i; i=1, \dots, V\}$ pour la séquence d'apprentissage considérée. La quantité C_a devient alors:

$$C_a = -\log_2 P_i \quad \text{bits/vecteur} \quad (2)$$

qui représente la quantité d'information apportée par le vecteur i du codebook (plus un vecteur apparait rarement, plus l'information qu'il apporte est grande). Nous nous intéressons alors à l'entropie H de ce codebook, définie par Shannon, qui représente la quantité d'information moyenne apportée par un vecteur:

$$H = -\sum_{i=1}^V P_i \times \log_2 P_i \quad \text{bits/vecteur} \quad (3)$$

Enfin, la redondance R du codebook peut étre calculée par:

$$R = C_a - H \quad (4)$$

II-4. Critères à optimiser

Le but que l'on cherche à atteindre lorsque l'on fait de la quantification vectorielle est d'obtenir un quantificateur comportant un codebook qui minimise un certain critère de distorsion. De nombreux critères ont été proposés dans la littérature pour la quantification vectorielle [1],[3]. Nous avons retenu, pour notre application, la mesure du carré de l'erreur (norme Euclidienne au carré) qui est la plus simple mesure de distorsion et la plus utilisée pour le codage des formes de signaux.

Soit X un vecteur de la séquence d'apprentissage et Y un vecteur du codebook. La distance entre X et Y est définie par:

$$d(X, Y) = \|X - Y\|^2 = \sum_{k=1}^L (X(k) - Y(k))^2 \quad (5)$$

où L est la dimension des vecteurs.

La distorsion totale du quantificateur pour un codebook donné est alors:

$$D = (1/M) \times \sum_{i=1}^M \left(\min_{Y_j \in \text{Codebook}} d(X_i, Y_j) \right) \quad (6)$$

Lorsque l'on utilise l'algorithme LBG, c'est cette distorsion D que l'on cherche à optimiser. Cependant, la distorsion D minimale fournie par l'algorithme correspond à un minimum local et est fortement conditionnée par le choix du codebook initial utilisé. Ce choix est en effet prépondérant si l'on ne veut pas converger vers un minimum local de distorsion "pauvre" (loin du minimum global [2]). De plus, l'algorithme LBG n'optimise pas la répartition des vecteurs de la séquence d'apprentissage lors de l'étape de classification. Soit, les probabilités d'apparition P_i des vecteurs du codebook (correspondant au nombre de vecteurs de la séquence d'apprentissage dans la classe S_i) ne sont pas optimisées. L'entropie H , définie précédemment (formule (3)), n'est alors pas maximisée et donc, le codebook obtenu est redondant.

La solution que nous avons adoptée pour chercher à améliorer la distorsion et l'entropie s'appuie sur le splitting des vecteurs du codebook correspondant aux centroides des classes les plus peuplées. Contrairement à la méthode de Gersho, la modification de la répartition des vecteurs se fait ici de manière guidée. Notre méthode est combinée avec l'algorithme LBG, elle est présentée dans le paragraphe II-5 suivant.

II-5. L'algorithme de SPLITTING (SPLIT-LBG)

Le principe du splitting en quantification vectorielle est de décomposer un vecteur Y_j en deux autres vecteurs Y_{j+p} et Y_{j-p} où p est un vecteur aléatoire de perturbation.

Cette opération est utilisée dans notre application de façon à modifier le codebook initial utilisé par le LBG en fonction des probabilités P_i d'apparition des vecteurs du codebook, obtenues lors d'une première classification. Un vecteur i du codebook est splitté si sa probabilité P_i est grande.

Le splitting des vecteurs du codebook qui sont le plus souvent utilisés permet de mieux équilibrer la répartition des vecteurs de la séquence d'apprentissage lors de la classification et donc, d'optimiser la partition du codebook. En effet, le splitting est "bon" si une classe S_i contenant V_i vecteurs est divisée en deux classes contenant chacune $V_i/2$ vecteurs: un vecteur qui a une grande probabilité d'apparition est alors remplacé par deux vecteurs équiprobables et de probabilité d'apparition plus faible. Ceci entraîne une augmentation de l'entropie H .

L'algorithme est le suivant:

(0) INITIALISATION: les données sont: un codebook A_0^{init} contenant V vecteurs, l'entropie H_0^{init} du codebook correspondant à la partition $P(A_0^{init}) = \{S_0^i; i=1, \dots, V\}$ (probabilités d'apparition des vecteurs du codebook: $P_i; i=1, \dots, V$), la distorsion D_0^{init} de A_0^{init} et une séquence d'apprentissage contenant M vecteurs.

$m = 0$ (nombre de splittings effectués), $N \leq V/2$ (nombre maximum de splittings à effectuer).

(1) $m = m + 1$

(2) SPLITTING du vecteur du codebook A_{m-1}^{init} qui a la P_i la plus grande.

SUPPRESSION du vecteur du codebook A_{m-1}^{init} qui a la P_j la plus faible.

A chaque vecteur splitté correspond un vecteur supprimé de façon à conserver le même nombre de vecteurs dans le codebook et donc la même limite C_a pour la capacité.

Un LBG LOCAL (seuil d'arrêt ϵ) est effectué dans la classe S_{m-1}^i splittée, avec un codebook local constitué par la paire de vecteurs résultant du splitting. La séquence d'apprentissage locale est constituée par les vecteurs appartenant à la classe S_{m-1}^i . Cette classification locale possède les propriétés d'optimisation d'un LBG.

⇒ Modification du codebook initial A_0^{init} (donc des conditions initiales du LBG effectué à l'étape (3)):

$A_0^{init} \rightarrow A_m^{init}$ et $D_0^{init} \rightarrow D_m^{init}$ avec $D_m^{init} > D_0^{init}$ en général

⇒ Amélioration de l'entropie:

$H_0^{init} \rightarrow H_m^{init}$ avec $H_m^{init} > H_0^{init}$, c'est-à-dire obtention d'une meilleure répartition des vecteurs: $P(A_m^{init})$

(3) Exécution de l'algorithme LBG sur les M vecteurs de la séquence d'apprentissage avec A_m^{init} comme codebook initial, $P(A_m^{init})$ comme partition initiale et ϵ comme seuil d'arrêt.

⇒ Optimisation de la distorsion:

$D_m^{init} \rightarrow D_m^{LBG}$ avec $D_m^{LBG} < D_0^{LBG}$



⇒ Evolution de l'entropie:

$$H_{m,init} \rightarrow H_{m,LBG} \text{ avec } H_{m,LBG} > H_{0,LBG}$$

La distorsion $D_{m,init}$ du nouveau codebook splitté est minimisée par l'algorithme LBG. Compte tenu de ce qui a été dit précédemment, la distorsion minimale $D_{m,LBG}$ obtenue après LBG est en général inférieure à la distorsion $D_{0,LBG}$ du codebook initial non splitté, après LBG avec le même seuil ϵ' (Cf. figure 1). De même, l'entropie $H_{m,LBG}$ est supérieure à l'entropie $H_{0,LBG}$ pour ϵ' donné.

(4) Si $m < N$ Alors retourner en (1) Sinon arrêter l'algorithme.
REMARQUES:

-La distorsion $D_{m,LBG}$ est fonction du nombre m de splittings; le test réel d'arrêt de l'algorithme devrait être en fait conditionné par la valeur de cette distorsion.

-L'algorithme permet, pour un codebook A_0^{init} donné, la recherche du nombre m_{opt} de splittings permettant une minimisation de la distorsion D et une amélioration de l'entropie H telles que:

$$D_{m_{opt},LBG} \leq D_m^{LBG} \quad \forall m \in [0, N]$$

et $H_{m_{opt},LBG} \rightarrow Ca$ (Capacité du codebook Cf. formule (1))

$D_{m_{opt},LBG}$ correspond alors au minimum minimorum que l'on peut obtenir en effectuant m_{opt} splittings sur le codebook initial A_0^{init} choisi, et compte tenu des valeurs retenues pour les vecteurs de perturbation p .

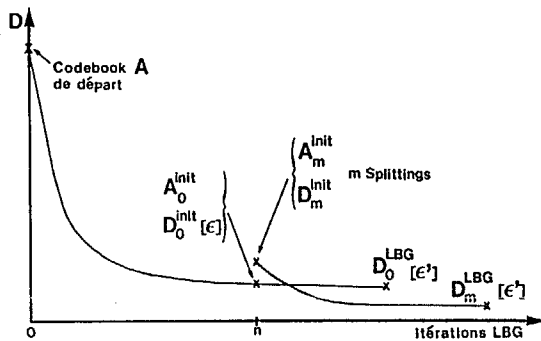


Figure 1

III. QUANTIFICATION VECTORIELLE D'UNE IMAGE RESIDUELLE

Un problème important rencontré en quantification vectorielle d'image est celui des effets de blocs résultant du raccord de deux vecteurs de quantification lors du codage. Ces effets sont dus à des discontinuités de moyennes locales qui s'avèrent très gênantes visuellement.

Nous présentons ici une nouvelle méthode hiérarchique de compression d'image qui utilise la quantification vectorielle en considérant une image résiduelle. Cette méthode a l'avantage d'effectuer une compression sur la moyenne à coder tout en diminuant les effets de blocs. L'image résiduelle $s_r(m,n)$ est définie par WOODS [11] de la manière suivante: soit $s(m,n)$ une image et $E[s(m,n)]$ l'image des moyennes calculées par blocs (moyennes locales) sur $s(m,n)$. Alors,

$$s_r(m,n) = s(m,n) - E[s(m,n)] \quad (7)$$

Or, cette opération entraîne sur l'image $s_r(m,n)$ des effets de blocs gênant visuellement à cause des discontinuités qui existent dans l'image $E[s(m,n)]$. Pour y remédier, nous avons apporté une solution qui consiste à effectuer au préalable un lissage de l'image $E[s(m,n)]$ au moyen d'un filtre passe-bas. Le filtre que nous avons utilisé est un pavé de dimension 9×9 pixels contenant les valeurs 1. On obtient alors, après ce prétraitement, une image résiduelle, contenant peu d'énergie dans les basses fréquences et dépourvue d'effets de blocs.

Dans un premier temps, les blocs de moyenne de l'image $E[s(m,n)]$ non lissée sont codés par un quantificateur scalaire simple. Une première opération de compression est ainsi effectuée sur les moyennes. Ensuite, le problème de codage 2D de l'image résiduelle $s_r(m,n)$, définie précédemment, de dimension (M lignes, N colonnes) est ramené à un codage de N signaux 1D décorrélés [7]. En effet, une FFT est effectuée sur l'image miroir résiduelle et chacune des colonnes de cette image transformée est quantifiée vectoriellement. Lors du codage, la somme des carrés des erreurs entre les vecteurs des colonnes et ceux du codebook est minimisée [8]; c'est-à-dire, nous cherchons à minimiser la distorsion D définie en (4). Les vecteurs des colonnes étant considérés centrés (image résiduelle centrée par blocs), le codebook utilisé par la quantification vectorielle est un codebook de formes. Seul le gain des vecteurs (variance), est considéré lors du codage et donc codé et transmis au même titre que l'index du vecteur du codebook et que la moyenne par blocs.

Les schémas de principe du système de codage/décodage sont présentés figures (2) et (3).

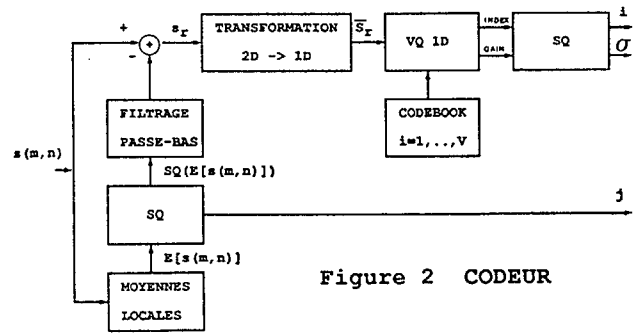


Figure 2 CODEUR

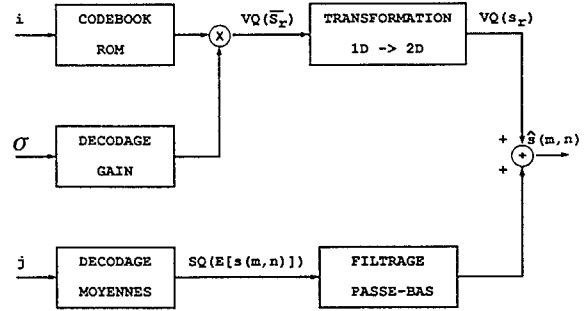


Figure 3 DECODEUR

Dans ce système de codage, les moyennes par blocs sont d'abord quantifiées de manière scalaire, lissées, puis soustraites de l'image $s(m,n)$. Ainsi, toutes les erreurs de quantification des moyennes sont incorporées dans l'image résiduelle $s_r(m,n)$ et sont donc codées. Contrairement à la méthode de Gray M/RVQ (Mean/Residual Vector Quantizer) [9], les effets de blocs sont supprimés ici grâce à l'utilisation du filtre passe-bas.

Cette méthode de compression a été peu développée dans cet article. En effet, elle est généralisée par la méthode de transformation en ondelette et développée dans un article que nous proposons aussi à ce colloque GRETSI 89 dans le cadre codage et compression d'image [10].

IV. RESULTATS EXPERIMENTAUX

IV-1. Splitting

Pour tester la méthode de splitting combinée avec l'algorithme LBG dans le but de créer un codebook, nous devons construire un quantificateur vectoriel. Ce quantificateur servira alors pour coder des images appartenant ou non à la séquence d'apprentissage. Les images que nous utilisons sont de dimension (256 lignes, 256 colonnes) et sont codées sur 8 bits. Le quantificateur vectoriel est présenté dans le paragraphe III; cependant, celui qui est utilisé ici ne considère pas l'image résiduelle mais l'image d'origine $s(m,n)$. Lors du codage, les moyennes des vecteurs 1D à coder sont donc prises en compte par le quantificateur.

Le codebook de départ, itération 0 de l'algorithme LBG (Cf. Figures 1 et 6) est obtenu en calculant le centroid de chacune des colonnes d'une image "Transformée". Ce codebook permet de créer par l'algorithme LBG traditionnel, avec un seuil de convergence $\epsilon = 2 \times 10^{-3}$, le codebook initial de l'algorithme de splitting A_0^{init} (Cf. Figures 1 et 6). Les vecteurs de la séquence d'apprentissage utilisée appartiennent à l'image qui a servi pour créer le codebook de départ à l'itération 0 du LBG. Cette séquence est constituée de 4096 vecteurs. Les codebooks contiennent 256 vecteurs 1D (formes) de dimension 16.

Les courbes (4) et (5) sont des courbes typiques qui montrent l'évolution de la distorsion D (Cf. formule (6)) et de l'entropie H (Cf. formule (3)) en fonction du nombre m de splittings. Ces courbes ont été obtenues avec un codebook et une séquence d'apprentissage créés à partir de l'image LENA ("The lady with hat"). Nous pouvons vérifier sur la courbe (4) que le nombre de minima locaux est important; ceci rend donc difficile la recherche du minimum minimorum de distorsion. Dans nos essais expérimentaux, comme le montre l'algorithme (Cf. paragraphe II-5), nous avons fixé un nombre maximum N de splittings à effectuer de façon à avoir l'allure générale des courbes $D_m^{LBG} = f(m)$ et $H_m^{LBG} = g(m)$; aucun test d'arrêt automatique sur D_m^{LBG} a été programmé. La valeur du seuil d'arrêt ϵ' utilisée par le SPLIT-LBG est égale à 1×10^{-3} . Nous pouvons constater que le minimum de distorsion se trouve pour $m=40$ splittings.

La courbe (6) montre l'évolution de la distorsion en fonction du nombre d'itérations effectuées par l'algorithme LBG et compte tenu des codebooks initiaux utilisés (avec ou sans splitting), le seuil de convergence est $\epsilon' = 1 \times 10^{-3}$ quelque soit le codebook initial utilisé. Elle correspond au cas réel pour $m=40$ splittings dont un schéma explicatif a été donné figure 1.

Le tableau figure 7 donne les résultats récapitulatifs obtenus lors du codage de différentes images appartenant ou non aux séquences d'apprentissage utilisées et fonctions des valeurs m du nombre de



splittings effectués. Le rapport signal à bruit considéré est le Peak SNR (PSNR exprimé en dB) [2]. Nous constatons que le SPLIT-LBG a permis, dans nos essais expérimentaux, une amélioration maximale au niveau PSNR d'environ 0,4 dB par rapport au LBG utilisé seul. Ces résultats numériques ne reflètent pas les espoirs que cette méthode laissait espérer, mais sont très proches de ceux que Gersho a obtenu avec le SA-GLA [2]. Toutefois, la qualité visuelle des images codées est améliorée de façon appréciable.

IV-2. Image résiduelle

Le quantificateur vectoriel utilisé est celui décrit dans le paragraphe III. Le codebook a été créé à partir d'une séquence d'apprentissage constituée par des vecteurs appartenant à LENA.

Les moyennes locales sont calculées sur des blocs d'image de 8x8 pixels, puis codées sur 8 bits. L'image des moyennes codées $E[s(m,n)]$ est alors lissée par un filtre passe-bas carré, contenant des 1 et de dimension 9x9; cependant, ce filtre introduit de la haute fréquence parasite du fait de sa coupure trop brutale dans le domaine spatial. La figure (8) montre l'image résiduelle non codée (moyennes filtrées) qui a été rendue visible en prenant les valeurs absolues des pixels puis leurs inverses vidéo.

Enfin, l'image de la figure (9) correspond à l'image $s(m,n)$ codée (PSNR=28,30 dB) avec un rapport de compression $C=8$ compte tenu du codage de la moyenne soit, un taux de bits transmis $R=1$ bit/pixel (8 bits de codage sont affectés au codage de l'index et autant pour celui de la variance).

V. CONCLUSION

Le splitting combiné avec l'algorithme LBG est une méthode efficace qui permet d'améliorer la distorsion et l'entropie d'un codebook utilisé par un quantificateur vectoriel. Cette amélioration a été constatée pour différentes séquences d'apprentissage. Le gain obtenu au niveau PSNR reste cependant assez faible bien que la qualité visuelle des images codées est meilleure.

La méthode de compression utilisant une image résiduelle est généralisée par la transformation en ondelette qui fait l'objet d'un article que nous présentons aussi à ce colloque dans le cadre de la compression d'image [10].

REFERENCES

- [1] Y. LINDE, A. BUZO, R.M. GRAY "An Algorithm for Vector Quantizer Design" IEEE Trans. on com., Vol. 28, n°1, January 1980, pp.84-95.
- [2] J. VAISEY, A. GERSHO "Simulated Annealing and Codebook Design" IEEE, ICASSP-88, April 11-14, New-York.
- [3] R.M. GRAY "Vector Quantization" IEEE ASSP Magazine, April 1984.
- [4] A.E. EL GAMAL, L.A. HEMACHANDRA, I. SHPERLING and V.K. WEI "Using Simulated Annealing to Design Good Codes" IEEE Trans. on Inform. Theory, vol. IT-33, n°1 pp.116-123, Jan. 1987.
- [5] S. KIRKPATRICK, C.D. GELATT, M.P. VECCHI "Optimization by Simulated Annealing" SCIENCE Vol.220, 13 May 1983.
- [6] N. METROPOLIS, A.W. ROSENBLUTH, M.N. ROSENBLUTH, A.H. TELLER, and E. TELLER, "Equation of State Calculations by Fast Computing Machines" J. Chem. phys., Vol. 21, June 1953.
- [7] L. BLANC-FÉRAUD, M. BARLAUD, P. MATHIEU "Ringling Reduction in Image Restoration Using Mirror Images and Adaptive Kalman Filtering" IEEE, ICASSP-88, April 11-14, New-York.
- [8] M. BARLAUD, L. BLANC-FÉRAUD, P. MATHIEU, J. MENEZ, M. ANTONINI "2D Linear Predictive Image Coding With Vector Quantization", EUSIPCO, Grenoble 1988.
- [9] N.M. NASRABADI, R.A. KING "Image Coding Using Vector Quantization: A Review". IEEE Trans. on Com. Vol.36 No 8, August 1988.
- [10] P. MATHIEU, M. BARLAUD, M. ANTONINI "Compression d'Images par Transformée en Ondelette" GRETSI, Codage et compression d'image, Juan-les-Pins 12-16 juin 1989.
- [11] F.C. JENG, J.W. WOODS "Inhomogeneous Gaussian Image Models for Estimation and Restoration", IEEE ASSP Vol.36 No 8, August 1988.

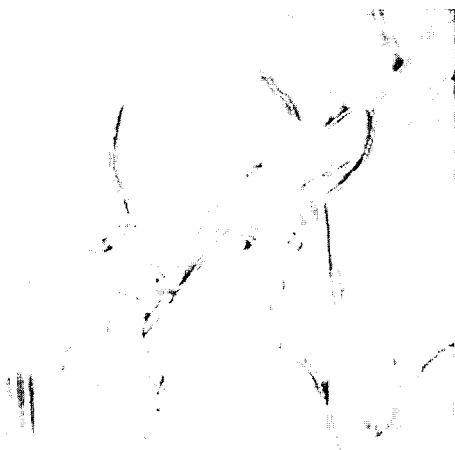


Figure 8

Image résiduelle (inverse vidéo)

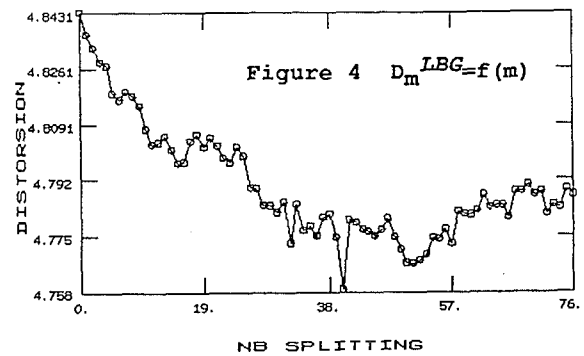


Figure 4 $D_m^{LBG}=f(m)$

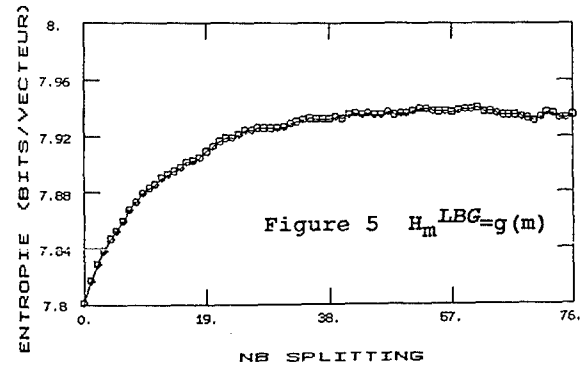


Figure 5 $H_m^{LBG}=g(m)$

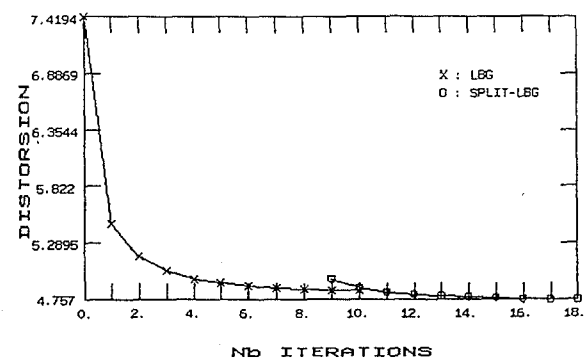


Figure 6 $D=fct(itérations\ LBG)$

PSNR (dB)

IMAGE \ TS	Codebook A_0^{LBG}		Codebook splitté A_m^{LBG}	
	LENA $m=0\ \epsilon'=10^{-3}$	CAMERAMAN $m=0\ \epsilon'=10^{-3}$	LENA $m=40\ \epsilon'=10^{-3}$	CAMERAMAN $m=24\ \epsilon'=10^{-3}$
LENA	29, 65	29, 12	30, 03	29, 21
CAMERAMAN	24, 22	25, 23	24, 43	25, 49

TS = Séquence d'apprentissage

Figure 7 Tableau récapitulatif



Figure 9 Image codée

PSNR=28,30 dB R=1 Bit/pixel