



Quantification Vectorielle Algébrique Contrainte en Débit: Un Nouvel Algorithme de Quantification au sens Débit-Distorsion

Patrice ONNO et Christine GUILLEMOT

C.C.E.T.T

4, rue du Clos Courtel

35 512 Cesson-Sévigné FRANCE

E-mail: onno@ccett.fr guillemo@ccett.fr

RÉSUMÉ

Ce papier décrit une nouvelle procédure de quantification vectorielle mettant en jeu à la fois la distorsion et le coût de codage d'un vecteur. Le signal de décomposition est issu de paquets d'ondelettes, ce qui permet de s'adapter de façon fine aux statistiques du signal. Les quantificateurs utilisés sont des quantificateurs vectoriels sur réseau contraints en débit. L'algorithme se décompose en 4 étapes. Les deux premières étapes font partie d'un processus itératif comme décrit dans [4] et permettent d'obtenir pour un débit donné le meilleur réseau de quantification et la transformation optimale. En plus de ces deux étapes classiques, deux étapes supplémentaires tirent avantage des propriétés des réseaux de quantification pour rechercher le point du réseau qui minimise non plus la distorsion seule mais une fonction objective faisant intervenir aussi le coût de codage du vecteur. Le calcul de ce dernier fait intervenir notamment une nouvelle partition de l'espace des vecteurs en constituant un code produit à trois composantes. Dans le plan débit-distorsion, l'introduction de cette nouvelle étape de quantification permet d'améliorer le rapport signal à bruit d'environ 0.5-0.6 dB pour un même débit.

ABSTRACT

This paper describes an image coding scheme using lattice vector quantizers where the emphasis is put on a new lattice vector quantization approach. The adaptive signal decomposition uses wavelet packets which allow to best match the decomposition to the signal nonstationary characteristics. The quantizers used here are data rate constrained lattice vector quantizers. A classical algorithm based on two nested optimization processes allowing to jointly optimize transformation and quantization is first performed. The algorithm is based on 4 processes. The two first steps are part of an iterative process as described in [4] and allow for a given bit rate to obtain the optimum lattice and decomposition tree. Two additional steps that take advantage of the lattice properties allow to search the lattice point providing the best compromise rate-distortion. For encoding a vector, a new partition of the vectors space is introduced by dividing the code into three components. In the rate distortion plane, this new quantization procedure brings a significant improvement both in SNR (about 0.5-0.6 dB) and in visual quality.

1 Introduction

La procédure de codage utilisée ici est guidée par l'idée de contrôler le débit alloué aux images. Pour atteindre de bonnes performances le choix s'est porté sur des quantificateurs vectoriels sur réseau et contraints en débit que l'on nommera EC-LVQ pour "Entropy Constrained Lattice Vector Quantization". La quantification vectorielle sur réseau est l'extension au cas multidimensionnel de la quantification scalaire uniforme et présente des avantages par rapport aux méthodes classiques de quantification vectorielle de type LBG [1]. Entre autres, elle implique des temps de calcul beaucoup moins importants pour des performances équivalentes. Couplée à des codeurs entropiques, la quantification vectorielle sur réseau a montré des résultats prometteurs pour la compression d'images fixes [2] ou animées [3]. Le débit de chaque vecteur est évalué au moyen de codes à longueur variable et fait intervenir un code produit. Dans la littérature [2], la procédure utilisée pour coder un vecteur consiste à décomposer ce code en deux parties, correspondant à une représentation du vecteur en coordonnées polaires (norme et phase). Ici, une nouvelle représentation en trois composantes est développée et permet de coder plus efficacement certaines directions privilégiées de l'espace.

Pour atteindre un débit fixe, on utilise dans un premier temps un algorithme basé sur deux optimisations successives

[4]. La première permet de trouver le meilleur quantificateur pour la sous-bande considérée en utilisant l'algorithme rapide de quantification proposée par [5]. La deuxième fait ensuite la recherche de la structure de décomposition optimale qui minimise la distorsion totale pour un débit donné.

Pendant, suivant la norme, le processus de quantification développé par [5] peut conduire au codage de points sur des hyperpyramides (isonormes L_1) ou des hypersphères (isonormes L_2) comportant de fortes densités de points. Ceci entraîne pour les vecteurs situés sur ces hyperpyramides ou hypersphères une longueur importante du code de phase. Il peut donc être préférable de choisir un point dont la distorsion engendrée est un peu plus forte, mais dont le coût de codage est nettement inférieur. Ceci fait l'objet de la nouvelle procédure que l'on introduit pour obtenir le meilleur compromis entre le débit et la distorsion pour chaque vecteur. On peut alors faire une analogie entre la technique présentée ici et la quantification vectorielle contrainte en entropie (ECVQ) [6]. En ce qui concerne l'ECVQ, le dictionnaire est construit de telle sorte que l'on obtiendra après quantification de la séquence d'entraînement la distorsion la plus faible pour une entropie donnée. La différence majeure se situe pour la quantification vectorielle sur réseau au niveau de l'opération de quantification elle-même où la contrainte de débit est introduite. Notons que l'EC-LVQ n'a pas besoin de transmettre de dictionnaire pour le décodage des vecteurs.



2 Quantification vectorielle sur réseau avec recherche du quantificateur optimal

Différents réseaux, D_n et E_n peuvent être considérés et les sous-bandes sont quantifiées suivant l'algorithme rapide de [5] qui recherche dans le réseau le point le plus proche d'un point arbitraire. On prend alors un ensemble de N quantificateurs possibles par sous-bande qui définissent la taille de la maille du réseau. A chaque quantificateur de cet ensemble est associé un couple débit-distorsion. Le débit de chaque vecteur est calculé en utilisant des codes à longueur variable. Le vecteur est représenté en coordonnées polaires et un code est attribué à sa norme, puis un autre code à sa phase. Dans notre cas, un code formé de trois parties comme décrit dans le paragraphe (3.3) est introduit

Partant de l'ensemble des quantificateurs, il est alors possible de construire un nuage de points dans le plan débit-distorsion pour l'ensemble des sous-bandes et de rechercher les quantificateurs donnant la configuration optimale: c'est à dire la combinaison de quantificateurs qui fournira globalement la distorsion minimale D pour un débit R proche de R_d . En d'autres termes, le problème consiste à minimiser l'expression $(D + \lambda R)$ où λ est un multiplicateur de Lagrange. L'algorithme utilisé s'inspire de [4] et repose sur les deux étapes suivantes imbriquées et itérées pour différentes valeurs de λ jusqu'à obtention de l'égalité $R = R_d$.

- étape 1: La première étape consiste à rechercher parmi l'ensemble des quantificateurs celui qui minimise la fonction: $d_i + \lambda r_i$. Elle fournit alors pour la $i^{ème}$ sous-bande le quantificateur qui minimise la distorsion d_i pour un débit r_i . En appliquant la même stratégie aux autres sous-bandes, on obtiendra le minimum de distorsion globale pour cette pente.
- étape 2: La seconde étape consiste ensuite à optimiser l'arbre de décomposition en adoptant une technique d'élagage qui fournira l'arbre de décomposition idéal pour l'image considérée; elle fournira en sortie un débit R pour une pente λ donné.

Dans le plan débit distorsion, illustré sur la figure (3), les étapes précédentes fournissent le point Ac qui donne le minimum de distorsion pour un débit R_d avec une pente optimale λ_{opt} .

3 Quantification Contrainte en Débit

3.1 L'idée

La comparaison des densités de points sur les hypersphères ou sur les hyperpyramides [2] nous amène à constater que le nombre de points sur une isonorme n'est pas une fonction croissante avec le rayon. Autrement dit, certaines sphères peu peuplées ont des sphères voisines beaucoup plus denses. Il est à noter aussi qu'en norme L_1 , la concentration de points devient très vite importante à mesure que m augmente. Il peut donc exister dans le voisinage des points obtenus par l'opération de quantification classique des points présentant une distorsion plus forte, mais un débit plus faible. L'objectif de

l'algorithme développé est donc de réaliser une quantification vectorielle qui sera contrainte en débit de manière à projeter les vecteurs sur les points du réseau dont le compromis débit distorsion est le meilleur. D'un point de vue algorithmique, l'opération de quantification au voisinage du vecteur préselectionné cherchera à minimiser la fonction suivante:

$$f(d_v, r_v) = d_v + \lambda_{opt} r_v \quad (1)$$

où d_v est la distorsion du vecteur, r_v son débit et λ_{opt} la pente optimale obtenue par l'algorithme de débit-distorsion précédent. Le voisinage n'a pas besoin d'être très étendu et l'on constate que la prise en compte des premiers voisins est généralement suffisante. En effet, si l'on étend le voisinage, le gain en débit n'est pas assez significatif pour contrer la distorsion importante engendrée dans ce cas.

3.2 L'algorithme de quantification

La figure (1) montre un exemple de configuration dans le cas d'un réseau de dimension 2. Partant du point V à quan-

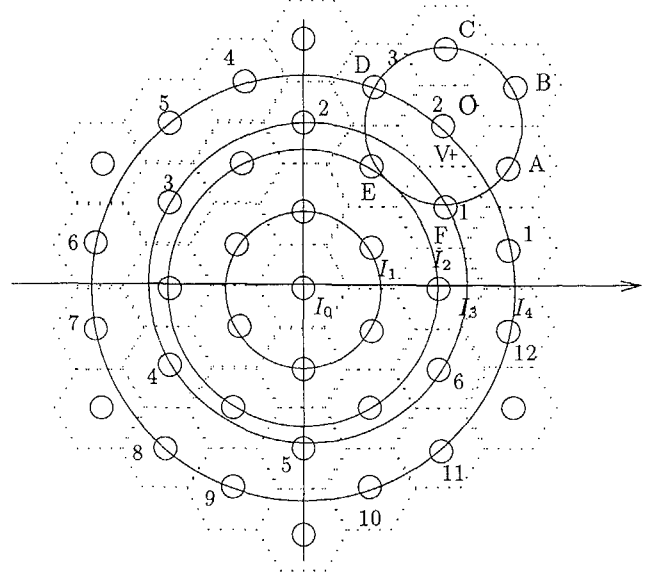


FIG. 1 - Zoom sur le réseau A_2 où les points A à F sont les plus proches voisins de O

tifier, l'algorithme de recherche du meilleur quantificateur représenté par les étapes 1 et 2 du paragraphe (2) permet dans un premier temps de régler le maillage du réseau et donc d'ajuster la taille de la cellule de Voronoï représentée par la surface des hexagones. Reprenant l'algorithme décrit par [5], le point V va être projeté après quantification sur le point le plus proche du réseau. Ainsi le point résultant sera le point O. Ensuite affecter un débit binaire à ce vecteur, consistera à coder la norme I_4 de ce point O et à coder son index qui est dans le cas présent 2.

La stratégie de l'EC-LVQ qui consiste à quantifier suivant un compromis débit-distorsion chaque vecteur doit tout d'abord calculer la fonction objective $d_v + \lambda_{opt} r_v$ pour tous les voisins du point O. Les six voisins (points A,...,F) qui appartiennent à des isonormes dont les densités de points varient, auront alors des codes de "phase" de longueur différente. Si la fonction est minimale pour un des voisins de

O, ce nouveau point dans le réseau sera retenu comme la quantification du vecteur V.

Il est donc intéressant pour tous les vecteurs quantifiés de savoir si dans leur voisinage il n'existe pas un vecteur qui appartient à une de ces isonormes peu denses. Le vecteur présélectionné dans la première étape minimise seulement la distorsion, au contraire, le nouveau vecteur issu de cette recherche aura donc un meilleur compromis débit-distorsion.

Le schéma synoptique de l'algorithme général est présenté sur la figure (2). Celui-ci peut être décomposé en deux blocs différents: l'un reprend l'algorithme existant recherchant le facteur d'échelle et l'arbre de décomposition pour un débit donné, l'autre encadré en pointillé schématise les nouveautés apportées.

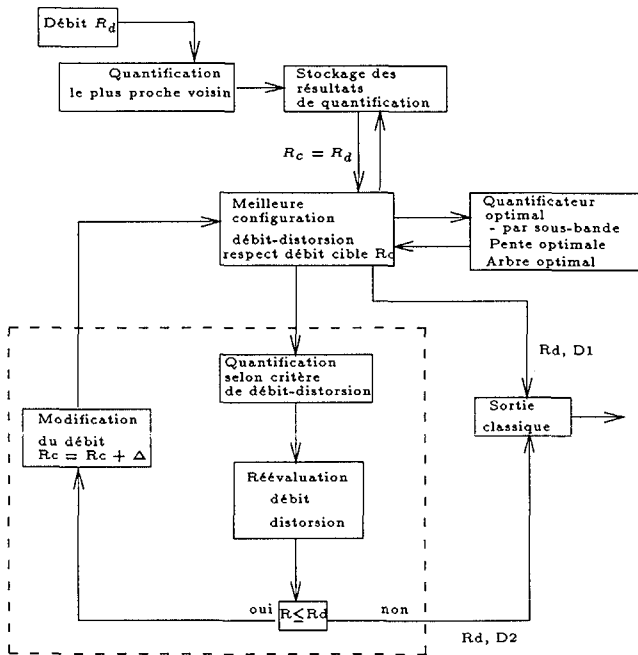


FIG. 2 - Schéma de l'algorithme de quantification. La partie encadré en pointillé englobe les nouvelles fonctions de quantification et de régulation

Ce nouvel algorithme peut être décrit en quatre étapes différentes:

- étape 1: identique à l'étape 1 du paragraphe (2).
- étape 2: identique à l'étape 2 avec rebouclage sur l'étape 1 comme au paragraphe (2).
- étape 3: Pour $\lambda = \lambda_{opt}$, on requantifie chaque sous-bande en reprojétant chaque vecteur sur le point minimisant la fonction: $d_v + \lambda_{opt} r_v$. Le nouveau débit obtenu est souvent inférieur et la distorsion un peu plus élevée.
- étape 4: La dernière étape permet de réajuster le débit cible en l'incrémentant légèrement: $R_c = R_c + \Delta$. Avec ce nouveau débit, on recherche à nouveau la configuration idéale au sens de la quantification classique. Ceci permet d'ajuster au mieux les quantificateurs et de trouver une nouvelle pente λ_{opt} . On boucle ainsi sur les quatre étapes précédentes de telle manière que le débit final voulu R_d soit atteint.

Globalement appliquées sur toutes les sous-bandes, les trois premières étapes permettent sur la figure (3) de déplacer

dans le plan débit-distorsion le point Ac vers le point Ao. Après quelques itérations de l'étape 4, l'algorithme de régulation converge vers le point Do qui respecte le débit R_d . Il fournit de cette manière un meilleur résultat que celui donné par la quantification classique représentée par le point Ac.

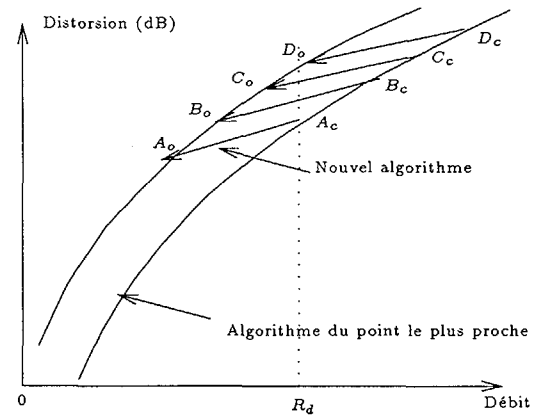


FIG. 3 - Apport de la nouvelle méthode

3.3 Code produit à trois composantes

L'attribution d'un code à chaque vecteur pour la quantification vectorielle sur réseau se fait généralement grâce à un code "produit". Ce code est ainsi dénommé parce qu'il code de façon indépendante la norme d'un vecteur et ensuite l'index sur cette isonorme. Lorsque les composantes des vecteurs sont très corrélées, il est important de pouvoir séparer les vecteurs d'une même isonorme en différents sous ensembles. En effet, d'un point de vue statistique, les vecteurs suivent dans ce cas des orientations privilégiées dans l'espace des index et il est judicieux d'en tirer partie pour les coder de façon efficace. Pour prendre en compte ces orientations, on introduit ici le *type* d'un vecteur. Le *type* d'un vecteur est défini par le nombre de composantes identiques qu'il comporte. Cette définition engendre ainsi un nombre de *types* différents suivant la dimension du réseau considéré, mais elle est indépendante de la norme choisie. Tous les vecteurs sont obtenus par la suite en réalisant toutes les combinaisons possibles des composantes du *type* considéré. Le but à atteindre est de prendre en considération les vecteurs dont les orientations sont les plus probables et ceci à des fins de réduction de débit.

Pour un réseau D_4 , dont la définition est la suivante:

$$D_4 = \{(x_1, x_2, \dots, x_4) \in Z^4 \mid x_1 + \dots + x_4 = k; k : \text{paire}\}, \quad (2)$$

on dénombre cinq *types* différents définis de la manière suivante:

- Le premier *type* que l'on définit est un vecteur dont les quatre composantes sont identiques.
- Le deuxième *type* correspond au cas où il y a deux couples de composantes identiques.
- Le troisième est défini par deux composantes identiques et deux différentes.
- Le quatrième possède trois composantes identiques et une composante différente.



- Enfin le cinquième et dernier *type* a des composantes toutes différentes.

L'attribution d'un code pour un vecteur se fera donc à partir d'un assemblage de trois codes. On appliquera dans un premier temps, un code à longueur variable pour coder sa norme, puis un autre code toujours à longueur variable pour coder son *type*. Le troisième code permet de coder l'index relatif au *type* du vecteur, ceci revient à affecter un numéro à chaque vecteur appartenant à un des cinq sous-ensembles. Le codage de cet index se fera ici en supposant l'équiprobabilité des index. A ce stade, l'hypothèse d'équiprobabilité est justifiée par le fait que toutes les permutations ont souvent une probabilité d'apparition équivalente. L'énumération des *types* est donnée dans le tableau (1) pour un réseau D_4 de norme L_1 . Ce tableau montre la faible proportion des *types* 0, 1 et 3. Or, lorsque les composantes des vecteurs sont corrélées, la probabilité d'apparition des *types* 0, 1 et 3 est très forte. De ce point de vue il est donc plus intéressant, une fois la norme codée, de coder le *type* du vecteur, puis son index que sa phase sur une isonorme (hypersphères ou hyperpyramides) dans le cas d'un code produit à deux composantes.

Norme	type 0	type 1	type 2	type 3	type 4	total
0	1	0	0	0	0	1
2	0	12	12	8	0	32
4	2	18	108	16	48	192
6	0	36	276	32	264	608
8	2	42	588	32	744	1408
10	0	60	924	56	1680	2720
12	2	66	1428	56	3120	4672
14	0	84	1980	72	5256	7392
16	2	90	2652	80	8184	11008
18	0	108	3396	96	12048	15648
20	2	114	4284	96	16944	21440

TAB. 1 - Tableau donnant le nombre de vecteurs pour chaque *type* sur un réseau D_4 de norme L_1

4 Résultats

Cet algorithme a été mis en œuvre dans un schéma de codage vidéo. La figure (4) montrent les résultats obtenus sur la première image codée en intra. Dans ce codeur, l'image est traitée en réalisant une décomposition sur 3 niveaux soit en ondelettes, soit en paquets d'ondelettes. Dans les deux cas, les filtres utilisés sont des filtres de longueur 8 synthétisés de telle manière qu'ils maximisent le gain de codage. Trois techniques de quantification différentes sont alors réalisées et les courbes montrent la supériorité de l'EC-LVQ par rapport à la LVQ ou l'algorithme de Zero Tree. Les résultats obtenus sur l'image "Mobile & Calendar" montrent une amélioration, à la fois en termes de SNR (environ 0.5-0.6 dB) et de qualité visuelle. A noter que l'algorithme du Zero Tree est appliqué aux sous-bandes issues d'une décomposition en ondelettes. De meilleurs résultats pourraient être obtenus en adaptant la technique de Zero Tree à une structure de décomposition en paquets d'ondelettes. Par ailleurs, les filtres utilisés ne sont peut-être pas optimaux pour une quantification reposant sur l'algorithme de Zero Tree. Enfin, l'utilisation d'un code produit à trois composantes est intéressante surtout à très bas débit où le gain par rapport à un code à deux composantes se situe autour de 0.2 dB. Celui-ci devient insignifiant pour des débits plus élevés.

5 Conclusion

Tenant compte à la fois du débit et de la distorsion de chaque vecteur, le nouveau procédé de quantification vectorielle sur réseau (EC-LVQ) présente l'avantage de quantifier les vecteurs de chaque sous-bande avec le meilleur compromis débit-distorsion. La pente optimale permettant d'ajuster le compromis est calculée dans un premier temps suivant une approche classique de quantification où l'on recherche le point le plus proche. Même si le nouveau vecteur obtenu par l'EC-LVQ présente l'inconvénient d'avoir une distorsion plus forte, le débit final est plus faible. Couplée à un codage en trois composantes des vecteurs, l'application de cette nouvelle méthode de quantification incluant une méthode de régulation de débit, permet de montrer pour un même débit global, un gain en qualité non négligeable.

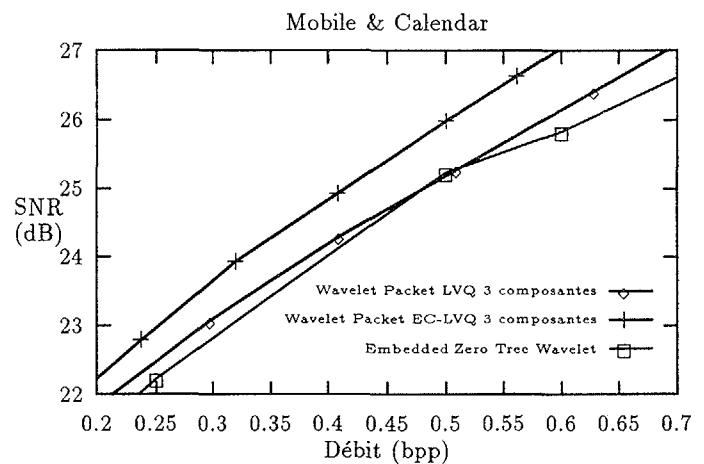


FIG. 4 - Résultats de codage

Références

- [1] Y. Linde, A. Buzo, and R.M. Gray. "An algorithm for vector quantizer design". *IEEE Trans. on Communications*, 28, January 1980.
- [2] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, and P. Mathieu. "Pyramidal lattice vector quantization for multiscale image coding". *IEEE Trans. on Image Processing*, July 1994.
- [3] C. Guillemot, F. Lallauet, and P. Onno. "Towards hybrid spatio-temporal segmentation/waveform coding for high compression of video signals". In *Workshop on Image Analysis and Synthesis in Image Coding*, Berlin, October 1994.
- [4] K. Ramchandran and M. Vetterli. "Best wavelet packet bases in a rate-distortion sense". *IEEE Transactions on Image Processing*, 2(2):160-175, April 1993.
- [5] J. H. Conway and N. J. A. Sloane. "Fast quantizing and decoding algorithms for lattice quantizers and codes". *IEEE Trans. on Information Theory*, 28(2), March 1982.
- [6] P. A. Chou, T. Lookabaugh, and R.M. Gray. "Entropy-constrained vector quantization". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(1), January 1989.