

Méthode de projection non-linéaire rapide pour la quantification vectorielle

P. Mathieu, D. Lebedeff, M. Barlaud

*Laboratoire I3S Ura CNRS
Université de Nice -Sophia Antipolis
Bât 4 SPI, 250 ave. A. Einstein, 06560 VALBONNE (FRANCE)*

L'augmentation du volume des transmissions numériques impose des méthodes de compression de plus en plus efficaces et rapides.

La quantification vectorielle a depuis longtemps montré sa supériorité sur la quantification scalaire, la complexité étant cependant augmentée. L'utilisation des réseaux algébriques permet de contourner cette difficulté, mais ils sont mal adaptés aux sources non-uniformes.

Nous proposons d'associer la vitesse de quantification des réseaux algébriques, à un traitement non-linéaire optimisé par un algorithme de type LBG. On ne fait plus ici l'hypothèse de quantification haute résolution, mais uniquement d'une répartition symétrique des vecteurs.

INTRODUCTION

L'augmentation du volume des transmissions numériques impose des méthodes de compression de plus en plus efficaces et rapides. La quantification vectorielle a depuis longtemps montré sa supériorité sur la quantification scalaire, la complexité étant cependant augmentée.

Notre étude a été entreprise pour quantifier à bas débit du signal brut SAR. Ce signal suit de près une loi gaussienne d'écart-type lentement variable. Il est très faiblement corrélé et donc peut être considéré comme indépendant et identiquement distribué. Des algorithmes de quantification scalaire [1, 2] puis vectoriel [3, 4, 5] ont été proposés. Ils utilisent soit l'algorithme de Linde, Buzo et Gray (LBG) [6], soit les réseaux algébriques sans traitement non-linéaire, soit plusieurs sous-réseaux dilatés de manières différentes [5].

Nous proposons d'associer la vitesse de quantification du réseau algébrique, à un traitement non-linéaire lui-même optimisé par un algorithme de type LBG. On ne fait plus ici l'hypothèse de quantification haute résolution, mais uniquement d'une répartition symétrique des vecteurs.

I. LE DICTIONNAIRE

1. LBG

Le dictionnaire optimal est obtenu par l'algorithme LBG qui à partir d'une séquence d'apprentissage place de manière optimale les vecteurs dans l'espace. L'algorithme de génération du

The increasing volume of transmission implies more and more efficient and fast compression techniques.

Vector quantization always outperforms scalar quantization, but it suffers from the exponential growth in encoding complexity. Lattice vector quantizers are highly structured, and hence allow a fast encoding procedure. Unfortunately, they are not adapted to non-uniform distributions.

We propose in this paper to combine the simplicity of the lattice quantizers with a non-linear processing, optimized by a training mode algorithm. In this approach, the high resolution approximation is not necessary; we just suppose the spherical symmetry of vector distribution.

dictionnaire est long mais fait de manière unique pour une classe de signaux (dimension des vecteurs, densité de probabilité, nombre de vecteurs dans le dictionnaire). C'est surtout au niveau de l'étape de quantification que la lenteur de l'algorithme empêche l'utilisation de dictionnaires de dimensions élevées. En effet l'absence de structuration interne, oblige à un scrutation complète du dictionnaire: c'est cette étape que nous proposons d'améliorer.

2. Réseaux algébriques

Les réseaux algébriques sont parfaitement structurés. Il s'agit d'une collection de sommets portés par une combinaison linéaire à coefficients entiers de vecteurs de base. On obtient, avec un choix judicieux de vecteurs de base, des réseaux réguliers qui optimisent le remplissage de l'espace tels que les réseaux A_2 , D_4 et E_8 pour les dimensions 2, 4 et 8 [7]. Ces réseaux tronqués par une hypersphère peuvent servir de dictionnaire. Leur grande structuration permet une quantification très rapide et surtout de durée indépendante du nombre de vecteurs dans le dictionnaire.

Utiliser un réseau algébrique pour la quantification vectorielle se met en oeuvre de la manière suivante [8]:

- Choix du débit souhaité R_s
- Choix de la dimension des vecteurs (contraintes dues à la taille des dictionnaires)
- Choix de l'énergie de troncature E_t (conséquence des 2 choix précédents)
- Choix d'un coefficient d'homothétie pour adapter de manière optimale le réseau à la dynamique de la source (compromis entre bruit granulaire et bruit de saturation).

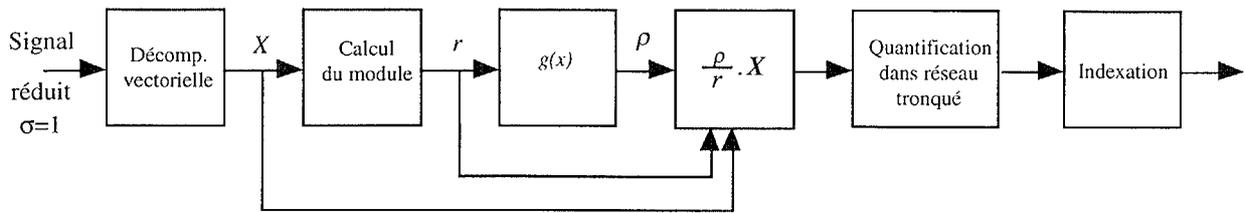


Figure n°1. Schéma de principe de la méthode proposée.

Par l'utilisation d'une méthode non-linéaire, nous voulons ajouter la possibilité de dilater le réseau de manière différente au centre et à la périphérie. C'est une technique classique en scalaire pour la quantification des sources non uniformes, jusqu'ici peu étudiée en vectoriel du fait d'un manque de généralité.

3. Le réseau à dilatation non-linéaire.

Le signal brut SAR se comporte comme un signal gaussien iid à moyenne nulle. On a donc une distribution de probabilité conjointe qui admet une symétrie de révolution: aucune direction n'est privilégiée.

Les réseaux algébriques ne sont pas aussi parfaits, mais si l'on considère des réseaux de dimensions suffisamment élevée, le nombre d'axes de symétrie augmente et l'on approche la symétrie de révolution.

Il suffit alors d'adapter radialement les réseaux algébriques.

II- LA MÉTHODE PROPOSÉE

1. Le codage

Le signal est découpé en blocs sur lesquels l'écart-type σ est quasi-constant. Chaque bloc est alors normalisé ($\sigma = 1$).

Pour chaque vecteur X de ce bloc, on calcule le module r , et on lui applique le traitement non-linéaire, la fonction g , pour obtenir ρ . C'est le vecteur $\frac{\rho}{r} \cdot X$ qui est quantifié dans le réseau choisi.

Le décodage s'effectue simplement en inversant chacune des étapes.

2. L'optimisation de la fonction g

Pour de hauts débits, la taille réduite des cellules de Voronoi permet l'hypothèse de haute résolution et donne une expression analytique approchée des différentes grandeurs, en particulier de la distorsion.

Pour un faible débit on ne peut plus faire cette hypothèse, il n'est donc plus possible d'obtenir les expressions analytiques. On va alors utiliser une séquence d'apprentissage pour optimiser le dictionnaire.

a) Approximation de la fonction

C'est la fonction inverse de g , $g^{-1}(x)$ qui peut être le plus facilement approchée de manière polynomiale. Le degré de ce polynôme dépend de la précision souhaitée et de la courbure de la fonction g . Si le réseau comporte peu de niveaux d'énergie, une approximation linéaire s'avère suffisante. Au delà de 4 couches d'énergie, il convient d'utiliser des fonctions quadratiques ou cubiques.

$$r = a_1 \rho$$

$$r = a_1 \rho + a_2 \rho^2$$

$$r = a_1 \rho + a_2 \rho^2 + a_3 \rho^3$$

C'est par rapport aux paramètres a_1, a_2 et a_3 que ce fait l'optimisation.

b) Structure du réseau

Dans une optimisation de dictionnaire avec séquence d'apprentissage (type LBG), on distingue deux étapes: La classification des vecteurs de la séquence au moyen d'un dictionnaire initial, puis une mise à jour du dictionnaire en fonction de la classification obtenue. Ces deux étapes sont itérées jusqu'à convergence d'un critère.

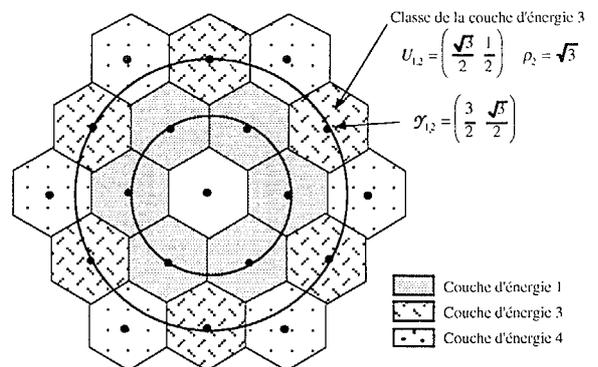
Les classes sont formées de vecteurs $X_{i,j,k}$ associés à un vecteur $Y_{j,k}$ du dictionnaire, le représentant de la classe. Si en général il n'y a pas d'ordre dans un dictionnaire optimisé, il n'en va pas de même ici car la structure du réseau algébrique est partiellement conservée lors de la dilatation.

Dans le réseau, on peut regrouper les classes de même énergie en couches. Dans une même couche k de rayon ρ_k , chaque vecteur $\mathcal{Y}_{j,k}$ du réseau peut-être séparé en $\mathcal{Y}_{j,k} = \rho_k U_{j,k}$ où j est l'indice de la classe dans la couche k . $U_{j,k}$ est un vecteur unitaire: Il représente l'orientation du vecteur $\mathcal{Y}_{j,k}$. Les $X_{i,j,k}$ sont donc les vecteurs de la séquence d'apprentissage regroupés dans la classe associée au vecteur $\mathcal{Y}_{j,k}$ de la k ème couche. Il faut noter que les vecteurs du dictionnaire $Y_{j,k}$ qui quantifient les vecteurs de la

source $X_{i,j,k}$ se déduisent de $\mathcal{Y}_{j,k}$ par la relation

$$\begin{cases} \mathcal{Y}_{j,k} = \rho_k U_{j,k} \\ Y_{j,k} = r_k U_{j,k} \end{cases}$$

et donc $Y_{j,k} = \frac{r_k}{\rho_k} \mathcal{Y}_{j,k}$

Figure n°2. Structure du réseau A_2
Les cellules de Voronoi sont hexagonales

On note dans la suite $I_{j,k}$ le nombre de vecteurs $X_{i,j,k}$ dans la classe j de la couche k , J_k le nombre de classes dans la couche k , et K le nombre de couches dans le réseau, ce qui représente aussi le nombre de niveaux d'énergie.

c) Optimisation

Le critère de qualité retenu est la distorsion que l'on estime par :

$$D = \sum_{k=1}^K \sum_{j=1}^{J_k} \sum_{i=1}^{I_{j,k}} (X_{i,j,k} - Y_{j,k})^T (X_{i,j,k} - Y_{j,k})$$

$$= \sum_{k=1}^K \sum_{j=1}^{J_k} \sum_{i=1}^{I_{j,k}} (X_{i,j,k} - r_k U_{j,k})^T (X_{i,j,k} - r_k U_{j,k})$$

Pour optimiser, il faut écrire que cette distorsion est minimale pour les paramètres a_n . On a ainsi 1, 2 ou 3 équations de la forme: $\frac{\partial D}{\partial a_n} = 0$ pour $n=1, 2$ ou 3 .

Dans le cas linéaire on trouve de manière simple:

$$a_1 = \frac{\sum_{k=1}^K \sum_{j=1}^{J_k} \sum_{i=1}^{I_{j,k}} \mathcal{Y}_{j,k}^T X_{i,j,k}}{\sum_{k=1}^K \sum_{j=1}^{J_k} I_{j,k} \rho_k^2}$$

on remarque que les vecteurs de la séquence d'apprentissage sont projetés sur les vecteurs du réseau.

Dans le cas quadratique il faut résoudre

$$\begin{cases} a_2 \sum_{i,j,k} \rho_k^4 + a_1 \sum_{i,j,k} \rho_k^3 = \sum_{i,j,k} \rho_k^2 U_{j,k}^T X_{j,k} \\ a_2 \sum_{i,j,k} \rho_k^3 + a_1 \sum_{i,j,k} \rho_k^2 = \sum_{i,j,k} \rho_k U_{j,k}^T X_{j,k} \end{cases}$$

ce qui s'écrit facilement mais que l'on ne détaillera pas ici.

La fonction $g(y)$ est évidente dans le cas linéaire $g(y) = \frac{1}{a_1} y$

et à peine plus compliquée dans le cas quadratique

$$g(y) = \frac{-a_1 + \sqrt{a_1^2 + 4a_2 y}}{2a_2}$$

la forme de la fonction $g^{-1}(x)$.

Dans le cas cubique les équations à résoudre ont la même forme que celles du système quadratique, par contre la fonction g a une expression plus complexe. Un exemple est présenté figure n°3.

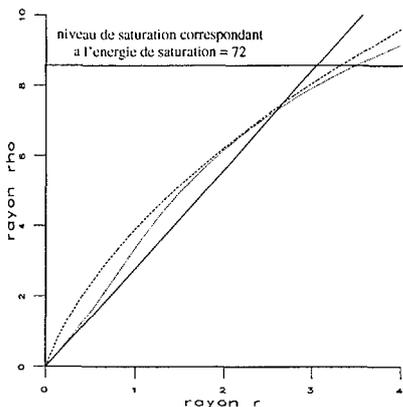


Figure n°3 . Fonctions g dans le cas du réseau A_2 pour 4 bits par échantillon

On remarque qu'aux faibles énergies, la fonction suit l'approximation linéaire et qu'au contraire pour les hautes énergies, elle suit l'approximation quadratique.

On dispose ainsi d'un algorithme pour optimiser la fonction g .

III- RÉSULTATS EXPÉRIMENTAUX

Les essais ont été effectués sur des séquences d'apprentissage gaussiennes de grande taille. Les réseaux mis en œuvre sont le A_2 , le D_4 et le E_8 .

La qualité est estimée par le SNR en variance et le débit R estimé pour un codage naturel: $R = \log_2 N$ avec N le nombre de vecteurs du réseau tronqué.

On compare les résultats obtenus entre la quantification vectorielle algébrique linéaire optimale qui sert de référence, la technique LBG standard et la quantification vectorielle algébrique avec projection non-linéaire optimale.

Les coefficients optimaux sont reportés dans les table 1, 2 et 3, ainsi que les valeurs du SNR obtenues dans le cas linéaire. On vérifie que la qualité augmente avec la dimension. Du fait de l'apprentissage, l'estimation de ces coefficients est très longue pour les dimensions et débits élevés. Par exemple, en E_8 pour 2 bits/échantillon il y a 56881 vecteurs dans le dictionnaire. Il ne nous a donc pas été possible de fournir de résultats pour les hauts débits autres que dans le réseau A_2 .

Cette étape d'apprentissage, indépendante de la quantification, permet d'obtenir des coefficients utilisables pour tous les signaux gaussiens réduits. Avec le même jeu de coefficients, les quantifications ultérieures sont à la fois adaptées et rapides.

L'examen des courbes 4 et 5 montre que pour les faibles débits (< 2 bits/échantillon) les trois méthodes donnent sensiblement les mêmes résultats. Pour les débits compris entre 2 et 3,5 bits par échantillon, on remarque que le traitement non-linéaire apporte un faible gain. En effet le nombre de couches d'énergie impliquées est encore faible. Pour les débits supérieurs, le gain par rapport au traitement linéaire devient nettement plus significatif du fait de l'augmentation du nombre de couches d'énergie.

Les performances de la méthode non-linéaire, ou linéaire à bas débits, sont proches de la méthode optimale LBG, pour une complexité bien moindre. On ne pourra cependant jamais obtenir un SNR égal, car la méthode rapide n'adapte pas l'orientation des vecteurs du dictionnaire.

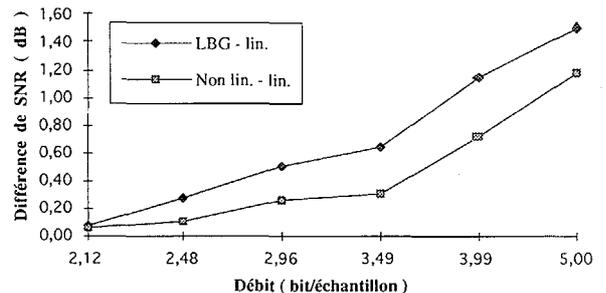


Figure n° 4. Comparaison des SNR entre la méthode LBG et non-linéaire par rapport au cas linéaire optimal, réseau A_2

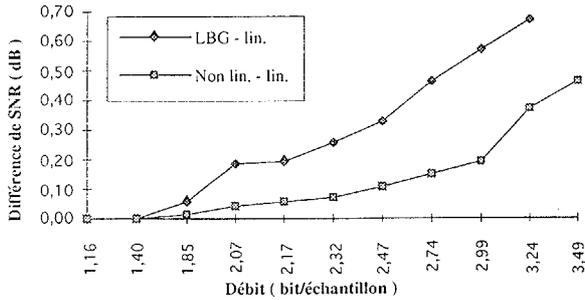


Figure n°5. Comparaison des SNR entre la méthode LBG et non-linéaire par rapport au cas linéaire optimal, réseau D₄

CONCLUSION

Une nouvelle méthode de quantification vectorielle rapide pour la compression des sources non-uniformes a été présentée. L'étude est basée sur la mise en œuvre de réseaux algébriques, qui ont l'avantage de quantifier très rapidement, mais sont mal adaptés aux sources non-uniformes. Pour améliorer leurs performances, nous proposons d'effectuer une transformation non-linéaire sur le module des vecteurs.

Cette technique, particulièrement bien adaptée aux signaux gaussiens qui présentent une symétrie de révolution, est une généralisation du "companding scalaire" et peut s'appliquer à n'importe quel type de signaux. L'optimisation est effectuée de manière paramétrique au moyen d'une séquence d'apprentissage caractéristique des signaux à quantifier.

Le traitement non-linéaire est d'autant plus efficace que le diamètre de l'hypersphère de troncature est important. Dans ce cas, les performances s'approchent de celles obtenues par la quantification vectorielle de type LBG.

A débit identique, le diamètre de la troncature augmente avec la dimension; on peut donc supposer de meilleurs résultats à des dimensions élevées. Il sera donc intéressant de modifier l'algorithme d'optimisation pour les dictionnaires de taille importante afin de faciliter son implantation sur calculateur.

RÉFÉRENCES

- [1] J.C. Curlander, R.N. McDonough, "Synthetic Aperture Radar" A Wiley-interscience publication, 1991.
- [2] R. Kwok and W.T.K. Johnson, "Block Adaptive Quantization of Magellan SAR Data", IEEE Trans. on Geoscience and Remote Sensing, pp.375-383, Jul. 1989.
- [3] D. Lebedeff, P. Mathieu, M. Barlaud, C. Lambert-Nebout, P. Bellemain, "Adaptive Vector Quantization for Raw SAR Data", IEEE ICASSP, Vol. 4, pp 2511-2514, Detroit, May 1995.
- [4] J.M. Moureaux, P. Gauthier, M. Barlaud and P. Bellemain, "Vector Quantization of Raw SAR data", IEEE ICASSP, Vol. 5, pp 189-192, Adelaïde, April 1994.
- [5] D.G. Jeong, J.D. Gibson, "Uniform and Piecewise Uniform Lattice Vector Quantization for Memoryless Gaussian and Laplacian Sources", IEEE Trans. on Information Theory, vol. 39, PP 786-803, Sept. 1989.
- [6] A. Gersho, R.M. Gray, "Vector Quantization and Signal Compression", Boston: Kluwer Academic Publishers, 1990.
- [7] J.H. Conway, N.J.A. Sloane, "Sphere Packings, Lattices and Groups", Springer-Verlag, 1988.
- [8] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, P. Mathieu, "Pyramidal Lattice Vector Quantization for Multiscale Image Coding", IEEE Trans. on Image Processing, vol. 3, n°4, PP 367-381, July. 1994.

Débit	R	2,12	2,48	2,96	3,49	3,99	5,00
Toncature	Et	4	7	16	36	72	279
Linéaire	a1	0,998	0,818	0,626	0,498	0,357	0,188
	SNR	10,15	11,99	14,45	17,16	19,81	25,08
Non linéaire	a1	0,781	0,539	0,445	0,354	0,360	0,207
	a2*100	12,39	11,54	5,76	-3,28	-3,55	-1,28
	a3*1000				10,30	4,80	0,81

Table n° 1. Coefficients des fonctions polynomiales, réseau A₂

Débit	R	1,16	1,46	1,85	2,08	2,16	2,32	2,47	2,74	2,98	3,24	3,49
Troncature	Et	2	4	8	10	12	16	18	28	40	56	78
Linéaire	a1	1,259	1,108	0,903	0,789	0,752	0,693	0,637	0,547	0,477	0,414	0,358
	SNR	5,52	6,78	9,09	10,33	10,83	11,64	12,42	13,87	15,16	16,46	17,77
Non linéaire	a1	•	1,016	0,695	0,550	0,509	0,420	0,383	0,282	0,200	0,481	0,392
	a2*100	•	4,87	8,93	8,46	8,34	8,19	6,93	5,84	4,89	-6,18	-4,05
	a3*1000	•									7,82	4,60

Table n° 2. Coefficients des fonctions polynomiales, réseau D₄

Débit	R	0,99	1,40	1,64	1,84	1,98
Troncature	Et	2	4	6	8	10
Linéaire	a1	1,664	1,359	1,194	1,077	0,995
	SNR	4,95	7,17	8,49	9,50	10,30

Table n° 3. Coefficients des fonctions linéaires, réseau E₈