



# QUANTIFICATION VECTORIELLE ORDONNÉE

G.Menez, J.Menez

E.Lançon, C.Raimondo

UNSA - I3S  
250, Av Albert Einstein  
06560 Valbonne

CER IBM  
06610 La Gaude

## RÉSUMÉ

Cet article présente une méthode de quantification vectorielle "rapide" dont l'objectif est la simplification de la recherche de l'élément (vecteur) du répertoire le plus représentatif de l'élément à quantifier, au sens d'un critère donné. Il s'agit d'une quantification vectorielle avec recherche organisée (OVQ), et basée sur la notion de distance entre les vecteurs du répertoire initial pour en organiser la recherche.

## ABSTRACT

This paper presents a "fast" vector quantization technique that aims to simplify the nearest reference vector codebook search of the incoming vector, according to a given criterion. It is called OVQ, for Organized Vector Quantization, because it is based on the introduction of a distance notion between codebook elements. Such a distance is used to organize the codebook and hence, to improve the search convergence.

## 1 Introduction

En s'imposant comme un élément clef en compression numérique de la parole et de l'image, la quantification vectorielle a fait l'objet d'une attention toute particulière au cours de la dernière décennie. Son utilisation s'est trouvée, dans un même temps, grandement favorisée par la très sensible montée en puissance et en capacité de stockage des supports de calculs (processeurs de signaux, calculateurs embarqués, ...). Bien qu'offrant des performances souvent remarquables, le traitement originel, mettant en œuvre un dictionnaire stochastique non structuré, induit un trop grand nombre de calculs pour que son utilisation se généralise encore aujourd'hui.

Pour pallier cet inconvénient, différents types de quantificateurs vectoriels, économes en terme de complexité, ont été proposés depuis. Parmi ceux-ci, il faut citer les méthodes reposant sur l'élaboration de dictionnaires structurés [2].

L'objectif de l'étude présentée dans cette communication est de montrer qu'il est possible de réduire, de façon significative, la charge de calculs requise par un quantificateur vectoriel sans mémoire à dictionnaire non structuré obtenu par apprentissage statistique (Algorithme LBG [1]). Notre méthode repose sur une technique d'ordonnement du dictionnaire permettant d'éviter le balayage systématique de la totalité (approche "full search") des vecteurs de ce dernier, pour la recherche de chaque vecteur représentatif [4] [5].

## 2 Constitution du dictionnaire

Dans un premier temps de la méthode, on détermine le dictionnaire  $C = \{C_1, C_2, \dots, C_L\}$  où  $C_i \in \mathbb{R}^N$  en supposant que  $N$  et  $L$  sont pré-déterminés. L'algorithme dit LBG [1] [3], généralement adopté, permet de résoudre ce problème.

Une fois le dictionnaire constitué, l'obtention d'un quantificateur optimal d'une suite de vecteurs à coder, nécessite

pour chacun d'eux le calcul de la distance le séparant de tous les vecteurs références du dictionnaire. Le vecteur sélectionné, appelé **vecteur représentatif**, pour quantifier le vecteur à coder, est celui qui présente la plus petite distance euclidienne avec ce dernier.

Ce type de recherche, qualifié de "full-search (FS)", nécessite un volume de calcul important, le plus souvent prohibitif pour les applications considérées.

## 3 Recherche Organisée (OVQ)

L'idée de base de la technique proposée dans cet article vise à diminuer très sensiblement cette complexité. Pour ce faire, nous introduisons la notion de **distance entre les vecteurs références** du dictionnaire initial afin de pouvoir organiser la recherche du vecteur représentatif.

### 3.1 Description de l'algorithme

Dans un premier temps on crée une relation d'ordre entre les divers vecteurs du répertoire initial.

#### 3.1.1 Etape 0 : Calcul des $d_i$

Nous avons choisi d'associer à chacun des  $C_i$  du dictionnaire, la distance  $d_i$  qui le sépare d'un vecteur constant,  $V_{ref}$ . La valeur d'initialisation de  $V_{ref}$  est indifférente dans la mesure où la relation d'ordre est établie par rapport à un même vecteur. Toutefois, nous avons choisi  $V_{ref} = (0, 0, \dots, 0)$  afin de réduire la complexité de calcul des  $d_i$ .

$$d_i = \sqrt{\sum_{j=1}^N (C_i^j - V_{ref}^j)^2} = \sqrt{\sum_{j=1}^N (C_i^j)^2} \quad (1)$$

avec  $V_{ref}^j$  la  $j^{\text{ème}}$  composante du vecteur  $V_{ref}$ .

L'étape 0 était une étape d'initialisation. A partir de l'étape 1, les opérations doivent être répétées pour chaque vecteur à coder.



**3.1.2 Etape 1: Calcul de  $d$**

Pour chaque vecteur à coder  $X$ , on calcule sa distance  $d$  par rapport à  $V_{ref}$ .

**3.1.3 Etape 2: Détermination de l'indice  $i$**

Soit  $L$  la taille du répertoire initial, nous avons associé à chacun des vecteurs références du répertoire  $C_{i,i=1..L}$ , la distance  $d_i$ , qui le sépare de  $V_{ref}$ . Ces distances sont stockées dans un tableau par ordre croissant  $d_1 < d_2 \dots < d_L$ .

On recherche l'indice  $i$  tel que  $d_i < d < d_{i+1}$ . Puisque les distances  $d_i$  sont ordonnées on peut procéder à une recherche dichotomique minimisant le nombre de comparaisons nécessaires.

La conjecture, validée a posteriori par les résultats des tests exposés ci-après, est que le vecteur représentatif du vecteur à coder se trouve au voisinage des vecteurs dont la distance à  $V_{ref}$  est voisine de celle de  $X$ .

Une fois  $i$  déterminé, on va chercher s'il existe à gauche ou à droite dans le voisinage de  $C_i$ , vecteur référence correspondant à la distance  $d_i$ , un vecteur de référence  $C_k$ , plus proche de  $X$  que les vecteurs  $C_i$  ou  $C_{i+1}$ . La définition du voisinage est basée sur l'inégalité triangulaire faisant intervenir les distances suivantes :

- $d = dist(X, V_{ref})$ , distance entre le vecteur à coder et  $V_{ref}$ .
- $d_i = dist(C_i, V_{ref})$ , distance entre le vecteur de référence  $i$  du dictionnaire et  $V_{ref}$ .
- $D_i = dist(C_i, X)$ , distance entre le vecteur de référence  $i$  du dictionnaire et le vecteur à coder.

Le voisinage correspond à la définition de bornes de recherche des indices au delà desquelles on est certain de ne plus trouver de vecteur référence minimisant davantage, que ceux trouvés jusqu'alors, la fonction de distorsion.

**3.1.4 Etape 3: Recherche à gauche de  $i$**

Pendant cette phase, on cherche en fait deux choses :

1. On cherche à déterminer si il existe un indice  $l$  tel que

$$dist(X, C_l) < dist(X, C_i) \iff D_l < D_i.$$

2. On cherche aussi à "arrêter la recherche" le plus vite possible. Il est en effet probable qu'il y ait plusieurs vecteurs de référence à gauche de  $C_i$  plus proche de  $X$  que ce dernier.

Pour déterminer  $l$ , on peut de façon simpliste parcourir les indices du dictionnaire  $l = i - 1 \dots 1$  à la recherche d'un vecteur  $C_l$  du répertoire plus proche de  $X$  que  $C_i$ , et ce, jusqu'au premier élément. On se ramène alors à une philosophie de type "full-search".

Il est dommage de se priver de l'information issue de l'ordonnancement que vient de subir le dictionnaire. Grâce à cette dernière, on peut définir un test permettant d'établir, de façon certaine, qu'une recherche d'un vecteur de référence **plus proche de X** est inutile au-delà d'un certain voisinage, et que par conséquent, on peut ainsi éviter des comparaisons coûteuses en calcul.

Ce test est bâti sur la vérification de l'inégalité triangulaire suivante :

$$dist(X, C_l) + dist(C_{l-k}, V_{ref}) < dist(X, V_{ref})$$

$$\iff D_l + d_{l-k} < d$$

avec  $C_l$  un vecteur de référence à gauche de  $C_i$  plus proche de  $X$  que ne le sont les  $C_{i-j,j=\{0..i-l-1\}}$ , et  $C_{l-k,k=\{1..l-1\}}$  un vecteur de référence à gauche de  $C_l$  dont on ne sait pas encore si il est plus proche de  $X$  que ne l'est  $C_l$ .

Dans le cas où cette inégalité est vérifiée, la sémantique est claire : " $C_{l-k}$  ne peut pas être plus proche de  $X$  que ne l'est  $C_l$ , et surtout il marque la borne inférieure (en terme d'indice) du voisinage de  $C_i$ ". On peut donc arrêter la recherche.

Dans le cas contraire,  $C_{l-k}$  pourrait être plus proche de  $X$  que ne l'est  $C_l$ , il deviendrait alors le nouveau  $C_l$ , et surtout il pourrait encore y avoir à sa gauche des vecteurs plus proche de  $X$ .

**3.1.5 Etape 4: Recherche à droite de  $i+1$**

On opère de la même manière pour la recherche à droite. On cherche à déterminer un indice  $r$  tel que

$$dist(X, C_r) < dist(X, C_{i+k}) \iff D_r < D_{i+k}.$$

Puis on résoud l'inégalité triangulaire suivante :

$$dist(X, C_r) + dist(X, V_{ref}) < dist(C_{r+k}, V_{ref})$$

$$\iff D_r + d < D_{r+k}$$

**Remarque:** Les figures qui suivent illustrent, dans un espace à deux dimensions, les différents cas pouvant subvenir lors des recherches à gauche et à droite. Les points sur les arcs de cercle servent à identifier les "orbites" déduites de la connaissance des distances propres à chaque vecteur. Néanmoins, et dans le but de faciliter la compréhension, leur position est choisie pour illustrer et souligner le cas de figure. Le lecteur devra rester conscient que leur position dans le plan est inconnue et que l'algorithme travaille au niveau des distances.

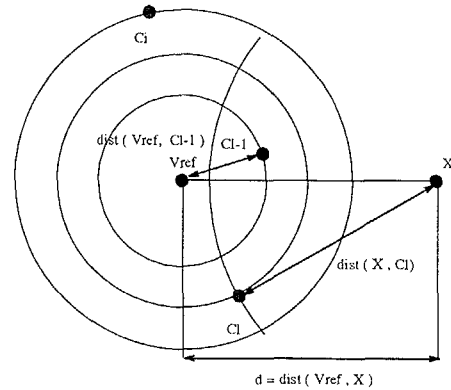


fig 1 : Inégalité triangulaire non vérifiée à gauche

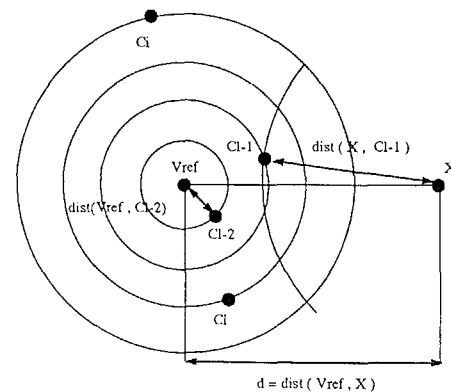


fig 2 : Inégalité triangulaire vérifiée à gauche.

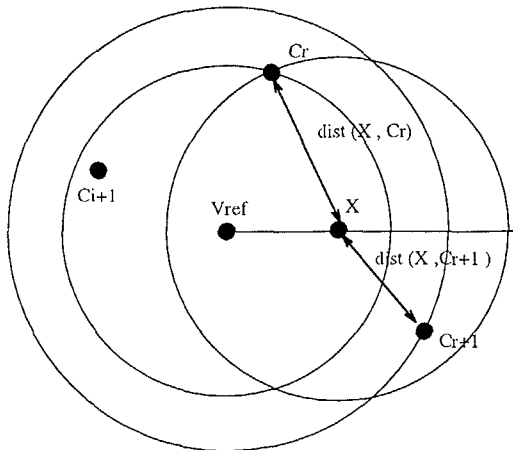


fig 3 : Inégalité triangulaire non vérifiée à droite

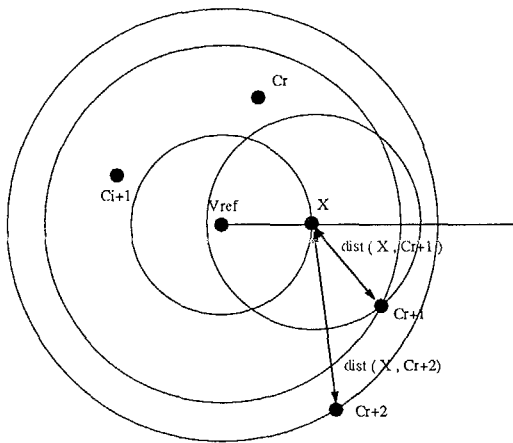


fig 4 : Inégalité triangulaire vérifiée à droite

### 3.1.6 Etape 5: Détermination finale du vecteur référence

Il nous suffit alors de comparer  $dist(X, C_l)$  et  $dist(X, C_r)$ , et de conserver le vecteur référence qui minimisera la fonction de distorsion entre le vecteur à coder  $X$  et  $C_l$  ou  $C_r$ .

## 3.2 Résultats

Après élaboration d'un dictionnaire LBG sur des séquences d'entraînement, cet algorithme a été testé sur des séquences d'images telles que Miss America, Table Tennis, ou le Présentateur.

Dans tous les cas, les adresses d'entrée du dictionnaire fournies par l'algorithme OVQ sont identiques à celles obtenues par l'utilisation d'une méthode optimale telle que le "full search". On obtient donc ainsi des séquences, de **qualité égale** comprimées à moindre coût de calcul.

Le tableau ci-dessous présente le nombre de comparaisons effectuées et le pourcentage de recherches évitées obtenus en utilisant la technique de recherche ordonnée (OVQ). Le format des images est de 256x256 en 256 niveaux de gris. Le dictionnaire est composé de 256 vecteurs de référence au format 4x4 et toujours avec 256 niveaux de gris.

Type de séquence	Nom de séquence	Nombre moyen de comparaisons	Pourcentage de recherches évitées
Head and Shoulders	Miss America	115	55
Head and Shoulders	Présentateur	104	59
Full Motion	Table Tennis	59	77

En conclusion, l'organisation du parcours du dictionnaire par la méthode **OVQ** permet de diminuer sensiblement le nombre total de recherches effectuées pour chaque vecteur à quantifier. Il s'agit donc d'une méthode simple et efficace qui, comme le montre ce qui suit, peut encore être améliorée.

## 4 Recherche Organisée Simplifiée (SOVQ)

Les expérimentations menées dans le cadre de OVQ nous ont conduit à certaines constatations intéressantes sur la convergence des recherches à gauche et à droite.

Le tableau ci-dessous fait apparaître le nombre de comparaisons effectivement réalisées lors des recherches à gauche et à droite.

Il montre que la vérification de l'inégalité triangulaire, qui constitue le critère d'arrêt de la recherche à gauche ou à droite, est réalisée le plus souvent bien après que l'on ait trouvé l'indice de déplacement à gauche ou à droite correspondant au vecteur référence le plus proche du vecteur à coder.

i	G-Dep	Nb-Cg	R-Dep	Nb-Cr
119	13	47	5	62
183	17	100	0	36
197	20	144	10	10
76	12	38	0	99
37	4	23	4	46

avec :

- i** Index de début de la recherche
- G-Dep** Indice de déplacement à gauche de  $i$  pour trouver le vecteur représentatif
- Nb-Cg** Indice de déplacement correspondant à la vérification de l'inégalité triangulaire à gauche
- R-Dep** Indice de déplacement à droite de  $i$  pour trouver le vecteur représentatif
- Nb-Cr** Indice de déplacement correspondant à la vérification de l'inégalité triangulaire à droite

La figure 5 illustre une configuration du dictionnaire où le vecteur représentatif est obtenu dès  $C_l$  et pourtant on ne peut être sûr que c'est le meilleur que lorsque l'on a testé  $C_{l-1}, C_{l-2}, C_{l-3}$ . La distance entre ces vecteurs et  $V_{ref}$  peut en effet laisser imaginer qu'ils sont plus représentatifs de  $X$  car ils vérifient l'inégalité triangulaire, or ceci s'avère faux a posteriori.

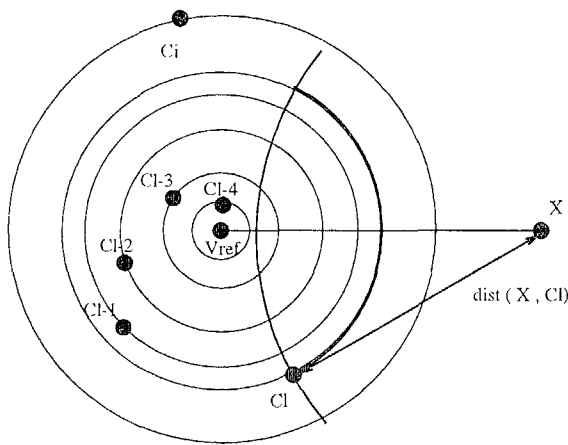


fig 5 : Exemple de recherche vers la gauche inutile ....

L'approche SOVQ vise à limiter au niveau de l'algorithme le nombre de comparaisons à gauche et à droite. Elle est directement issue de la constatation que dans la majorité des cas :

$$Nb - Cg \gg G - Dep$$

et

$$Nb - Cr \gg R - Dep.$$

Dans la mesure où les contraintes de compression sont compatibles avec une quantification sous-optimale, il est possible de réduire encore davantage le nombre de comparaisons à effectuer pour obtenir l'adresse du vecteur du dictionnaire le plus proche du vecteur à coder. Pour ce faire on restreint, on simplifie de façon arbitraire, le voisinage de recherche du vecteur minimisant la fonction de distorsion et se situant à gauche de  $C_i$  ou à droite de  $C_{i+1}$ .

Cette réduction de la complexité de calcul n'induit pas pour autant de dégradation notable des performances de l'algorithme en terme de qualité.

Soit  $G\text{-Max}$  la valeur maximale de l'indice de déplacement à gauche et  $R\text{-Max}$  la valeur maximale de l'indice de déplacement à droite, le tableau ci-dessous fait apparaître le pourcentage d'adresses du répertoire identiques à celles obtenues lors de l'utilisation d'une recherche "full-search" (V-ld).

L-Max	R-Max	V-ld
30	30	87.8 %
30	10	72.3 %

On remarquera que ce pourcentage reste très élevé. De plus, les vecteurs dont l'indice diffère correspondent à des distances excessivement proches de celles que l'on obtient par une recherche "full-search".

## 5 Conclusion

Nous avons présenté deux techniques issues d'une même philosophie d'ordonnement du dictionnaire vectoriel. Elles permettent d'obtenir aisément, dans le cadre de la compression d'images 256x256 et en terme de comparaisons, un gain de 6 sur la complexité de l'approche "full search". La légère distorsion alors induite par la quantification sous-optimale SOVQ n'apparaît pas au niveau des expérimentations menées sur les séquences tests.

Ces méthodes se révèlent donc efficaces dans l'obtention de

séquences d'images de bonne qualité (équivalente au "full-search") à moyenne complexité.

## Références

- [1] Y. Linde, A. Buzo & A. Gray  
"An algorithm for vector quantizer design"  
IEEE Trans. Commun., Vol. COM 28, pp.84-95, Jan. 1980.
- [2] A. Gersho & R. M. Gray  
"Vector Quantization and Signal Compression"  
Kluwer Academic Publisher, 1993.
- [3] N. Morcau  
"Techniques de compression des signaux"  
Collection Technique et Scientifique des  
Télécommunications, MASSON, 1995.
- [4] C. Raimondo  
"Contribution à l'étude et au développement de méthodes de compression numérique de séquence d'images"  
Thèse de doctorat, mention "Sciences de l'Ingénieur",  
présentée le 31 Mai 1991 à l'Université de Nice-Sophia  
Antipolis.
- [5] E. Goubault, E. Lançon, J. Menez & C. Raimondo  
"A vector quantizing method for coding signals and system for implementing said method"  
Demande de brevet européen n° 91480054.5- déposé  
par la compagnie IBM en date du 29 Mars 1991.



Séquence "Table Tennis" -&- Séquence "Présentateur"



Séquence "Miss America".