

# Accélération de la quantification vectorielle d'images par combinaison d'une transformation linéaire et d'une évaluation partielle de distorsion

Christophe FOUCHER<sup>1</sup>, François DURBIN<sup>2</sup>, Daniel LE GUENNEC<sup>1</sup>, Pierre LERAY<sup>1</sup>,  
André TISSOT<sup>2</sup>, Gilles VAUCHER<sup>1</sup> et Jacques WEISS<sup>1</sup>

1. Supélec/ETSN, avenue de la Boulaie, BP 28, 35511 Cesson Sévigné Cédex  
christophe.foucher@supelec.fr, gilles.vaucher@supelec.fr
2. CEA-DIF/Service équipement instrumentation métrologie,  
BP 12, 91680 Bruyères-le-Châtel

## Résumé

L'utilisation de la quantification vectorielle (QV) en compression d'images est freinée par la complexité algorithmique de l'opération de recherche du plus proche voisin (RPPV), vecteur minimisant une distorsion. Les travaux réalisés dans le domaine de la simplification de la QV peuvent être séparés en deux classes d'approches complémentaires: les contraintes sur le dictionnaire et la simplification de la RPPV elle-même. Dans le cadre de nos travaux sur l'amélioration des performances des algorithmes de QV, nous proposons ici une méthode appartenant à la seconde catégorie. Pour l'évaluer, elle a été testée en combinaison avec la QV Moyenne/Résiduel. Le temps de codage a été réduit dans un rapport allant jusqu'à 4,75.

## Mots Clef

Quantification vectorielle, compression d'images, complexité.

## 1 Introduction

Les algorithmes de compression avec pertes se composent généralement de trois étapes: prétraitement, quantification et compression sans pertes. Une transformée linéaire telle que la transformée en cosinus discrète (TCD) ou la transformation en ondelettes est parfois utilisée au cours de la première étape pour concentrer la variance du signal sur un plus petit nombre de composantes vectorielles.

La quantification vectorielle (QV) permet des taux de compression plus élevés que la quantification scalaire[5]. Rappelons qu'un quantificateur vectoriel est une application de  $\mathbb{R}^k$  dans  $[1, N]$  qui associe à tout vecteur  $k$ -dimensionnel  $\mathbf{x}$  un indice  $\alpha(\mathbf{x})$ . A chaque indice de quantification  $i$ , est associée une valeur de reconstruction  $\mathbf{y}_i$  appelée *vecteur code*, l'ensemble des vecteurs codes étant appelé *dictionnaire*. La fonction de codage  $\alpha$  réalise une *Recherche du Plus Proche Voisin*

(RPPV) en associant à  $\mathbf{x}$  l'indice de son plus proche voisin parmi les vecteurs codes.

L'utilisation de la QV est freinée par la complexité de cette opération. Pour un dictionnaire donné  $C = \{\mathbf{y}_i \mid i \in [1, N]\}$ , le vecteur code  $\mathbf{y}_i$  est le plus proche voisin de  $\mathbf{x}$  s'il minimise la distorsion:

$\forall j \in [1, N] d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j)$ . L'erreur quadratique est généralement utilisée pour  $d: d(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^k (x(n) - y(n))^2$ , où  $x(n)$  la  $n$ -ième composante de  $\mathbf{x}$ . Ainsi, la RPPV nécessite  $kN$  produits par vecteur, ce qui est coûteux car  $N = 2^{rk}$  ( $r$  étant la résolution en bits par vecteur).

Les algorithmes de réduction de la complexité appartiennent principalement à deux classes. Dans la première, appelée «QV contrainte»[5], différentes contraintes sont appliquées au dictionnaire. La QV moyenne/résiduel (QVMR), la QV classifiée et la QV par réseau régulier de points en sont des exemples[5]. La position des vecteurs codes doit respecter certaines contraintes induisant une baisse de qualité par rapport à un positionnement totalement libre des vecteurs codes. En contrepartie, ces contraintes permettent d'organiser le dictionnaire et de simplifier la RPPV (par exemple, recherche arborescente).

Les algorithmes de la seconde classe, parfois désignés sous le terme de *QV rapide* sont décrits dans la section suivante. Ils sont partiellement indépendants de tout type de contrainte sur le dictionnaire. En conséquence, des techniques de ces deux classes peuvent être combinées pour former un quantificateur vectoriel encore plus rapide.

Dans cet article, nous proposons d'améliorer la *Recherche par Distorsion Partielle* (RDP) qui appartient à la seconde classe, en utilisant des transformations linéaires. La section suivante décrit certains algorithmes existants et notre contribution est présentée dans la section 3. La section 4 présente les tests effectués et les résultats correspondants.

## 2 Algorithmes de QV rapide existants

Certaines techniques de la seconde classe utilisent une mesure de distorsion approximée qui demande moins de calculs. Par exemple, PO et CHAN[4] proposent de projeter les vecteurs dans un espace de dimension réduite et d'utiliser l'erreur quadratique dans cet espace. Les directions de projection sont choisies afin que les vecteurs projetés soient une aussi bonne approximation que possible des vecteurs originaux. PO et CHAN suggèrent d'utiliser les premières directions de la TCD ou de la transformation de Walsh-Hadamard (TWH). Ainsi, la mesure de distorsion obtenue est également une approximation de l'erreur quadratique dans l'espace initial, et le codage n'est pas une exacte RPPV. Il en résulte une baisse de qualité d'autant plus grande que la dimension de l'espace de projections est petite.

Dans d'autres techniques de QV rapide, on utilise des *inégalités géométriques* permettant d'éliminer des vecteurs codes candidats sans calculer l'erreur quadratique. Par exemple, la *Recherche par Distorsion Partielle*(RDP), de BEI et GRAY[2], utilise le fait que l'erreur quadratique est une somme de termes positifs. On appelle *distorsion partielle* la somme des premiers de ces termes. Lors de la RPPV, si la distorsion partielle dépasse la distorsion minimale déjà rencontrée, on passe directement au vecteur code suivant sans calculer la distorsion totale. D'autres algorithmes[1, 3], utilisent des inégalités géométriques plus explicites telles que l'inégalité triangulaire.

Dans [6] et [7], il est proposé d'améliorer l'efficacité de la RDP par une transformation linéaire préalable. Sa matrice est obtenue par *analyse en composantes principales* (ACP) des vecteurs traités ou des vecteurs codes. La variance des données se retrouve concentrée sur les premières composantes des vecteurs transformés. Ainsi, durant la RDP, la probabilité de sortie de boucle sur les premières composantes est augmentée et l'efficacité améliorée.

## 3 Notre proposition d'algorithme

PO et CHAN ont montré que le coût de calcul de la distorsion pouvait être réduit par projection des vecteurs sur un espace de dimension réduite. Cela signifie que l'information nécessaire à la discrimination entre les vecteurs codes est portée par un nombre réduit de directions vectorielles. Cependant, un nombre arbitraire de directions de projection doit être choisi sur la base d'un compromis entre la réduction du temps de codage et la baisse de qualité due à l'approximation de la RPPV. Nous avons testé l'algorithme de PO et CHAN, qui suggèrent 4 projections pour une dimension initiale de 16, et constaté une sévère baisse de qualité. De plus, le nombre de composantes nécessaires

à la discrimination peut varier d'un vecteur à l'autre. Un vecteur code très éloigné d'un vecteur  $\mathbf{x}$  à coder pourrait être éliminé à partir d'une seule composante de projection, alors que le choix entre deux vecteurs codes proches de  $\mathbf{x}$  nécessite une meilleure précision de l'approximation.

D'où l'idée de combiner la concentration d'information suggérée par PO et CHAN et la discrimination par nombre variable de composantes de la RDP. Les algorithmes décrits dans [6, 7] utilisent cette approche avec une matrice de transformation linéaire obtenue par ACP. Cependant, l'ACP préalable induit un surcoût algorithmique important, et une telle transformation ne possède pas de symétrie permettant d'optimiser les calculs à la manière de la FFT par exemple.

Notre algorithme s'inspire des deux approches précédentes. Comme dans celle de PO et CHAN, la variance est concentrée par TCD ou TWH. Ces transformations sont indépendantes du dictionnaire et de l'image codée et leur calcul peut être optimisé. Au lieu de négliger les composantes de faible variance, nous proposons d'appliquer la RDP aux vecteurs transformés. La concentration de variance est exploitée, tout en réalisant une recherche équivalente à la recherche exhaustive, sans aucune approximation. Cette méthode présente l'avantage de ne pas nécessiter de choix arbitraire du nombre de directions de projection. Autrement dit, le nombre de vecteurs de projection est adaptatif.

Comme d'autres algorithmes de QV rapide, celui-ci est potentiellement indépendant de tout type de contraintes imposées au dictionnaire. Il peut donc être combiné à tout type d'algorithme de QV contrainte. Il faut toutefois analyser l'influence de la contrainte sur la distribution statistique des vecteurs, et choisir une transformation linéaire correspondant à cette distribution. Comme PO et CHAN, nous avons travaillé sur la QVMR plutôt que sur la QV sans contrainte pour la réduction substantielle de complexité qu'elle permet. Le calcul de moyenne est intégré à celui de la transformation : avec la TCD ou la TWH, la moyenne est la première composante transformée.

Lors de la conception du système, trois dictionnaires doivent être construits, un pour la moyenne,  $C_m = \{u_i\}$ , un pour le résiduel au codage,  $C_r = \{y_i\}$  et un pour le résiduel au décodage,  $C'_r = \{y'_i\}$ . Les vecteurs d'apprentissage sont transformés (TWH ou TCD, notée  $f$ ) et  $C_m$  est construit. Ensuite, leur moyenne est quantifiée et leur résiduel est extrait. La variance des composantes est alors calculée. L'ordre des composantes est déterminé et il sera utilisé directement lors du codage. Il est important de noter que la variance de la moyenne résiduelle dépend de la quantification scalaire. Une quantification précise produira une variance réduite de la moyenne résiduelle. Les dictionnaires sont construits par un algorithme classique tel que le *splitting*[5]. Le décodeur n'a pas besoin de calculer la trans-

formation inverse. Elle est appliquée une fois pour toutes aux vecteurs codes lors de la construction du système. Autrement dit,  $C_r'$  est défini par :  $\mathbf{y}'_i = f^{-1}(\mathbf{y}_i)$ . Lors du codage,  $\mathbf{x}$  est codé par l'indice  $\alpha_m(\mathbf{x})$  ( $\hat{z}(0) = u_{\alpha_m(\mathbf{x})}$ ) de la valeur de quantification de la moyenne et par l'indice  $\alpha_r(\mathbf{x})$  du vecteur de quantification du résiduel. Les composantes sont réordonnées par ordre décroissant de variance, déterminé durant la phase de conception.

En résumé, la procédure de codage d'un vecteur  $\mathbf{x}$  comprend les étapes suivantes :

1. transformer  $\mathbf{x}$  en  $\mathbf{z}$  par  $f$  (TCD ou TWH),
2. quantification scalaire de la première composante  $z(0)$ , produisant  $\hat{z}(0)$ ,
3. remplacer  $z(0)$  par la valeur résiduelle  $z(0) - \hat{z}(0)$ ,
4. réordonner les composantes de  $\mathbf{z}$  par variance décroissante,
5. quantification vectorielle du résiduel avec la RDP.

On combine ainsi l'apport de PO et CHAN[4] (points 1 et 4), celui de la QVMR[5] (points 2 et 3) et celui de la RDP[2] (point 5).

Les étapes du décodage sont :

1. décoder la moyenne, reconstruite par  $u_{\alpha_m(\mathbf{x})}$ ,
2. décoder le résiduel, reconstruit par  $\mathbf{y}'_{\alpha_r(\mathbf{x})}$ ,
3. calculer  $\hat{\mathbf{x}} = \mathbf{y}'_{\alpha_r(\mathbf{x})} + u_{\alpha_m(\mathbf{x})} \times \mathbf{1}_k$ .

## 4 Expériences

Une première expérimentation nous a permis de vérifier que notre algorithme réduit le temps de codage par rapport à la QVMR simple avec recherche exhaustive. Les expériences suivantes ont été menées afin de vérifier qu'une transformation linéaire peut améliorer la RDP dans ce contexte.

### 4.1 Protocole expérimental

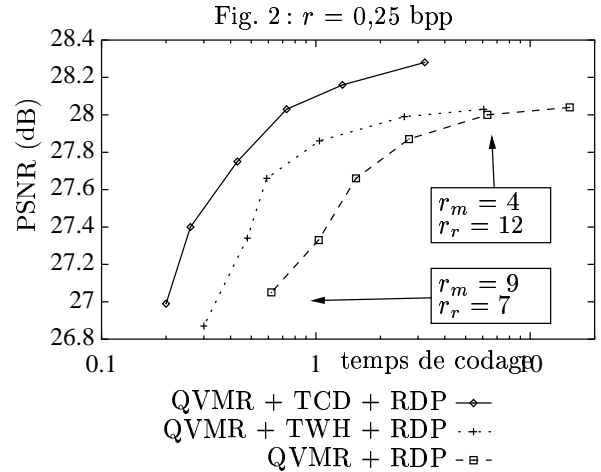
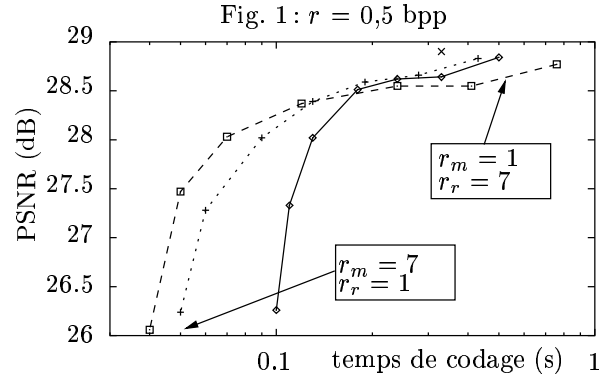
Nous avons testé notre algorithme avec trois résolutions  $r$  classiques différentes : 0,25, 0,5 et 1 bit par pixel (bpp) et deux tailles de blocs :  $4 \times 4$  et  $8 \times 8$ . La RDP est associée à la TCD ou la TWH, et comparée à la QVMR classique accélérée par RDP seule, sans transformation. Dans chaque expérience, différentes combinaisons (résolution des moyennes  $r_m$ , résolution des résiduels  $r_r$ ) sont testées avec la relation :  $(r_m + r_r)/k = r$ . Ainsi, les dictionnaires de moyenne et de résiduel comportent respectivement  $2^{r_m}$  valeurs et  $2^{r_r}$  vecteurs codes.

Les dictionnaires sont construits avec 18 images d'apprentissage en niveaux de gris sur 8 bits. L'image classique Lena ( $512 \times 512$  pixels), non incluse dans la base d'apprentissage, est l'image de test.

La qualité de reconstruction est mesurée par le *Peak-Signal-to-Noise Ratio* :

$$PSNR = 10 \log \left( \frac{255^2}{EQM} \right).$$

Le temps de codage est mesuré, sur un PC équipé d'un processeur Pentium II 300MHz.



### 4.2 Résultats

Dans la première expérience, les blocs sont de taille  $4 \times 4$  pixels et  $r = 0,5$ . L'algorithme est testé avec des valeurs entières de  $r_m$  allant de 1 à 7. La figure 1 montre les résultats obtenus avec chaque algorithme et chaque combinaison  $(r_m, r_r)$ . L'échelle de temps est logarithmique.

Pour un couple de résolutions  $(r_m, r_r)$  donné, le PSNR de reconstruction est similaire pour les trois algorithmes. Avec une quantification grossière du résiduel, QVMR+RDP demandent moins de temps de calcul. En fait, il y a peu de vecteurs codes pour le résiduel, et la partie vectorielle du codage consomme peu de temps. Au contraire, si plus de bits sont alloués au résiduel, utiliser une transformée devient rentable. Dans tous les cas, la TWH est plus efficace que la TCD. La TWH n'utilise que des additions alors que la TCD utilise quelques produits. Cette différence peut expliquer l'écart observé de temps de codage.

Pour comparer, nous avons implémenté l'algorithme décrit dans[1]. Comme dans cet article, nous avons appliqué la QV non contrainte au lieu de la QVMR. Avec le même protocole expérimental que dans la première expérience, nous avons obtenu un temps de codage de 0,33 secondes et un PSNR de 28,9dB ( $\times$  sur la fig. 1). La figure 1 montre un PSNR légèrement inférieur pour ce temps de codage. Ainsi, l'algorithme de BAEC est

préférable si un temps de codage plus faible n'est pas nécessaire à l'application. Inversement, si un temps de codage réduit est très important, notre algorithme est meilleur. De plus, le compromis qualité/temps de calcul ne peut pas être modulé avec la QV non contrainte. Pour poursuivre la comparaison, on pourrait essayer d'accélérer la QVMR avec l'algorithme de BAEK, mais son application à la QVMR n'est pas directe car il utilise la grande variance de la moyenne des blocs. Une autre direction de discrimination que  $\mathbf{1}_k$  pourrait être plus appropriée, selon la résolution de la moyenne.

Dans la deuxième expérience, les blocs sont de taille  $4 \times 4$  et la résolution totale est de 1 bpp. La moyenne est quantifiée sur 4 à 9 bits. Contrairement à la valeur des pixels, la moyenne est un réel, et il n'est pas absurde de la coder sur plus de bits que la valeur des pixels.  $r_r$  va de 12 à 7 bits.

A ce niveau de résolution, notre algorithme d'accélération réduit significativement le temps de codage, comparativement à la QVMR+RDP avec tous les couples  $(r_m, r_r)$ . De plus, la TCD, tout en demandant plus de calculs que la TWH, permet des temps de codage plus faibles que la TWH grâce à sa meilleure propriété de concentration de la variance.

En fait, dans ce test, même avec la RDP, le temps de calcul des transformations est négligeable devant celui de la RPPV.

Finalement, un troisième test applique la QVMR à des blocs  $8 \times 8$ , conduisant à une basse résolution de 0,25 bpp, avec les mêmes résolutions de moyenne et de résiduel que dans l'expérience précédente. La figure 2 montre les résultats obtenus. De nouveau, la combinaison (transformation+RDP) apporte un gain substantiel de temps de codage. Ce gain est plus important que dans les test précédents. Par exemple, avec  $r_m = 5$  et  $r_r = 11$ , QVMR+TCD+ RDP ont donné un temps de codage 4,75 fois plus court qu'avec QVMR+RDP. Dans le précédent test, ce ratio était de 1,60.

## 5 Conclusion

Nous avons proposé dans ce papier un algorithme d'accélération de la recherche du plus proche voisin pour la quantification vectorielle d'image. Cet algorithme combine une transformation linéaire, TWH ou TCD, et la recherche par distorsion partielle. La transformation linéaire concentre la variance sur un nombre réduit de composantes. Grâce à un réordonnement par variance décroissante, l'efficacité de la RDP est améliorée.

Dans la plupart des cas, nous avons observé que notre algorithme a réduit le temps de codage pour un niveau de qualité donné. A basse résolution, à cause du faible nombre de vecteurs codes, la QVMR simple est plus rapide. A partir d'une résolution de 4 bits par vecteur résiduel, le coût de calcul de la transformation devient inférieur au gain obtenu sur le temps de codage.

Comme les autres algorithmes de QV rapide, il peut être utilisé avec tout type de QV à dictionnaire contraint. Nous l'avons testé avec la QVMR car elle permet des taux de codage plus bas que la QV sans contrainte. Pour compléter ce travail, il serait intéressant d'appliquer cette approche à d'autres algorithmes de QV, en gardant à l'esprit que les gains obtenus dépendent de la complexité de l'algorithme de QV envisagé.

Les expériences ont été menées sur des images naturelles. Les transformées linéaires pourraient être moins efficaces sur d'autres types d'images, comme les images de synthèse ou les images d'erreur en codage vidéo.

Il pourrait également être intéressant de vérifier si toutes les composantes transformées sont réellement nécessaires, et si une réduction supplémentaire du temps de codage ne pourrait pas être obtenue par élimination des composantes inutiles, tout en appliquant la RDP sur les composantes conservées.

## Références

- [1] SeongJoon BAEK, BumKi JEON, et Koeng-Mo SUNG. « A Fast Encoding Algorithm for Vector Quantization ». *IEEE Signal Processing Letters*, 4(12):325–327, décembre 1997.
- [2] Chang-Da BEI et Robert M. GRAY. « An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization ». *IEEE Transactions on Communications*, COM-33(10):1132–1133, octobre 1985.
- [3] J. CARDINAL. « A Fast Full Search equivalent for Mean-Shape-Gain Vector Quantizers ». Dans *20th Symposium on Information Theory in the Benelux*, pages 39–46, 1999.
- [4] C.-K. CHAN et L.-M. PO. « A complexity reduction technique for image vector quantization ». *IEEE Transactions on Image Processing*, 1(3):312–321, juillet 1992.
- [5] Allen GERSHO et Robert M. GRAY. *Vector quantization and signal compression*. Kluwer Academic, 1992.
- [6] James MCNAMES. « Rotated Partial Distance Search for Faster Vector Quantization Encoding ». *IEEE Signal Processing Letters*, 7(9), septembre 2000.
- [7] Władysław SKARBEK et Krystian IGNASIAK. « Fast VQ codebook search in KLT space ». *Neural Network World*, 6(3):383–386, 1996.