

Algorithmes de recherche rapide du plus proche mot-code pour le codage vectoriel 3D des séquences d'images

A. Majda, M. Janati-Idrissi et H. Youlal

Université Mohammed V, Faculté des Sciences,

B.P. 1014, RABAT, MAROC

Fax : + 212 7 778973

E-mail : youlal@mtds.net.ma

Résumé

Dans le codage vectoriel des séquences d'images se pose le problème de temps d'exécution trop élevé, ce qui limite les implantations temps réel de ces méthodes. Ainsi, diverses techniques de codage vectoriel 2D d'images statiques dont l'objectif est d'optimiser les temps de construction du dictionnaire et de codage, en faisant subir à chaque mot - code des tests préliminaires, ont été récemment proposées. Dans ce papier on propose d'étendre ces algorithmes au cas 3D pour le codage des séquences d'images et de comparer leurs performances. Un aperçu dans le cas d'images statiques de leur rapidité par comparaison à l'algorithme de base est d'abord présenté. Ensuite, les performances des versions 3D sont illustrées par simulations de séquences d'images artificielles.

Introduction

La quantification vectorielle s'est révélée un moyen attrayant et efficace pour atteindre des taux de compression numérique très prometteurs. Elle peut être définie comme une surjection Q d'un sous-espace A de l'espace euclidien de dimension K , R^K , vers un ensemble Y de R^K , où $Y = \{ y_i / i=1, \dots, N_c \}$ est le dictionnaire de taille N_c et de mot-codes $y_i = (y_{i1}, \dots, y_{iK})$. Elle comporte trois étapes: la construction du dictionnaire, le codage et le décodage. Le dictionnaire doit contenir les mot-codes les plus représentatifs et doit être placé au niveau du codeur et du décodeur. Le codeur assigne à chaque vecteur d'entrée x de A un indice p qui correspond au mot-code $y_p = Q(x)$ le plus proche de x . Le décodeur utilise l'indice p transmis pour trouver dans le dictionnaire le mot-code y_p qui représentera le vecteur x . Le quantificateur vectoriel a été utilisé en codage de la parole (signal

Abstract

In vector quantization of image sequences arises the problem of too high computation times which limit real time implementations of these methods. Recently, several 2D vector quantization techniques of static images with the objective of optimizing the times needed to construct the dictionary and for coding, by performing appropriate preliminary tests on each code-word, have been reported. In this paper, one propose to extend these fast algorithms to the 3D case for image sequences coding and to compare their performances. An outline in case of static images of their fastness compared to the basic algorithm, is first presented. Then, performances of the 3D versions are illustrated through simulations of artificial image sequences.

1D) et d'images (signal 2D) [1]. En quantification de la parole le signal 1D est subdivisé en vecteurs considérés comme des entités indivisibles durant le processus de codage. En codage d'images le signal 2D est subdivisé en blocs rectangulaires. Le contenu de chaque bloc est restitué dans un vecteur considéré comme unité de quantification. La taille du bloc doit être assez large pour grouper les pixels qui ont une corrélation spatiale, mais suffisamment petite pour éviter les effets de blocs. La première démarche pour l'application du quantificateur vectoriel en codage des séquences d'images est d'utiliser un vecteur de quantification 2D, où chaque image de la séquence est quantifiée indépendamment des autres images. L'inconvénient majeur de cette utilisation est que seule la corrélation spatiale est prise en compte. Dans le cas où on veut exploiter les corrélations spatiale et temporelle contenues dans l'image, il est nécessaire d'utiliser un vecteur de quantification 3D. La

troisième composante est celle du temps. Pour cela, la séquence considérée comme un signal 3D est subdivisée en cubes de taille $K=N_xM_xS$ (Fig.1). Chaque cube contient les pixels voisins d'une petite région de l'image (corrélacion spatiale), et les pixels ayant les mêmes coordonnées dans les $S-1$ images suivantes dans la séquence (corrélacion temporelle). Donc le vecteur de quantification 3D permet de grouper un nombre plus élevé de pixels corrélés que dans le cas 2D, tout en évitant les effets de blocs que le vecteur de quantification 2D présente lorsqu'on y groupe le même nombre de pixels [2]. Pour optimiser le volume de calcul, diverses techniques pour accélérer la recherche du plus proche mot-code ont été récemment proposées [3,4,5,6]. Il s'agit de faire subir à chaque vecteur du dictionnaire des tests préliminaires. De cette façon, on limite la zone de recherche du plus proche mot-code à ceux qui vérifient certaines conditions. L'extension de ces méthodes au cas 3D pour la compression des séquences d'images et la comparaison de leurs performances est présentée dans ce qui suit.

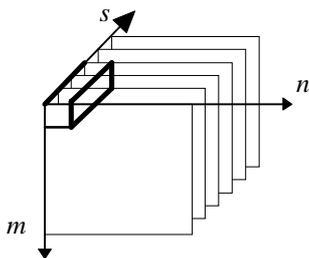


Figure 1 : Cube tridimensionnel d'une séquence d'images pour le vecteur de quantification 3D
 m, n : coordonnées spatiales,
 s : coordonnée temporelle.

Algorithme LBG standard

L'algorithme le plus utilisé pour la compression vectorielle est celui de Linde, Buzo et Gray [1], dit algorithme LBG. C'est un algorithme itératif qui conduit à un dictionnaire sous-optimal. Il calcule, pour chaque vecteur d'entrée x , la distance $d(x, y_j)$ pour tous les mot-codes du dictionnaire, afin de trouver le vecteur qui lui est le plus proche. Cette distance est souvent choisie euclidienne quadratique. Dans le cas 3D, elle est donnée par la quantité:

$$d(x, y_i) = \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{s=0}^{S-1} (x(m, n, s) - y_i(m, n, s))^2}$$

Le coût de calcul pour chaque vecteur x est de l'ordre de $N_c(M_xN_xS)$ multiplications et $N_c(2(M_xN_xS)-1)$ additions/soustractions. Ce coût croît alors avec la dimension du dictionnaire. Cependant, plus on augmente la taille du dictionnaire plus on assure une meilleure reconstruction de l'image. Ce qui mène à faire un compromis (qualité de l'image reconstruite / complexité de calcul).

Extension 3D de l'algorithme MFNNS (Modified Fast Nearest Neighbour Search)

L'algorithme se base sur l'idée suivante: Deux vecteurs qui sont très proches l'un de l'autre ont des normes voisines, [3]. Le mot-code y_p qui peut être proche au vecteur d'entrée x vérifie:

$$|r_x - r_p| = \min_{1 \leq j \leq N_c} |r_x - r_j|$$

La recherche du plus proche mot-code se fera alors parmi les y_j qui vérifient :

$r_x - d(x, y_p) \leq r_j \leq r_x + d(x, y_p)$ où r_x et r_j sont respectivement les normes de x et y_j .

Dans le cas 3D, elles sont données par :

$$r_x = \sqrt{\left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{s=0}^{S-1} x^2(m, n, s) \right)}$$

$$r_j = \sqrt{\left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{s=0}^{S-1} y_j^2(m, n, s) \right)}$$

Extension 3D de l'algorithme MENNS (Modified Equal-Average Nearest Neighbour Search)

Cet algorithme utilise deux critères : un relatif à la moyenne et l'autre à la variance des vecteurs, [6]. le mot-code y_p susceptible d'être proche du vecteur d'entrée x est tel que:

$$\left| m_x - m_{y_p} \right| \leq \min_{1 \leq j \leq N_c} \left| m_x - m_{y_j} \right|$$

La zone de recherche du plus proche mot-code se limite alors aux mot-codes y_j ayant :

- Une moyenne m_{y_j} comprise entre $m_x - \frac{d(x, y_p)}{\sqrt{K}}$

et $m_x + \frac{d(x, y_p)}{\sqrt{K}}$ avec $K=N_xM_xS$ dans le cas 3D.

- Une variance V_{y_i} telle que : $|V_x - V_{y_i}| < d(x, y_p)$.

Dans le cas 3D, la moyenne et la variance d'un vecteur x sont données par les expressions suivantes :

$$m_x = \frac{1}{M \times N \times S} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{s=0}^{S-1} x(m, n, s)$$

$$V_x = \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{s=0}^{S-1} (x(m, n, s) - m_x)^2}$$

Performances dans le cas d'images 2D

L'analyse et l'évaluation des performances des techniques de codage vectoriel 3D est entreprise en deux étapes. En premier lieu et afin de rendre compte de leur comportement dans le cas des images fixes, les tableaux 1 et 2 récapitulent quelques résultats de traitement d'images archéologiques et d'archives du patrimoine marocain. En effet, pour $s=1$, on retrouve les techniques développées dans la littérature [3-7]. Les échantillons d'images traitées par ces algorithmes sont illustrées sur les figures 2, 3 et 4. Les résultats montrent que les deux algorithmes qui utilisent les techniques de recherche rapide se détachent nettement de la version de base. Ces techniques permettent des temps de calcul nettement plus réduits sans introduire davantage d'erreur que l'algorithme de base ne le fait, comme on peut l'apprécier au vue des résultats (tableau 2).

Tableau 1. Temps d'exécution en secondes sur ordinateur PC486/DX2-66MHz

Taille Nc	Algorithme	td	tc Palais	tc Volubilis 1	tc Volubilis 2
256	LBG	3328	43	42	42
	MFNNS	1352	6	6	5
	MENNS	1286	5	5	5
512	LBG	65032	83	81	82
	MFNNS	2201	9	9	8
	MENNS	2093	8	8	7

Tableau 2. Valeurs des taux PSNR en dB

Taux de bits/pixel	Algorithme	Palais	Volubilis 1	Volubilis 2
0.5	LBG	23.30	22.54	22.56
	MFNNS	23.31	22.52	22.57
	MENNS	23.32	22.53	22.57
0.56	LBG	23.98	23.27	23.26
	MFNNS	23.96	23.27	23.23
	MENNS	23.97	23.27	23.25

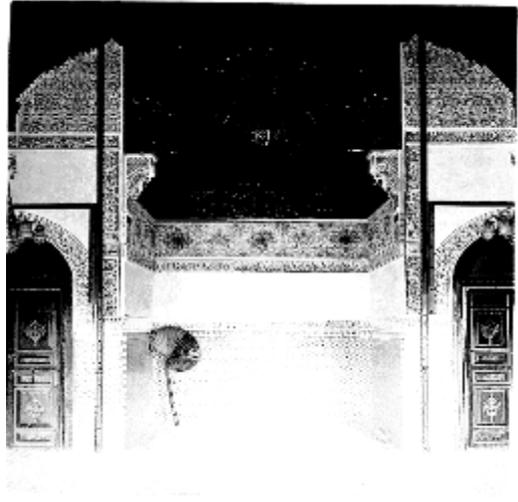


Figure 2 : Image archéologique « palais » codée sur 8 bits



Figure 3 : Image archéologique « Volubilis 1 » codée sur 8 bits



Figure 4 : Image archéologique « Volubilis 2 » codée sur 8 bits

Codage 3D de séquences d'images et résultats

La séquence d'images teste est une séquence artificielle communément connue sous le

nom « Eye », générée par la relation suivante:

$$f(x, y, t) = 25 \times \exp\{-0.01\|R\|\} \times \cos(2\pi R / p) + 31$$

où $f(x, y, t)$ est l'intensité à la position (x, y) au temps t , R est la distance entre (x, y) et le centre de l'œil, avec $p = \left(1 - \frac{1}{60} R\right) \times 10 + 10$.

La séquence est subdivisée en cubes distincts de taille $4 \times 4 \times 2$. La taille des images de la séquence étant (64×64) . Le dictionnaire est construit à partir des huit images de la séquence (séquence d'apprentissage). Chaque paire d'images donne ainsi 256 mot-codes. Les algorithmes sont comparés entre eux en terme des temps de construction du dictionnaire, td , et de codage d'une paire d'images ($s=2$) de la séquence, tc , et du rapport signal sur bruit de crête $PSNR$.

Les tableaux 3 et 4 illustrent cette comparaison pour différentes tailles (N_c) du dictionnaire. La rapidité des algorithmes MENNS et MFNNS pendant les phases de construction du dictionnaire et de codage est bien mise en évidence. Notons aussi que pour un même taux de compression, on obtient pratiquement la même qualité de l'image reconstruite pour les trois algorithmes. Ce qui conforte la supériorité des versions accélérées en ce sens qu'elles n'introduisent pas plus d'erreur que ne le fait l'algorithme de base. Un autre aspect de la comparaison est celui relatif aux méthodes hybrides 3D et 2D avec estimation du mouvement [8].

Tableau 3 : Temps d'exécution en secondes sur ordinateur PC486/DX2-66MHz

Taille N_c	Algorithme	td	tc (I1,I2)	tc (I3,I4)
256	LBG	115	7	7
	MFNNS	40	1	1
	MENNS	36	1	1
512	LBG	225	14	14
	MFNNS	62	2	2
	MENNS	52	2	2

Tableau 4 : Valeurs du PSNR en (dB)

Taux de bits/pixel	Algorithme	I1	I2	I3	I4
0.25	LBG	42.85	43.21	42.49	43.16
	MFNNS	42.72	43.11	42.30	42.40
	MENNS	42.85	43.38	42.55	43.16
0.28	LBG	52.94	53.67	50.01	49.78
	MFNNS	52.65	53.53	49.81	49.68
	MENNS	52.82	53.74	49.83	49.82

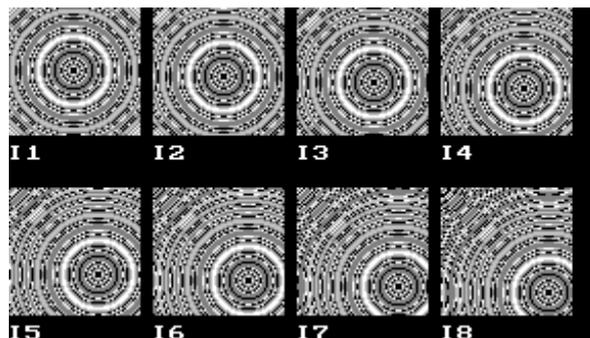


Figure 5 : La séquence d'images artificielle "Eye"

Conclusion

Les résultats préliminaires présentés dans ce papier montrent que le meilleur compromis entre le temps d'exécution et la valeur du rapport PSNR, revient à l'algorithme MENNS, suivi de près par l'algorithme MFNNS. Les deux extensions surclassent la version standard en terme de temps de codage à taux de compression égaux. Ces résultats sont par ailleurs confirmés par d'autres simulations comparatives sur une séquence d'images satellitaires.

Remerciements : Les images archéologiques ont été fournies par la direction du patrimoine culturel du Maroc.

Références

- [1] R.M. Gray "Vector quantization", IEEE ASSP Magazine, Vol.1, 1984, pp.4-29.
- [2] J. Huguet, L. Torres "Vector quantization in image sequence coding", Signal Processing V: Theories and applications, 1990, pp.1079-1082.
- [3] C.M.Huang, Q.Bi, G.S.Stiles, R.W.Harris "Fast full search equivalent encoding algorithms for image compression using vector quantization", IEEE Trans. on Image Processing, Vol.1, N° 3, 1992, pp.413-416.
- [4] L Torres, J. Huguet "An improvement on codebook search for vector quantization " IEEE Trans. on Communications, Vol.42, 1994, pp.208-210.
- [5] A. Xiau " Acceleration of the LBG algorithm", IEEE Trans. on Communications, Vol.42,1994, pp.1518-1523.
- [6] C.H.Lee, L.H.Chen " Fast closest codeword search algorithm for vector quantization ", IEE Proc.-V, Image Signal Process. Vol 141, n° 3, 1994. pp.143-148.
- [7] A. Majda, M. Janati-Idrissi, H. Youlal, "Techniques d'optimisation du dictionnaire de mot-codes en quantification vectorielle d'images", Colloque JNEETS-96, Marrakech, 1996, pp. 342-345.
- [8] H. Youlal, M. Janati-Idrissi, K. Hakam, " Performance of 2D/3D prediction based image sequences coding methods for MRI medical images ", Colloque JTEA'94, Tunis, 1994.