

COMPRESSION DE SEQUENCES D'IMAGES AVEC ESTIMATION DE MOUVEMENT ET QUANTIFICATION VECTORIELLE ALGEBRIQUE

T. Gaidon, M. Barlaud, P. Mathieu

Laboratoire I3S du CNRS
 Université de NICE - SOPHIA ANTIPOLIS
 250 Av. Albert Einstein 06560 VALBONNE (FRANCE)
 et Groupement De Recherche - Traitement Du Signal et Images (GDR TDSI)

RÉSUMÉ

Nous présentons dans cet article une méthode de compression de séquences d'images numériques. Pour réduire l'information redondante spatiale et temporelle nous utilisons un codeur hybride composé d'un estimateur de mouvement et d'un quantificateur vectoriel algébrique.

Une technique pel-réursive améliorée est utilisée pour réaliser une estimation et compensation de mouvement entre deux images d'une séquence. Nous présentons une approche Markovienne tenant compte des discontinuités du champ de déplacement.

Les images d'erreur de prédiction sont quantifiées vectoriellement. L'utilisation de dictionnaires définis par des réseaux réguliers de points permet d'utiliser des algorithmes de quantification rapides pour le traitement des séquences.

I. INTRODUCTION

Les séquences d'images sont de plus en plus utilisées comme moyen de communication (télévision, vidéoconférence, ...). Les informations visuelles sont souvent plus expressives que la parole ou l'écriture. Malheureusement, leur utilisation entraîne la manipulation d'une grande quantité de données numériques.

Le propos de cet article est la compression des séquences d'images numériques pour réduire ce volume de données.

L'algorithme proposé utilise une *technique pel-réursive d'estimation-compensation du mouvement* et une *quantification vectorielle* des images d'erreurs de prédiction.

Nous présentons au paragraphe II le schéma de compression. Le paragraphe III est consacré à l'estimation de mouvement et le suivant à la quantification vectorielle des images d'erreurs de prédiction avec les réseaux réguliers de points. Finalement les résultats sur des séquences réelles de type télévision sont présentés au paragraphe V.

II. SCHEMA DE COMPRESSION

Le schéma de compression est décrit aux figures 1 et 2. Une *estimation* de mouvement est effectuée entre les images codées-décodées \hat{I}_{t-1} et \hat{I}_t aux instants $t-1$ et t . En reproduisant ce mouvement estimé sur l'image \hat{I}_t , on *prédit* l'image suivante. L'image prédite \tilde{I}_t est comparée à l'image originale I_t . L'erreur de prédiction $\epsilon = \tilde{I}_t - I_t$ est quantifiée et transmise.

ABSTRACT

Image sequences are more and more used in multimedia. They are a favorite support in communication applications. Nevertheless, using digital image sequences involves manipulation of an enormous quantity of numerical data. Therefore, a compression scheme is needed.

In this paper a digital image sequences compression scheme is presented. It consists in two main functions: a pel-recursive motion estimation and a lattice vector quantization. We propose an improved motion estimation algorithm taking into account discontinuities. A lattice vector quantizer using pyramidally shaped codebook is used.

These two improved technics permit to obtain better results on TV sequences: high compression ratio with good visual quality.

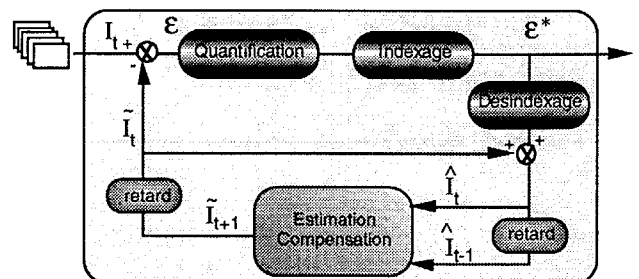


Figure 1 : Schéma du codeur.

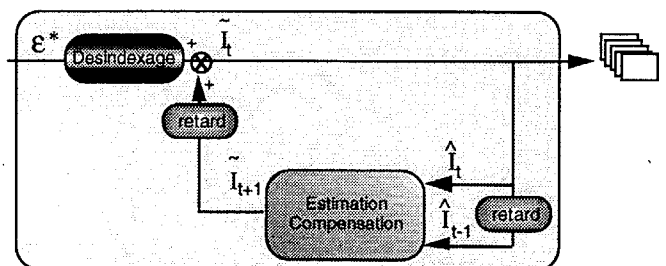


Figure 2: Schéma du décodeur

L'estimation de mouvement est réalisée à partir des images codées-décodées au lieu des images originales. Ainsi, côté codeur et décodeur les modules estimation et compensation disposent des mêmes entrées. Par conséquent, il n'est pas nécessaire de transmettre d'information relative aux vecteurs de mouvement.



III. ESTIMATION DE MOUVEMENT

Le but de l'estimation de mouvement est d'éliminer la redondance entre des images successives dans une séquence. L'idée principale est de prédire l'image à l'instant $t+1$, connaissant les images aux instants $t-1$ et t . On recherche ce qui a bougé entre $t-1$ et t pour prédire ce qui bougera en $t+1$.

Nous utilisons un algorithme "pel-récursif". A chaque pixel est associé un vecteur de déplacement \mathbf{d} . On note $I(i, j, t)$ l'intensité du pixel à la ligne i , colonne j et à l'instant t .

Nous reprenons les hypothèses couramment utilisées pour l'estimation de mouvement [8]: 1) la luminosité est constante au cours du temps et 2) le mouvement \mathbf{d} est un mouvement de translation.

$$\begin{aligned} 1) \quad & I(i, j, t-1) = I(i', j', t) \\ 2) \quad & i' = i + u \quad j' = j + v \quad \text{avec } \mathbf{d}_{i,j}^t = (u(i,j), v(i,j)) \end{aligned} \quad (1)$$

Considérons la fonction DFD (Displaced Frame Difference) relative à chaque pixel de l'image :

$$DFD_{i,j} = I(i, j, t-1) - I(i', j', t) \quad (2)$$

Rechercher le champ de vecteurs mouvement de l'image, c'est chercher \mathbf{d} pour tout (i,j) en minimisant la DFD . [8].

Malheureusement cette minimisation appartient à l'ensemble des problèmes "mal posés" car il n'y a pas unicité et stabilité de la solution par rapport aux observations.

Le problème est reformulé en utilisant la théorie de la régularisation au sens de Tikhonov. Il s'agit de rajouter une contrainte a priori de douceur sur le champ de vecteurs. On suppose alors que le champ est régulier, i.e. que le gradient du champ déplacement est faible. Donc un vecteur dépend fortement de ses voisins.

Le critère à minimiser s'écrit alors:

$$C(\mathbf{d}) = \sum_{i,j} DFD_{i,j}^2 + \lambda \sum_{i,j} grad^2(\mathbf{d}_{i,j}) \quad (3)$$

Les solutions régularisées classiques obtenues après minimisation de fonctionnelles convexes (3) sont globalement lisses. Elles ne rendent pas compte des ruptures locales du champ de mouvement réel.

Pour y remédier, nous modélisons \mathbf{d} , la solution à estimer par des champs de Markov. Les processus de ligne ou de colonne permettent de tenir compte des discontinuités locales du champ [6].

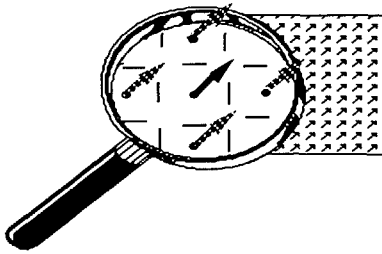


Figure 3: Champ de vecteurs déplacement avec réseau dual des processus de ligne.

On désigne par le terme général $p_{m,n} \in \{0,1\}$ les termes des processus de ligne ou de colonne (cf. figure 3). Lorsque $p_{m,n} = 0$ le pixel est lié avec son voisin (forte interaction locale). Lorsque $p_{m,n} = 1$ les pixels voisins sont indépendants, il y a présence d'une discontinuité.

Maximiser la probabilité a posteriori (MAP) revient à minimiser le critère suivant en fonction de \mathbf{d} (déjà minimisé par rapport à $p_{m,n} \in \{0,1\}$):

$$C(\mathbf{d}) = \sum_{i,j} DFD_{i,j}^2 + \sum_{i,j} \left\{ g_{\alpha,\lambda} \left(grad^2(\mathbf{d}_{i,j}) \right) \right\} \quad (4)$$

$$\text{où } g_{\alpha,\lambda} = \begin{cases} \lambda x^2 & \text{si } |x| < \sqrt{\alpha}/\lambda \\ \alpha & \text{si } |x| > \sqrt{\alpha}/\lambda \end{cases} \text{ est une fonction d'interaction}$$

locale (cf. figure 4). Si le gradient est inférieur à $h = \sqrt{\alpha}/\lambda$, la pondération est une loi quadratique, alors que si le gradient est supérieur à h , la pondération vaut α qui correspond au coût d'une discontinuité.

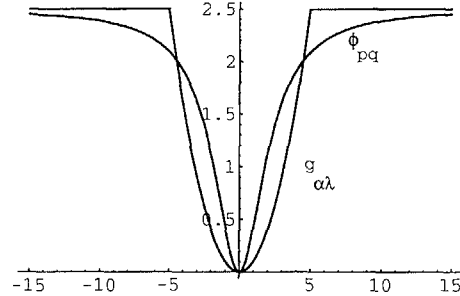


Figure 4: Fonctions de pondération correspondant à un processus booléen $g_{\alpha,\lambda}$ et à un processus continu $\phi_{p,q}$.

Cette quadratique tronquée n'est pas dérivable au point $h = \sqrt{\alpha}/\lambda$ ce qui rend la solution sensible au bruit et, le réglage des paramètres délicat.

Nous proposons une nouvelle fonction d'interaction locale ϕ dérivable en tout point et associée à des processus de ligne continus [3]. La relation entre un pixel et son voisinage est continue (cf. figure 4).

$$\phi_{p,q}(x) = \frac{x^2}{p^2 + q^2 x^2} \quad (5)$$

Elle appartient à la famille des ϕ fonctions de Geman et McClure [6]. Le critère à minimiser devient:

$$C(\mathbf{d}) = \sum_{i,j} DFD_{i,j}^2 + \sum_{i,j} \phi_{p,q} \left(grad^2(\mathbf{d}_{i,j}) \right) \quad (6)$$

Le critère est non convexe puisque le terme de régularisation n'est pas quadratique en \mathbf{d} . Nous proposons un algorithme de *relaxation déterministe* qui permet de minimiser la fonctionnelle par minimisations successives d'une suite de fonctions convexes.

Le principe est de séparer la fonction ϕ en deux

$$\phi_{p,q}(x) = \frac{1}{p^2 + q^2 x^2} x^2 = \varphi(x) \cdot x^2 \quad (7)$$

Lorsque φ est fixée pour une valeur de $x = x_s$, alors ϕ est quadratique par rapport à la variable x . φ rend compte de la nature des discontinuités.

$$C(\mathbf{d}) = \sum_{i,j} DFD_{i,j}^2 + \sum_{i,j} \varphi \left(grad(\mathbf{d}_{i,j_s}) \right) \cdot grad^2(\mathbf{d}_{i,j}) \quad (8)$$

Algorithme de minimisation [3]:

- 1/- Initialisation
Numéro d'étape $s = 0$ et
Initialisation de la solution $\mathbf{d} = \mathbf{d}_0$,
- 2/- Etape s
minimiser la fonctionnelle convexe (9) connaissant \mathbf{d}_s
avec un algorithme itératif (Gauss Seidel),
- 3/- Changement d'étape
 $s = s + 1$, jusqu'à convergence.

La solution à l'étape s est calculée avec un algorithme de type

Gauss-Seidel. A l'itération $k+1$, la solution au site i,j est donnée par :

$$u_{i,j}^{k+1} = \overline{u_{i,j}^{k+1,k}} - \frac{DFD_{i,j} I_x(i-u_{i,j}^k, j-v_{i,j}^k, t-1)}{4\lambda} \quad (9)$$

$$v_{i,j}^{k+1} = \overline{v_{i,j}^{k+1,k}} - \frac{DFD_{i,j} I_y(i-u_{i,j}^k, j-v_{i,j}^k, t-1)}{4\lambda}$$

Où $\overline{u_{i,j}^{k+1,k}}$ et $\overline{v_{i,j}^{k+1,k}}$ sont des moyennes pondérées [3].

Par ailleurs, nous utilisons une analyse multirésolution par transformée en ondelettes qui accélère la minimisation. Les images sont décomposées sur plusieurs niveaux de résolution (stratégie "coarse-to-fine"). Le champ vectoriel de déplacement est d'abord estimé sur les basses résolutions, puis raffiné avec les résolutions supérieures. A une résolution donnée, le champ initial d_s est celui qui a été trouvé à la résolution inférieure.

IV. QUANTIFICATION VECTORIELLE ALGEBRIQUE

Dans ce chapitre nous présentons une méthode de quantification vectorielle d'erreur de prédiction en utilisant la quantification vectorielle algébrique (QVA).

IV.1. Principe

Les images d'erreur de prédiction (IEP) sont quantifiées vectoriellement puis transmises au récepteur. Les vecteurs source sont des blocs de pixels des IEP et ont des densités de probabilité qui peuvent être approchées par des lois laplaciennes.

$$f_X(x) = \frac{1}{(\sigma\sqrt{2})^n} e^{-\frac{\sqrt{2}}{\sigma} \sum_{k=1}^n |x_k|} \quad (10)$$

où σ est l'écart type et n la dimension des vecteurs.

La QVA est une quantification vectorielle dont le dictionnaire est une partie d'un réseau régulier, c'est-à-dire un ensemble fini de points régulièrement disposés dans l'espace. L'intérêt de ces quantificateurs est la vitesse de quantification par des algorithmes rapides [10]. C'est la différence avec les techniques basées sur l'algorithme LBG pour lequel les vecteurs de reproduction sont disposés dans l'espace en fonction de la séquence d'apprentissage. Ce manque d'ordre oblige à utiliser des méthodes de "full-search" plus ou moins élaborées pour rechercher le vecteur de reproduction.

La QVA a été largement étudiée pour la quantification de sources gaussiennes [10] et laplaciennes pour des réseaux simples[9]. Nous avons étendu l'étude à des réseaux denses pour des sources dont le modèle de la densité de la probabilité est une laplacienne.

IV.2. Définition

Un réseau n -dimensionnel λ_n est défini comme l'ensemble des vecteurs satisfaisant

$$\lambda_n = \{Y / Y = u_1 a_1 + \dots + u_n a_n\} \quad (11)$$

où a_i sont des vecteurs linéairement indépendants dans R^m avec $m \geq n$ et u_i des entiers.

Dans nos applications nous utilisons les réseaux D_n , E_8 , Λ_{16} et Z^n [10].

Pour $n \geq 3$, le réseau D_n est défini par

$$D_n = \left\{ Y = (y_1, y_2, \dots, y_n) \in Z^n / \sum_{k=1}^n y_k \text{ est paire} \right\} \quad (12)$$

Les réseaux E_8 et Λ_{16} sont construits à partir des réseaux D_8 et D_{16} . Le réseau E_8 est défini par $E_8 = D_8 \cup \left(\frac{1}{2} + D_8\right)$ où $\frac{1}{2}$ est le vecteur tout à 1/2. Le réseau de Barnes-Wall, Λ_{16} est construit par $\Lambda_{16} = \bigcup_{i=0}^{32} (r_i + 2D_{16})$ avec r_i les vecteurs de translation définis à partir d'une matrice de Hadamard.

IV.3. Troncature et Dénombrement

Le réseau choisi est tronqué suivant une hyperpyramide de rayon m . L'ensemble des points du réseau d'énergie inférieure à m , forme le dictionnaire \mathcal{Y} .

$$\mathcal{Y} = \left\{ Y \in \Lambda_n / \sum_{k=1}^n |x_k| \leq m \right\} \quad (13)$$

Une hyperpyramide définie par $\sum_{k=1}^n |x_k| = \text{constante}$ correspond à

une surface d'équiprobabilité pour les vecteurs de source. Chaque vecteur de la source est projeté à l'intérieur du dictionnaire avec un facteur d'échelle. Ce facteur d'échelle est défini par $\gamma = E_{\max}/m$ pour que les vecteurs d'énergie E_{\max} soient quantifiés par les points en surface du dictionnaire. Les vecteurs d'énergie supérieure sont traités séparément. Pour plus de détails, le lecteur est invité à lire [2].

Les vecteurs sont quantifiés puis codés et transmis. Côté récepteur, il suffit de décoder et d'appliquer l'inverse du facteur d'échelle.

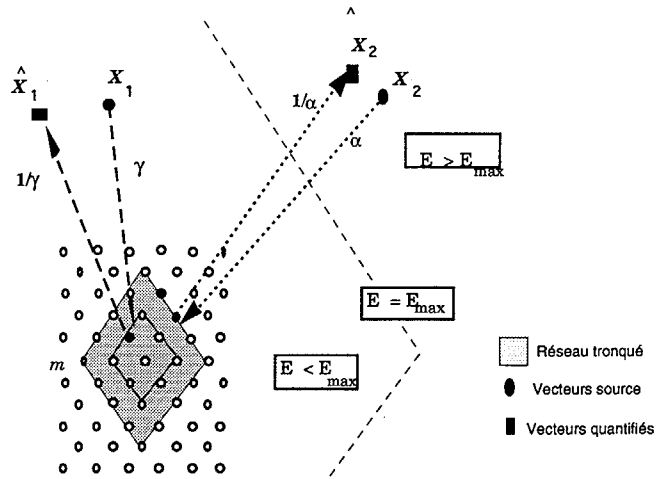


Figure 5: Méthode de quantification.

Le réseau choisi est tronqué suivant une surface pyramidale pour former le dictionnaire. La source X est dilatée par un facteur d'échelle γ pour que les vecteurs source d'énergie inférieure à E_{\max} soient quantifiés par un vecteur du dictionnaire. Les autres sont traités à part. Côté décodeur, il suffit d'appliquer $1/\gamma$ pour obtenir le vecteur décodé.

Pour énumérer les points du dictionnaire, nous avons introduit dans [2] les fonctions génératrices V .

$$V_{\Lambda}(z) = \sum_{m=0}^{+\infty} N_m z^m \quad (14)$$

Cette fonction de la variable z , indique le nombre N_m de points du réseau Λ appartenant à une hyperpyramide de rayon m . Le nombre total



de points du dictionnaire est $\mathcal{X}_m = \sum_{i=0}^m N_i$.

Séries V des réseaux utilisés:

$$V_{D_4} = 1 + 32z^2 + 192z^4 + 608z^6 + 1408z^8 + 2720z^{10} + \dots$$

$$V_{E_8} = 1 + 128z^4 + 2944z^8 + 1024z^{12} + \dots$$

$$V_{\Lambda_{16}} = 1 + 512z^4 + 47872z^8 + 92160z^{10} + \dots$$

V. RESULTATS

La séquence test, de 16 images de type non entrelacées, a été comprimée à un débit de 0,25 bits par pixels, soit une compression de 32. L'estimateur de compensation a été choisi avec une modélisation markovienne tenant compte des discontinuités de champ, comme nous l'avons présenté au paragraphe III. Les minimisations successives ont été réalisées par une méthode de relaxation auto-adaptative. Deux champs de vecteurs, avec et sans gestion des discontinuités, sont comparés à la figure 7.

La figure 8 représente la densité de probabilité d'une image d'erreur de prédiction. On remarquera la bonne similarité avec une loi laplacienne.

Le réseau D_4 a été choisi pour la quantification vectorielle algébrique. Nous avons figé les paramètres du quantificateur et travaillé à débit constant sur toute la séquence. Par conséquent, la qualité (mesurée par le Pic SNR) du résultat décroît en fonction du temps. L'erreur de quantification s'accumule au fil des images. Pour conserver une qualité constante sur toute la séquence il faudrait, soit rafraîchir les données (blocs de 16 images par exemple), soit asservir le quantificateur en fonction de la qualité du résultat.

De plus une transformée en ondelettes, sur peu de niveaux de résolution, a été utilisée pour décomposer les IEP. On peut ainsi réorganiser l'information contenue dans ces images pour mieux quantifier. Il suffit d'allouer plus ou moins de bits selon les résolutions.



Figure 6: Image n°1 de la séquence Interview.

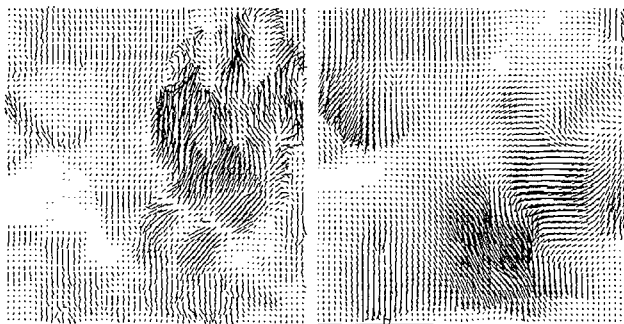


Figure 7: Champ de vecteurs déplacement avec et sans gestion des discontinuités.

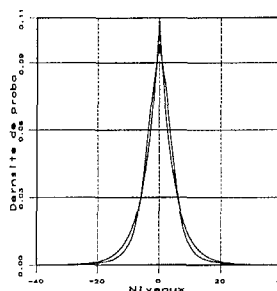


Figure 8: Comparaison entre la densité de probabilité d'une image d'erreur de prédiction et une loi laplacienne.

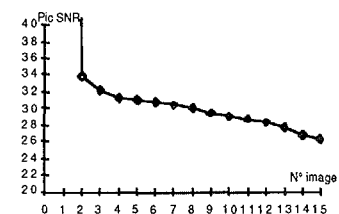


Figure 9: Courbe de qualité en fonction du n° image

VI. CONCLUSION

Nous avons présenté un algorithme de compression de séquences d'images comprenant un module d'estimation-compensation de mouvement et une quantification vectorielle géométrique.

L'estimation de mouvement proposée prend en compte de manière continue les discontinuités du champ de mouvement. Ceci améliore les résultats d'estimation dans les régions de contours et ainsi diminue les erreurs de prédiction.

L'utilisation de la quantification vectorielle algébrique permet d'utiliser des algorithmes de quantification rapide. Les dictionnaires définis par troncature de réseaux suivant une forme pyramidale sont "adaptés" à la statistique des images d'erreur de prédiction.

Références:

- [1] N. Baaziz, "Approches d'Estimation et de Compensation de Mouvement Multirésolutions pour le Codage de Séquences d'Images", Thèse de doctorat, Université de Rennes I, Octobre 1991.
- [2] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, P. Mathieu, "Pyramidal Lattice Vector Quantization for Multiscale Image Coding", accepté avec des révisions mineures à IEEE Transaction on Image Processing
- [3] M. Barlaud, L. Blanc-Féraud, T. Gaidon, "Motion Estimation Involving Discontinuities in a Multiresolution Scheme" Revue Optical Engineering de Juillet 1993.
- [4] A. Blake, A. Zisserman, "Visual Reconstruction", MIT Press : Cambridge, MA. 1987
- [5] T. Gaidon, M. Barlaud, "Image Sequence Algorithm based on Motion Compensation and Pyramidal Lattice Vector Quantization", Picture Coding Symposium'93, Lausanne, Suisse, 17-19 Mars 1993.
- [6] S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, pp. 721-741, Nov. 1984.
- [7] B.K.P Horn, B.G. Schunck, "Determining Optical Flow", Artificial Intelligence, Vol 17 pp 185-203, 1981.
- [8] A.N. Netravali, J.D. Robbins, "Motion Compensated Television Coding part I", Bell System Technical Journal, 58(3) : 629-668, 1979.
- [9] T. R. Fischer "Entropy - Constrained Geometric Vector Quantization for Transform Image Coding", IEEE ICASSP, pp 2269-2272, Toronto CANADA, May 1991.
- [10] J.H. Conway, N.J.A. Sloane, "Sphere Packings, Lattices and Groups", Springer Verlag, 1988.
- [11] T. Gaidon, M. Barlaud, P. Mathieu, "Image Sequence Coding using Quincunx Wavelet Transform, Motion Compensation and Lattice Vector Quantization", VCIP'92, Boston USA, November 1992.