

THEME 12 : IMAGES

PRESENTATION D'UN CODEC VISIOPHONE A 64 Kbit/s
PRESENTATION OF A 64 Kbit/s VIDEOPHONE CODEC

ROBERT Philippe

THOMSON-CSF - LABORATOIRES ELECTRONIQUES DE RENNES
AVENUE DE BELLE-FONTAINE - 35510 CESSON-SEVIGNE - FRANCE

Nous présentons une méthode de compression d'images pour le service visiophonique, à un débit de 64Kbit/s. Un tel débit nécessite une compression considérable et l'emploi de techniques de codage complémentaires performantes. Le codec présenté est basé sur les principes suivants :

- Codage à compensation de mouvement
- Transformée cosinus discrète
- Quantification adaptative
- Sous échantillonnage de trame + interpolation adaptée au mouvement

La juxtaposition de telles techniques permet d'obtenir des résultats satisfaisants compte tenu du taux de compression requis.

1 - INTRODUCTION

Le RNIS (Réseau Numérique à Intégration de Services) doit fournir prochainement un débit numérique de 2 canaux à 64 kbit/s (+ une voie de signalisation et de données à 16 kbit/s) [1]. Pour le service visiophonique, la transmission d'images à un débit de 64 kbit/s (l'autre canal étant destiné au son) s'intégrerait parfaitement dans ce contexte, et est, par conséquent très attrayante.

Cependant, un tel débit nécessite une compression considérable (environ 360 en luminance pour le format d'images employé : 352 pixels x 288 lignes à 25 Hz) et donc un codage -décodage très performant.

Par ailleurs, le service visiophonique concerne "la transmission réciproque des portraits des correspondants". Pratiquement, l'image est générée par le constituant de la tête ou du buste d'un personnage s'animant modérément. Le fond est fixe.

Dans ces conditions, la réduction de débit nécessite la juxtaposition de plusieurs techniques de codage performantes et complémentaires.

Le codec présenté dans cet article exploite des techniques de compression temporelles et spatiales, connues pour la plupart mais qui ont été peu associées jusqu'à présent [2]. La structure de base du codeur est une boucle DPCM inter-images. L'erreur de prédiction, obtenue par compensation de mouvement, est comprimée spatialement sous forme de blocs 16 x 16 par transformée cosinus. Par ailleurs, la classification des blocs et la quantification des coefficients s'adaptent au contenu de l'image pour répartir au mieux l'information binaire ; les paramètres associés sont ajustés en fonction du contenu de la mémoire tampon.

La fréquence des trames codées et transmises au récepteur est de 8,33 Hz : ceci permet un taux de compression de 6. Au décodeur, les trames manquantes sont reconstruites par interpolation adaptée au mouvement.

An image codec for videophone at a 64 Kbit/s rate is presented. Such a rate implies a high compression with complementary and effective coding techniques. This codec is based on the following points :

- Motion compensation coding
- Discrete Cosine Transform
- Adaptive Quantization
- Field subsampling + Motion adaptive interpolation

Joining such techniques provides satisfying results in comparison with the compression rate.

Le principe du codec est illustré en figures 2 et 3. Ses différents blocs fonctionnels sont précisés dans les paragraphes suivants. Il s'agit de :

- Compensation de mouvement
- Transformée Cosinus Discrète
- Classification de Bloc
- Quantification Adaptative
- Régulation
- Codage
- Interpolation adaptée au Mouvement

2 - COMPENSATION DE MOUVEMENT

Elle constitue l'étape de décorrélation inter-images. Le signal de sortie est l'erreur de prédiction :

$$DFD(x,y,t) = I(x,y,t) - \tilde{I}(x-dx, y-dy, t-\tau)$$

La trame $t - \tau$ étant celle précédemment codée puis décodée et stockée en mémoire M1, et dont dispose également le récepteur.

La corrélation exploitée est celle des zones fixes ou déplacées.

Le déplacement inter-trames est déterminé par Block Matching : le modèle du mouvement est une translation des blocs dans le plan d'image [3].

La trame courante t étant considérée comme un ensemble de blocs $M \times N$ adjacents, chaque bloc B est traité ainsi :

- recherche du bloc B_m dans $t - \tau$ qui minimise la fonction d'erreur $C(dx, dy)$:

$$C(dx, dy) = \sum_{\text{Bloc}} |I(x,y,t) - I(x-dx, y-dy, t-\tau)|$$

- un vecteur mouvement $\vec{D} = (dx, dy)^T$ est ainsi attribué à chaque bloc B. Les vecteurs sont codés par un code d'Huffman et transmis au récepteur



La taille du bloc résulte en partie d'un compromis entre la minimisation de l'information mouvement à transmettre (grande taille) et d'une minimisation effective de l'information luminance par compensation de mouvement (petite taille) : couramment 8x8 ou 16x16.

La recherche du minimum s'effectue ainsi :

- détection de mouvement : application du mouvement nul et comparaison de C(0,0) au seuil S1 :

- . si $C(0,0) < S_1$ alors $\vec{D} = (0,0)^T$ (bloc fixe)
- . si $C(0,0) \geq S_1$ le bloc est détecté en-mouvement

- estimation du mouvement : application d'une prédiction spatiale du mouvement \vec{D}_p :

- . si $C(\vec{D}_p) < S_1$ $\vec{D} = \vec{D}_p$
- . si $S_1 \leq C(\vec{D}_p) \leq S_2$: recherche fine (pas de 1) autour de \vec{D}_p dans la fenêtre [-3, +3] en x et y.
- . si $C(\vec{D}_p) > S_2$: recherche grossière (pas de 2) autour de $\vec{D}_0 = (0,0)^T$ dans la fenêtre [-7, +7] en x et y.

De meilleurs résultats sont obtenus avec une fonction de correction calculée entre images originales plutôt qu'entre images courante originale et prédente reconstruite : en particulier, en début de séquence, le codage converge beaucoup plus vite dans le premier cas.

3 - TRANSFORMEE COSINUS DISCRETE (DCT)

Elle a pour effet de concentrer l'énergie des blocs sur un petit nombre de coefficients regroupés spatialement. Elle est décrite ainsi :

$$F(u,v) = \frac{4c(u)c(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} I(x,y) \cdot \cos \left[\frac{(2j+1)u\pi}{2N} \right] \cdot \cos \left[\frac{(2k+1)v\pi}{2N} \right]$$

$$\text{pour } u,v = 0,1,\dots,N-1 \text{ où } C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } w=0 \\ 1 & \text{pour } w = 1, 2, \dots, N-1 \end{cases}$$

L'efficacité de la DCT relativement aux autres transformées a déjà été largement démontrée. Il existe par ailleurs des algorithmes de calcul rapide [4].

4 - CLASSIFICATION DES BLOCS

On distingue 4 classes de blocs selon 2 critères :

- . bloc fixe ou en mouvement : le critère est la valeur du vecteur mouvement associé au bloc : le bloc est fixe si les composantes dx et dy sont nulles.
- . compensé ou non compensé : pour cela, un critère d'activité est défini sur le bloc :

$$\text{ENERG} = \sum_{\text{Bloc}} |F(u,v)|$$

si $\text{ENERG} < \text{SEU1}$ le bloc est compensé
sinon le bloc est non compensé.

L'information transmise au décodeur dépend du type du bloc : aucune information concernant les coefficients (classés non significatifs) n'est transmise pour les blocs compensés (fixes ou en mouvement).

Pour chacune des 4 classes, l'information suivante est transmise :

- bloc non compensé, fixe : coefficients quantifiés
- bloc compensé, en mouvement : vecteur déplacement
- bloc non compensé, en mouvement : vecteur déplacement + coefficients quantifiés.
- bloc compensé, fixe : aucune information n'est transmise. Les blocs des 3 autres classes sont adressés relativement au bloc précédemment codé sur la même ligne, ou à défaut, au début de ligne.

5 - QUANTIFICATION DES COEFFICIENTS

Si la compensation de mouvement est adéquate, les erreurs de prédiction (DFD, avant Transformation DCT) sont décorrelés (le Bloc est compensé). L'imprécision du modèle crée des défauts de présence de contours. Ceci se traduit par une répartition structurée des "coefficients significatifs" très différente selon la structure des défauts dans le bloc d'erreurs de prédiction. Le quantificateur-codeur, tout en sélectionnant les "coefficients significatifs", doit s'adapter à leur répartition dans le bloc.

Un quantificateur à allocation de bits [4] est trop coûteux en débit binaire pour notre cas. On lui préfère une quantification unique pour tous les coefficients (sauf F(0,0)) [5], qui s'adapte mieux à la situation décrite ci-dessus.

Les coefficients sont seuillés, normalisés, puis quantifiés ainsi :

$$\begin{cases} \text{si } F(u,v) > \text{SEU2} : F_q(u,v) = \text{Partie entière de } [(F(u,v) \cdot \text{SEU2}) / \text{PAS} + 0,5] \\ \text{sinon} : F_q(u,v) = 0 \end{cases}$$

Les niveaux de décision sont les suivants :

$$\alpha_n = \text{SEU2} + (n-1) \text{PAS}$$

F(0,0) est codé linéairement sur 9 bits.

Les coefficients nuls sont codés par plage. Du fait des diverses structures de répartitions des coefficients non nuls observables, le bloc est parcouru de F(0,0) à F(15,15) par 3 chemins différents, chacun privilégiant une structure afin de minimiser la quantité et la longueur des plages de 0 (voir figure 1). Le balayage retenu est celui qui minimise la quantité de bits transmise.

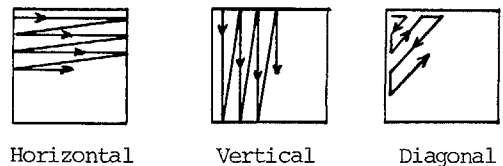


FIGURE 1 - BALAYAGE DU BLOC

6 - REGULATION

Les valeurs des paramètres SEU1 (classification des blocs), SEU2 et PAS (quantification) sont fixés selon l'état de la mémoire tampon à chaque fin de ligne, et transmis au récepteur. La mémoire tampon est virtuellement découpée en 9 intervalles pour chacun desquels un triplet (SEU1, SEU2, PAS) est fixé (voir figure 4). SEU2 est en général fixé à 1,5xPAS sauf lorsque la capacité de la mémoire tampon est dans l'état 1 ou 2 : dans ce cas, SEU2 = PAS

7 - CODAGE DE L'INFORMATION TRANSMISE

La plupart des informations sont codées par des codes à longueur variable. Quatre codes d'Huffman ont été définis.

Premier code :

- . adressage du bloc en mouvement ou/et compensé : si le bloc précédent est lui aussi à coder, aucune adresse n'est envoyée.
- . classe du bloc : non compensé fixe, non compensé en mouvement, compensé en mouvement
- . fin de ligne, fin d'image.

Second code : vecteurs mouvement (Blocs en mouvement)

Troisième code :

- . coefficients non nuls
- . plages de zéros
- . fins de blocs : la dernière plage de zéro n'étant pas transmise.

Quatrième code : les paramètres de régulation

Le mode de balayage du bloc (horizontal, vertical ou diagonal) est codé sur 2 bits.

Les informations se succèdent conformément à la figure 5.

8 - RECONSTRUCTION DES TRAMES MANQUANTES AU RECEPTEUR

Cinq trames sur six sont reconstruites à partir des trames codées transmises au récepteur. L'interpolation de ces trames est adaptée au mouvement estimé entre les trames codées. Elle comprend les étapes suivantes :

- a) l'estimation du mouvement entre les 2 trames-mères successivement codées. Pour cela, un algorithme récursif point à point a été développé (du type développé en [6]). Il fournit un champ de mouvement dense et précis représentant l'évolution de la scène entre les 2 trames-mères.
- b) la définition du champ de mouvement pour la trame omise et son interpolation dans la direction du mouvement associé.

9 - SIMULATIONS - CONCLUSIONS

Le codec a été simulé sur une séquence visio-phonie de 250 images ("Alexis"). On peut y distinguer 2 parties selon la quantité de mouvement inter-images : une première partie (images 1 à 170) peu animée, la 2^{ème} partie (images 170 à 250) étant plus mouvementée.

Les résultats présentés concernent le codage de la luminance à 56 Kbit/s.

Concernant le Block Matching, 2 configurations ont été envisagées :

- a) taille de bloc 8x8 : les blocs 8x8 sont ensuite concaténés par 4 pour constituer un bloc 16x16 de DFD. Quatre vecteurs mouvement sont envoyés pour chaque bloc 16x16 en mouvement.
- b) taille de bloc 16x16.

L'emploi d'un Block Matching 16x16 conduit à un rapport signal sur bruit supérieur d'environ 0,5 dB (en moyenne) à celui d'un Block Matching 8x8 (figure 6, Photos 1 et 2). Son intérêt par rapport à la différence inter-images est clair dès que le mouvement devient important : c'est le cas dans la 2^{ème} partie de la séquence (Figure 6, Photos 2 et 3).

De même, la supériorité de l'interpolation adap-

tée au mouvement apparaît dans les photos 4 et 5 correspondant à l'image 180, où le déplacement est le plus élevé.

La quantité d'informations mouvement transmise fluctue entre 0 % et 10 % du débit. L'adressage présente un débit stable, la juxtaposition de blocs codés ne créant pas d'adressage supplémentaire (3 à 4 % du débit).

Le retard entre codage et restitution, créé essentiellement par la mémoire tampon (6000 bits) et l'interpolation adaptée au mouvement, est de l'ordre de 270 ms, ce qui est acceptable.

Les résultats sont satisfaisants compte tenu du taux de compression. La poursuite de l'étude va porter essentiellement sur l'amélioration de la distribution des bits entre les blocs et entre les coefficients, par introduction de nouveaux critères subjectifs.

10 - BIBLIOGRAPHIE

- (1) - F. du CASTEL - "SERVICES ET RESEAUX DE TELECOMMUNICATIONS"
L'ECHO DES RECHERCHES N° 123
1er trimestre 1986
- (2) - G. KUMMERFELDT, F. MAY, W. WOLF
"CODING TELEVISION SIGNALS AT 320 AND 64 KBIT/S"
2nd INTERNATIONAL TECHNICAL SYMPOSIUM ON OPTICAL AND ELECTRO-OPTICAL APPLIED SCIENCE AND ENGINEERING - CANNES, DECEMBRE 1985
- (3) - J.R. JAIN & A.K. JAIN
"DISPLACEMENT MEASUREMENT AND ITS APPLICATION IN INTERFRAME IMAGE CODING"
IEEE VOL. COM-29 N° 12 DECEMBRE 1981
- (4) - W.H. CHEN & C.H. SMITH
"ADAPTIVE CODING OF MONOCHROME AND COLOR IMAGES"
IEEE VOL. COM-25 N° 11 NOVEMBER 1977
- (5) - W.H. CHEN & W.K. PRATT
"SCENE ADAPTIVE CODER"
IEEE VOL. COM-32, N° 3 MARCH 1984
- (6) - P. ROBERT, C. CAFFORIO & P. ROCCA
"TIME AND SPACE RECURSIONS FOR DIFFERENTIAL MOTION ESTIMATION"
2nd INTERNATIONAL TECHNICAL SYMPOSIUM ON OPTICAL AND ELECTRO-OPTICAL APPLIED SCIENCE AND ENGINEERING CANNES, DECEMBRE 1985

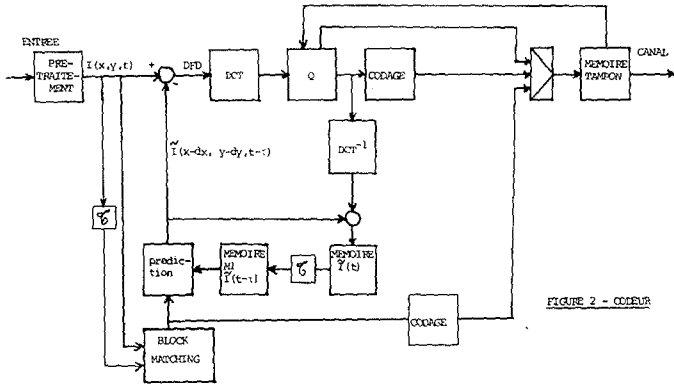


FIGURE 2 - CODEUR



PHOTO N° 1 - Trames 1 des images 181 et 211 (Block Matching 8x8)

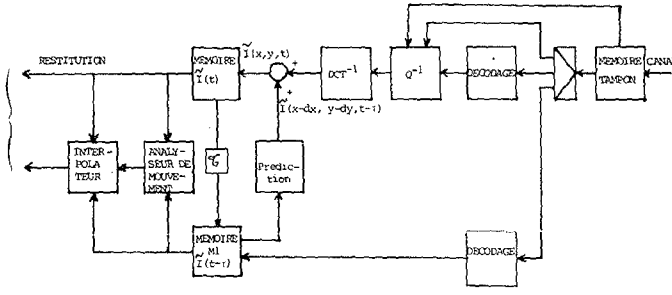


FIGURE 3 - DECODEUR

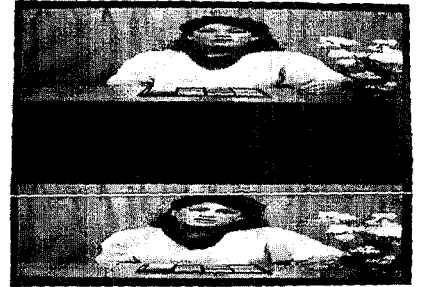


PHOTO N° 2 - idem (Block Matching 16x16)

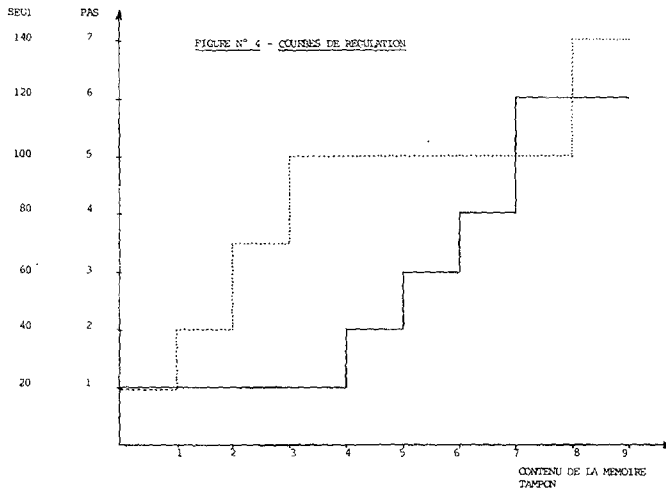


FIGURE N° 4 - COURSES DE REGULATION



PHOTO N° 3 - idem (différence inter-images)

CLASSE DU BLOC	INFORMATION TRANSMISE AU RECEPTEUR	
	(ADRESSE)	MOUVEMENT
FIXE COMPENSE	AUCUNE INFORMATION	
EN MOUVEMENT, COMPENSE		
FIXE, NON COMPENSE		BALAYAGE COEFFICIENTS NON NULS LONGUEURS DE PLACES DE ZERO FIN DE BLOC
EN MOUVEMENT, NON COMPENSE	CLASSE	MOUVEMENT

FIGURE N° 5 - INFORMATIONS TRANSMISES SELON LA CLASSE DU BLOC



PHOTO N° 4 - Interpolation adaptée au mouvement (Image N° 198)

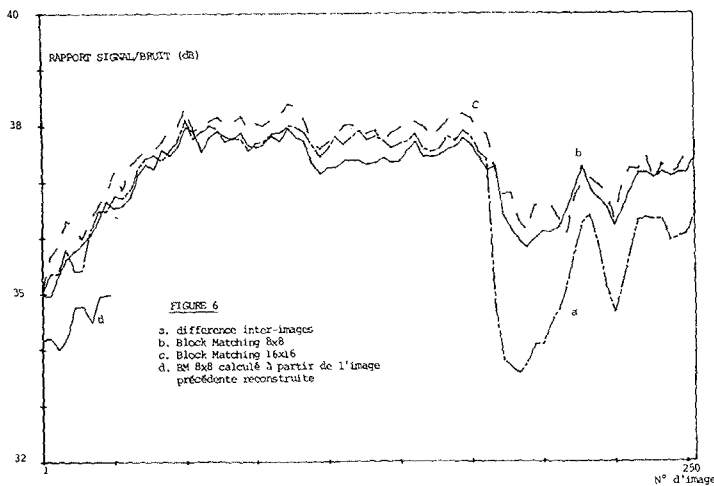


FIGURE 6
a. différence inter-images
b. Block Matching 8x8
c. Block Matching 16x16
d. 8x8 calculé à partir de l'image précédente reconstruite



PHOTO N° 5 - Interpolation linéaire simple (image N° 198)