DIXIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

PRESENTATION D'UNE ARCHITECTURE VLSI EN SYSTOLIQUE POUR LA DCT

C. BOZZO - M. FOUQUES - R. GABRIEL - A. LORENZI

CSEE/CETIA - 150 Rue Marcellin Berthelot - Z.I. Toulon-Est - 83008 TOULON CEDEX

RESUME

Dans le cadre du problème de la compression d'images, on propose de mettre en évidence une architecture VLSI spécialisée pour une transformation fonctionnelle particulière. (DCT ou transformée discrète en cosinus)

On présente, dans un premier temps, différentes approches possibles pour le calcul d'une transformation fonctionnelle appliquée à la compression d'images.

Puis, en fonction de certains critères de performances, le choix de la DCT est justifié le schéma algorithmique équivalent présenté.

On adopte enfin une approche systolique pour déterminer l'architecture VLSI associée.

SUMMARY

In the context of image coding, propose a specialized VLSI architecture for a special functional transform :

The Discrete Cosine Transform.

We present, in a first step, different possible approaches for the computation of a functional transform for image coding.

Then, taking into account certain performance criteria, we will explain our choice for the DCT and will give the equivalent algorithmic sheme.

Finally, we will adopt a systolic approach to determine the associed VLSI architecture.

PRESENTATION D'UNE ARCHITECTURE VLSI EN SYSTOLIQUE POUR LA DCT

I - INTRODUCTION

De nombreuses applications de traitement d'image comme la télévision numérique, la téléconférence ou l'imagerie médicale nécessitent la mise en oeuvre de CODECS appropriés à une compression efficace des données caractérisant l'information véhiculée par l'image.

Ce type d'images devra être restitué aussi fidèlement que possible après stockage ou transmission en fonction de critères de qualité visuelle souvent très subjectifs.

Des exigences excessives au niveau de la mémoire de stockage et de la largeur de bande passante sont alors implicites et un CODEC doit par conséquent servir à réduire de façon efficace le flot des données afin de diminuer les contraintes au niveau du stockage et de la transmission.

Les nombreuses études effectuées dans le cadre du problème de la compression d'images ont conduit à distinguer deux classes principales de méthodes :

- la première classe repose sur la notion de "codage prédictif" et la technique de mise en oeuvre est alors la Modulation d'Impulsions Codées Différentielle (MICD) qui utilise les propriétés de corrélation au voisinage d'un point de l'image,
- la deuxième classe réalise le codage à partir d'une transformation fonctionnelle de l'image originale (en général par blocs élémentaires) qui vont être présentés dans le chapitre suivant.

Pour ce type de transformation, les propriétés de corrélation entre les points de l'image originale se traduisent par une concentration de l'énergie (et donc de l'information) contenue dans l'image et impliquent un petit nombre de coefficients caractéristiques de l'image transformée.

Ce phénomène de compaction de l'énergie et de décorrélation des données conduit à l'élimination d'un certain nombre de coefficients peu énergétiques et permet un codage efficace en vue de la transmission ou du stockage.

Bien que plus coûteux à réaliser, ce type de méthode apparaît moins sensible aux erreurs de codage dont l'effet sera réparti uniformément sur l'image après transformation inverse, ce qui est visuellement moins gênant que dans le cas d'erreurs localisées résultant de l'exploitation des techniques de codage prédictif directement appliquées sur l'image originale.

En particulier, on effectue souvent une transformation fonctionnelle par blocs de l'image originale, suivie d'un codage différentiel (voir figure 1).

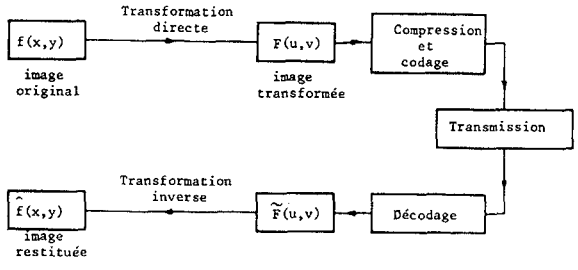


Figure 1 - Schéma fonctionnel simplifié d'un système de compression d'image

On notera le rôle important de la transformation fonctionnelle de l'image originale et de son inverse qui interviennent respectivement en entrée et en sortie du dispositif général de codage.

Dans ces conditions, nous allons, dans un premier temps, nous intéresser aux différents types de transformations fonctionnelles envisageables et nous mettrons en évidence les différents critères qui nous ont conduit au choix de la DCT (cf § 2).

Enfin (cf § 3) nous présenterons plusieurs architectures élaborées à partir d'une approche systolique.

II - PRESENTATIONS DES PRINCIPALES TRANSFORMATIONS FONCTIONNELLES

2.1 - Transformation d'HADAMARD [1]

Introduite par le mathématicien HADAMARD, cette transformation est définie à partir d'une matrice carrée H qui ne contient comme coefficients que les valeurs + 1 et - 1.

Si H est une matrice d'HADAMARD de dimension N, la matrice $G = \begin{bmatrix} H & H \\ H & -H \end{bmatrix}$ est aussi une matrice d'Hadamard

dimension 2N. Il est alors possible, à partir de la matrice élémentaire de dimension 2

$$H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

de construire successivement les matrices de dimension 4, 8, ...2^n.

Cette transformation apparaît alors comme effectuant la décomposition du signal pour un ensemble de signaux rectangulaires analogues aux signaux sinusoïdaux utilisés dans la transformation de Fourier. De plus, sa structure modulaire et la nature des coefficients (entiers signés) mis en jeu conduisent à des architecturs simples et rapides.

2.2 - Transformation de HAAR [2]

La transformation introduite par le mathématicien HARR, est une des méthodes les plus rapides rapides actuellement connues ; elle est exploitée notamment pour la transmission et l'identification des scènes.

Il existe de nombreuses sortes de fonctions de HAAR. Les plus usuelles, notées har (j,k,t), sont des ondes rectangulaires binaires, définies sur l'intervalle $[0, 1[$, et dont l'amplitude est une puissance de $\sqrt{2}$.

L'expression analytique des fonctions de Haar est :

$$\text{haar}\ (j,k,t) = \begin{cases} 2^{j/2} & \text{pour } (k-1)/2^j \leqslant t \leqslant (k-0,5)/2^j \\ -2^{j/2} & \text{pour } (k-0,5)/2^j \leqslant t \leqslant (k)/2^j \\ 0 & \text{ailleurs} \end{cases}$$

avec $j = 0,r \qquad k = 1,2^j$

Cette relation permet de trouver facilement les huit premières fonctions et de définir, en outre, les suivantes. La présence de racine carrée pose des problèmes évidents lors de l'implémentation de cet algorithme.

2.3 - Transformation de SLANT [3]

Nous présentons ici, une nouvelle transformée unitaire, la transformée de SLANT, utilisée spécialement pour le codage d'image.

Les matrices, mises en jeu dans cette transformée, sont construites itérativement comme produits de matrices peu denses. Pour une matrice de dimension 2x2, la transformée de SLANT est identique à la transformée d'HADAMARD d'ordre 2. Ainsi

$$S_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

La matrice d'ordre 4 est obtenue par l'opération suivante :

$$S_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & b_4 \\ 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{bmatrix} \times \left[\begin{array}{c|c} S_2 & \emptyset \\ \hline \emptyset & S_2 \end{array}\right]$$

Les valeurs a_4 et b_4 sont des constantes réelles qui sont déterminées suivant les conditions de S_4. La transformée de SLANT est une transformée orthogonale qui possède les propriétés suivantes :

- matrice de base constante,
- calcul récurrent,
- rapidité de calcul,
- compaction de l'énergie.

Remarque :

Il faut noter la présence de racine carrée dans le développement des calculs, ce qui rend plus complexe la réalisation "Hardware" de cet algorithme.

2.4 - Transformation de HARTLEY [4]

La transformation de HARTLEY est aussi rapide que la transformation de Fourier. Elle est utilisée dans deux applications principales :

- l'analyse spectrale,
- la convolution d'image.

Soit une fonction réelle f(z) pour z= 0,1, ... N-1. La transformée de HARTLEY est définie comme la somme des transformées en cosinus et en sinus :

$$H(\nu) = \frac{1}{N} \sum_{z=o}^{N-1} f(z) \cos(2\pi\nu z/N) \quad \text{avec } \nu = 0,1.... N-1$$

où $\cos\theta = \cos\theta + \sin\theta$

Remarque :

Partant de la transformée de HARTLEY, nous pouvons obtenir la transformée de FOURIER de la façon suivante. La partie réelle $R(\upsilon)$ est égale à la partie paire de la DHT et la partie imaginaire $X(\upsilon)$ est égale à la partie négative et impaire de $H(\upsilon)$. Le spectre de puissance peut être calculé directement à partir de

$$z^2 = \{[H(\upsilon)]^2 + [H(-\upsilon)]^2\}/2$$

Cette formule évite de passer par la quantité complexe R+jX, ce qui facilite l'implantation de l'algorithme de transformée de Fourier rapide.

2.5 - Transformation de Fourier [2]

La transformée discrète de Fourier s'écrit :

$$F(u,v) = \frac{1}{N^2} \sum_{x=o}^{N-1} \sum_{y=o}^{N-1} f(x,y) \exp\left[-2\pi j(ux+vy)/N\right]$$

dans le cas d'une transformation à deux dimensions d'une image de surface carrée.

Le calcul de cette transformée peut s'effectuer à l'aide de l'algorithme de FFT (Fast Fourier Transform)

2.6 - Transformation SVD

La transformation SVD est une transformation unitaire de dimension 2 basée sur la décomposition en valeurs singulières des matrices.

Nous ne développerons pas davantage ce type de transformation qui conduit à des calculs lourds ainsi qu'à des représentations complexes.

2.7 - Transformation en cosinus [5]

La transformée en cosinus, notée DCT, fut introduite en traitement numérique du signal par AHMED et AL en 1974. Cette transformation peut être utlisée en reconnaissance de formes et en filtrage de Wiener ; elle sera plus particulièrement étudiée dans le chapitre 3.

La DCT d'une séquence de données X(m) avec m = 0,1, ..., (M-1) est définie par :

$$G_x(o) = \frac{\sqrt{2}}{M} \sum_{m=o}^{M-1} X(m)$$

$$G_x(k) = \frac{\sqrt{2}}{M} \sum_{m=o}^{M-1} X(m) \cos\frac{(2m+1)k\pi}{2M}$$

avec k = 1,2,.. M - 1

L'ensemble des vecteurs de base $\{1/\sqrt{2}, \cos(2m/1)k\pi/2M\}$ est une classe des polynomes discrets de Chebyshev. Cet ensemble permet une bonne approximation des vecteurs propres de la classe des matrices de TOEPLITZ définies par :

$$\psi = \begin{vmatrix} 1 & \rho & \rho^2 ..... & \rho^{M-1} \\ \rho & & & \\ \vdots & & & \\ \rho^{M-1} & \rho^{M-2} & & 1 \end{vmatrix} \quad \text{avec } 0 < \rho < 1$$

Les vecteurs propres de ψ pour M = 8 et ρ = 0.9 sont représentés par la figure 2 ainsi que les vecteurs de base de la DCT.

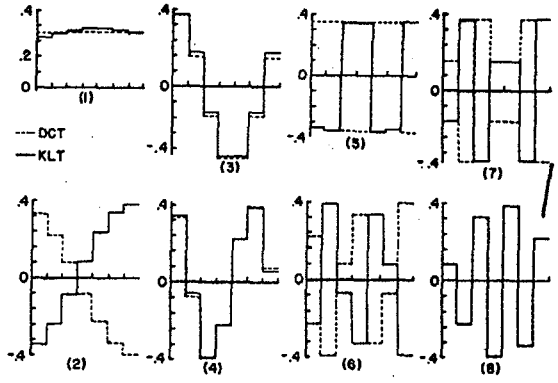


Figure 2 - Vecteurs propres de la matrice de TOEPLITZ (8x8) et vecteurs de base de la DCT

D'après la figure 2, nous pouvons remarquer que la DCT est une bonne approximation de la transformée de KARHUMEN-LOEVE.

La transformée discrète, bidimensionnelle en cosinus est représentée par l'équation suivante :

$$F(u,v) = \frac{2}{N} C(u)\, C(v) \sum_{j=0}^{N-1} \sum_{K=0}^{N-1} f(j,k) \cos\left[\frac{\pi}{N}[u(j+\frac{1}{2})]\right] \cos\left[\frac{\pi}{N}[v(k+\frac{1}{2})]\right]$$

La condition de normalisation impose :

$$C(k) = \sqrt{2}/N \text{ pour } k \neq 0$$
$$C(k) = 1/N \text{ pour } k = 0$$

2.8 - Transformation de KARHUMEN-LOEVE

Pour la classification et la compression d'images, on exploite souvent la transformation de KARHUMEN-LOEVE.

Cette transformation possède deux propriétés importantes qui sont :

- décorrélation des coefficients transformés,
- répartition de l'énergie maximale sur un nombre donné d'échantillons.

Selon les valeurs des paramètres de corrélation, on peut montrer qu'il existe des transformations sinusoïdales, fournissant de très bonnes approximations de la KLT (voir figure 3).

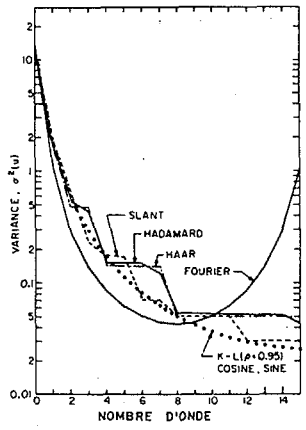


Figure 3 - Variances des coefficients de transformation unitaire N = 16, ρ= 0,9

La figure suivante représente l'erreur quadratique moyenne pour un certain nombre de transformations.

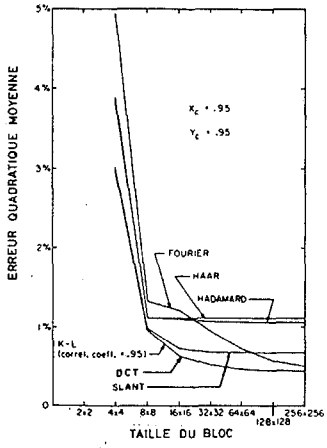


Figure 4 - Pourcentage de l'erreur quadratique moyenne en fonction de la taille du bloc

Notons à ce niveau que la DCT, beaucoup plus simple à mettre en oeuvre que la KLT, autorise des performances pratiquement équivalentes et en tous cas supérieures à celles que peuvent présenter les autres méthodes citées.

III - ARCHITECTURES DU CALCUL DE LA DCT

3.1 - Architecture simple

En utilisant la propriété de séparabilité, nous obtenons la figure 5 basée sur la décomposition de la transformée en cellules élémentaires.

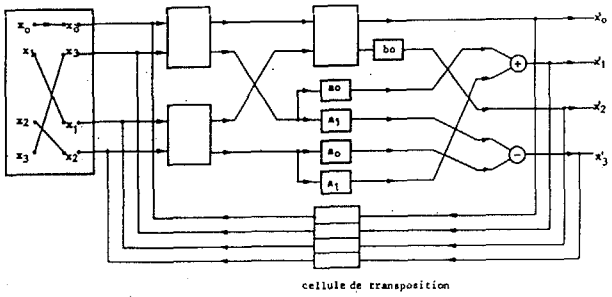
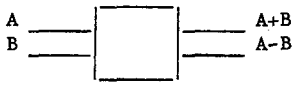


Figure 5 - Décomposition en cellules élémentaires de la DCT (N = 4)

Remarque :

La cellule élémentaire effectue les opérations suivantes :



Le tableau ci-dessus récapitule le nombre d'opérations nécessaires pour réaliser cette transformation.

Fenêtre	Nombre de cellules	de multiplications	d'additions et de soustractions
4 x 4	24	40	16
8 x 8	112	272	224
16 x 16	480	2720	1120

Tableau 1 : nombre d'opérations par blocs

3.2 - Représentation systolique

3.2.1 - Introduction [6]

Les besoins dans le domaine du traitement numérique rapide ont conduit au développement d'architectures associant des processeurs et des opérateurs élémentaires selon les modalités qui permettent de distinguer différentes classes de dispositifs dont la finalité est d'obtenir des structures à haut parallélisme intrinsèque.

Une architecture systolique est définie de la façon suivante :

- c'est une machine parallèle spécialisée, faite de processeurs ou cellules construits sur quelques modèles simples,

- les processeurs sont connectés de façon régulière et locale,

- les calculs effectués par une telle machine utilisant à la fois la notion de pipeline et de parallélisme vrai sont exécutés de façon synchrone.

Les architectures systoliques sont peu complexes, modulaires et extensives, et permettent d'obtenir des performances de traitement élevées.

3.2.2 - Etude de réseaux systoliques appliqués au calcul de la DCT

La notation matricielle de la DCT est représentée par la formule suivante :

$$D \cdot X^T \cdot D^T = Y$$

avec X et Y représentant un bloc de l'image d'entrée et de l'image de sortie, respectivement.

a) Schéma d'utilisation d'opérateurs de forme carrée: (schéma 1)

Le calcul s'effectue de la façon suivante :

produit de D et de X $\quad D \cdot X$
transposition $\quad (D \cdot X)^T$
produit de D et de $(D.X^T)$ $D(D.X)^T = (D \times D^T)^T$

$$\text{avec } D = \begin{vmatrix} 1 & 1 & 1 & 1 \\ a_o & a_1 & -a_1 & -a_o \\ b_o & -b_o & -b_o & b_o \\ a_1 & -a_o & a_o & -a_1 \end{vmatrix} \quad \text{et} \quad X = \begin{vmatrix} x_o & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{vmatrix}$$

Le fonctionnement global de l'opérateur est décrit à la figure 6. Sa description détaillée est donnée à la figure 7.

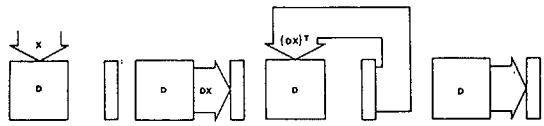


Figure 6 - Fonctionnement global de l'opérateur

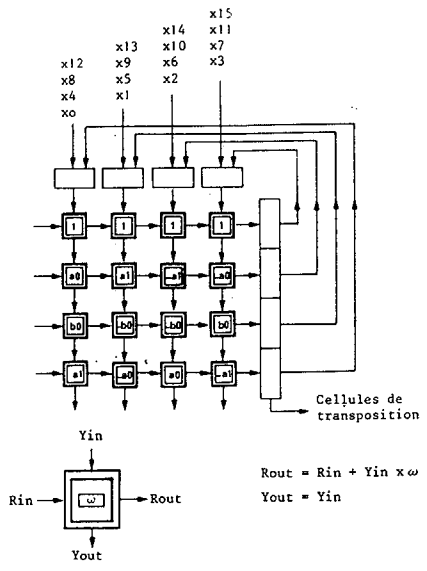


Figure 7 - Présentation de l'opérateur de dimension (4x4) et du processeur de produit intérieur élémentaire

b) Schéma d'utilisation d'opérateurs de forme hexagonale (schéma 2)

Le fonctionnement global de l'opérateur est décrit à la figure 8. Sa description détaillée est donnée à la figure 9.

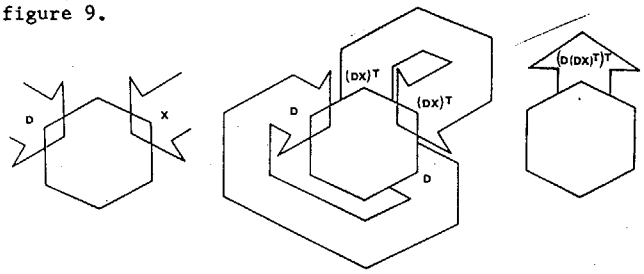


Figure 8 - Fonctionnement global de l'opérateur

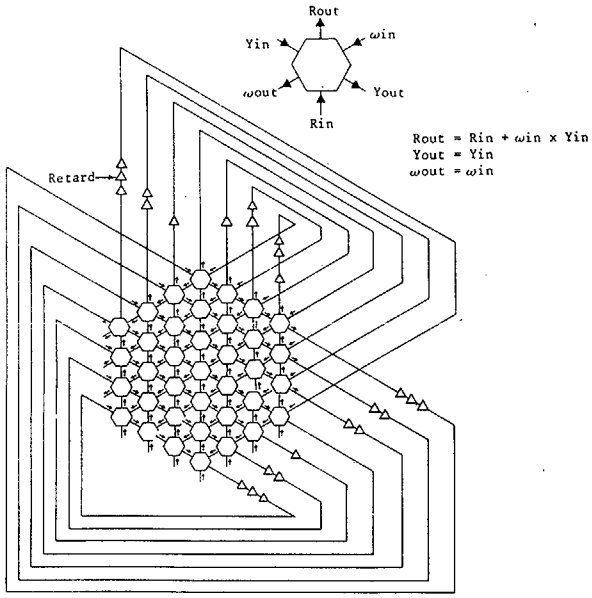


Figure 9 - Présentation de l'opérateur de dimension (4x4) et du processeur de produit intérieur élémentaire

c) Schéma d'utilisation d'opérateurs de forme linéaire (schéma 3)

La description détaillée de l'opérateur ainsi que des processeurs de produit intérieur élémentaire est donné à la figure 10.

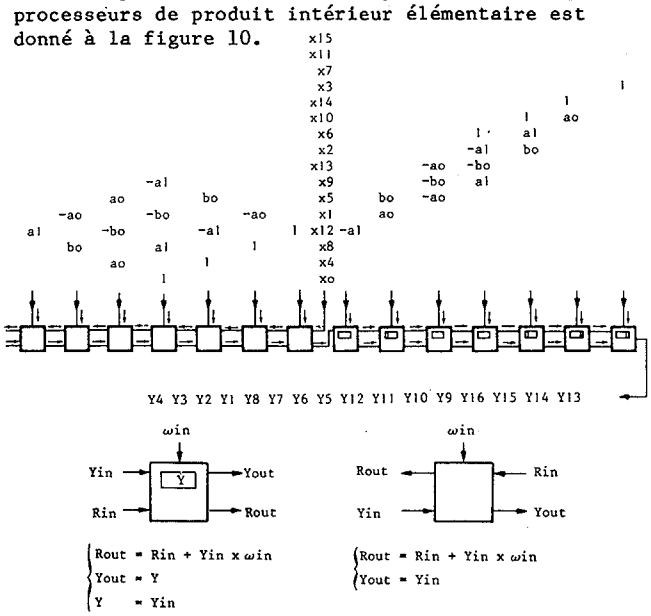


Figure 10 - Présentation de l'opérateur et des processeurs utilisés

3.3 - Elaboration de différents critères de qualité pour chacune des formes proposées

On présente la comparaison des différentes représentations systoliques (cf tableau 2), retenues pour la DCT, en fonction de 5 principaux critères de qualité.

	SCHEMA 1	SCHEMA 2		SCHEMA 3
		1 EQUATION	3 EQUATIONS	
Nombre de cellules	N^2	$3N^2 - 3N + 1$		$2(2N-1)$
Latence	$2(N+1)$	$3(2N-1)$	$9N$	$4(2N-1)$
Temps de calcul des sorties	$2(N-1)$	$3N-2$	$3N-2$	$2(N-1)$
Taux de parallélisme	$\frac{N3}{N+1}$	$\frac{N3}{2N-1}$	N^2	
Efficacité	$\frac{N}{N+1}$	$\frac{N3}{(2N-1)\ (3N^2-3N+1)}$	$\frac{N^2}{3N^2-3N+1}$	

Remarque :

N représente la taille de la fenêtre de traitement considérée.

IV - CONCLUSION

Nous avons présenté, dans un premier temps, différentes approches possibles pour le calcul d'une transformation fonctionnelle appliquée à la compression d'images : cette analyse a montré que la DCT était beaucoup plus simple à mettre en oeuvre que la KLT, tout en représentant une bonne approximation de celle-ci.

Nous avons alors proposé différents types d'architectures.

La première architecture, réduisant le nombre d'opérations, utilise les propriétés de la matrice associée à la transformation (celle-ci est constituée de trois coefficients différents dans le cas où N = 4). Nous pouvons remarquer que cette architecture est moins régulière que les trois architectures systoliques présentées dans le dernier paragraphe.

L'architecture de type linéaire est constituée d'un petit nombre de processeurs mais nécessite un cadencement assez complexe au niveau des données et des processeurs.

L'architecture construite avec des cellules de type carré nécessite un préchargement des coefficients mais comporte beaucoup moins de processeurs que les architectures exploitant des cellules hexagonales. Par contre, ces dernières permettent de modifier les coefficients sans arrêter le cadencement.
On notera qu'il paraît difficile de conclure sur l'utilisation générale d'une architecture donnée dans la mesure où la philosophie systolique conduit à des représentations multiples qui sont fonction des contraintes imposées par le milieu d'intégration.

BIBLIOGRAPHIE :

[1] J. PONCIN "Utilisation de la transformation de Hadamard pour le codage et la compression de signaux d'images".
Annales de télécommunication.

[2] J. LIFERMANN "Les méthodes rapides de transformation du signal : Fourier, Walsh, Hadamard, Haar"
Edition MASSON, Paris 1980

[3] W. PRATT, W. CHEN, L. WELCH "Slant transform Image coding"
IEEE transaction on communication
Vol. COM-22 - August 74

[4] R.N BRACEWELL "The fast Hartley transform"
proceeding of the IEEE, Vol. 72, August 84

[5] N. AHMED, T. NATARAJAN, K. RAD "Discrete cosine transform" IEEE transactions on computers,
January 74

[6] H. KUNG "Why Systolic Architectures ?"
Computer, Vol. 15, n° 1, Janvier 82