(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 680 259**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **91 12450**

(51) Int Cl[5] : G 06 F 15/66

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt** : 09.10.91.

(30) **Priorité** : 09.08.91 JP .

(43) **Date de la mise à disposition du public de la demande :** 12.02.93 Bulletin 93/06.

(56) **Liste des documents cités dans le rapport de recherche :** *Se reporter à la fin du présent fascicule.*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *Société dite: RICOH COMPANY; LTD.* — JP.

(72) **Inventeur(s) :** Blonstein Steven M., Allen James D. et Boliek Martin P.

(73) **Titulaire(s) :**

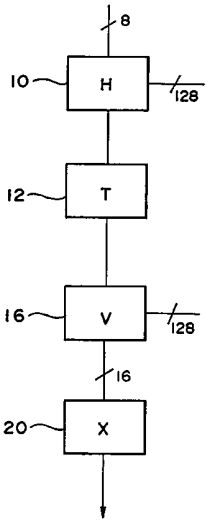
(74) **Mandataire :** Cabinet Beau de Loménie.

(54) Appareil et procédé de compression d'image.

(57) L'invention concerne les techniques de compression d'image.

Un appareil de compression d'image comprend des moyens de transformation en direction horizontale (10) qui transforment horizontalement des pixels d'entrée en utilisant seulement un réseau d'additionneurs; une mémoire de transposition (12) qui fait tourner les pixels transformés en direction horizontale pour les amener en direction verticale; des moyens de transformation en direction verticale (16) qui reçoivent les pixels verticaux et qui les transforment verticalement; et un seul multiplieur (20) qui accomplit une seule fonction de multiplication sur les pixels verticaux transformés, pour fournir des données de pixels comprimées qui sont représentatives des pixels d'entrée.

Applications aux photocopieurs en couleurs.



FR 2 680 259 - A1

La présente invention concerne un appareil et un procédé correspondant pour la compression d'images fixes, qui sont compatibles avec une Norme de Compression d'Images Fixes JPEG (Joint Photographic Experts Group).

Lorsqu'on doit comprimer des images de haute qualité pour réduire des exigences de mémoire ou de transmission, il est de pratique courante de transformer tout d'abord les images dans un autre espace dans lequel on peut représenter l'information de façon plus compacte. On effectue habituellement ceci bloc par bloc au moyen d'une transformation linéaire (multiplication matricielle); une configuration caractéristique consiste à effectuer des transformations à 8 points sur des segments de rangée de 8 pixels, et à effectuer ensuite des transformations à 8 points sur les segments de colonne à 8 éléments de cette image transformée dans la direction des rangées. De façon équivalente, on peut effectuer une seule transformation à 64 points sur un bloc de pixels comprenant 64 pixels disposés en un bloc de 8 par 8.

La transformation de Tchebychev discrète est un bon choix pour une transformation unidimensionnelle :

$$F(u) = c(u) * \sum_{i=0}^{7} f(i) * \cos u(2i+1)pi/16$$

avec

$$C(u) = \sqrt{2}/8 \text{ pour } u = o$$
$$2/8 \text{ dans les autres cas}$$

cette transformation présente plusieurs avantages parmi lesquels : a) le fait que la compression est presque optimale à certains égards, b) le fait qu'il existe des algorithmes de calcul rapides pour effectuer cette transformation et son inverse, et

c) le fait qu'on peut réduire aisément le flou (amélioration de l'image initiale) dans l'espace de la transformée, sous certaines hypothèses qui sont décrites dans la Référence [1].

L'invention a pour but de procurer un appareil et un procédé correspondant pour la compression d'images fixes.

Un but plus particulier est de procurer un appareil et un procédé correspondant pour la compression d'images fixes qui demeurent compatibles avec une Transformation JPEG.

D'autres buts, caractéristiques et avantages de l'invention seront mieux compris à la lecture de la description détaillée qui va suivre d'un mode de réalisation, donné à titre d'exemple non limitatif. La suite de la description se réfère aux dessins annexés dans lesquels :

La figure 1A montre un schéma synoptique d'un compresseur et la figure 1B montre un schéma synoptique d'un extenseur, conformes à l'invention.

Les figures 2A-2C montrent la façon dont sont ordonnés les pixels d'entrée, les caractéristiques temporelles de bloc et les caractéristiques temporelles de vecteur de données conformes à l'invention.

Les figures 3A,3B montrent une transformation à trois points de données RGB en données XYZ.

Les figures 4A et 4B montrent des configurations possibles de circuits à très haut niveau d'intégration (VLSI) conformes à l'invention.

La figure 5 montre un schéma d'un registre à décalage qui est utilisé dans l'invention.

La figure 6A montre un schéma d'un réseau de décalage conforme à l'invention.

La figure 6B montre un exemple du réseau de

décalage de la figure 6A.

La figure 7 montre un schéma de la circulation combinée de données.

Les figures 8A et 8B montrent des schémas de réseaux d'addition en sens avant conformes à l'invention.

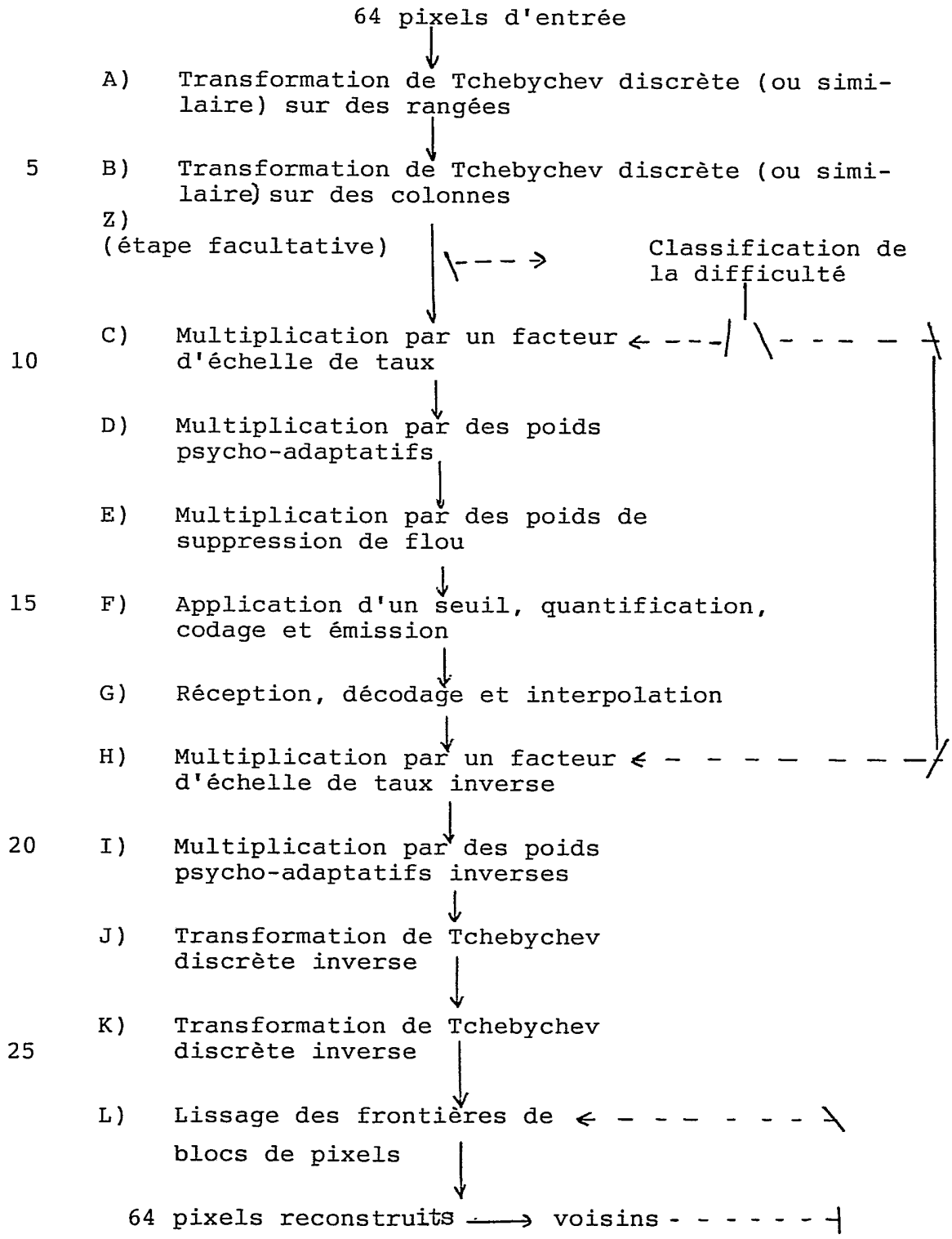
La figure 9 montre un schéma d'une transformation de Chen généralisée bidimensionnelle conforme à l'invention.

La figure 10 montre un schéma synoptique d'un mode de réalisation préféré de l'invention.

Exposé théorique de l'invention

Un système complet pour la compression et la reconstruction d'images peut se présenter comme le montre le Tableau 1.

TABLEAU 1


64 pixels d'entrée
A) Transformation de Tchebychev discrète (ou similaire) sur des rangées
B) Transformation de Tchebychev discrète (ou similaire) sur des colonnes
Z) (étape facultative)
Classification de la difficulté
C) Multiplication par un facteur d'échelle de taux
D) Multiplication par des poids psycho-adaptatifs
E) Multiplication par des poids de suppression de flou
F) Application d'un seuil, quantification, codage et émission
G) Réception, décodage et interpolation
H) Multiplication par un facteur d'échelle de taux inverse
I) Multiplication par des poids psycho-adaptatifs inverses
J) Transformation de Tchebychev discrète inverse
K) Transformation de Tchebychev discrète inverse
L) Lissage des frontières de blocs de pixels
64 pixels reconstruits
voisins

Le Tableau 1 ci-dessus décrit l'invention, et également la technologie actuelle lorsqu'on omet les étapes facultatives (L, Z).

La multiplication par les poids de suppression de flou (étape E) peut également être effectuée sous la forme d'une étape de décodage (par exemple après l'étape I).

On effectue la suppression du flou pour compenser la fonction d'étalement de point du dispositif d'entrée. Elle doit être adaptée au dispositif ou omise lorsque l'image d'entrée a déjà été améliorée. Il existe de meilleures manières de supprimer le flou de l'image, mais le procédé qui est représenté ici est économique du point de vue du calcul et il est approprié dans certaines applications, par exemple pour un photocopieur en couleurs.

Il est possible de réaliser le calcul de la transformation directe ((A, B) de façon que la majeure partie de la charge de travail de calcul consiste en une phase de multiplication finale. En pré-calculant les produits de ces multiplicateurs et ceux des étapes C, E, on peut accélérer le processus de compression.

De façon similaire, il est possible de réaliser le calcul de la transformation inverse de façon que la majeure partie de la charge de travail de calcul soit concentrée dans une phase de multiplication préliminaire. Ici encore, en pré-calculant les produits, on élimine effectivement l'effort de calcul des étapes H, I.

En outre, on substitue une autre transformation à la transformation de Tchebychev discrète bidimensionnelle, ce qui simplifie encore davantage les calculs.

De plus, on peut faire varier sélectivement les poids psycho-adaptatifs, pour que les multiplicateurs

combinés pour les étapes B, D, soient efficaces au point de vue du calcul, c'est-à-dire qu'ils correspondent par exemple à des puissances de deux. De petits changements dans les poids psycho-adaptatifs des éléments de transformation de sortie à faible énergie ont peu d'effet sur la qualité de l'image ou le taux de compression.

Enfin, on considérera les étapes L, Z du Tableau 1, c'est-à-dire Classification de Difficulté de l'Image, et Lissage des Frontières de Blocs. du fait que ces opérations sont facultatives et indépendantes de l'invention principale, on ne les envisagera ici que de façon minimale.

Algorithme de Chen

L'algorithme de Chen unidimensionnel [3] indique que :

$$X = 2/_N \, A_N \, x \qquad (1)$$

en désignant par x un vecteur de données, par X un vecteur transformé et par A_N la quantité suivante :

$$A_N = c(k) \cos((2j+1)k\pi/2N);\ j,\ k,\ = 0,1,2,...N-1$$

En outre, on peut décomposer A_N de la façon suivante :

$$A_N = Z_N \begin{vmatrix} A_{N/2} & 0 \\ 0 & R_{N/2} \end{vmatrix} B_N \qquad (2)$$

avec pour $R_{N/2}$ la valeur suivante :

$$R_{N/2} = c(2k+1) \cos((2j+1)(2k+1)\pi/2N);\ j,\ k = 0,1,2,...,N/2-1. \qquad (3)$$

On note que la matrice Z est la matrice P de la transformation de Chen. On a changé la notation pour éviter une confusion avec la matrice P dans la présente demande.

Exemple de transformation de Chen unidimensionnelle à huit points (N = 8)

Pour mettre en oeuvre l'algorithme de Chen à huits points, on utilise deux fois l'équation 2, de façon récursive. La première itération utilise les matrices Z_8, R_4 et B_8. La seconde itération effectue une résolution visant à déterminer A_4, et elle utilise les matrices Z_4, R_2, A_2 et B_4. On obtient aisément ces matrices à partir des équations ci-dessus ou de l'article de Chen |3|.

$$A_8 = Z_8 \begin{vmatrix} & | & A_2 & 0 & | & & | & \\ & Z_4 & | & & | & B_4 & 0 & | & \\ & | & 0 & R_2 & | & & | & B_8 \\ & | & & & | & & | & \\ & | & & 0 & | & & R_4 & | & \end{vmatrix}$$

avec

$$Z_8 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$B_8 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{vmatrix}$$

$$R_4 = \begin{vmatrix} C1 & C3 & C5 & C7 \\ C3 & -C7 & -C1 & -C5 \\ C5 & -C1 & C7 & C3 \\ C7 & -C5 & C3 & -C1 \end{vmatrix} = \begin{vmatrix} C1 & C4(C7+C1) & C4(C1-C7) & C7 \\ C3 & C4(C5-C3) & -C4(C3+C5) & -C5 \\ C5 & -C4(C5+C3) & C4(C3-C5) & C3 \\ C7 & C4(C7-C1) & C4(C1+C7) & -C1 \end{vmatrix}$$

$$= \begin{vmatrix} C1 & 0 & C7 & 0 \\ 0 & C3 & 0 & C5 \\ 0 & C5 & 0 & -C3 \\ C7 & 0 & -C1 & 0 \end{vmatrix} \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & C4 & 0 & 0 \\ 0 & 0 & C4 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix}$$

$$= RA_4 \; RB_4 \; RC_4 \; RD_4$$

$$Z_4 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$B_4 = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{vmatrix}$$

$$R_2 = \begin{vmatrix} C2 & C6 \\ C6 & -C2 \end{vmatrix}$$

$$A_2 = \begin{vmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{vmatrix}$$

avec, d'après l'équation 3 :

$$Cn = \cos(n\pi/16)$$

Transformation de Chen-Wu (modifiée) ou paramétrisée

La transformation de Chen est tout ce qui a été fait jusqu'à présent. On pourrait effectuer les multiplications nécessaires et réaliser une économie de calcul par rapport à une mise en oeuvre grossière de la transformation de Tchebychev discrète , exigeant des multiplications intensives. Ce n'est cependant pas ce que propose la demanderesse. Pour réduire le nombre de multiplications au strict minimum, on reprend la paramétrisation des matrices, de la façon suivante. Ceci constitue ce que la demanderesse appelle la transformation de Chen-Wu (modifiée), qui est une création de la demanderesse.

$$R_4 = \begin{vmatrix} a & r(1+a) & r(a-1) & 1 \\ c & r(1-c) & -r(1+c) & -1 \\ 1 & -r(1+c) & r(c-1) & c \\ 1 & r(1-a) & r(a+1) & -a \end{vmatrix} \begin{vmatrix} 1\sqrt{(a^2+1)} & 0 & 0 & 0 \\ 0 & 1/\sqrt{(c^2+1)} & 0 & 0 \\ 0 & 0 & 1/\sqrt{(c^2+1)} & 0 \\ 0 & 0 & 0 & 1/\sqrt{(a^2+1)} \end{vmatrix}$$

$$= RE_4 \; RF_4$$

$$R_2 = 1\sqrt{(b^2+1)} \begin{vmatrix} b & 1 \\ 1 & -b \end{vmatrix}$$

$$A_2 = 1/\sqrt{2} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

avec

$$a = C1/C7 = \cos(\pi/16)/\cos(7\pi/16) = \tan(7\pi/16),$$
$$b = C2/C6 = \tan(6\pi/16),$$
$$c = C3/C5 = \tan(5\pi/16),$$
$$r = C4 = \cos(4\pi/16). \qquad (4)$$

Il faut noter que la matrice diagonale RF_4 contient les facteurs de normalisation de la matrice non paramétrisée RA_4. Il faut également noter qu'une matrice diagonale peut être constituée par les constantes dans R_2 et A_2.

Au moment de la reconstruction de la matrice A_8, deux matrices sont conservées distinctes. Les matrices diagonales sont conservées séparées de la matrice principale. La matrice principale est multipliée par les termes B_N. Après le réarrangement approprié et la multiplication par le terme constant, l'équation 1 se réduit à : $X=Q(a, b, c)\ P(a,b,c,r)\ x$, avec:

$$
Q(a,b,c)=\begin{vmatrix}
1/2\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/2\sqrt{(a^2+1)} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2\sqrt{(b^2+1)} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/2\sqrt{(c^2+1)} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2\sqrt{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/2\sqrt{(c^2+1)} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/2\sqrt{(b^2+1)} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2\sqrt{(a^2+1)}
\end{vmatrix} \qquad (5)
$$

$$
P(a,\ b,\ c,\ r)=\begin{vmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
a & r(a+1) & r(a-1) & 1 & -1 & r(1-a) & -r(a+1) & -1 \\
b & 1 & -1 & -b & -b & -1 & 1 & b \\
c & r(1-c) & -r(c+1) & -1 & 1 & r(c+1) & r(c-1) & -c \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -r(c+1) & r(c-1) & c & -c & r(1-c) & r(c+1) & -1 \\
1 & -b & b & -1 & -1 & b & -b & 1 \\
1 & r(1-a) & r(a+1) & -a & a & -r(a+1) & r(1-a) & -1
\end{vmatrix} \qquad (6)
$$

Transformation généralisée

La transformation DCT à 8 points généralisée est déterminée par quatre paramètres a, b, c et r, et on peut l'écrire sous la forme :

$$T(a,b,c,r,) = P(a,b,c,r) \times Q(a,b,c)$$

dans laquelle P() et Q() sont les mêmes que ci-dessus.

La transformation d'image exige deux de ces transformations T, à savoir Tv et Th, pour transformer l'image respectivement en direction verticale et en direction horizontale. La transformation bidimensionnelle complète est représentée par

$$[F] = [T_v]\ [f]^t\ [T_h]$$

en désignant par f le bloc d'image d'entrée, et par F les coefficients de la transformée de sortie, tandis que l'indice supérieur "t" désigne l'opération de transposition de matrice. Toutes les matrices ont ici des dimensions de 8 par 8.

Du fait qu'une matrice diagonale (telle que Q) est sa propre transposée, et que :

$$[A]^t\ [B]^t = ([B]^t\ [A])$$

pour toutes les matrices, et que :

$$[T_v] = [F_v]\ [Q\ 1_v],$$

$$[T_h 1] = [P_h]\ [Q_h],$$

on peut écrire :

$$[F] = [Q_v]\ [P_v]\ [f]\ [P_h]\ [Q_h]$$

ce qui se réduit à :

$$F(i,j) = q(i,j) * g(i,j)$$

avec

$$[g] = [F_v]\ [f]\ [P_h^t]$$

et

$$q(i,j) = Q_v(i,i) * Q_h(j,j) \qquad (7)$$

Lorsqu'on transforme un bloc d'image, on doit effectuer une résolution donnant [g], en utilisant l'algorithme de Chen-Wu, et multiplier ensuite par les facteurs q(i,j). Connaissant

$$P_v = P(a, b, c, r_v)$$

et

$$P_h = P(a, b, c, r_h)$$

l'inverse de la transformation ci-dessus s'exprime par :

$$[f] = [P'_v]\ [Q_v]\ [F]\ [Q_h]\ [P'^t_h]$$

avec

$$P'_v = P(a, b, c, 1 / 2r_v)$$

et

$$P'_h = P(a, b, c, 1 / 2\ r_h)$$

Ici encore, on obtient la solution par l'intermédiaire de l'algorithme de Chen-Wu.

Algorithme de Chen

Plusieurs procédés ont été imaginés pour accélérer le calcul des transformations de Tchebychev unidimensionnelles ou bidimensionnelles et de leurs transformations inverses;

Il existe un algorithme bien connu (Chen) [2,3] qui multiplie un 8-uple arbitraire par la ma-

trice T ci-dessus, en utilisant seulement 16 multiplications, 13 additions et 13 soustractions. Cet algorithme ne repose sur aucune propriété spéciale des paramètres a, b, c et r.

Algorithme de Chen-Wu (modifié)

En factorisant [T] = [P] [Q] comme ci-dessus, on décompose l'algorithme de Chen en deux étages, avec 8 multiplications utilisées dans la multiplication par [Q], et 8 multiplications et le reste des opérations arithmétiques utilisées dans la multiplication par [P]. Ceci est une conséquence de notre choix pour [Q]; plusieurs éléments de [P] deviennent égaux à 1 ou -1 et une multiplication disparaît.

Comme indiqué ci-dessus, des simplifications similaires s'appliquent à la transformation inverse, à la transformation bidimensionnelle et à la transformation bidimensionnelle inverse. Pour des blocs de 8 par 8, on utilise 128 multiplications pour la transformation bidimensionnelle directe ou inverse (à l'exclusion des multiplications par [q]). Lorsqu'on observe le flux de données interne de l'algorithme de Chen, on note que ces multiplications sont incorporées dans une structure de huit étages d'addition/soustraction et de quatre étages de multiplication.

Il est important de souligner le fait que l'algorithme de Chen fonctionne indépendamment des paramètres a, b, c et r. Cependant, la transformation DCT à 8 points qui est employée dans l'art antérieur utilise les paramètres de la "transformation en cosinus vraie" :

a = tangente (7*pi/16)
b = tangente (6*pi/16°
c = tangente (5*pi/16)
r = racine carrée de (1/2) = 0,70710678 ...

avec le choix de r nécessaire et suffisant pour que la matrice T soir orthogonale.

Choix des valeurs des paramètres

La transformation de Chen fonctionne indépendamment des valeurs qui sont sélectionnées pour les paramètres a, b, c et r. Ceci vient du fait que la transformée qui est créée par QP est orthogonale. Il est entièrement possible d'utiliser n'importe quels nombres et d'avoir une transformation qui accomplit la décorrélation désirée des données d'image qui est nécessaire pour la compression. On note que cette transformation n'est ni une transformation en cosinus discrète, ni une approximation d'une transformation DCT. C'est une transformation spécifique.

On admet cependant de façon générale que la transformation DCT est très souhaitable pour réaliser une décorrélation efficace de l'image d'entrée, et pour effectuer une transformation en coefficients de fréquence spatiale relativement significatifs [5]. Par conséquent, pour bénéficier de la transformation DCT, on fixe les paramètres de façon qu'ils constituent des approximations de ceux de la transformation DCT qui sont donnés par l'équation 4. Le facteur qui s'oppose à ceci est l'efficacité du calcul. Du fait qu'une addition est moins coûteuse qu'une multiplication (dans le cas du matériel, l'économie porte sur l'aire de silicium qui est occupée, tandis que dans le cas du logiciel elle porte sur le nombre de cycles), les paramètres sont choisis pour être efficaces au point de vue du calcul.

Autres algorithmes

D'autres solutions de calcul ont été imaginées pour la Transformation de Tchebychev Discrète. Par exemple, un algorithme dû à Lee effectue la transformation unidimensionnelle à 8 points et la transfor-

mation bidimensionnelle à 64 points avec respectivement 12 et 144 multiplications [4,5].

Ces algorithmes "plus rapides" présentent cependant plusieurs inconvénients en comparaison avec l'algorithme de Chen :

a) La simplification T = P x Q (et la factorisation similaire pour la transformation inverse) ne fonctionne plus. La séparation de la matrice diagonale Q est essentielle pour les simplifications qui suivent.

b) Ces algorithmes ne fonctionnent pas avec des paramètres a, b, c et r arbitraires. Ils reposent en effet sur diverses identités trigonométriques qui sont spécifiquement valides pour les véritables paramètres de la transformation en cosinus.

c) Ces algorithmes ont une structure plus difficile. Ceci peut créer des difficultés dans leur mise en oeuvre et augmenter le risque d'instabilité numérique.

Exposé de l'invention

A) En se référant à nouveau au Tableau 1, on note que les étapes C, D, E peuvent être combinées avec les post-multiplicateurs de transformation directe déduits de [Q]. De façon similaire, les étapes H, I peuvent être combinées avec les pré-multiplicateurs de transformation inverse. Ceci vient du fait que l'opération de multiplication par un facteur d'échelle de taux, l'opération de multiplication par des poids psycho-adaptatifs (correspondant à ce que l'on appelle couramment des valeurs de quantification), et l'opération de multiplication par des poids de suppression du flou, sont toutes des opérations de multiplication de point. Si b, c, d, e sont les informations de sortie des étapes respectives B, C, D, E, on a :

$$c(i,j) = b(i,j) * q(i,j)$$

$$d(i,j) = c(i,j) * r(i,j) = b(i,j) * q(i,j) * r(i,j)$$

$$e(i,j) = d(i,j) * u(i,j) = b(i,j) * q(i,j) * r(i,j) * u(i,j)$$

ou

$$e(i,j) = b(i,j) * tous(i,j)$$

avec

$$tous(i,j) = q(i,j) * r(i,j) * u(i,j) \qquad (8)$$

et q(i,j) est le facteur d'échelle de taux, r(i,j) sont des poids de quantifications choisis de façon psycho-adaptative (ou même choisis par l'utilisateur), et u(i,j) sont des poids des suppression de flou.

On peut combiner de façon similaire les étapes H et I.

Ceci signifie effectivement que les fonctions d'application de facteur d'échelle de taux, de pondération adaptative et de suppression de flou sont réalisées sans aucune charge de calcul générale supplémentaire. Comme indiqué ci-dessus, cette façon de procéder n'est pas applicable aux algorithmes "plus rapides", tels que celui de Lee.

B) Du fait que l'algorithme de Chen fonctionne avec n'importe quels paramètres a, b, c et r, on choisira des valeurs qui offrent une qualité et une compression similaires à celles de la transformation DCT, mais qui conduisent à une multiplication rapide.

Les paramètres suivants sont raisonnablement proches de ceux de la transformation DCT, mais beaucoup plus efficaces au point de vue du calcul :

$$a = 5,0$$

$$b = 2,5$$

$$c = 1,5$$

$$r = 0,75$$

La multiplication est maintenant remplacée par des opérations arithmétiques beaucoup plus simples. Par exemple, la multiplication par 5 devient : copie; décalage à gauche de 2; addition. La multiplication par 1,5 devient : copie; décalage à droite de 1; addition.

Selon une variante, le numérateur inverse d'un multiplicateur rationnel peut être factorisé dans le multiplicateur combiné [q]. Ainsi, la multiplication par 2,5 peut devenir des multiplications par 5 et 2, respectivement pour des termes affectés et non affectés.

En mettant en oeuvre cette dernière idée, la manipulation du paramètre r = 0,75 dans l'algorithme de Chen de base exige 96 multiplications par 4 et 32 multiplications par 3. Avec l'algorithme de Wu-Paolini dans un perfectionnement correspondant à une forme de réalisation bidimensionnelle, un étage de multiplication complet est éliminé, et on obtient ainsi 36 multiplications par 16, 24 multiplications par 12 et 4 multiplications par 9. (La transformation inverse utilise 36 multiplications par 9, 24 multiplications par 6 et 4 multiplications par 4.)

Au prix d'une diminution de la vitesse de calcul, on peut sélectionner des valeurs de pramètres encore plus proches de celles de la transformation en cosinus. On peut effectuer les substitutions : b = 12/5 et/ou r = 17/24. Une autre possibilité intéressante est :

rRow = 0,708333 (17/24)

rCol = 0,7 (7/10)

On utilise ici des transformations légèrement différentes (paramètre r différent) pour les rangées et les colonnes. Ceci a pour but de simplifier les multiplicateurs qui sont obtenus dans le procédé de Wu-Paolini. Ce procédé donne ici 36 multiplications par 15, 12 multiplications par 85/8, 12 multiplications par 21/2 et 4 multiplications par 119/16. (La transformation inverse utilise 36 multiplications par 119/16, 12 multiplications par 85/16, 12 multiplications par 21/4 et 4 multiplications par 15/4.)

De la façon que l'on vient de décrire, toutes les multiplications ont été rendues rapides et peu coûteuses, sauf pour le multiplicateur combiné [q] dans le compresseur et le multiplicateur combiné [q] dans l'extenseur. Chacun d'eux exige une multiplication par élément de transformation. Ce dernier cas est simplifié par le fait que la majorité des coefficients de transformation seront égaux à zéro, et que la majeure partie des coefficients différents de zéro seront des entiers très proches de zéro que l'on pourra traiter de façon spécifique.

C) On utilise une technique supplémentaire pour réduire le coût du calcul du multiplicateur combiné [q] dans le compresseur. Du fait que le facteur d'échelle de taux est en réalité une valeur arbitraire, on l'ajustera point par point pour donner à tous les éléments de matrice [q] des valeurs simples au point de vue du calcul, par exemple des puissances de deux. Ces 64 ajustements ne doivent être effectués qu'une seule fois (après la spécification du facteur d'échelle de taux et des filtres de suppression de flou).

Par exemple, si un élément (C) du multipli-

cateur combiné et l'élément du multiplicateur d'extension correspondant (D) ont les valeurs :

$$C = 0,002773$$

$$D = 0,009367$$

on peut trouver l'approximation $C \simeq 3 / 1024 = 0,002930$, et on peut l'utiliser pour simplifier la multiplication. Ceci donne $C' = 3 / 1024$ et $D' = D * C/C' \simeq 0,008866$.

Description détaillée du processus (de base)

Remarques :

a) Dans l'espace de transformation quantifié, il est commode et efficace de choisir une largeur constante (w) pour les étapes différentes de zéro de la quantification des coefficients "AC", et de prendre la largeur w*q pour l'étape zéro.

En outre, q = 2 est commode d'un point de vue arithmétique et est presque optimal pour la qualité sur une gamme étendue de facteurs de compression. Dans la description, on prend q = 2 ("zéro à double largeur"), bien que l'invention englobe n'importe quelle valeur possible de q.

b) L'algorithme suivant est conçu pour une arithmétique portant sur des entiers binaires en complément à deux à précision limitée, sauf pour les déterminations intermédiaires aux étapes (2), (4) et (8), qui sont effectuées une seule fois en arithmétique à haute précision.

En outre, et avec l'exception supplémentaire de l'étape (9.1), les multiplications de nombres entiers qui sont décrites ici sont optimisées du point de vue du coût et de la vitesse. On considère par exemple les multiplications par :

Nrr * Nrc = Drr' * Drc' = 1,75 * 4,25 = 7,4374

En choisissant l'identité 7,4375 = (8-1) * (1+1/16), la multiplication peut être effectuée efficacement avec des décalages et des additions.

c) Les multiplications de suppression du flou sont représentées ici à l'étape 8, mais elles doivent habituellement être effectuées à l'étape 4, si encore elles le sont. Dans de nombreuses applications, l'extenseur ne "sait" pas comment supprimer le flou de l'image, ou s'il doit le faire. Il faut noter que les meilleures valeurs de Thr( ) dépendent du dispositif d'entrée et du procédé de suppression du flou.

Une technique recommandée consiste à calculer la valeur m(i,j) (voir l'étape 8) au moment de la compression (étape 4), et de la transmettre ou de l'enregistrer dans le cadre de l'image comprimée.

d) Il existe plusieurs moyens évidents pour combiner en parallèle, en séquence temporelle ou de façon entrelacée, les calculs qui suivent. Le procédé préféré pour une architecture de matériel donnée ressort clairement.

Exemple de mode de réalisation en pseudo-code

Cette partie de la présente demande consiste fondamentalement en un mode de réalisation de l'invention expliqué en texte et en pseudo-code. On y trouve plusieurs sections, comprenant la paramétrisation, le calcul de tous(i,j) comme dans l'équation 8 ci-dessus, l'exécution du corps principal de la transformation de Chen généralisée directe, le calcul de tous(i,j) inverse, et l'exécution du corps principal de la transformation de Chen généralisée inverse.

1. Les paramètres a, b, c et r sont indiqués ci-dessus. On note qu'il existe une valeur de r à la fois pour les rangées et les colonnes. Bien que la

transformation de Chen généralisée bidimensionnelle soit une transformation séparable et puisse être exécutée en deux passes, il n'y a aucune restriction exigeant qu'elle soit symétrique. Les facteurs d'échelle peuvent donc être dissymétriques, comme représenté.

Les équations pour les numérateurs N et les dénominateurs D montrent des combinaisons possibles de numérateur et de dénominateur qui peuvent être égales aux valeurs ci-dessus. Le concepteur de la mise en oeuvre de la transformation de Chen généralisée dispose d'une grande liberté dans le choix des valeurs réelles qui sont utilisées dans le réseau additionneur. Les choix de valeurs sont corrigés dans l'étage de multiplication final.

On choisit :

tan 7*pi/16	~=a	=	Na / Da
tan 6*pi/16	~=b	=	Nb / Db
tan 5*pi/16	~=c	=	Nc / Dc
$\sqrt{(0,5)}$	~=rRow	=	Nrr/ Drr
$\sqrt{(0,5)}$	~=rCol	=	Nr / Drc
0,5 / rRow	= rRow'	=	Nrr'/Drr'
0,5 / rCol	= rCol'	=	Nrc'/Drc',

pour les paramètres de la transformation de Chen généralisée, comme envisagé ci-dessus. Il n'est pas nécessaire que les "numérateurs" N et les "dénominateurs" D soient entiers, bien qu'on les choisisse entiers pour la commodité du calcul. Parmi plusieurs possibilités utiles, on peut mentionner :

Na = 5	Da = 1
Nb = 3	Db = 1,25
Nc = 1,5	Dc = 1
Nrr = 1,75	Drr = 2,5
Nrc = 4,25	Drc = 6
Nrr' = 1,25	Drr' = 1,75
Nrc' = 3	Drc' = 4,25

mais ici encore, l'invention englobe toutes les approximations rationnelles des tangentes ci-dessus

On calcule ainsi les facteurs d'échelle de normalisation nécessaires.

2. On écrit également :

$$U(0) = U(4) = \sqrt{(0,5)}$$

$$U(I) = U(7) = 1/\sqrt{(Na*Na + Da*Da)}$$

$$U(2) = U(6) = 1/\sqrt{(Nb*Nb + Db*Db)}$$

$$U(3) = U(5) = 1/\sqrt{(Nc*Nc + Dc*Dc)}$$

3. On adopte les notations suivantes :

i désigne un index sur 0,1,2,3,4,5,6,7 représentant la position verticale (dans l'espace d'image) ou la séquence de variation verticale (dans l'espace de la transformée)

j désigne un index sur 0,1,2,3,4,5,6,7 représentant la position horizontale (dans l'espace d'image) ou une séquence de variation horizontale (dans l'espace de la transformée)

Debl(i,j) désigne les facteurs de suppression de flou, ou Debl() = 1 quand il n'y a pas de suppression de flou.

Thr(i,j) désigne les poids psycho-adaptatifs inverses, par exemple ceux recommandés par le CCITT.

M désigne le facteur d'échelle de taux; on a ici M = 1 (approximativement) pour des taux de compression caractéristiques.

v(i,j) désigne les différentes valeurs de luminance dans l'espace d'image (spatiale).

L(i,j) désigne les valeurs de luminance transformées dans l'espace de la transformée (avec compression).

S est un entier arbitraire de valeur faible désignant la précision arithmétique qui est utilisée dans la reconstruction.

Les poids psycho-adaptatifs 1 / Thr(i,j) doivent être réoptimisés pour chaque ensemble de paramètres de la transformation de Chen généralisée. Cependant, les paramètres qui sont donnés à l'étape (1) ci-dessus sont suffisamment proches des paramètres du CCITT pour que la même matrice Thr() soit optimale.

4. g(i,j) équivaut ici à tous(i,j). On effectue une itération passant par les 64 positions de transformation (i,j), en calculant k(i,j) et s(i,j) pour satisfaire la relation :

$$g(i,j) < \frac{2^{s(i,j)} * M * U(i) * U(j)}{k(i,j) * Zr(i) * Zc(j) * Thr(i,j)}$$

avec le membre de droite aussi proche que possible de g(i,j), avec s(i,j) entier, et avec :

g(i,j) = 1,0, k(i,j) dans {1,3,5,7,9}
pour i+j < 4

g(i,j) = 0,9, k(i,j) dans {1,3,5}
pour i+j < 4

g(i,j) = 0,7, k(i,j) = 1
pour i+j < 4

Zr(i) = 1, lorsque i = 0, 1, 2 ou 3
Zr(i) = Drr, lorsque i = 4, 5, 6 ou 7
Zc(j) = 1, lorsque j = 0, 1, 2 ou 3
Zc(j) = Drc, lorsque j = 4, 5, 6 ou 7
Zr'(i) = 1, lorsque i = 0, 1, 2 ou 3
Zr'(i) = Drr', lorsque i = 4, 5, 6 ou 7
Zc'(j) = 1, lorsque j = 0, 1, 2 ou 3
Zc'(j) = Drc, lorsque j = 4, 5, 6 ou 7

Les facteurs g(i,j) sont destinés à rendre le biais de quantification indépendant de la taille choisie.

5. Exécution de la transformation de Chen généralisée directe

L'étape 5 est l'exécution du pseudo-code de la transformation directe. Les étapes suivantes accomplissent une transformation bidimensionnelle sous une forme entrelacée.

On effectue une itération dans l'image en accomplissant les opérations qui suivent sur chaque bloc de valeurs de luminance v(,) de 8 par 8 :

5.1 Préparer les valeurs :

```
M(i, 0)    = V(i, 0) + V(i,7)
M(i, 1)    = V(i, 1)  + V(i,6)
M(i, 2)    = V(i, 2)  + V(i,5)
M(i, 3)    = V(i, 3)  + V(i,4)
M(i, 4)    = V(i, 3)  - V(i,4)
M5(i)      = V(, 2)   - V(i, 5)
M6(i)      = V(i, 1)  - V(i, 6)
M(i, 5)    = M6(i)    + M5(i)
M(i, 6)    = M6(i)    - M5(i)
M(i, 7)    = v(i, 0)  - v(i,7)
```

pour i = 0, 1, 2, ..., 7

5.2 Préparer les valeurs :

```
H(0, j)    = M(0, j) + M(7,j)
H(1, j)    = M(1, j)  + M(6,j)
H(2, j)    = M(2, j)  + M(5, j)
H(3, j)    = M(3, j)  + M(4, j)
H(4, j)    = M(3, j)  - M(4, j)

H5(j)      = M(2, j)  - M(5, j)
H6(j)      = M(1, j)  - M(6, j)
H(5, j)    = H6(j)    + H5(j)
H(6, j)    = H6(j)    = H5(j)
H(7, j)    = M(0, j)  - M(7,j)
```

pour j = 0, 1, 2, ..., 7

5.3 Multiplier chaque H(i,j) par

(si i = 0, 2, 3 ou 4 :)

Nrc	si j = 5 ou 6
Drc	si j = 4 ou 7
1 (pas d'action)	si j = 0, 1, 2 ou 3

(si i = 4 ou 7 :)

Drr Nrc	si j = 5 ou 6
Drr Drc	si j = 4 ou 7
Nrr	si j = 0, 1, 2 ou 3

(si i = 5 ou 6 :)

Nrr Nrc	si j = 5 ou 6
Nrr Drc	si j = 4 ou 7
Nrr	si j = 0, 1, 2 ou 3

5.4 Préparer les valeurs

```
E(0, j) = H(0,j) + H(3,j)
E(1, j) = H(7,j) + H(5,j)
E(2, j) = H(0,j) - H(3,j)
E(3, j) = H(7,j) - H(5,j)
E(4, j) = H(I,j) + H(2,j)
E(5, j) = H(6,j) - H(4,j)
E(6, j) = H(I,j) - H(2,j)
E(7, j) = H(6,j) + H(4,j)
```

```
F(0, j) = E(4, j) + E(0, j)
F(4, j) = E(0, j) - E(4, j)
F(2, j) = Db * E(6,j) + Nb * E(2, j)
F(6, j) = Db * E(2,j) - Nb * E(6, j)
F(1, j) = Da * E(7,j) + Na * E(1, j)
F(7. j) = Da * E(1,j) - Na * E(7, j)

F(3, j) = Dc * E(5,j) + Nc * E(3, j)
F(5, j) = Dc * E(3,j) - Nc * E(5, j)
```

pour j = 0, 1, 2, ..., 7

5.5 préparer les valeurs

```
Z(i, 0) = F(i,0) + F(i,e,)
Z(i, 2) = F(i,0) - F(i,3)
Z(i, 4) = F(i,1) + F(i,2)
Z(i, 6) = F(i,1) + F(i,2)
Z(i, 1) = F(i,7) + F(i,5)
Z(i, 3) = F(i,7) - F(i,5)
Z(i, 5) = F(i,6) - F(i,4)
Z(i, 7) = F(i,6) + F(i,4)
```

$$G(i,0) = Z(i,4) + Z(i,0)$$
$$G(i,4) = Z(i,0) - Z(i,4)$$
$$G(i,2) = Db * Z(i,6) + Nb * Z(i,2)$$
$$G(i,6) = Db * Z(i,2) - Nb * Z(i,6)$$
$$G(i,1) = Da * Z(i,7) + Na * Z(i,1)$$
$$G(i,7) = Da * Z(i,1) - Na * Z(i,7)$$
$$G(i,3) = Dc * Z(i,5) + Nc * Z(i,3)$$
$$G(i,5) = Dc * Z(i,3) - Nc * Z(i,5)$$

pour $i = 0, 1, 2, \ldots, 7$

Selon une variante, on peut séparer la transformation en deux passes dans une transformation unidimensionnelle. Ce qui suit est un exemple d'une passe d'une transformation unidimensionnelle. La figure 8 montre ces étapes.

$A1 = X0 + X7$	$B1 = A1 - A2$	$C1 = 1,25\ B1$
$A2 = X3 + X4$	$B2 = A1 + A2$	$C2 = 3\ B1$
$A3 = X2 + X5$	$B3 = A3 + A4$	$C3 = 1,25\ B4$
$A4 = X1 + X6$	$B4 = A4 - A3$	$C4 = 3\ B4$
$A5 = X0 - X7$	$B5 = A6 + A7$	$C5 = 1,5\ A5$
$A6 = X1 - X6$	$B6 = A6 - A7$	$C6 = 1,0625\ B5$
$A7 = X2 - X5$		$C7 = 1,0625\ B6$
$A8 = X3 - X4$		$C8 = 1,5\ A8$

D1 = C5 + C6	E1 = 2,5 D1	Y0 = B2 + B3
D2 = C5 - C6	E2 = 1,25 D2	Y1 = E1 + (0,5 D3)
D3 = C7 + C8	E3 = 2,5 D3	Y2 = C2 + C4
D4 = C7 - C8	E4 = 1,5 D4	Y3 = E2 + D4
		Y4 = B2 - B3
		Y5 = D2 - E3
		Y6 = C1 - C4
		Y7 = (0,5 D1) - E4

On note que toutes les multiplications dans ces équations sont accomplies avec des opérations de décalage et d'addition.

Pour lier ceci à la forme matricielle de la transformation de Chen généralisée, on montre à titre d'exemple le cas du point de vecteur Y6.

$$Y6 = C1 - C4 = (1,25\ B1) - (3\ B4) = 1,25\ (A1 - A2) - 3\ (A4 - A3)$$
$$= 1,25\ ((X0 + X7) - (X3 + X4)) - 3\ ((X1 + X6) - (X2 + X5))$$
$$= 1,25\ X0 - 3\ X1 + 3\ X2 - 1,25\ X3 + 1,25\ X4 + 3\ X5 - 3\ X6 + 1,25\ X7$$
$$Y6/1,25 = X0 - 2,4\ X1 + 2,4\ X2 - X3 + X4 + 2,4\ X5 - 2,4\ X6 + X7$$

$$= \begin{vmatrix} 1 & -b & b & -1 & 1 & b & -b & 1 \end{vmatrix} x$$

avec b = 2,4. Ceci correspond à la sixième ligne de la matrice P dans l'équation. On note que la division par 1,25 correspond à un facteur d'échelle qui est collecté dans la matrice de facteurs d'échelle de taux.

On fait passer par ce réseau additionneur les données de rangées d'un bloc de 8 x 8 pixels. On transpose les composantes de fréquence unidimension-

nelles résultantes, et on les fait passer à nouveau par le même réseau.

6. Après l'étape 5.1 dans chaque sous-bloc d'image, et pour chacune des 64 positions (i,j), en utilisant k(i,j) et s(i,j) provenant de l'étape (4), préparer la valeur :

$$L(i,j) = G(i,j) * k(i,j) * 2^{-s(i,j)}$$

mais si cette valeur est négative (ou si i = j = 0), additionner 1 à cette valeur. Ce résultat est le coefficient de transformée L(i,j).

Commentaires concernant l'étape 6 :

Les calculs ici sont simples du fait que :

- k(i,j) est toujours égal à 1, 3, 5, 7 ou 9 et est habituellement égal à 1.
- La multiplication par 2^(-s(i,j)) est simplement un décalage à droite (ou éventuellement un décalage à gauche si on a choisi M très grand).

Les décalages à droite arithmétiques donnent toujours un arrondi vers le bas. On désire en réalité un arrondi vers zéro; d'où la condition "si (valeur négative) additionner 1".

L'addition de 1 lorsque i = j = 0 repose sur le fait que v(i,j) >= 0, et est juste un moyen pour simplifier la condition de l'étape (9.1) ci-dessous.

7. Coder, enregistrer et/ou émettre les valeurs L(i,j). En fin de compte, ces valeurs seront récupérées et l'image sera reconstruite par les étapes suivantes.

8. Ceci est la version inverse de tous(i,j). Effectuer une itération passant par les 64 positions de transformation (i,j), en calculant m(i,j) sous la forme de l'entier le plus proche de

$$m(i,j) = \frac{U(i)^2 * U(j)^2 * Zr(i) * Zc(j) * Debl(i,j)}{Zr'(i) * Zc'(j) * k(i,j) * 2^{4-S-s(i,j)}}$$

Dans cette expression, s(i,j) et k(i,j) sont calculés à l'étape (4) ci-dessus et les termes "Z" sont définis à l'étape (4).

De plus, choisir A(i,j) sous la forme de l'entier le plus proche de

$$A(0,0) = \frac{2^{S-2}}{Drc' * Drr'} - 0,5 * m(0,0)$$

$$A(i,j) = m(i,j) * (25 - i - j) / 64$$

pour i = 0 ou j = 0

Commentaires concernant l'étape 8 :

Les valeurs m(i,j) peuvent avoir été précalculées ci-dessus à l'étape (4) et transmises avec l'image comprimée. Ceci n'est pas nécessaire pour A(i,j), qui dépend seulement de constantes et de m(i,j). Dans des applications dans lesquelles le facteur d'échelle de taux et les poids de suppression de flou sont fixés, les valeurs m(i,j) et A(i,j) seront constantes. Le facteur 2^S représentent des bits de précision supplémentaires qui seront ultérieurement éliminés par des décalages arithmétiques à droite dans les étapes (9.2) et (10).

L'ajustement relatif à A(0,0) corrige un biais d'arrondi, pour permettre l'utilisation des informations de sortie ci-dessous sans correction d'arrondi.

Dans le cas considéré ici, l'obtention de A(0,0) repose sur l'addition de 1 à L(0,0) à l'étape (6).

L'interpolation "(25 - i - j) / 64" est heuristique, mais elle est approximativement optimale au sens des moindre carrés de l'erreur. Ici encore, on utilise la version entrelacée d'ordre 20.

9. Effectuer une itération dans l'image transformée en accomplissant ce qui suit sur chaque bloc de huit par huit des valeurs de luminance transformées L(__,__) qui sont obtenues à l'étape (5) ci-dessus :

9.1 Préparer les valeurs :

$$E(i,j) = L(i,j) * m(i,j) + A(i,j)$$

pour $L(i,j) > 0$

$$E(i,j) = L(i,j) * m(i,j) - A(i,j)$$

pour $L(i,j) < 0$

$$E(i,j) = 0$$

pour $L(i,j) = 0$

pour chaque (i,j), $i = 0,1,2,...,7$ et $j = 0,1,2,...,7$.

On doit toujours additionner A(0,0). La présente invention couvre également le cas dans lequel le test $L(0,0) > 0$ n'est pas effectué et les étapes (6) et (8) ci-dessus sont simplifiées (facultativement).

En pratique, on doit reconnaître de petites multiplications, par exemple $-11 < L(i,j) < 11$ comme des cas spéciaux, pour éviter le coût de calcul que représente une multiplication.

9.2 (Si ceci est commode pour réduire le coût du dispositif à semiconducteurs, décaler les nombres E(i,j) d'un nombre arbitraire de positions S1 vers la droite. Il faut noter que ces décalages sont "libres" dans certaines formes de réalisation du procédé. Dans des formes de réalisation dans lesquelles le décalage n'est pas libre, on peut choisir de l'omettre lorsque E(i,j) est égal à zéro. On peut également choisir d'éliminer tous les décalages en fixant S1 = 0.)

9.3 Dans la forme bidimensionnelle, préparer à nouveau les valeurs suivantes :

```
F(0, j) = E(4, j) + E(0, j)
F(4, j) = E(0, j) - E(4, j)
F(2, j) = Db * E(6, j) + Nb * E(2, j)
F(6, j) = Db * E(2, j) - Nb * E(6, j)
F(1, j) = Da * E(7, j) + Na * E(1, j)
F(7, j) = Da * E(1, j) - Na * E(7, j)
F(3, j) = Dc * E(5, j) + Nc * E(3, j)
F(5, j) = Dc * E(3, j) - Nc * E(5, j)

H(0, j) = F(0, j) + F(2, j)
H(1, j) = F(4, j) + F(6, j)
H(2, j) = F(4, j) - F(6, j)
H(3, j) = F(0, j) - F(2, j)
H(4, j) = F(7, j) - F(5, j)
H5(j)   = F(7, j) + F(5, j)
H6(j)   = F(1, j) - F(3, j)
H(5,j)  = H6(j)   + H5(j)
H(7, j) = F(1, j) + F(3, j)
```

pour j = 0, 1, 2, ..., 7

9.4 Préparer les valeurs :

```
G(i, 0) = H(i, 4) + H(i, 0)
G(i, 4) = H(i, 0) - H(i, 4)
G(i, 2) = Db * H(i, 6) + Nb * H(i, 2)
G(i, 6) = Db * H(i, 2) - Nb * H(i, 6)
G(i, 1) = Da * H(i, 7) + Na * H(i, 1)
G(i, 7) = Da * H(i, 1) - NA * H(i, 7)
G(i, 3) = Dc * H(i, 5) + Nc * H(i, 3)
G(i, 5) = Dc * H(i, 3) - Nc * H(i, 5)

M(i, 0) = G(i, 0) + G(i, 2)
M(i, 1) = G(i, 4) + G(i, 6)
M(i, 2) = G(i, 4) - G(i, 6)
M(i, 3) = G(i, 0) - G(i, 2)
M(i, 4) = G(i, 7) - G(i, 5)
M5(i)   = G(i, 7) + G(i, 5)
M6(i)   = G(i, 1) - G(i, 3)
M(i, 5) = M6(i) - m5(i)
M(i, 6) = M6(i) + M5(i)
M(i, 7) = G(i, 1) + G(i, 3)
```

pour i = 0, 1, 2, ..., 7

9.5 Multiplier chaque M(i,j) par

(si i = 0, 2, 3 ou 4 :)

Nrc'	si j = 5 ou 6
Drc'	si j = 5 ou 7
1 (pas d'action)	si j = 0, 1, 2 ou 4

(si i = 4 ou 7 :)

Drr' Nrc'	si j = 5 ou 6
Drr' Drc'	si j = 4 ou 7
Drr'	si j = 0, 1, 2 ou 3

(si i = 5 ou 6 :)

Nrr' Nrc'	si j = 5 ou 6
Nrr' Drc'	si j = 4 ou 7
Nrr'	si j = 0, 1, 2 ou 3

9.6. Préparer les valeurs

$$Z(i, 0) = M(i, 0) + M(i, 7)$$
$$Z(i, 1) = M(i, 1) + M(i, 6)$$
$$Z(i, 2) = M(i, 2) + M(i, 5)$$
$$Z(i, 3) = M(i, 3) + M(i, 4)$$
$$Z(i, 4)\ M(i, 3) - M(i, 4)$$
$$Z(i, 5) = M(i, 2) - M(i, 5)$$
$$Z(i, 6) = M(i, 1) - M(i, 6)$$
$$Z(i, 7) = M(i, 0) - M(i, 7)$$

pour i = 0, 1, 2, ..., 7

9.7 Préparer les valeurs

$$Y(0, j) = Z(0, j) + Z(7, j)$$
$$Y(1, j) = Z(1, j) + Z(6, j)$$
$$Y(2, j) = Z(2, j) + Z(5, j)$$
$$Y(3, j) = Z(3, j) + Z(4, j)$$
$$Y(4, j) = Z(3, j) - Z(4, j)$$
$$Y(5, j) = Z(2, j) - Z(5, j)$$
$$Y(6, j) = Z(1, j) - Z(6, j)$$
$$Y(7, j) - Z(0, j) - Z(7, j)$$

pour j = 0, 1, 2, ..., 7

10. Après l'étape 9.7 dans chaque sous-bloc d'image, et pour chacune des 64 positions (i,j), préparer la valeur :

$$V(i,j) = Y(i,j) * 2^{S1 - S}$$

dans laquelle S et S1 sont des entiers arbitraires définis aux étapes (7) et (9.2) ci-dessus. Ici encore, la multiplication est en réalité un décalage à droite.

11. En fonction de systèmes particuliers, il peut maintenant être nécessaire d'effectuer un contrôle de plage. Par exemple, si la plage de luminance admissible est $0 \leq v(i,j) \leq 255$, on doit remplacer les valeurs de V(i,j) inférieures à zéro ou supérieures à 255, respectivement par 0 et 255.

Les valeurs v(i,j) sont maintenant les valeurs de luminance de l'image reconstruite.

Considérations concernant des processus secondaires

On complète habituellement le processus de base par des mesures supplémentaires visant à améliorer la compression ou la qualité de l'image.

Après l'étape (10), on peut améliorer l'exactitude de l'image par une itération passant par toutes les paires de pixels V(8I + 7, j), V(8I + 8, j) et par toutes les paires de pixels V(k, 8J + 7), V(i, 8J + 8) (c'est-à-dire des pixels adjacents qui étaient séparés dans des blocs d'image séparés), et en incrémentant et en décrémentant respectivement leurs valeurs v1, v2, par exemple par (v2 - V1) / max (2, 11 $\sqrt{(M)}$), en désignant par M le facteur d'échelle de taux qui est utilisé à l'étape (4), et l'expression qui figure au dénominateur étant ici encore simplement une approximation commode de l'optimum.

Avant d'accomplir l'étape (6), on peut classifier la difficulté subjective de la zone d'image locale, de façon à la faire correspondre de préférence à un type parmi trois, à savoir simple précision, double précision ou quadruple précision, avec respectivement le préfixe codé "0", "10" ou "11" pour l'information de sortie. Le calcul de l'étape (6) est maintenant remplacé par

$$L(i,j) = G(i,j) * J(i,j) * 2^{P-s(i,j)}$$

avec p = 0, 1 ou 2 respectivement pour la simple précision, la double précision et la quadruple précision. Ceci est compensé ultérieurement à l'étape (9.2) à laquelle la précision ajoutée doit être éliminée avec un décalage à droite (augmenté).

Malheureusement, on n'a trouvé aucune technique de classification qui soit simple et très efficace. On utilise actuellement une technique malcommode qui déduit la mesure de difficulté de P de quatre sources :

a) P_gauche et P_haut, qui sont les mesures de difficulté de zones d'image voisines,

b) $\sum(i+j)G(i,j)^2) / \sum G(i,j)^2$, qui est le biais d'énergie de la transformée

c) -G(0,0), qui est la luminance moyenne inverse, et

d) max ($\sum$ (Histogramme (v(i,j))) _de_largeur_fixe), qui est l'uniformité.

A l'étape (7), les données de transformée L(,) à enregistrer ou à transmettre peuvent faire l'objet d'une réduction supplémentaire en utilisant un procédé de codage par entropie. On utilise et on recommande de mettre en oeuvre le code de plage en zigzag/gabarit du CCITT, avec plusieurs tables de Huffman prises par défaut, en fonction du débit binaire. Pour que l'exposé soit complet, on décrit en détail un exemple de ceci au paragraphe suivant.

Exemple de format de fichier comprimé

Une image comprimée est représentée par les informations suivantes :

1) Préface (largeur et hauteur d'image, facteur d'échelle de taux M, etc.)

2) Bloc de pixels 0
Bloc de pixels 1
Bloc de pixels 2
...
Bloc de pixels N-1
3) Posface (éventuellement)

et chaque Bloc de pixels est représenté par les informations suivantes :

1) Code de précision (déterminé par l'étape facultative Z)
2) Code delta de coefficient continu
3) Code de coefficient alternatif (répété zéro fois ou plus)
4) Code de fin de bloc

chaque Code de coefficient alternatif étant représenté par les informations suivantes

1) Extension neuf-zéro (répétée E fois, E 0)
2) Code de plage/gabarit désignant (R, T)
3) Valeur de signe de coefficient (1 bit)
4) Valeur absolue de coefficient avec le bit de plus fort poids supprimé (T bits).

R+(*E) est le nombre de coefficients de valeur zéro qui précèdent le coefficient considéré dans un ordre en "zigzag" (séquence basée sur la somme i+j), et T est la position de bit du bit de plus fort poids (MSB) de la valeur absolue du coefficient, par exemple T = 3 lorsque le coefficient est égal à 11 ou -11 :

position de bit : 876543210

11 = 000001011 (en binaire)

← bit de plus fort poids

On ne décrira pas en dédail le choix ou le codage concernant le code delta de coefficient continu, mais on donnera effectivement un exemple de code de Huffman utile à des débits binaires élevés, pour les codes de plage/gabarit alternatifs.

Code	R	T
0xx	0	w
100x	0	4+w
111110	0	6
1111110{0}	0	7+n
1010	1	0
10110	1	1
10111	2	0
1100xx	1+w	max(0,2-w)
11010{0}1xx	1+w	n+1+max(0,2-w)
11011xx	5+w	0
111100{0}>1xx	1+w	n+1+max(0,2-w)
11011xx	5+w	0
111100{0}>1xx	%+w	1+n

1111111 = Réservé
111101 = Extension Neuf-Zéro
1110 = Code de fin de bloc

avec les notations suivantes :

0 désigne n zéros consécutifs, n = 0, 1, 2, 3, ...
xx désigne 2 bits interprétés comme w = 1, 2, ou 3
x désigne 1 bit interprété comme w = 0 ou 1

Transformations à 128 points et 256 points

On peut utiliser le procédé précédent avec une plus grande transformation de Chen généralisée (GCT), de type 8 par 16 ou 16 par 16. Le procédé pour généraliser encore davantage la transformation de Chen apparaît clairement lorsqu'on note que la transformation GCT unidimensionnelle à 16 points est donnée (avec les rangées en "ordre papillon" et sans les post-multiplicateurs de normalisation qui sont nécessaires) par :

GCT_16 (a, b, c, e, f, g, h, r, s, t) =

GCT_8(a, b, c, r) GCT_*(a, b, c, r)
GQ(e,f,g,h,r,s,t,) - GQ(e,f,g,h,r,s,t)

avec GCT8(a, b, c, r) =

1	1	1	1	1	1	1	1
1	-1	-1	1	1	-1	-1	1
b	1	-1	-b	-b	-1	1	b
b	1	-1	-b	-b	-1	1	b
a	ar+r	ar-r	1	-1	r-ar	-r-ar	-a
1	-cr-r	cr-r	c	-c	r-cr	cr+r	-1
c	r-cr	-cr-r	-1	1	cr+r	cr-r	-c
1	r-ar	ar+r	-a	a	-r-ar	ar-r	-1

et avec GQ8(e, f, g, h, r, s, t) =

Les paramètres de type "cosinus vrai" ont ici les valeurs :

g = tangente 15pi/32 ~=10,1532.
a = tangente 14pi/32 ~= 5,0273
f = tangente 13pi/32 ~= 3,2966
b = tangente 12pi/32 ~= 2,4142
g = tangente 11pi/32 ~= 1,8709
c = tangente 10pi/32 ~= 1,4966
h = tangente 9pi/32 ~= 1,2185
r = cosinus 8pi/32 ~= 0,7071
t = cosinus 2pi/32 ~= 0,3827
s = cosinus 4pi/32 = t * b

Les paramètres que l'on utilise sont :

e=10
a=5
f=3,25
b=2,4
g=1,875
c=1,5
h=1,25
r=17 / 240,708333
r=17 / 240,708333
t = 5 / 13 ~= 0,384615
s=t * b=12/ 13

L'inverse de GQ8(e, f, g, h, r, s, t) est la transposée de GQ8(e, r, g, h, 1/2r, t', b, t')
avec :

b = s / t

t' = 1 / (t+t*b*b)

Exemples de matrices

Transposée de la matrice TP

Matrice de transformation en cosinus
(a = 5,02734 b = 2,41421 c = 1,49661 r = 0,70711) :

	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768
	0,2452	0,2079	0,1389	0,0488	-0,0488	-0,1389	-0,2079	-0,2452
	0,2310	0,0957	-0,0957	-0,2310	-0,2310	-0,0957	0,0957	0,2310
	0,2070	-0,0488	-0,2452	-0,1389	0,1389	0,2452	0,0488	-0,2079
5	0,1768	-0,1768	-0,1768	0,1768	0,1768	-0,1768	0,1768	0,1768
	0,1389	-0,2452	0,0488	0,2079	-0,2079	-0,0488	0,2452	-0,1389
	0,0957	-0,2310	0,2310	-0,0957	-0,0957	-0,2310	-0,2310	0,0957
	0,0488	-0,1389	0,2079	-0,2452	0,2452	0,2452	-0,2079	0,1389
10	Transformation de Chen associée			(a = 5,0 b = 2,4 c = 1,5 r = 0,7)				
	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768	0,1768
	0,2451	0,2059	0,1373	0,0490	-0,0490	-0,1373	-0,2059	-0,2451
	0,2308	0,0962	-0,0962	-0,2308	-0,2308	-0,0962	0,0962	0,2308
	0,2080	-0,0485	-0,2427	-0,1387	0,1387	0,2427	0,0485	-0,2080
15	0,1768	-0,1768	-0,1768	0,1768	0,1768	-0,1768	-0,1768	0,1768
	0,1387	-0,2427	0,0485	0,2080	-0,2080	-0,0485	0,2427	-0,1387
	0,0962	-0,2308	0,2308	-0,0962	-0,0962	0,2308	-0,2308	0,0962
	0,0490	-0,1373	0,2059	-0,2451	0,2451	-0,2059	0,1373	-0,0490

Description de l'appareil

20 Maintenant qu'on a examiné l'invention en détail, on va décrire un appareil qui met en oeuvre des aspects de l'invention.

Dans ce qui suit, on utilise le terme "point" pour désigner un registre de facteurs ou un 25 chemin de données de précision arbitraire, comprenant de façon caractéristique 8 à 12 bits. On connaît un procédé pour déterminer la précision appropriée [6].

Dans le procédé par logiciel, on combine des étages de transformation, et on a adopté l'amélioration de Wu-Paolini. Pour l'appareil à semiconducteurs, 30 il est plus commode d'utiliser simplement deux unités de transformation à 8 points, l'une pour la direction horizontale et l'autre pour la direction verticale. Il est nécessaire de prévoir un réseau de décalage à 64 35 points entre les transformations verticale et horizontale, ainsi qu'une structure tampon similaire entre la

section de transformation et la section de codage.

Bien que l'invention puisse utiliser un appareil monochrome et / ou des appareils séparés pour la compression et l'extension, un mode de réalisation préféré (figure 7) comprend à la fois un compresseur (figure 1A) et un extenseur (figure 1B) qui travaillent sur des données correspondant à trois couleurs. Des données sont admises dans le compresseur (figure 2A) sous la forme de vecteurs de 8 pixels qui sont en outre arrangés selon un ordre lexicographique en blocs de 64 pixels. Le traitement des blocs est accompli en une configuration pipeline (figure 2B).

Un pixel qui est appliqué au compresseur comprend des facteurs d'échelle "R" (rouge); "G" (vert) et "B" (bleu). Ceux-ci sont immédiatement transformés dans un espace de luminance-chrominance. (Les raisons d'une telle transformation sont bien connues.)

La transformation peut utiliser des coefficients arbitraires, fixes ou programmables (figure 3A), ou bien elle peut être "câblée" en correspondance avec des valeurs simples dans une application spécialisée (figure 3B). On désigne ici par XYZ l'espace de la transformation, mais on peut utiliser n'importe quelle forme linéaire de l'information d'entrée à trois couleurs, comme par exemple la forme standard du CCITT : (Y, R-Y, B-Y). Chacune des trois valeurs X, Y et Z est ensuite effectivement appliquée à un compresseur monochrome séparé. L'extenseur utilise un circuit identique ou similaire à celui des figures 3A, 3B, à l'exception du fait que maintenant un vecteur XYZ est transformé en un vecteur RGB.

Les valeurs Y, X et Z sont ensuite appliquées à trois registres à décalage (figure 5), pour attendre l'application à la première unité de trans-

formation. La première unité de transformation travaille en une durée de 2,6 périodes de pixel, ce qui fait que certaines des données doivent être retardées, comme représenté. La désignation "XYZ" est un peu malheureuse; des procédés de codage optimisés exigent que la liminance ("Y") soit traitée en premier.

Pendant l'extension, le problème du biais XYZ est inversé. On note que 5 points de registres sont économisés dans le mode de réalisation préféré, en inversant l'utilisation des registres à décalage Y et Z pendant l'extension.

En se référant à la figure 1A, on note que les principales sections du compresseur comprennent une section d'entrée (1, 2) qui transforme l'information d'entrée en une information dans l'espace XYZ et qui l'enregistre en tampon pour son transfert ultérieur vers l'unité de transformation (3). Pour chaque période de huit pixels, l'unité de transformation 1 doit accomplir trois cycles (un pour chacune des informations X, Y et Z). L'information de sortie de l'unité de transformation 1 est placée dans le réseau de décalage (4) dans lequel elle est conservée jusqu'à ce que le bloc de 8 x 8 pixels ait été entièrement lu. L'unité de transformation 2 (5, 6) travaille sur le bloc de pixels qui a été lu précédemment, en accomplissant également trois cycles pendant chaque période de huit pixels, et elle fournit des données au Tampon d'Entrée de Codeur (7, 8). Le Codeur (9, 10, 11) est également utilisé de façon partagée entre les trois coordonnées de couleurs, mais un bloc de luminance entier sera codé sans interruption, en étant suivi par chacun des blocs de chrominance. Si le traitement de ces trois blocs ne peut pas être entièrement accompli au cours de 64 périodes de pixel, la logique de commande et de définition des conditions temporelles

bloquera l'horloge de pixel qui est appliquée au circuit d'entrée externe.

Les zones de mémorisation d'information (Registre à Décalage d'Entrée (2), Réseau de Décalage (4) et Tampons d'Entrée de Codeur (7, 8)) doivent être triplées pour les trois couleurs, mais les unités de calcul (3, 5, 6, 9, 10, 11) sont utilisées en temps partagé (multiplexage temporel) entre les données Y, X et Z.

Le codeur (9, 10, 11), le Tampon d'Entrée de Codeur (7, 8), la programmation de code (12, 13, 14) et la logique de commande et de définition des conditions temporelles (non représentées) peuvent être conformes à la technique ou à la pratique existante. De façon similaire, le procédé pour le multiplexage temporel de trois couleurs dans un seul circuit est bien connu. La section de transformation à 3 points (1) (figures 3A, 3B) et les Registres à Décalage (2) (figure 5) sont également bien connus.

Le circuit d'application de facteur d'échelle (6) utilise une mémoire vive ou une mémoire morte programmée et un système de décalage (implicite), de multiplexeurs et d'additionneurs. La mise en oeuvre de ceci ne présente aucune difficulté. La conception du Dispositif de Transformation à 8 Points (figures 8A, 8B) n'offre également aucune difficulté, connaissant la définition de la Transformation de Chen généralisée et les paramètres appropriés.

Le Réseau de Décalage (figure 6A) mérite un examen spécial. Des vecteurs verticaux (transformés) provenant du bloc de pixels d'entrée courant sont assemblés pendant que des vecteurs horizontaux provenant du bloc de pixels précédent sont appliqués au dispositif de transformation en direction horizontale. En l'absence d'une conception spéciale, ceci exigerait

128 registres (64 pour chaque bloc comprenant le bloc courant et le bloc précédent), du fait que les points sont utilisés dans un ordre différent de celui dans lequel ils sont reçus. On peut cependant éliminer cette nécessité en décalant les données de la gauche vers la droite pendant des blocs de pixels de numéro pair, et du haut vers le bas pendant des blocs de pixels de numéro impair.

Le réseau de décalage qui est décrit est bidirectionnel. Un réseau de décalage quadridirectionnel est préférable dans certains modes de réalisation.

La figure 6B montre de façon plus détaillée l'aspect du réseau de décalage de la figure 6A. Sur la figure 6B, des vecteurs sont retirés du réseau de décalage au bas de celui-ci, un par un, et ils sont émis vers la section DECT8 de la figure 1A. Pendant ce temps, des vecteurs verticaux provenant de l'autre section DCT8 sont introduits dans le réseau de décalage, en haut de celui-ci. Progressivement, les anciens vecteurs sont retirés du réseau de décalage, et ce dernier se remplit complètement avec des vecteurs verticaux provenant du bloc de pixels suivant.

Pour le bloc de pixels suivant, la direction de circulation des données diffère maintenant de 90° de la direction de circulation des données dans un bloc de pixels précédent. De cette manière, les vecteurs horizontaux seront retirés à droite du réseau de décalage et ils seront émis vers la section DCT8, et de nouveaux vecteurs verticaux arriveront du côté gauche. Au passage au bloc N + 2, une autre rotation de 90° ramène à la forme d'origine, et ainsi de suite.

L'extenseur (figure 1B) a une structure tout à fait similaire à celle du compresseur (figure 1A),si on excepte que la direction de circulation des données est maintenant inversée. Dans un mode de réali-

sation préféré, un seul appareil fonctionne selon deux modes, en compresseur ou en extenseur.

Différentes configurations possibles de circuits à très haut niveau d'intégration (VLSI) (figures 4A, 4B) conduisent à différents flux de données pour la compression (figures 4A, 4B) et l'extension (figures 4A, 4B). On note que le fonctionnement des unités de transformation et de réseau de décalage présente le même sens pour la compression et l'extension dans une configuration (figure 4B), mais non dans l'autre (figure 4A). Ceci apparaît plus clairement lorsqu'on considère le flux de données du compresseur/extenseur combiné (figure 7). Lorsque les deux unités de transformation sont respectivement associées avec des données RGB et des données comprimées (figure 4A), des difficultés liées à la configuration apparaissent, sauf si on utilise un réseau de décalage à quatre directions. Par conséquent, on associe aux deux unités de transformation (figure 4B) respectivement les sections d'entrée et de sortie du réseau de décalage.

Dans un mode de réalisation, l'unité de transformation qui est utilisée dans le compresseur (figure 8A) emploie 38 additionneurs. Le décalage à droite d'une position ("R1"), de deux positions ("R2") ou de quatre positions ("R4"), ou le décalage à gauche d'une position ("L1") s'effectuent aisément. Le circuit qui est représenté utilise les paramètres (a,b,c,r) = (5, 2,4, 1,5, 17/24). Une forme de réalisation avec b = 2,5 n'exigerait que 36 additionneurs, dans un autre mode de réalisation (figure 8B).

Un circuit associé est nécessaire pour l'unité de transformation inverse dans l'extenseur. En utilisant soigneusement des signaux de "validation de sortie", il est possible de réutiliser la majeure partie des additionneurs dans l'unité de transformation

directe. La réalisation de ceci n'offre aucune difficulté pour l'homme de l'art.

Le dispositif d'application de facteurs d'échelle utilise une mémoire vive ou une mémoire morte programmée, et un système de décalage implicité, de multiplexeurs et d'additionneurs. La réalisation de ceci n'offre aucune difficulté.

Le dispositif d'application de facteurs d'échelle inverses peut être réalisé de diverses manières, en employant de préférence un petit multiplieur câblé avec une mémoire vive, un accumulateur, une logique de commande et de définition de conditions temporelles, et un petit dispositif de découpage à gabarit. Dans une application spécialisée à faible coût, on peut simplifier le dispositif d'application de facteurs d'échelle inverses, en notant que les poids de suppression de flou sont presque optimaux sur une plage étendue; on peut donc utiliser une opération simple, comme dans le dispositif d'application de facteurs d'échelle.

Le dispositif d'application de facteurs d'échelle inverses peut être placé soit entre le codeur et son tampon de sortie, soit entre le tampon de sortie et une unité de transformation, comme indiqué sur les figures 1B et 7.

On peut réaliser de diverses manières le tampon d'entrée de codeur, celles-ci comprenant l'utilisation d'une structure de réduction de registres par partage de cycles, similaire à celle du réseau de décalage. Une structure plus directe utilise une mémoire vive de 384 par 10 bits, avec une mémoire morte de 64 par 7 bits pour fournir les adresses de la mémoire vive.

On va maintenant décrire un exemple de cycle de fonctionnement, en se référant aux figures 1A et 1B.

Sur la figure 1A, des données entrent dans le compresseur sous la forme d'une information correspondant à trois couleurs (rouge, vert et bleu). Elles sont immédiatement transformées en données dans un autre espace, que l'on appelle XYZ. Chacun des trois éléments X, Y et Z entre dans son propre registre à décalage.

A partir du registre à décalage (Etape 2), les données passent à une unité de transformation DCT à 8 points. Il pourrait y avoir une seule unité de transformation DCT à 8 points, multiplexée entre les trois couleurs X, Y et Z, ou bien chaque couleur pourrait avoir sa propre unité de transformation DCT 8 individuelle.

L'information entre ensuite dans le réseau de décalage à 64 points (4). Il y a un réseau de décalage individuel pour chaque couleur. A partir du réseau de décalage (bloc 4), l'information passe à une autre unité de transformation DCT (bloc 5) qui est similaire au bloc 3. On doit ensuite appliquer un facteur d'échelle à l'information, ce qui correspond à une couche supplémentaire de décalage ajouté.

L'information est seulement transformée en direction horizontale et en direction verticale. Le réseau de décalage fait en réalité tourner théoriquement les données de 90°, de façon qu'elles puissent être transformées dans l'autre direction. Après l'application d'un facteur d'échelle, les données entrent dans un autre tampon, correspondant aux blocs 7 et 8 (Z1 et Z2), qui est destiné à conserver les données de façon qu'elles puissent finalement être codées et émises par la puce (Z1, Z2 correspondent au traitement en zigzag).

D'un point vue théorique, ceci est semblable au réseau de décalage (bloc 4), à l'exception du fait

du fait que maintenant les données ne subissent pas une rotation de 90°. A la place, elles sont transformées dans l'ordre en zigzag, qui est habituellement utilisé dans ce domaine, et qui est utilisé par la norme CCITT. L'information est ensuite présentée au bloc d'unité de commande de plage et de gabarit, 9, qui détecte des zéros et crée des plages pour les zéros, qui détecte des valeurs différentes de zéro et qui produit une estimation du logarithme de la valeur, que l'on appelle le gabarit. La combinaison de plage/gabarit est recherchée dans une mémoire vive ou une mémoire morte, correspondant à ce qu'on appelle le code RT, et elle est ensuite émise par la puce.

La mantisse, qui correspond aux bits de fort poids des coefficients de transfert, est également émise par la puce. Du fait que la mantisse et le code de plage/gabarit ont une longueur arbitraire, à savoir un bit, deux bits, ou un nombre quelconque, et l'information de sortie de la puce a toujours 16 bits, ou 8 bits, 32 bits ou un nombre quelconque, le bloc 11 (alignement) facilite ceci.

Les autres blocs qui sont représentés sur la figure 1A sont des blocs de programmation (facultatifs), 12, 13 et 14, qui permettent respectivement de fixer une transformation de RGB en XYZ arbitraire, des facteurs d'échelle de taux et des poids psycho-adaptatifs arbitraires, et un code de Huffman modifié arbitraire pour la plage et le gabarit.

La figure 1B est très similaire à la figure 1A. Le code de plage et de gabarit doit maintenant être décodé en une combinaison de plage et de gabarit arbitraire, et le nombre de zéros nécessaire doit être supprimé. Sur la figure 1A, le dispositif d'application de facteur d'échelle est un simple réseau d'additionneurs et de circuits de décalage. Sur la figure 1B,

le dispositif d'application de facteurs d'échelle inverses est réalisé sous la forme d'un très petit multiplicateur, par matériel.

La figure 9 montre un schéma relatif à la transformation de Chen bidimensionnelle généralisée. Des pixels entrent par le haut et ont une largeur caractéristique de 8 bits. Les pixels traversent un réseau d'additionneurs de grande largeur dans l'unité de transformation horizontale 10, avec une largeur de données d'une valeur caractéristique de 128 bits. L'information de sortie de l'unité de transformation horizontale 10 traverse une mémoire vive de transposition 12 qui fait tourner l'information de la direction horizontale vers la direction verticale. Les données entrent ensuite dans l'unité de transformation verticale 16 qui ne comprend également que des additionneurs (d'une largeur caractéristique de 128 bits). Les coefficients de sortie sont finalement réduits à une largeur d'approximativement 16 bits, et ils traversent un seul multiplieur 20 qui, conformément à l'invention, est compatible avec la norme JPEG.

La figure 10 montre un schéma synoptique pour la réalisation sous forme d'un circuit à très haut niveau d'intégration (VLSI) conformément à l'invention. Sur la figure 10, les données entrent dans un composant 40 et sont mémorisées dans le réseau de bascules d'entrée 42, et elles traversent un multiplexeur 44 pour entrer dans la première moitié de l'unité de transformation GCT 50 (qui est un réseau additionneur). La seconde moitié du réseau additionneur 60 se trouve à droite des bascules de milieu d'étage 54. L'information de sortie est transmise par le multiplexeur 62 à la mémoire vive de transposition 66, qui effectue la transformation faisant passer de la direction horizontale à la direction verticale.

L'information de sortie de la mémoire vive de transposition 66 est renvoyée vers le premier étage de l'unité de transformation GCT 50, dans le but de former la première moitié de la transformée verticale, dans une configuration de partage de temps ou de découpage du temps en tranches.

L'information de sortie de l'unité de transformation GCT 50 est appliquée à l'entrée du second étage de l'unité de transformation verticale 60. Enfin, l'information de sortie de l'unité de transformation GCT 60 est transmise par le multiplexeur de sortie avec mémorisation, 70, et elle est dirigée par l'intermédiaire du multiplieur 74 et du dispositif d'arrondi 76 au dispositif d'arrangement en zigzag 80, dont l'information de sortie est émise sous la forme d'un coefficient à douze bits 84.

On va maintenant décrire brièvement le processus de transformation inverse conforme à l'invention, en se référant toujours à la figure 10.

Sur la figure 10, les coefficients à 12 bits sont appliqués par l'intermédiaire du bloc 84 à l'entrée Y du dispositif d'arrangement en zigzag 80. L'information de sortie du dispositif d'arrangement en zigzag 80 traverse le multiplieur 74 et le dispositif d'arrondi qui accomplissent un processus de quantification inverse similaire à celui qui est accompli dans le processus direct. L'information de sortie du multiplieur 74 est appliquée au réseau de bascules 42, qui est le premier étage du processus de transformation inverse.

A partir du réseau de bascules 42, le processus de transformation inverse suit le même chemin en multiplex temporel à deux étages que le processus direct. L'information de sortie apparaît sur le réseau de bascules de sortie 70, et cette information de

sortie consiste en pixels qui sont arrondis par le dispositif d'arrondi 76, dont l'information de sortie est appliquée au bloc 40 pour être présentée en sortie.

La description précédente du mode de réalisation préféré de l'invention a été présentée dans un but d'illustration et de description. Elle ne vise pas à être exhaustive ou à limiter l'invention à la forme précise qui est décrite, et de nombreux changements et modifications sont possibles sur la base de l'information qui précède. De plus, l'invention est compatible avec des normes existantes, telles que la norme JPEG. On a choisi et décrit le mode de réalisation préféré dans le but d'expliquer le mieux possible les principes de l'invention et ses applications pratiques, pour permettre à l'homme de l'art d'utiliser au mieux l'invention ainsi que divers modes de réalisation et diverses modifications qui conviennent pour l'utilisation particulière envisagée.

Références

(1) Acheroy, M. "Use of the DCT for the restoration of an image sequence", SPIE vol 593 Medical Image Processing (1985).

(2) Cooley, et Tukey, JW, "An algorithm for (fast) Fourier series", Math. Comput., XIX N° 90, page 296-301, 1965.

(3) Chen, W. et al., "A fast computational algorithm for the DCT", IEEE Trans. Commun. vol.COM-25 (1977).

(4) Wu, HR et Paolini, FJ, "A 2D Fast Cosine Transform", IEEE Conf. on Image Processing, vol.I (1989).

(5) Lee, BC, "A Fast Cosine Transform", IEEE ASSP, vol XXXIII (1985).

(6) Jalali et Rao, "Limited Wordlength and

FDCT Processing Accuracy", IEEE ASSP-81, vol.III, pages 1180-2.

(7) Wu, H.R. et Paolini, F.J., "A Structural Approach to Two Dimensional Direct Fast Discrete Cosine Transform algorithms", International Symposium on Computer Architecture and Digital Signal Processing Hong Kong, octobre 1989.

REVENDICATIONS

1. Appareil de compression d'image, caractérisé en ce qu'il comprend : des moyens de transformation en direction horizontale (10) qui sont destinés à recevoir des pixels d'entrée ayant une certaine largeur, exprimée en nombre de bits, et à transformer horizontalement des pixels d'entrée en utilisant seulement un réseau d'additionneurs; des moyens de mémoire de transposition (12) destinés à faire tourner les pixels transformés en direction horizontale, pour les amener en direction verticale; des moyens de transformation en direction verticale (16) qui sont destinés à recevoir les pixels verticaux et à transformer verticalement les pixels verticaux en utilisant seulement un second réseau d'additionneurs; et un multiplieur unique (20) qui est destiné à recevoir les pixels verticaux transformés et à accomplir une seule fonction de multiplication sur ces pixels verticaux transformés, pour fournir des données de pixels comprimées qui sont représentatives des pixels d'entrée.

2. Procédé prévu pour être mis en oeuvre dans un appareil de compression d'image, caractérisé en ce qu'il comprend les étapes suivantes : on reçoit des pixels d'entrée ayant une certaine largeur, exprimée en nombre de bits, et on transforme horizontalement ces pixels d'entrée en utilisant seulement un réseau d'additionneurs (10); on fait tourner les pixels transformés en direction horizontale, pour les amener en direction verticale; on reçoit les pixels verticaux et on transforme verticalement les pixels verticaux en utilisant seulement un second réseau d'additionneurs (16); et on reçoit les pixels verticaux transformés et on applique une seule fonction de multiplication à ces pixels verticaux transformés, pour fournir des données de pixels comprimées qui sont

représentatives des pixels d'entrée.

3. Système de compression d'image, caractérisé en ce qu'il comprend : des moyens (40, 42) qui sont destinés à recevoir des données de pixels d'image d'entrée représentatives d'une image; des moyens de transformation de Chen généralisée (ou GCT) (50, 60) qui sont destinés à comprimer les données d'image, ces moyens de transformation GCT comprenant : des moyens additionneurs de transformation GCT (50, 60) qui sont destinés à transformer horizontalement les données d'image en utilisant seulement des additionneurs; des moyens de mémoire de transposition (66) qui sont destinés à faire tourner les pixels transformés en direction horizontale, pour les amener en direction verticale; les moyens additionneurs de transformation GCT (50, 60) comprenant des moyens pour transformer verticalement les pixels verticaux, en utilisant seulement les additionneurs précités; et des moyens multiplieurs (74) pour accomplir une fonction de multiplication sur les pixels verticaux transformés, pour donner des données de pixels comprimées représentatives des pixels d'entrée.

4. Système de compression d'image selon la revendication 3, caractérisé en ce que les moyens additionneurs de transformation GCT comprennent un premier étage de réseau d'additionneurs de transformation GCT (50) qui est destiné à accomplir la première moitié des transformations horizontale et verticale, et un second égage de réseau d'additionneurs de transformation GCT (60) qui est destiné à accomplir la seconde moitié des transformations horizontale et verticale.

5. Système de compression d'image selon la revendication 4, caractérisé en ce que les premiers et seconds moyens additionneurs (50, 60) transforment

horizontalement et verticalement les pixels d'image selon un mode de fonctionnement en temps partagé.

6. Système de compression d'image selon la revendication 5, caractérisé en ce que les moyens multiplieurs (74) comprennent des moyens d'arrangement en zigzag (80).

7. Système de compression d'image selon la revendication 6, caractérisé en ce que les moyens multiplieurs (80) comprennent des moyens effectuant une opération d'arrondi (76).

8. Système de compression d'image selon la revendication 7, caractérisé en ce que les moyens multiplieurs (74) comprennent une structure de table de multiplieur.

FIG. 1A

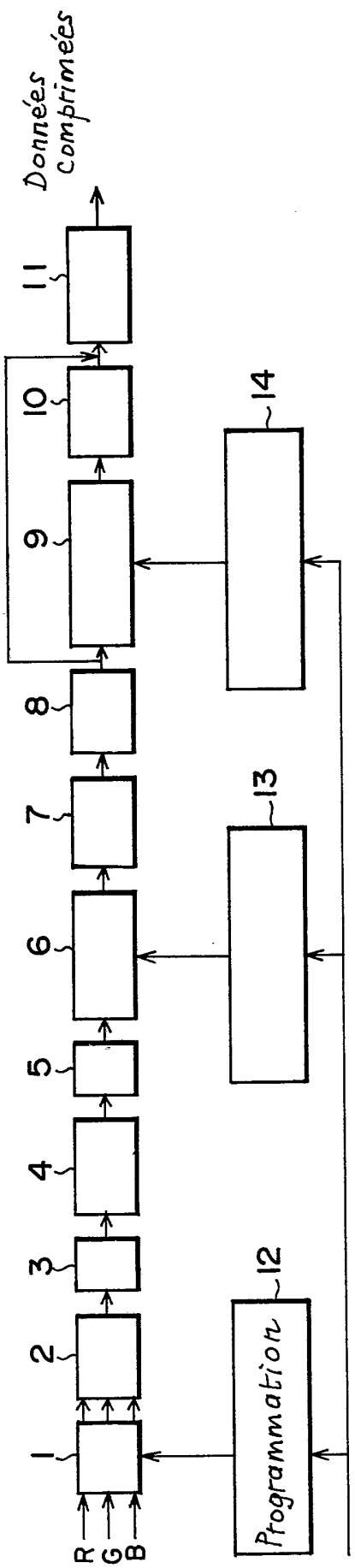

Données comprimées
11
10
9
8
7
6
5
4
3
2
1
14
13
12
Programmation
R
G
B

FIG. 1B

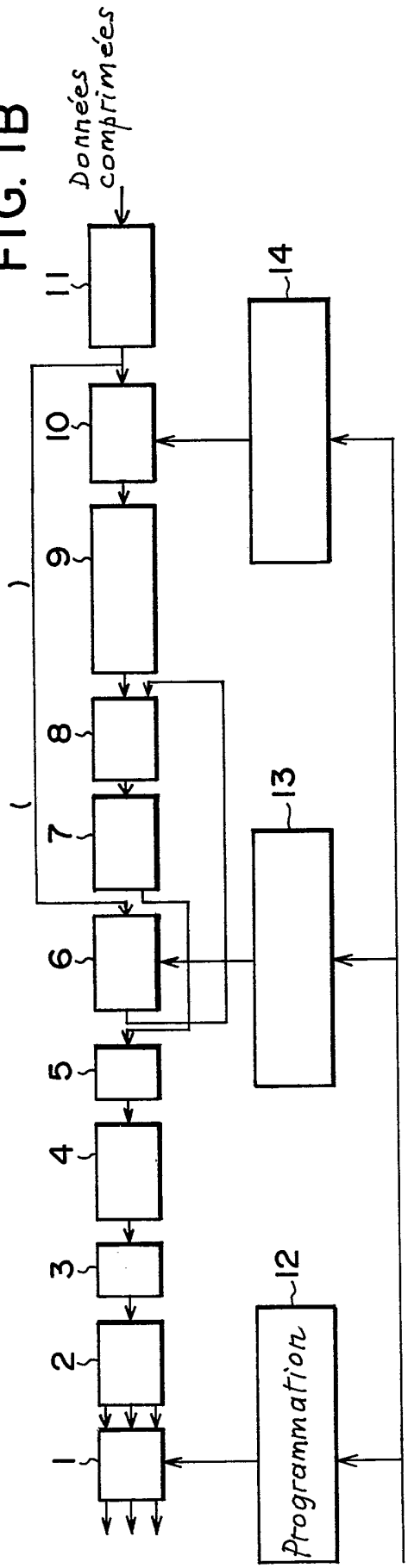
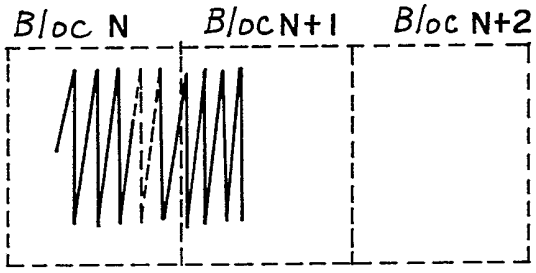

Données comprimées
11
10
9
8
7
6
5
4
3
2
1
14
13
12
Programmation

FIG. 2A



Arrangement des pixels d'entrée

FIG. 2B

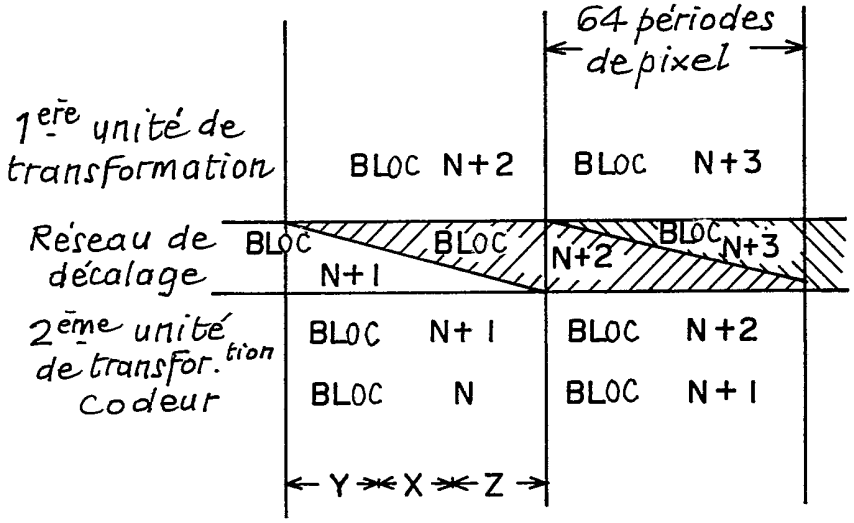


FIG. 2C

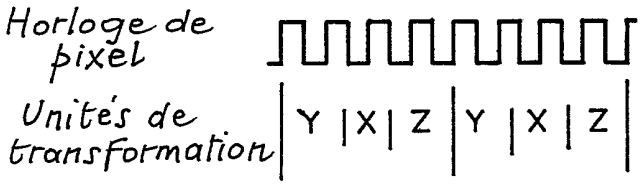


FIG. 3B

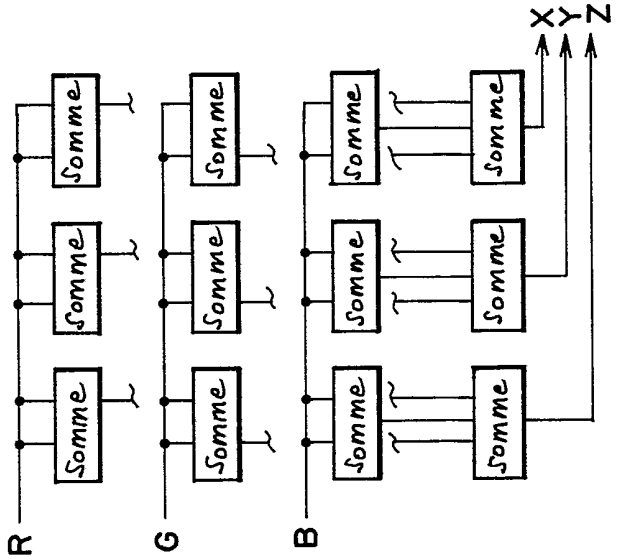

Somme
R
G
B
X
Y
Z

FIG. 3A

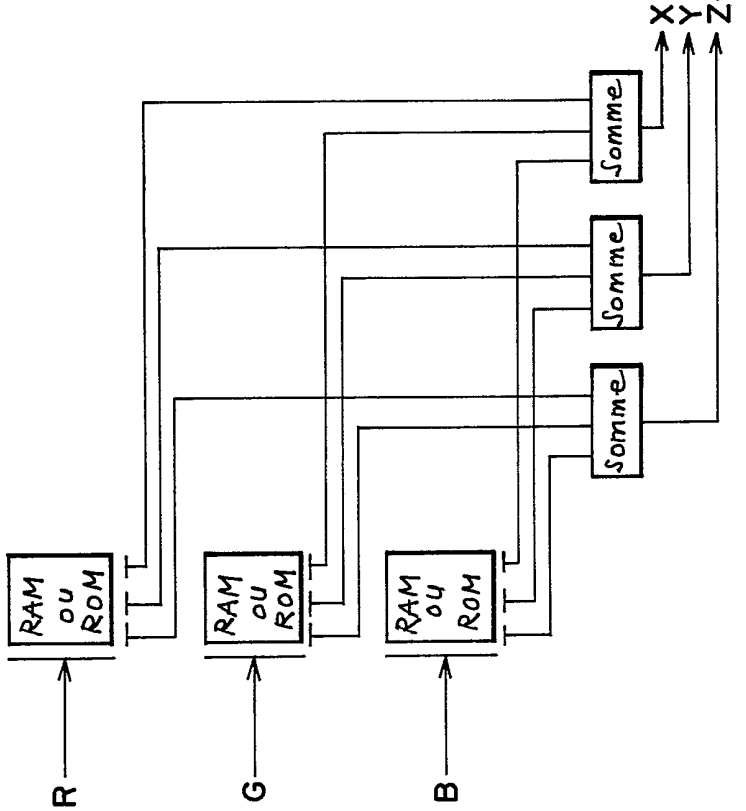

RAM ou ROM
Somme
R
G
B
X
Y
Z
vers le registre à décalage

FIG. 4A

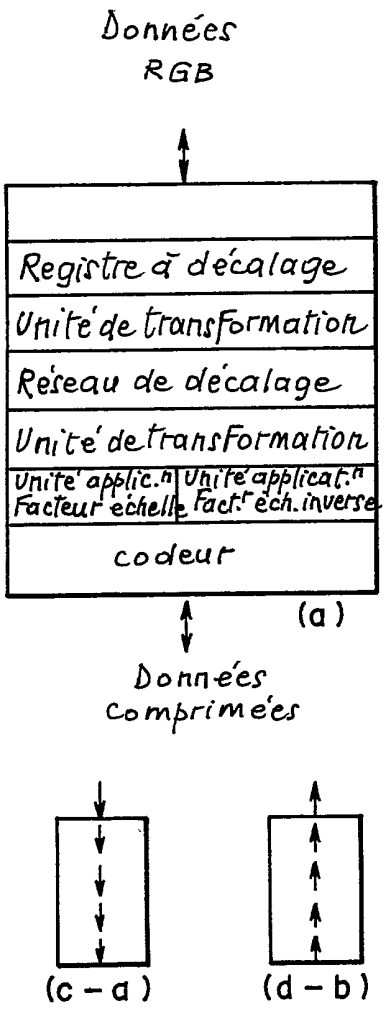

Données
RGB
Registre à décalage
Unité de transformation
Réseau de décalage
Unité de transformation
Unité applic.n Facteur échelle
Unité applicat.n Fact.r éch. inverse
codeur
(a)
Données
comprimées
(c - a)
(d - b)

FIG. 4B

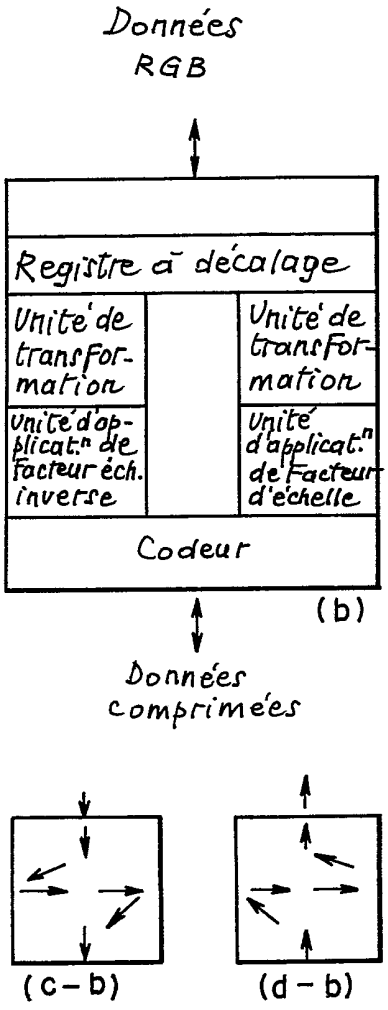

Données
RGB
Registre à décalage
Unité de transformation
Unité de transformation
Unité d'applicat.n de facteur éch. inverse
Unité d'applicat.n de Facteur d'échelle
Codeur
(b)
Données
comprimées
(c - b)
(d - b)

FIG. 5

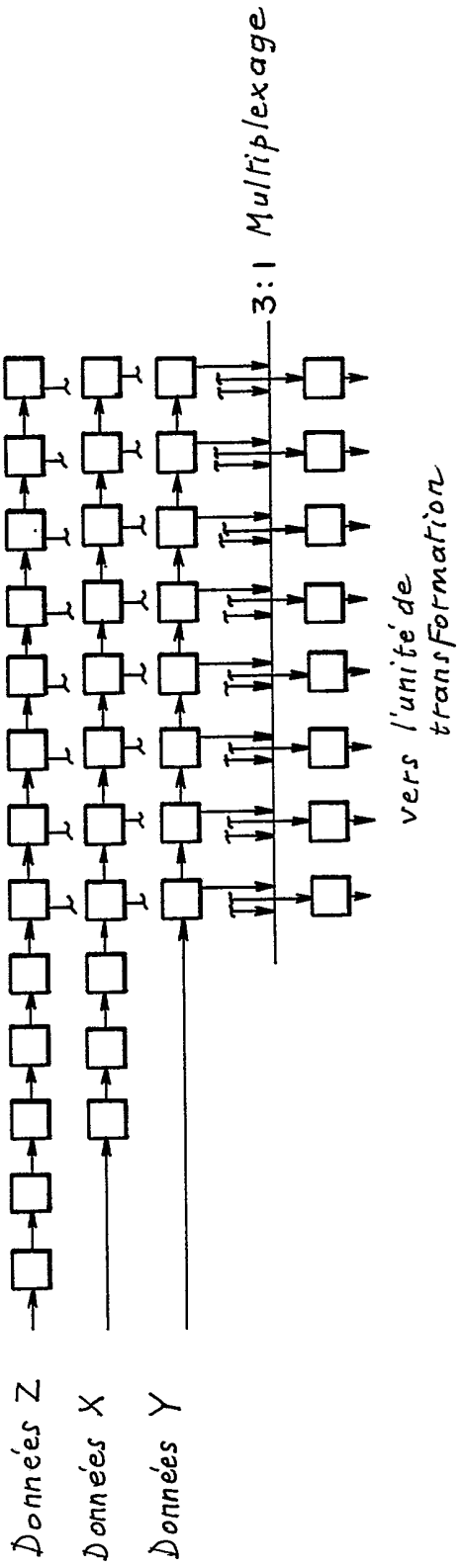

Données Z
Données X
Données Y
3:1 Multiplexage
vers l'unité de transformation

FIG. 6A

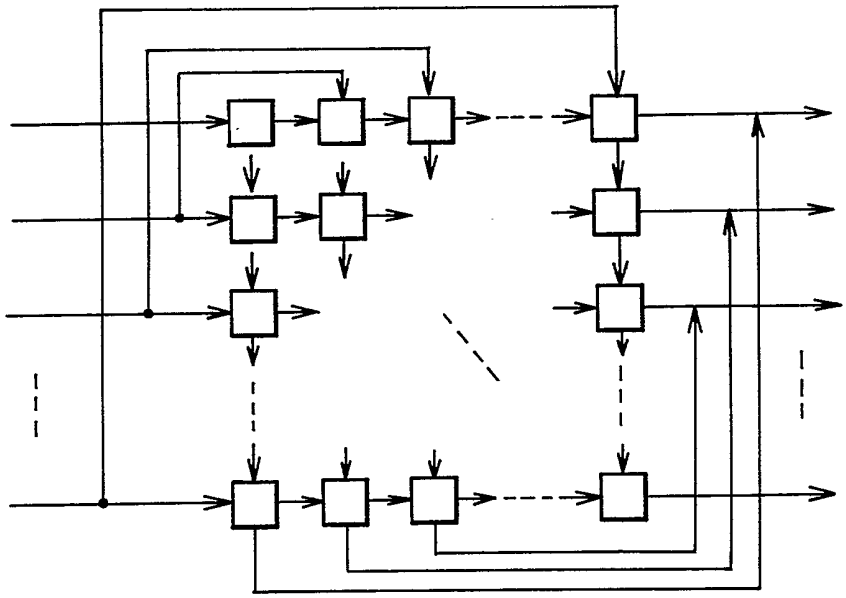


FIG. 6B

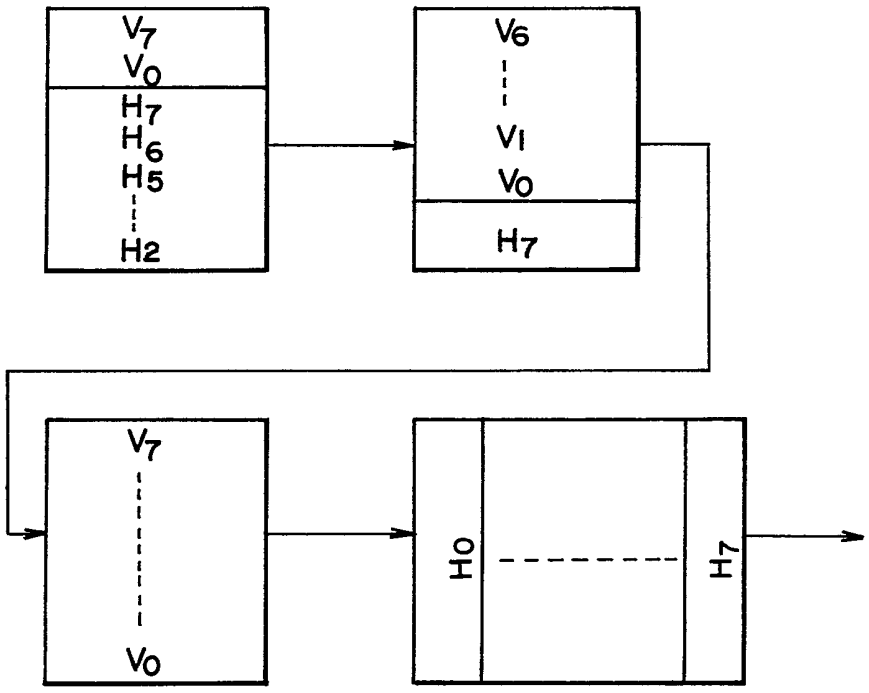


FIG. 7

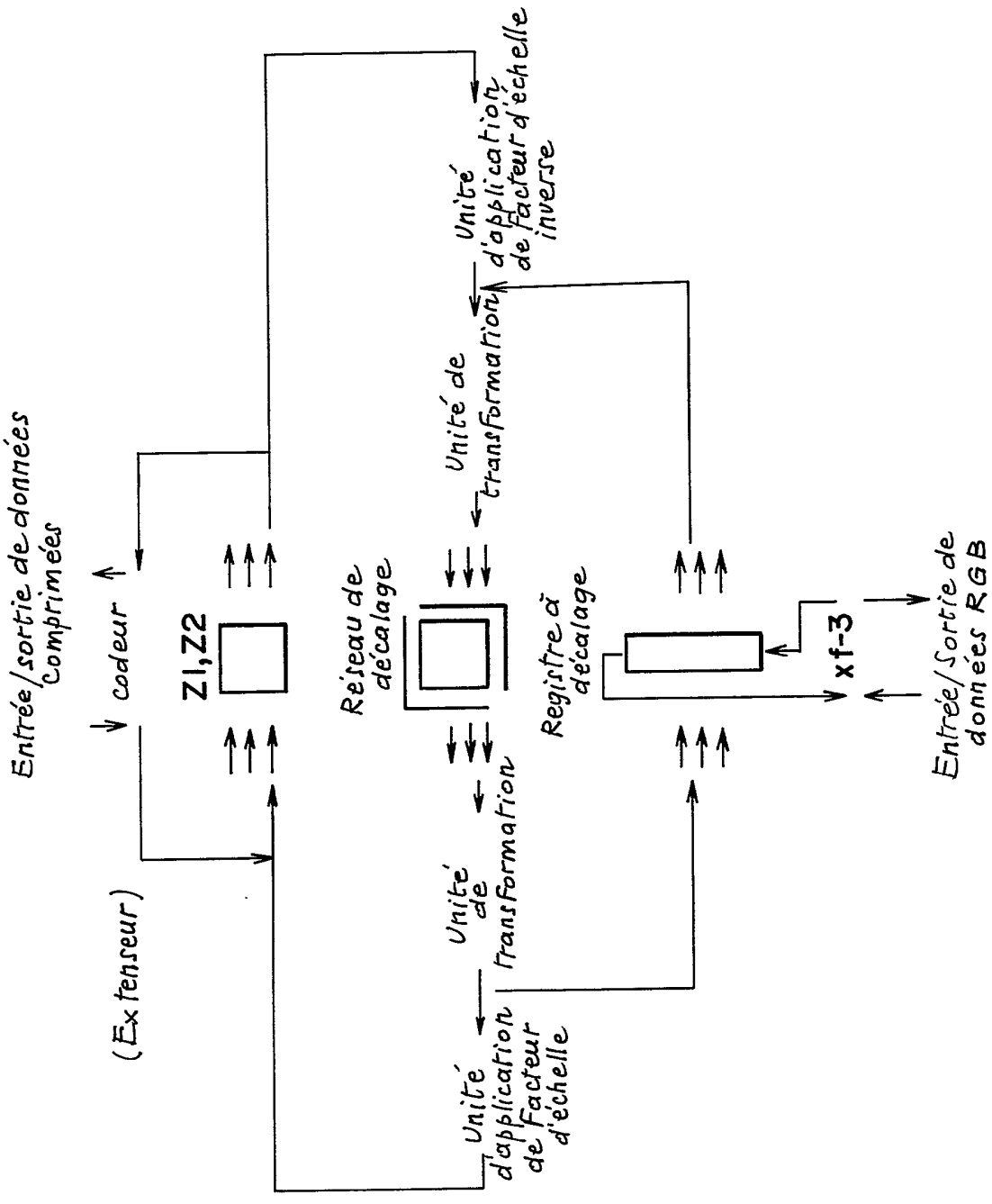

Entrée/sortie de données Comprimées
codeur
(Extenseur)
Z1,Z2
Réseau de décalage
Unité de transformation
Unité d'application de facteur d'échelle inverse
Unité de transformation
Unité d'application de Facteur d'échelle
Registre à décalage
xf-3
Entrée/Sortie de données RGB

FIG. 8A

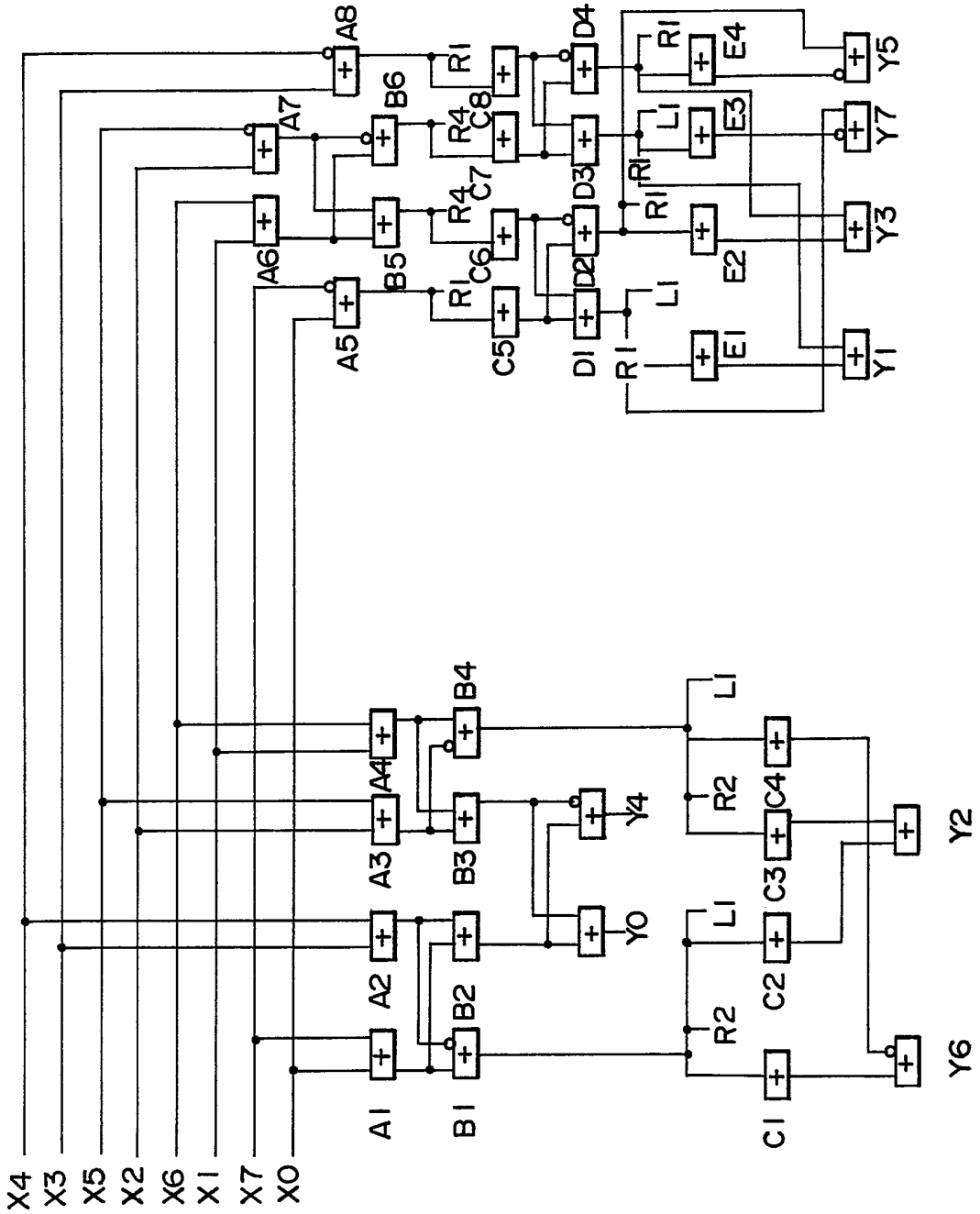

X4
X3
X5
X2
X6
X1
X7
X0
A1
A2
A3
A4
A5
A6
A7
A8
B1
B2
B3
B4
B5
B6
C1
C2
C3
C4
C5
C6
C7
C8
D1
D2
D3
D4
E1
E2
E3
E4
R1
R2
R4
L1
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7

FIG. 8B

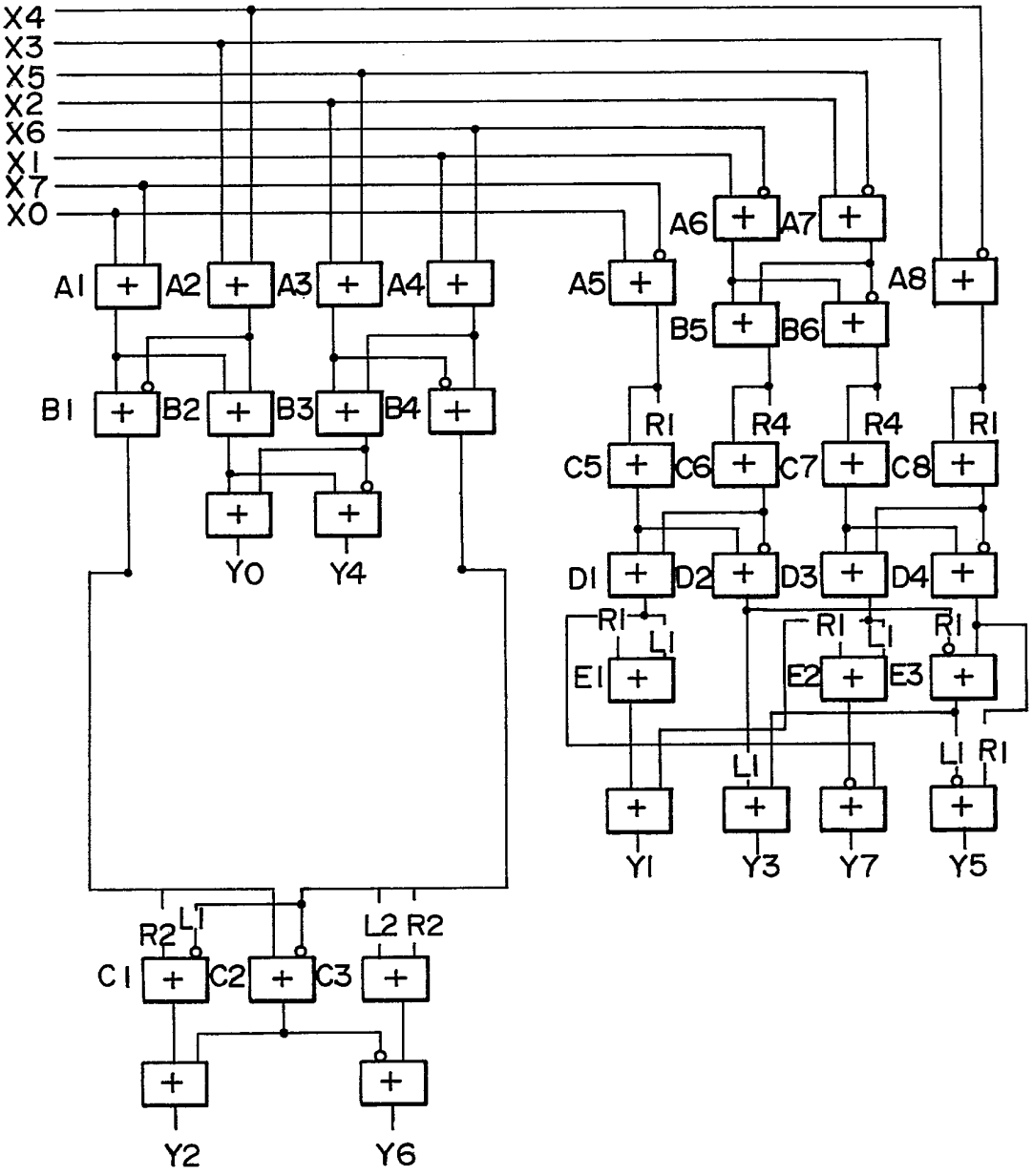

X4
X3
X5
X2
X6
X1
X7
X0
A1
A2
A3
A4
A5
A6
A7
A8
B1
B2
B3
B4
B5
B6
C1
C2
C3
C5
C6
C7
C8
D1
D2
D3
D4
E1
E2
E3
R1
L1
R2
L2
Y0
Y4
Y1
Y3
Y7
Y5
Y2
Y6

FIG. 9

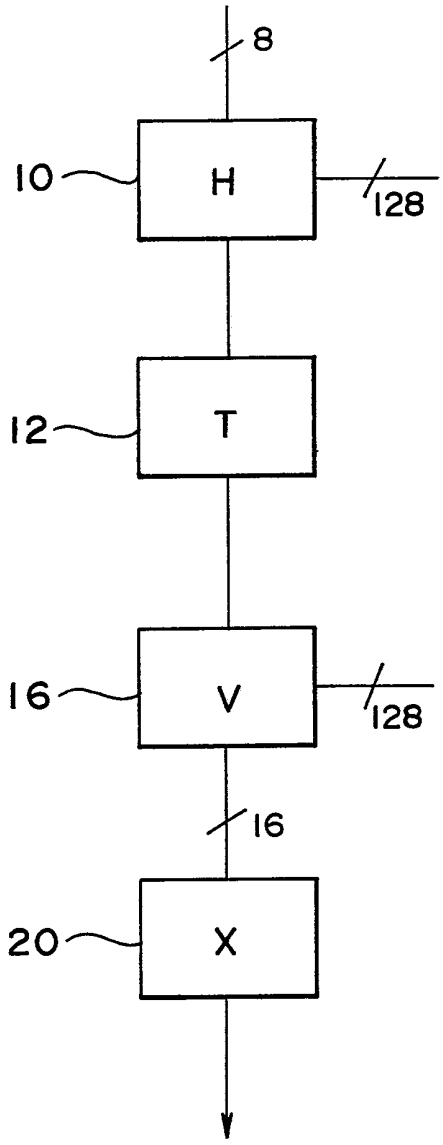
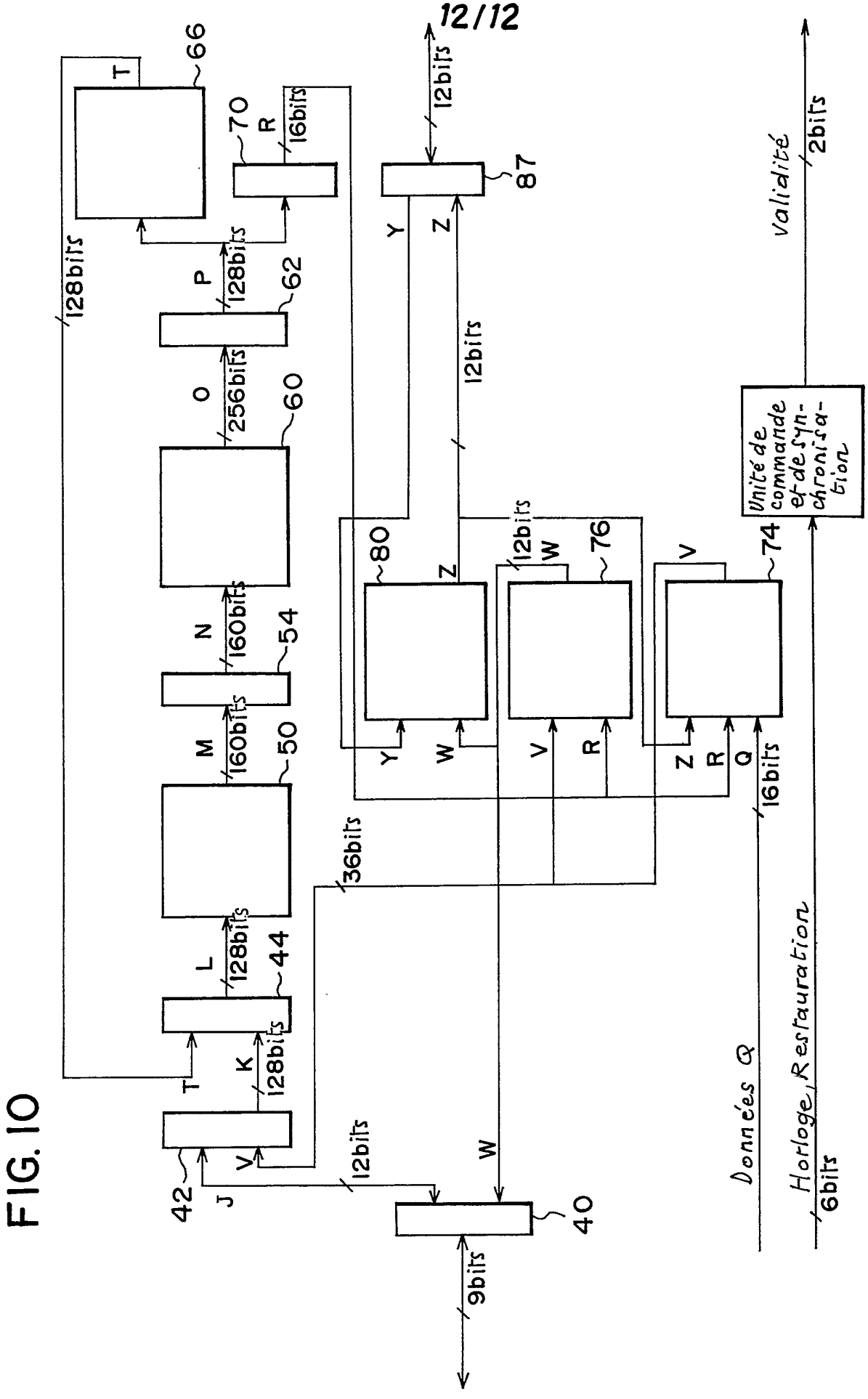


FIG. 10


66
T
128bits
70
R
16bits
P
128bits
62
O
256bits
60
N
160bits
54
M
160bits
50
L
128bits
44
K
128bits
42
J
V
T
12bits
36bits
W
12bits
40
9bits
87
Y
Z
12bits
80
Z
Y
W
12bits
W
76
V
R
V
74
Z
R
Q
16bits
Validité
2bits
Unité de commande et de synchronisation
Données Q
Horloge, Restauration
6bits

REPUBLIQUE FRANÇAISE

2680259

N° d'enregistrement national

INSTITUT NATIONAL de la PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

FR 9112450
FA 462838

DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendications concernées de la demande examinée
Y	IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, vol. 36, no. 4, Avril 1989, NEW YORK US pages 610 - 617; MING-TING SUN ET AL.: 'vlsi implementation of a 16x16 discrete cosine transform' * le document en entier * ---	1-4
Y	EP-A-0 412 252 (TELETTRA TELEFONIA ELETTRONICA E RADIO S.P.A) * abrégé * * colonne 5, ligne 20 - ligne 37 * * colonne 12, ligne 50 - colonne 14 * -----	1-4

DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5)

G06F

Date d'achèvement de la recherche	Examinateur
25 JUIN 1992	CHATEAU J.P.

1

EPO FORM 1503 03.82 (P0413)

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant