

A1

**DEMANDE
DE BREVET D'INVENTION**

(21)

N° 83 01157

(54) Procédé pour composer les adresses d'une mémoire.

(51) Classification internationale (Int. Cl. 3). G 11 C 8/00; G 06 F 7/38; H 04 N 9/535.

(22) Date de dépôt..... 26 janvier 1983.

(33) (32) (31) Priorité revendiquée : JP, 27 janvier 1982, n° 57-11180.

(41) Date de la mise à la disposition du
public de la demande..... B.O.P.I. — « Listes » n° 30 du 29-7-1983.

(71) Déposant : DAINIPPON SCREEN SEIZO KABUSHIKI KAISHA, société de droit japonais. —
JP.

(72) Invention de : Mitsuhiro Yomada, Tsukasa Nishida et Toshifumi Inoue.

(73) Titulaire : *Idem* (71)

(74) Mandataire : Cabinet Pierre Loyer,
18, rue de Mogador, 75009 Paris.

PROCEDE POUR COMPOSER LES ADRESSES D'UNE MEMOIRE

La présente invention a trait à un procédé pour composer les adresses d'une mémoire dans laquelle plusieurs données correspondant à plusieurs adresses sont accessibles simultanément.

5 Habituellement, lorsqu'une donnée est introduite ou extraite sur une adresse d'une mémoire, il n'est pas possible d'accéder aux autres adresses pour une opération d'écriture ou de lecture. S'il devient possible d'accéder simultanément à plusieurs adresses, on peut en attendre de nombreux avantages, tels qu'un accroissement de la vitesse de traitement résultant de la réduction du temps
10 d'accès, et ainsi de suite.

On a déjà proposé une telle méthode d'accès simultané à plusieurs adresses. Dans ce cas, lorsque des adresses (x) signifiant 0, 1, 2, 3, ..., sont à une dimension, une mémoire 1 ayant des adresses composées de manière continue, comme représenté Fig. 1, est divisée en plusieurs blocs, et les adresses ne sont plus
15 mémorisées séquentiellement dans les divers blocs (ici, au nombre de deux) numérotés 2 et 3 dans la Fig. 2a, mais sont mémorisées selon une opération entrelacée dans les blocs 4 et 5 par un procédé dit d'entrelacement de mémoire, comme le représente la Fig. 2b. Dans cette forme de mise en oeuvre du procédé d'entrelacement de mémoire, les adresses adjacentes telles que (2) et (3), ou
20 (15) et (16) dans un espace logique d'adresse d'une mémoire sont accessibles simultanément.

Toutefois, dans une mémoire classique telle qu'une mémoire-table ayant des adresses à deux dimensions (x,y), où x signifie 0, 1, 2, 3, ... et y signifie 0, 1, 2, 3, ..., il n'est pas possible d'accéder simultanément à plusieurs
25 adresses. Il n'a été proposé aucune méthode telle que la méthode d'entrelacement de mémoire décrite ci-dessus en ce qui concerne les adresses unidimensionnelles, de telle sorte que dans une mémoire ayant des adresses à au moins deux dimensions, il faut accéder à ces adresses une par une à chaque fois.

La présente invention a notamment pour but de définir un procédé pour
30 composer les adresses d'une mémoire ayant des adresses à n dimensions dans un espace logique d'adresses constitué de plusieurs sections unitaires ayant chacune des intersections avec les adresses à n dimensions, ce procédé étant exempt des inconvénients évoqués ci-dessus, et capable de réaliser un traitement rapide.

35 Le procédé selon l'invention est caractérisé par le fait de diviser une mémoire, ayant des adresses à n dimensions dans un espace logique d'adresses constitué de plusieurs sections unitaires ayant chacune des intersections avec

les adresses à n dimensions, en un nombre 2^n de blocs de mémoire ayant chacun des lignes indépendantes de données et d'adresses dans lesquelles l'adresse de chaque intersection de la section unitaire correspond à une adresse de chaque bloc de mémoire, de manière à accéder simultanément aux adresses des inter-

5 sections de la section unitaire.

Une forme de mise en oeuvre du procédé selon l'invention est caractérisée en ce que les blocs de mémoire sont repérés consécutivement par des numéros tels que 0, 1, 2, 3 ... et 2^n-1 , et en ce que chacune des adresses à n dimensions telles que $A^1, A^2, \dots A^{n-1}$ et A^n correspond à un bloc de mémoire dont le numéro

10 d'ordre est égal à un reste obtenu en divisant par 2^n le terme $\sum_{k=1}^n 2^{k-1} \cdot A_k$

Une autre forme de mise en oeuvre du procédé selon l'invention est caractérisée en ce que l'adresse de chaque bloc de mémoire est déterminée en divisant par 2 les adresses de la mémoire et en omettant leurs fractions.

Une autre forme de mise en oeuvre du procédé selon l'invention s'applique à

15 la réalisation d'une table d'index multidimensionnelle utilisable pour la conversion de signaux d'analyse d'une image en couleurs.

Selon l'invention, cette forme de mise en oeuvre est caractérisée par le fait de stocker dans une mémoire-table des ensembles de signaux secondaires des couleurs R, G et B pour le rouge, le vert et le bleu, ces signaux correspondant

20 à des ensembles de signaux primaires des couleurs C, M, Y et K pour des encres de couleurs cyan, magenta, jaune et noir, les signaux secondaires des couleurs R, G et B étant lus dans la mémoire-table par adressage sur une adresse (C, M, Y, K), cette mémoire-table étant divisée en 2^4 blocs de mémoire ayant chacun des lignes indépendantes de données et d'adresses repérées consécutivement par des

25 numéros d'ordre 0, 1, 2, 3 ... et 15, tandis que chaque adresse (C, M, Y, K) correspond au bloc de mémoire dont le numéro d'ordre est déterminé par une division $(C+2M+4Y+8K)/2^4$.

Des formes préférées de mise en oeuvre de la présente invention seront décrites ci-après de manière plus détaillée, en se référant aux dessins annexés,

30 dans lesquels :

la Fig. 1 représente des adresses d'une mémoire, composées d'une manière classique;

la Fig. 2 représente des adresses stockées dans deux blocs composant une mémoire d'un type classique, la Fig. 2(b) représentant des adresses stockées

35 selon une méthode classique d'entrelacement de mémoire;

la Fig. 3 représente quatre blocs de mémoire dans lesquels des adresses à deux

dimensions sont composées conformément à la présente invention;
 la Fig. 4 représente une mémoire d'index qui exprime les numéros des blocs de
 mémoire et des adresses à deux dimensions composées selon l'invention;
 la Fig. 5 représente une mémoire d'index qui exprime les numéros des blocs de
 5 mémoire et des adresses tridimensionnelles composées selon l'invention;
 la Fig. 6 est un schéma expliquant une interpolation selon l'invention, et
 la Fig. 7 est un schéma de principe d'un circuit arithmétique pour exécuter
 l'interpolation selon l'invention.

En se référant aux Fig. 3-7, on voit un espace logique d'adressage d'une
 10 mémoire ayant des adresses à n dimensions. Cet espace est composé de plusieurs
 sections unitaires qui ont chacune des intersections avec les adresses à n
 dimensions. Dans un système à deux dimensions, les adresses des intersections
 d'une section unitaire ayant une forme quadratique sont exprimées sous la forme
 d'un ensemble de (1,1), (2,1), (1,2) et (2,2), et ainsi de suite, et dans un
 15 système à trois dimensions, les adresses des intersections d'une section unitaire
 ayant une forme cubique sont exprimées sous la forme d'un ensemble de (1,1,1),
 (2,1,1), (1,2,1), (1,1,2), (2,2,1), (2,1,2), (1,2,2) et (2,2,2), et ainsi de suite.

Tout d'abord, pour faciliter la compréhension de l'invention, celle-ci sera
 expliquée en termes d'adresses à deux dimensions (x,y), où x signifie 0, 1, 2, 3,
 20 ..., et y signifie 0, 1, 2, 3,

Dans cette forme de mise en oeuvre, telle qu'elle est représentée à la Fig.3,
 la mémoire est divisée en $2^2 = 4$ blocs 6, 7, 8 et 9, ci-après référencés t0, t1,
 t2 et t3, respectivement, chaque bloc ayant des lignes d'adresses et des lignes
 de données indépendantes, et chaque donnée d'adresse des quatre intersections de
 25 la section unitaire est stockée dans chaque bloc t0, t1, t2 ou t3.

C'est à dire que lorsque y est égal à zéro, les données d'adresse (x,y) sont
 stockées dans les blocs de mémoire, par exemple :

(0,0) dans le bloc t0, (1,0) dans le bloc t1
 (2,0) dans le bloc t2, (3,0) dans le bloc t3
 30 (4,0) dans le bloc t0, (5,0) dans le bloc t1 et ainsi de suite. En
 général, la donnée d'adresse (x,y) est stockée dans un bloc de mémoire tp_2 où
 p_2 signifie un reste qui est obtenu par une division $(x+2y)/2^2$.

Cette relation est représentée visuellement dans la Fig. 4 qui montre les
 numéros des blocs de mémoire et les adresses à deux dimensions. C'est à dire
 35 que les intersections des deux lignes qui s'étendent dans les direction x et y
 signifient les adresses bidimensionnelles (0,0), (0,1), ..., et que les numéros
 "0", "1", "2" et "3" apparaissant sur les intersections signifient les numéros

d'ordre affectés aux blocs de mémoire, c'est à dire que "0", "1", "2" et "3" signifient respectivement les blocs de mémoire t0, t1, t2 et t3.

Par suite, le numéro d'ordre p_2 du bloc-mémoire tp_2 dans lequel il y a lieu de stocker une adresse (x,y) s'obtient au moyen de la formule (I) ci-après, dans laquelle $\text{MODE}_2 ()$ signifie un reste obtenu en divisant par 2^2 un nombre de ().

$$p_2 = \text{MODE}_2(x+2y) \dots\dots\dots (I)$$

En outre, ainsi que cela a été décrit ci-dessus, les adresses des intersections de la section unitaire sont stockées séparément dans les différents blocs de mémoire, et par conséquent les adresses stockées dans chacun des blocs sont dispersées. Toutefois, pour répartir convenablement les adresses dans les différents blocs, la relation entre l'adresse (x,y) de la mémoire et l'adresse (x',y') du bloc est déterminée par la formule (II) ci-après, dans laquelle les nombres $\frac{x}{2}$ et $\frac{y}{2}$ dans () sont des nombres entiers obtenus en omettant leurs fractions.

$$\left. \begin{array}{l} x' = (\frac{x}{2}) \\ y' = (\frac{y}{2}) \end{array} \right\} \dots\dots\dots (II)$$

Autrement dit, les adresses des intersections de chaque section unitaire sont réparties dans les différents blocs de mémoire ayant des lignes d'adresses indépendantes et des lignes de données indépendantes.

L'invention va maintenant être exposée en se référant à un système tridimensionnel dans lequel une section unitaire de forme cubique possède huit intersections ayant des adresses tridimensionnelles (x,y,z) dans lesquelles x signifie 0, 1, 2, 3, ..., y signifie 0, 1, 2, 3, ... et z signifie 0, 1, 2, 3, ...

Dans cette forme de mise en oeuvre, la mémoire est divisée en $2^3 = 8$ blocs t0, t1, t2, t3, t4, t5, t6 et t7, ayant chacun des lignes d'adresses et des lignes de données indépendantes, et les adresses des intersections de chaque section unitaire sont composées de manière à être stockées dans les différents blocs de mémoire.

Par exemple, lorsque y est égal à zéro et que z est égal à zéro, les données d'adresses (x,y,z) sont stockées dans les blocs de mémoire, de la manière suivante :

(0,0,0) dans le bloc t0,	(1,0,0) dans le bloc t1
(2,0,0) dans le bloc t2,	(3,0,0) dans le bloc t3
(4,0,0) dans le bloc t4,	(5,0,0) dans le bloc t5
(6,0,0) dans le bloc t6,	(7,0,0) dans le bloc t7

(8,0,0) dans le bloc t0 et ainsi de suite.

D'une manière générale, une donnée d'adresse (x,y,z) est stockée dans un bloc de mémoire tp_3 , où un numéro d'ordre p_3 signifie un reste obtenu par une division $(x+2y+4z)/2^3$. Cette relation est représentée dans la Fig. 5 qui montre les 5 numéros des blocs de mémoire et les adresses tridimensionnelles.

Le numéro d'ordre p_3 du bloc tp_3 dans lequel il y a lieu de stocker l'adresse (x,y,z) est obtenu au moyen de la formule (III) ci-après dans laquelle $MODE_3$ () signifie un reste obtenu en divisant par 2^3 un nombre contenu dans ().

$$P_3 = MODE_3(x+2y+4z) \dots\dots\dots (III)$$

10 En outre, la relation entre l'adresse (x,y,z) de la mémoire et l'adresse (x',y',z') de chaque bloc de mémoire est déterminée au moyen de la formule (IV) ci-après, dans laquelle les nombres $\frac{x}{2}$, $\frac{y}{2}$ et $\frac{z}{2}$ dans () sont des nombres entiers obtenus en omettant leurs fractions.

$$15 \quad \left. \begin{array}{l} x' = (\frac{x}{2}) \\ y' = (\frac{y}{2}) \\ z' = (\frac{z}{2}) \end{array} \right\} \dots\dots\dots (IV)$$

Comme décrit ci-dessus, les adresses des intersections de chaque section unitaire sont réparties dans les divers blocs ayant chacun des lignes d'adresses et des lignes de données indépendantes.

20 L'invention sera maintenant expliquée en se référant à un système à n dimensions, dans lequel une section unitaire possède n intersections avec des adresses à n dimensions ($A_1, A_2, A_3, \dots$ et A_n) ($n = 1, 2, 3, \dots$).

Dans cette forme de mise en oeuvre, la mémoire est divisée en 2^n blocs de mémoire t0, t1, t2, t3, et $t(2^n-1)$, ayant chacun des lignes d'adresses et 25 des lignes de données indépendantes. D'une manière générale, une donnée d'adresse ($A_1, A_2, A_3, \dots, A_n$) est stockée dans un bloc tp_n , où un numéro d'ordre p_n est obtenu au moyen de la formule (V) ci-après qui est établie à partir des formules (I) et (III) précitées, et dans laquelle $MODE_n$ () signifie un reste obtenu en divisant par 2^n un nombre contenu dans ().

$$30 \quad \begin{aligned} p_n &= MODE_n(A_1+2A_2+4A_3+8A_4+ \dots + 2^{n-1}A_n) \\ &= MODE_n(\sum_{k=1}^n 2^{k-1}A_k) \dots\dots\dots (V) \end{aligned}$$

En outre, la relation entre l'adresse ($A_1, A_2, A_3, A_4, \dots, A_n$) de la mémoire et l'adresse ($A_1', A_2', A_3', A_4', \dots, A_n'$) de chaque bloc est déterminée au moyen de la formule (VI) ci-après de la même manière que pour les formules (I) et (III) des formes de mise en oeuvre décrites précédemment.

$$\begin{array}{lcl} 5 & A_1' = \left(\frac{A_1}{2} \right) & \\ & A_2' = \left(\frac{A_2}{2} \right) & \dots\dots\dots (VI) \\ & \vdots & \\ 10 & A_n' = \left(\frac{A_n}{2} \right) & \end{array}$$

Par conséquent, selon la présente invention, les adresses à n dimensions des intersections de chaque section unitaire sont réparties dans les différents blocs de mémoire ayant chacun des lignes d'adresses et des lignes de données indépendantes dans le système à n dimensions.

15 On comprendra aisément d'après la description ci-dessus que les adresses des intersections de chaque section unitaire sont accessibles simultanément, à la différence des procédés classiques avec lesquels on ne peut accéder aux adresses que séquentiellement, même quand ces adresses sont multi-dimensionnelles et quand
20 il existe un grand nombre d'adresses d'intersections de chaque section unitaire. Par conséquent, le temps d'accès pourra être largement diminué grâce au procédé selon la présente invention.

Dans les formes de mise en oeuvre de l'invention qui viennent d'être décrites, l'accès simultané vers plusieurs blocs de mémoire ne peut pas toujours être
25 exécuté sur les mêmes adresses, en commun, et l'accès aux blocs de mémoire devrait se faire sur leurs adresses respectives.

Un exemple de la présente invention sera décrit ci-après, à titre non limitatif.

Lorsque des signaux primaires sont convertis, très fréquemment, en signaux
30 secondaires par un calcul approprié, il faut beaucoup de temps si le calcul est exécuté chaque fois sur un seul signal. Il est donc préférable d'obtenir les signaux secondaires à l'avance, en traitant les signaux primaires par un calcul de simulation, puis de les stocker dans une mémoire d'index sous la forme de données pouvant être lues à n'importe quel moment dans cette mémoire d'index en
35 utilisant les signaux primaires comme signaux d'adressage, ce qui permet de réaliser la conversion des signaux à une grande vitesse.

Lorsque des signaux primaires (x, y) à deux dimensions sont convertis en signaux secondaires $F(x, y)$ à deux dimensions, et que ces derniers sont ensuite stockés dans une mémoire d'index comme représenté Fig. 6, les signaux secondaires $F(x_i + x_f, y_i + y_f)$, où x_i et y_i signifient des parties entières et x_f et y_f des parties décimales correspondant aux signaux primaires $(x_i + x_f, y_i + y_f)$ en un point G, sont obtenus à partir des valeurs $F(x_i, y_i)$, $F(x_i + 1, y_i)$, $F(x_i, y_i + 1)$ et $F(x_i + 1, y_i + 1)$, ces valeurs étant désignées ci-après comme données d'interpolation des signaux secondaires des intersections d'une surface d'interpolation ABCD, selon la formule (VII) ci-après pour une méthode d'interpolation linéaire :

$$\begin{aligned}
 10 \quad F(x_i + x_f, y_i + y_f) = & F(x_i, y_i) (1 - x_f) (1 - y_f) \\
 & + F(x_i + 1, y_i) x_f (1 - y_f) \\
 & + F(x_i, y_i + 1) (1 - x_f) y_f \\
 & + F(x_i + 1, y_i + 1) x_f y_f \dots\dots\dots (VII)
 \end{aligned}$$

15 Pour exécuter une telle interpolation selon une méthode classique, les données d'interpolation sont stockées dans une mémoire d'index ayant une composition d'adresses classique, et il faut alors chaque fois lire ces données une à une sur la table d'index.

20 Par contre, il devient possible, grâce au procédé selon l'invention, dans lequel un ensemble de données d'interpolation correspondant aux adresses des intersections de la section unitaire sont stockées dans les différents blocs de mémoire, de lire simultanément et d'une manière commode l'ensemble des données d'interpolation dans une mémoire dont les adresses sont composées de la manière décrite ci-dessus conformément à la présente invention.

25 Ceci signifie qu'il devient possible, selon l'invention, de lire tout un ensemble de données d'interpolation pendant le temps qu'il faudrait autrement pour lire une seule donnée d'interpolation selon une méthode classique. Par conséquent, dans un système à deux dimensions selon la présente invention, le temps de lecture pourra être réduit à un quart de celui qu'exige la méthode
30 classique.

De plus, le procédé selon l'invention pourra être appliqué d'une manière beaucoup plus performante à une mémoire d'index multi-dimensionnelle pour utiliser cette mémoire lors de la conversion de signaux d'images en couleurs, de la manière décrite en détail ci-après.

35 Lorsque des signaux primaires numériques de séparation de couleurs C, M, Y et K appartenant au premier système de signaux, et correspondant aux couleurs

- cyan, magenta, jaune et noir des encres d'impression, sont convertis en signaux secondaires numériques de séparation de couleurs R, G et B appartenant au deuxième système de signaux, pour les couleurs rouge, verte et bleue, en vue de la reproduction d'une image en couleurs, en même temps qu'on simule, au moyen de
- 5 l'écran cathodique de contrôle d'un dispositif à balayage pour l'analyse des couleurs, l'aspect final de l'impression d'après les signaux primaires de l'image, ceci pouvant se faire aussi au moyen d'un dispositif électronique de vérification des tirages en couleur, les valeurs $U(C_i, M_i, Y_i, K_i)$ des signaux secondaires d'image R, G et B obtenues à l'avance en exécutant un calcul arithmétique de
- 10 simulation et une conversion de signal des quatre bits supérieurs C_i, M_i, Y_i et K_i correspondant aux parties entières des signaux d'image primaires C, M, Y et K exprimés en 256 gradations (8 bits), sont stockées à l'avance dans une mémoire d'index dont les adresses sont composées conformément au procédé selon l'invention, comme décrit ci-dessus. Par conséquent, l'interpolation pourra ensuite
- 15 être exécutée à grande vitesse de la manière décrite ci-après :

- Selon cette forme de mise en oeuvre, étant donné que les adresses (C, M, Y, K) de la mémoire d'index sont à quatre dimensions, la mémoire est divisée en $2^4 = 16$ blocs de mémoire $t_0, t_1, t_2, t_3 \dots$ et t_{15} . Un numéro d'ordre p_4 du bloc de mémoire t_{p_4} s'obtient au moyen de la formule (IIX) ci-après qui est
- 20 elle-même obtenue par substitution de $n=4, A_1=C_i, A_2=M_i, A_3=Y_i$ et $A_4=K_i$ dans la formule (V) décrite ci-dessus, et la relation entre l'adresse (C_i, M_i, Y_i, K_i) de la mémoire et l'adresse (C_i', M_i', Y_i', K_i') du bloc de mémoire est déterminée par la formule suivante (IX) basée sur la formule (VI) :

$$p_4 = \text{MODE}_4(C_i + 2M_i + 4Y_i + 8K_i) \dots\dots\dots \text{(IIX)}$$

25

$$\left. \begin{aligned} C_i' &= \left(\frac{C_i}{2} \right), & M_i' &= \left(\frac{M_i}{2} \right) \\ Y_i' &= \left(\frac{Y_i}{2} \right), & K_i' &= \left(\frac{K_i}{2} \right) \end{aligned} \right\} \dots\dots\dots \text{IX)}$$

- 30 Donc, dans cette forme de mise en oeuvre où les adresses sont composées selon le procédé de l'invention, lorsque les signaux primaires d'image $C = C_i + C_f, M = M_i + M_f, Y = Y_i + Y_f$ et $K = K_i + K_f$, où C_f, M_f, Y_f et K_f correspondent aux quatre bits inférieurs, $0 \leq C_f, M_f, Y_f, K_f < 1$, sont obtenus sous la forme de la division unitaire de un par l'interpolation, un ensemble de 16 valeurs
- 35 $U(C_i, M_i, Y_i, K_i), U(C_i+1, M_i, Y_i, K_i), \dots$ et $U(C_i+1, M_i+1, Y_i+1, K_i+1)$ c'est à dire 16 données d'interpolation correspondant à 16 adresses $(C_i, M_i, Y_i, K_i), (C_i+1, M_i, Y_i, K_i) \dots$ et $(C_i+1, M_i+1, Y_i+1, K_i+1)$ des

intersections de la section unitaire incluant un point devant être interpolé, pourra être lu simultanément dans les différents blocs de mémoire. Ainsi, le temps d'accès pourra être considérablement réduit grâce à l'invention.

En utilisant les données d'interpolation lues dans la mémoire, on peut
 5 obtenir les signaux d'image secondaires correspondants R, G et B par la méthode d'interpolation linéaire exprimée par la formule (X) ci-après ou par une autre méthode simple d'interpolation décrite dans la demande de brevet japonais N° 52-57198 (demande ouverte à l'inspection sous le N° 53-123201).

10

$$\begin{aligned}
 & U(C_i + C_f, M_i + M_f, Y_i + Y_f, K_i + K_f) \\
 & = U(C_i, M_i, Y_i, K_i) (1 - C_f) (1 - M_f) (1 - Y_f) (1 - K_f) \\
 15 & + U(C_i + 1, M_i, Y_i, K_i) C_f (1 - M_f) (1 - Y_f) (1 - K_f) \\
 & + U(C_i, M_i + 1, Y_i, K_i) (1 - C_f) M_f (1 - Y_f) (1 - K_f) \\
 & + U(C_i, M_i, Y_i + 1, K_i) (1 - C_f) (1 - M_f) Y_f (1 - K_f) \\
 & + U(C_i, M_i, Y_i, K_i + 1) (1 - C_f) (1 - M_f) (1 - Y_f) K_f \\
 20 & + U(C_i + 1, M_i + 1, Y_i, K_i) C_f M_f (1 - Y_f) (1 - K_f) \\
 & + U(C_i + 1, M_i, Y_i + 1, K_i) C_f (1 - M_f) Y_f (1 - K_f) \\
 & + U(C_i + 1, M_i, Y_i, K_i + 1) C_f (1 - M_f) (1 - Y_f) K_f \\
 25 & + U(C_i, M_i + 1, Y_i + 1, K_f) (1 - C_f) M_f Y_f (1 - K_f) \\
 & + U(C_i, M_i + 1, Y_i, K_f + 1) (1 - C_f) M_f (1 - Y_f) K_f \\
 & + U(C_i, M_i, Y_i + 1, K_i + 1) (1 - C_f) (1 - M_f) Y_f K_f \\
 & + U(C_i + 1, M_i + 1, Y_i + 1, K_i) C_f M_f Y_f (1 - K_f) \\
 30 & + U(C_i, M_i + 1, Y_i + 1, K_i + 1) (1 - C_f) M_f Y_f K_f \\
 & + U(C_i + 1, M_i, Y_i + 1, K_i + 1) C_f (1 - M_f) Y_f K_f \\
 & + U(C_i + 1, M_i + 1, Y_i, K_i + 1) C_f M_f (1 - Y_f) K_f \\
 & + U(C_i + 1, M_i + 1, Y_i + 1, K_i + 1) C_f M_f Y_f K_f \dots\dots\dots (X)
 \end{aligned}$$

35

.../...

La Fig. 7 représente un circuit arithmétique pour exécuter l'interpolation exprimée par la formule (X), ce circuit comprenant 16 blocs de mémoire $10_1 - 10_{16}$ qui fournissent les données d'interpolation, des sélecteurs $11_1 - 11_{16}$, $12_1 - 12_{16}$, $13_1 - 13_{16}$ et $14_1 - 14_{16}$ qui choisissent C_f ou $1-C_f$, M_f ou $1-M_f$, Y_f ou $1-Y_f$, et K_f ou $1-K_f$, des soustracteurs $15_1 - 15_{16}$, $16_1 - 16_{16}$, $17_1 - 17_{16}$ et $18_1 - 18_{16}$ qui calculent respectivement $1-C_f$, $1-M_f$, $1-Y_f$ et $1-K_f$ des multiplicateurs $19_1 - 19_{16}$, $20_1 - 20_{16}$, $21_1 - 21_{16}$ et $22_1 - 22_{16}$ pour multiplier les valeurs de sortie des sélecteurs $11 - 14$ et des blocs de mémoire 10 , et enfin des additionneurs $23_1 - 23_{15}$ qui totalisent les produits des multiplications, obtenant ainsi la valeur interpolée $U(C_1+C_f, M_1+M_f, Y_1+Y_f, K_1+K_f)$.

Dans cette forme de mise en oeuvre, les données d'interpolation peuvent être lues simultanément dans les blocs de mémoire, et le calcul de l'interpolation peut être exécuté en même temps. Par suite, le temps de traitement de l'interpolation pourra être considérablement diminué par rapport à la méthode classique.

Tandis que l'invention a été décrite et représentée avec suffisamment de détails pour qu'elle puisse être clairement comprise, il reste bien entendu que différentes modifications peuvent être apportées à la forme, aux détails et à la disposition des divers éléments sans sortir du cadre de la présente invention.

RE V E N D I C A T I O N S

1. Procédé pour composer les adresses d'une mémoire ayant des adresses à n dimensions dans un espace logique d'adresses constitué de plusieurs sections unitaires ayant chacune des intersections avec les adresses à n dimensions,
 5 caractérisé par le fait de diviser cette mémoire en un nombre 2^n de blocs de mémoire ayant chacun des lignes indépendantes de données et d'adresses, dans lesquelles l'adresse de chaque intersection de la section unitaire correspond à une adresse de chaque bloc de mémoire de manière à accéder simultanément
 10 aux adresses des intersections de la section unitaire.
2. Procédé selon la revendication 1, caractérisé en ce que les blocs de mémoire sont repérés consécutivement par des numéros d'ordre tels que 0, 1, 2, 3, et 2^n-1 , et en ce que chacune des adresses à n dimensions telles que A^1 , A^2 , ..., A^{n-1} et A^n correspond à un bloc de mémoire dont le numéro d'ordre
 15 est égal à un reste obtenu en divisant par 2^n le terme $\sum_{k=1}^n 2^{k-1} \cdot A_k$
3. Procédé selon la revendication 1, caractérisé en ce que l'adresse de chaque bloc de mémoire est déterminée en divisant par deux les adresses de la mémoire et en omettant leurs fractions.
4. Procédé selon la revendication 1, caractérisé par le fait de stocker dans une
 20 mémoire-table des ensembles de signaux secondaires des couleurs R, G et B pour le rouge, le vert et le bleu, ces signaux correspondant à des ensembles de signaux primaires des couleurs C, M, Y et K pour des encres de couleurs cyan, magenta, jaune et noir, les signaux secondaires des couleurs R, G et B étant lus dans la table par adressage sur une adresse (C, M, Y, K), cette
 25 table étant divisée en 2^4 blocs de mémoire ayant chacun des lignes indépendantes de données et d'adresses repérées consécutivement par des numéros d'ordre 0, 1, 2, 3 ... et 15, tandis que chaque adresse (C, M, Y, K) correspond au bloc de mémoire dont le numéro d'ordre est déterminé par une division $(C+2M+4Y+8K)/2^4$.

FIG. 1

(0)
(1)
(2)
(3)
(4)
⋮
(65534)
(65535)

FIG. 2

(a)

(0)	(32767)
(1)	(32768)
(2)	(32769)
⋮	⋮
(32766)	(65535)
2	3

(b)

(0)	(1)
(2)	(3)
(4)	(5)
⋮	⋮
(65534)	(65535)
4	5

FIG. 3

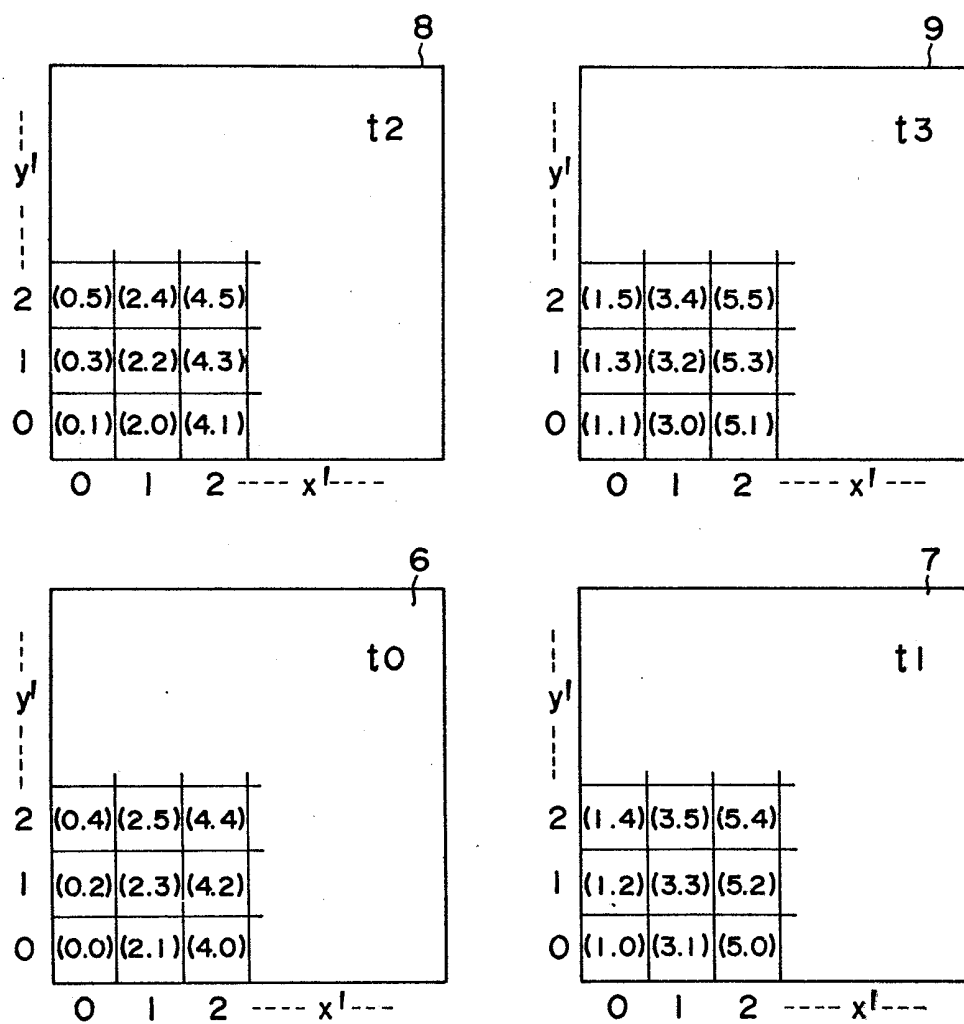


FIG. 4

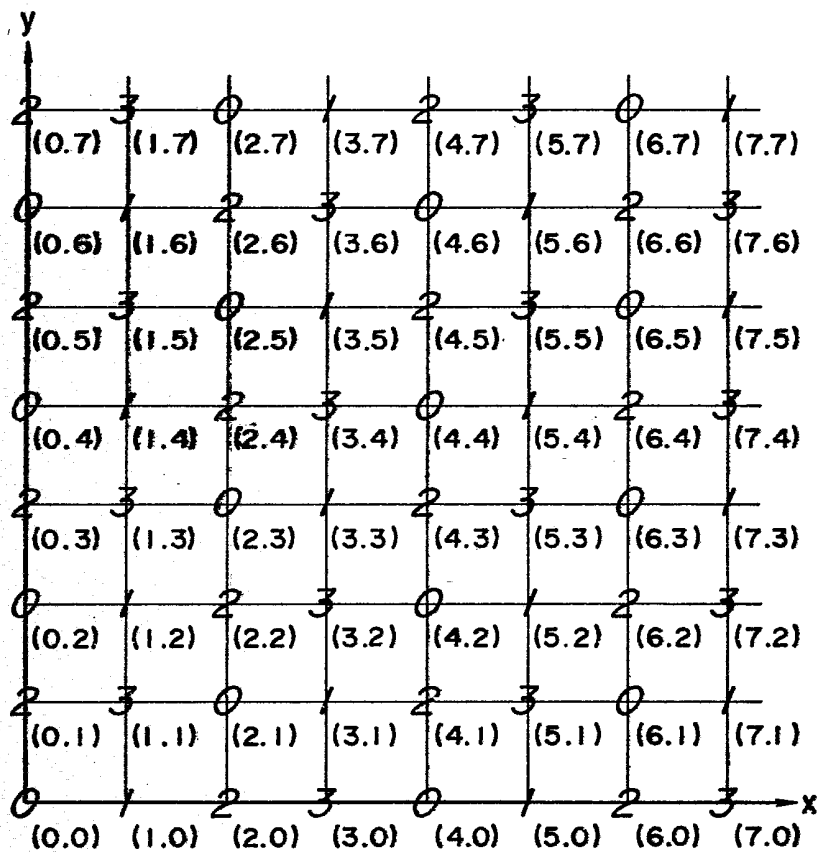


FIG. 6

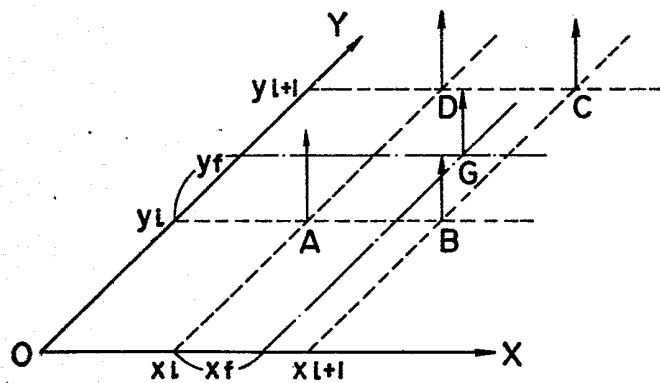


FIG. 5

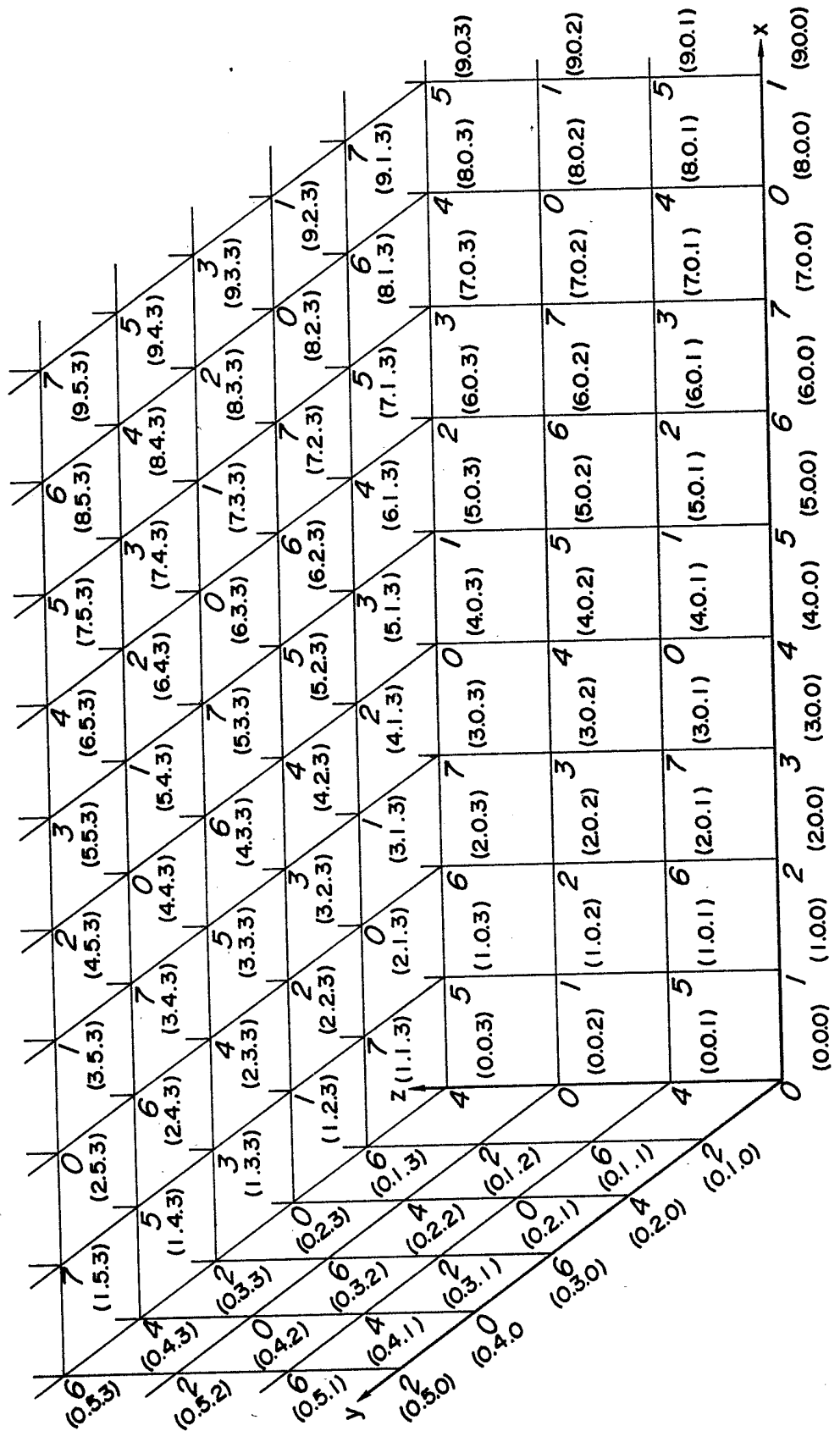


FIG. 7

