

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4366250号
(P4366250)

(45) 発行日 平成21年11月18日 (2009.11.18)

(24) 登録日 平成21年8月28日 (2009.8.28)

(51) Int. Cl.		F I	
HO 4 N	1/41	(2006.01)	HO 4 N 1/41 B
HO 3 M	7/30	(2006.01)	HO 3 M 7/30 A
HO 4 N	7/30	(2006.01)	HO 4 N 7/133 Z

請求項の数 6 (全 17 頁)

(21) 出願番号	特願2004-174595 (P2004-174595)	(73) 特許権者	000001007
(22) 出願日	平成16年6月11日 (2004.6.11)		キヤノン株式会社
(65) 公開番号	特開2005-39798 (P2005-39798A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成17年2月10日 (2005.2.10)	(74) 代理人	100076428
審査請求日	平成19年1月24日 (2007.1.24)		弁理士 大塚 康徳
(31) 優先権主張番号	特願2003-178610 (P2003-178610)	(74) 代理人	100112508
(32) 優先日	平成15年6月23日 (2003.6.23)		弁理士 高柳 司郎
(33) 優先権主張国	日本国 (JP)	(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(72) 発明者	中山 忠義
			東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

最終頁に続く

(54) 【発明の名称】 データ変換処理装置及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

それぞれが重み付けのない正規化されたデータを受け取り、重み付けされた非正規化回転変換データを出力する2段の梯子演算処理を実行する2つの第1変換手段と、

それぞれが前記第1変換手段からの前記重み付けされた非正規化回転変換データを受け取り、前記重み付けと逆の重み付けを行って重み付けのない正規化ロスレス回転変換データを出力する2段の梯子演算処理を実行する2つの第2変換手段とを有し、

前記2つの第1変換手段の一方から出力される2つのデータを前記2つの第2変換手段に別々に入力し、前記2つの第1変換手段の他方から出力される2つのデータを前記2つの第2変換手段に別々に入力してロスレス4点直交変換を実行することを特徴とするデータ変換処理装置。

10

【請求項 2】

前記2つの第1変換手段のそれぞれは、互いに同じ方向に回転させた非正規化回転変換データを出力することを特徴とする請求項1に記載のデータ変換処理装置。

【請求項 3】

前記第1変換手段は、重み付けされた非正規化ロスレス回転変換データを出力することを特徴とする請求項1又は2に記載のデータ変換処理装置。

【請求項 4】

前記第1変換手段と前記第2変換手段はそれぞれ同じ角度の回転処理を実行することを特徴とする請求項1乃至3のいずれか1項に記載のデータ変換処理装置。

20

【請求項 5】

前記 2 つの第 1 変換手段と前記 2 つの第 2 変換手段との間に、前記 2 つの第 1 変換手段と前記 2 つの第 2 変換手段との接続を切り替える出力データ交換手段を更に有することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載のデータ変換処理装置。

【請求項 6】

入力したデータに対してロスレス 4 点直交変換を実行するデータ変換処理装置のデータ変換処理をコンピュータに実行させるために、該コンピュータを、

それぞれが重み付けのない正規化されたデータを受け取り、重み付けされた非正規化回転変換データを出力する 2 段の梯子演算処理を実行する第 1 及び第 2 変換手段と、

それぞれが前記第 1 及び第 2 変換手段からの前記重み付けされた非正規化回転変換データを受け取り、前記重み付けと逆の重み付けを行って重み付けのない正規化ロスレス回転変換データを出力する 2 段の梯子演算処理を実行する第 3 及び第 4 変換手段とを有し、

前記第 1 変換手段から出力される 2 つのデータを前記第 3 及び第 4 変換手段に別々に入力し、前記第 2 変換手段から出力される 2 つのデータを前記第 3 及び第 4 変換手段に別々に入力してロスレス 4 点直交変換を実行するデータ変換処理装置として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、例えば整数データを出力する可逆変換が可能なロスレス 4 点直交変換処理を実行するデータ変換処理装置及びプログラムに関する。

【背景技術】

【0002】

画像、特に多値画像は非常に多くの情報を含んでおり、その画像を蓄積或いは伝送する際には、その膨大なデータ量が問題になる。このため画像の蓄積或いは伝送に際しては、画像の持つ冗長性を除く、或いは画質の劣化が視覚的に認識し難い程度で画像の劣化を許容することによって、その画像データの量を削減する高能率符号化が用いられる。例えば、静止画像の国際標準符号化方式として ISO と ITU-T により勧告された JEPG では、画像データをブロック（8 画素 × 8 画素）毎に離散コサイン変換（DCT）して DCT 係数に変換した後に、各 DCT 係数を量子化し、更に、その量子化した結果をエントロピー符号化することにより画像データを圧縮している。この DCT を利用した圧縮技術には JEPG 以外にも、H261、MPEG1/2/4 等がある。

【0003】

JEPG において、圧縮・伸長後の画像が元の画像と完全に一致するようロスレス符号化モードが規格化されたが、当時はロスレス変換技術が十分研究されておらず、DCT を用いてロスレス変換を実現することができなかった。そのため、DCT を用いたブロック変換符号化とは異なる技術で、ある画素単位の予測符号化という方法でロスレス符号化を実現していた。

【0004】

その後、ロスレス符号化専用の標準符号化技術（JEPG-LS）が規格化されたが、更に後に規格化された JEPG2000 では、ウェーブレット変換を用いることにより、ロスレス変換と劣化を伴う通常圧縮（ロッシー変換）の両方を実現することが可能となった。

【0005】

近年、DCT においてもロスレス変換の研究が進み、現在広く普及している DCT ベースの JEPG でロスレス圧縮を実現しようとする試みがなされている。JEPG で用いられる DCT は 8 点 DCT である。この 8 点 DCT は、図 1 に示すように、8 点 DCT は 4 つの 2 点変換、4 点 DCT 及び 4 点直交変換とに分解できる。4 点 DCT と 4 点変換は更に細かい 2 点変換に分解できるが、ここでは 4 点 DCT に着目する。

【0006】

4点DCTは、図2に示すように4つの2点変換201～204に分解できる。ロスレス変換を実現するには、分解した各2点変換をロスレス変換にすればよい。2点変換のロスレス変換は、非特許文献1に紹介されているラダー回路網と整数化(丸め)という手法で実現できる。

【0007】

2点変換行列を行列式の値が「1」になるように、入力又は出力データを入れ替えると2点変換は回転変換となる。2点回転変換が2次元のせん断変換3回で実現できることは幾何学分野ではよく知られている。2次元のせん断変換における 2×2 変換行列は、2つの対角要素が共に「1」で、2つの非対角要素の1つが「0」、もう1つが傾斜角に対応するパラメータである。

【0008】

このせん断変換をシグナルフローで表わすと、1回のせん断変換は乗算処理と加算処理からなる1段の梯子演算に置き換えられる。よって、2点回転変換は、図3に示すように3段の梯子演算で実現できる。図3において、各梯子演算における乗算処理後の値を丸め処理して整数化することにより容易にロスレス変換にすることができる。よって、ロスレス変換における梯子演算では、乗算処理部311, 321, 331、丸め処理部313, 323, 333、加算処理部315, 325, 335(減算処理の場合もある)の3つの処理からなる。回転角が θ であるときの乗算係数は、乗算処理部311, 321, 331の順にそれぞれ、 $\tan(\theta/2)$ 、 $-\sin(\theta)$ 、 $\tan(\theta/2)$ である。

【0009】

そして梯子演算の1段ごとに乗算結果を整数化するための丸め処理を行なうことにより、乗算結果が整数でない限り必ず丸め誤差が発生し、各梯子演算で発生する誤差は出力データに含まれる。

【0010】

従来、4つの2点回転変換処理からなる4点直交変換のロスレス変換は、図4のように構成されていた。

【0011】

図4において、401～404は各々2点回転変換で、それぞれは図3に示すように3段の梯子演算で構成されている。このロスレス4点直交変換全体でみると12段の梯子演算があり、同じく12回の丸め処理(Rで示す)が行なわれる。当然のように、丸め処理の回数に比例して丸め誤差が増える。

【0012】

一方、非特許文献1では、4点直交変換を4次元のせん断変換5回に分割して、ロスレス変換を実現している。 n 次元の1回のせん断変換は $(n-1)$ 個の梯子演算に対応するので、4点直交変換では $(4-1) \times 5 = 15$ 個の梯子演算が必要になる。この数はそのまま乗算処理の回数になる。しかし、せん断変換の性質上、丸め処理を大幅に減らすことができる。多次元のせん断変換では、梯子演算の梯子の先(加算対象となるデータ)が1つのデータに集中するので、それらのデータを合算してから丸め処理を行なうことができ、せん断変換1回につき1回の丸め処理で済ませることができる。よって、非特許文献1の4点直交変換では、合計5回の丸め処理で済む。

【0013】

ロスレス変換ではない変換、即ち、線形変換の結果を基準とすると、上記ロスレス変換したデータには、丸め処理の回数に比例して丸め誤差が重畳して変換精度が悪くなる。

【0014】

ロスレス変換後にエントロピー符号化を行って生成した符号化データを復号する際、必ず、そのロスレス変換に対応した逆ロスレス変換を行なうという状況しかありえない場合には何ら問題はない。しかし例えば、ロスレスDCTを用いてJPEG符号化した符号化データを、通常のJPEGデコーダで復号するといった状況が考えられる。その際、ロスレスDCTの変換精度の違いが、復号した画像信号との差となって表れ画質に影響を与える。これは、ロスレス変換が線形変換に近ければ近い方が良いことを示す典型的な例であ

10

20

30

40

50

る。

【 0 0 1 5 】

その他、ロスレス符号化とロッシー符号化とで同じ種類の変換を用いる場合、ロスレス変換が必要となる。そしてロッシー符号化時の符号化効率を考慮すると、該ロスレス変換が線形変換に近ければ近い方が良いといった技術的な背景がある。

【非特許文献 1】小松邦紀、瀬崎薫、「可逆的離散コサイン変換とその画像情報圧縮への応用」、信学技報、I E 9 7 - 8 3 , p p . 1 ~ 6 、 1 9 9 7 年 1 1 月

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 6 】

これまでのロスレス 4 点直交変換の処理方法では、乗算処理が 1 2 回或いは 1 5 回と多く必要であった。乗算回数を少ない方を選べば丸め処理の回数が 1 2 回になり、丸め処理が 5 回で済む方を選べば乗算回数が 1 5 回になる。線形 4 点直交変換に対して誤差が少なくなるように変換精度を上げようとするれば、丸め処理の少ない方法を選ぶ必要があるが、これでは演算回数が増えて処理速度が遅くなったりハードウェア規模が大きくなるという問題がある。

【 0 0 1 7 】

また処理速度やハードウェア規模を優先すれば、丸め処理が 1 2 回に増えて変換精度がかなり悪くなる。このように変換精度と処理速度（ハードウェア規模）を両立させることは困難であった。

【 0 0 1 8 】

本発明は上記従来例に鑑みてなされたもので、本発明の特徴は、少ない演算処理で、或は小さい回路規模でロスレス直交変換処理を実行できるデータ変換処理装置及びプログラムを提供することにある。

【 0 0 1 9 】

また本発明の特徴は、変換精度の良いロスレス直交変換処理を実行できるデータ変換処理装置及びプログラムを提供することにある。

【課題を解決するための手段】

【 0 0 2 0 】

本発明の一態様に係るデータ変換処理装置は以下のような構成を備える。即ち、
それぞれが重み付けのない正規化されたデータを受け取り、重み付けされた非正規化回転変換データを出力する 2 段の梯子演算処理を実行する 2 つの第 1 変換手段と、
それぞれが前記第 1 変換手段からの前記重み付けされた非正規化回転変換データを受け取り、前記重み付けと逆の重み付けを行って重み付けのない正規化ロスレス回転変換データを出力する 2 段の梯子演算処理を実行する 2 つの第 2 変換手段とを有し、
前記 2 つの第 1 変換手段の一方から出力される 2 つのデータを前記 2 つの第 2 変換手段に別々に入力し、前記 2 つの第 1 変換手段の他方から出力される 2 つのデータを前記 2 つの第 2 変換手段に別々に入力してロスレス 4 点直交変換を実行することを特徴とする。

【 0 0 2 1 】

また本発明の一態様に係るプログラムは以下のような構成を備える。即ち、
入力したデータに対してロスレス 4 点直交変換を実行するデータ変換処理装置のデータ変換処理をコンピュータに実行させるために、該コンピュータを、
それぞれが重み付けのない正規化されたデータを受け取り、重み付けされた非正規化回転変換データを出力する 2 段の梯子演算処理を実行する第 1 及び第 2 変換手段と、
それぞれが前記第 1 及び第 2 変換手段からの前記重み付けされた非正規化回転変換データを受け取り、前記重み付けと逆の重み付けを行って重み付けのない正規化ロスレス回転変換データを出力する 2 段の梯子演算処理を実行する第 3 及び第 4 変換手段とを有し、
前記第 1 変換手段から出力される 2 つのデータを前記第 3 及び第 4 変換手段に別々に入力し、前記第 2 変換手段から出力される 2 つのデータを前記第 3 及び第 4 変換手段に別々に入力してロスレス 4 点直交変換を実行するデータ変換処理装置として機能させるための

10

20

30

40

50

プログラム。

【発明の効果】

【0022】

本発明によれば、少ない演算処理でロスレス直交変換処理を実行できるという効果がある。

【0023】

また本発明によれば、小さい回路規模でロスレス直交変換処理を実行できる。

【発明を実施するための最良の形態】

【0024】

以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。

10

【0025】

[実施の形態1]

上記従来技術で説明したように、前述の非特許文献1において、ロスレス2点変換を実現するための構成が図3に示されている。図3の簡単な説明は従来技術のところでも述べたが、ここでは本実施の形態への展開を考慮して回転角を $(-\pi/2)$ として再度説明する。

【0026】

回転角を $(-\pi/2)$ とした場合、1段目の梯子演算部の乗算部311において、一方のデータ(X1)に $(-\tan(\pi/4))$ を乗算した後、この乗算で発生した小数点以下のデータを整数化するための丸め処理を丸め処理部313にて行ない、この丸め処理した結果を加算部315により、もう一方のデータ(X0)に加算する。

20

【0027】

2段目の梯子演算部における乗算係数を $\sin(\pi/4)$ 、3段目の梯子演算部における乗算係数を $(-\tan(\pi/4))$ にして、2段目、3段目の梯子演算部においても同様の処理を行なう。他の文献等を見ても、2点ロスレス変換を梯子演算で構成した例は、このような3段のものしか示されていない。

【0028】

ここで、2段目の梯子演算の乗算部321の乗算係数を半分 $(\sin(\pi/4)/2)$ にして2段に置き換えた構成を図5(A)に示す。丸め処理を無視すれば、図5(A)の構成の処理内容は次のように解釈できる。

30

【0029】

変換処理による回転角度を $(-\pi/2)$ とすると、前段2段の梯子演算501で $(-\pi/4)$ 回転して、後段の2段の梯子演算502で $(-\pi/4)$ 回転し、全体で $(-\pi/2)$ 回転する。ここでは前段2段の梯子演算501と後段2段の梯子演算502の回転角は同じだが、前段2段の梯子演算501による回転変換では変換データが正規化されておらず、回転角 $(-\pi/4)$ に依存したスケーリング係数 $(\cos(\pi/4))$ が2つの変換データに重み付けされている。このスケーリング係数は、梯子演算501の上側出力では $1/\cos(\pi/4)$ で、下側出力では $\cos(\pi/4)$ である。そして後段2段の梯子演算502では、上述の重み付けされた非正規化データを回転処理しながら、逆の重み付けをして、最終的に正規化した回転変換データを生成する。

40

【0030】

従来は、図4の回転処理の内容をここまで分解しても何も得るものがなかった。そればかりか、逆に乗算や丸め処理が増えてしまい、このような構成は無駄が多くて価値のないものであった。ところが、本願発明人はこの分解によって、新しい見方とそれに基づいた新たなロスレス4点変換の構成を見い出した。その構成では、図5(A)の構成を基本的な構成要素として用いる。また後述する実施の形態3も図5(A)の構成なくしては考えられない。よって、図5(A)の構成そのものに発明性があると考え、本発明の実施の形態1とする。

【0031】

[実施の形態1の変形例]

50

図 5 (A) (B) 及び図 6 (A) (B) は、本発明の実施の形態 1 に係るロスレス 4 点直交変換を説明するブロック図である。

【 0 0 3 2 】

図 5 (A) の構成に対して、更に図 5 (B) 及び図 6 (A) , (B) に示すような変形例が考えられる。

【 0 0 3 3 】

図 5 (B) は、図 5 (A) の梯子演算における乗算係数の正負を反転すると共に、梯子演算の梯子の向きを、図 5 (A) に対して全て反転したものである。従って図 5 (B) の機能は、図 5 (A) と全く同じである。

【 0 0 3 4 】

図 6 (A) は、図 5 (A) と梯子演算の向きを同じにし、梯子演算における乗算係数の正負を図 5 (A) に対して反転させたものである。

【 0 0 3 5 】

図 6 (B) は、図 5 (A) の梯子演算における乗算係数を同じにし、梯子演算の梯子の向きを全て図 5 (A) に対して反転させたもので、言い換えると、図 5 (B) の乗算係数の正負を反転したものである。図 6 (B) の機能は図 6 (A) と全く同じである。

【 0 0 3 6 】

次に、上記変形例に対する補足説明を行う。

【 0 0 3 7 】

回転処理における回転方向を反転するには 2 つの方法があり、1 つは梯子演算における乗算係数の正負を反転する方法、もう 1 つは梯子演算の梯子の向きを反転する方法である。このうち前者を図 5 (A) に適用したものが図 6 (A) 、後者を図 5 (A) に適用したものが図 6 (B) 、両方を図 5 (A) に適用したものが図 5 (B) である。図 5 (B) のように両方を適用すると、回転方向が反転の反転で元に戻るため、図 5 (B) の回転方向は図 5 (A) と同じになる。

【 0 0 3 8 】

図 5 (A) と図 5 (B) は機能的には同一であるが、内部データの重み付けが異なる。既に説明したように、図 5 (A) のロスレス変換 5 0 1 の出力データには、スケーリング係数 $1 / \cos()$ と $\cos()$ がそれぞれ重み付けされている。それに対し、図 5 (B) のロスレス変換 5 0 3 の出力データには、前記スケーリング係数のそれぞれの逆数である $\cos()$ と $1 / \cos()$ がそれぞれ重み付けされている。そして図 5 (B) のロスレス変換 5 0 4 は、これに対応した回転と正規化を行なうようになっている。これが図 5 (A) と図 5 (B) の構成上の違いである。

【 0 0 3 9 】

同様に図 6 (A) と図 6 (B) も機能的には同一であるが、上述したように内部データの重み付けが逆数の関係になっている。

【 0 0 4 0 】

図 5 (A) , (B) 及び図 6 (A) , (B) はフローチャートの形式で示されていないが、左の梯子演算から順番に演算するだけで容易にロスレス直交変換がソフトウェアで実現できるし、ハードウェア化も容易である。

【 0 0 4 1 】

一般的に、論文等においても、DCT や直交変換等の処理をフローチャートで表現することは少なく、図 1 ~ 図 6 に示したシグナルフロー形式で表現されることが多い。それは、この表現形式がソフトウェアでの実現とハードウェアでの実装の両方に対応できるからである。よって、以下に示す図もすべてシグナルフロー形式とする。

【 0 0 4 2 】

[実施の形態 2]

本発明の実施の形態 2 では、前述の実施の形態 1 の基本構成を組み合わせた 4 点直交変換方法と変換装置について説明する。まず、実施の形態 2 の基本形を図 7 に示す。同図において係数 $a = \tan()$ である。

【 0 0 4 3 】

図 7 は、本発明の実施の形態 2 に係るロスレス 4 点直交変換を説明するブロック図である。

【 0 0 4 4 】

図 7 の構成は、前述の実施の形態 1 で説明した図 5 (A) と図 5 (B) とを用いて、(正規化データ入力、正規化データ出力の) 正規 4 点直交変換のロスレス変換を行なうものである。各基本構成における回転角は 2 とする。

【 0 0 4 5 】

4 つの入力データは、ロスレス変換 501 , 503 にてロスレス変換され、重み付きの中間データが生成される。この中間データは上から順に、 $1 / \cos ()$ 、 $\cos ($ 10
 $)$, $\cos ()$, $1 / \cos ()$ という重み付けがなされている。ここで同じ重みを持つ 2 番目と 3 番目のデータを入れ替えて、次のロスレス変換 502 , 504 に入力することにより、その重みを取り除くと共にロスレス回転変換を行なう。

【 0 0 4 6 】

ここで丸め処理を無視した時、即ち、線形変換としてみたときの変換処理結果は、上から順に、以下のような結果となる。

[数 1]

$$Y0 = (X0 - a X1 - a X2 + a^2 X3) / (1 + a^2)$$

$$Y1 = (a X0 - a^2 X1 + X2 - a X3) / (1 + a^2)$$

$$Y2 = (a X0 + X1 - a^2 X2 - a X3) / (1 + a^2)$$

$$Y3 = (a^2 X0 + a X1 + a X2 + X3) / (1 + a^2)$$

入力データに対する乗算係数をベクトルとしてみると、4 つの変換式に対応する 4 つのベクトルは全て互いに直交している (内積が「0」になる) ことが分かる。また、ベクトルの絶対値も「1」になっているので、4 点正規直交変換が実現できている。 20

【 0 0 4 7 】

従来の方式では、4 つの回転処理からなる 4 点正規直交変換は、4 つの回転処理の回転角が同じ場合でも、各々の回転処理を 3 段の梯子演算に置き換えて、計 12 個の梯子演算で実現する方法しかなかったが、本実施の形態では、8 段の梯子演算で実現できる。

【 0 0 4 8 】

従来のロスレス変換においては、各々の梯子演算に丸め処理が入るので、従来は 12 回の丸め処理が必要であったものが、本実施の形態 2 によれば、図 7 に示すように 8 回の丸め処理で済み、線形変換に対する変換誤差を減らすことができる。 30

【 0 0 4 9 】

[実施の形態 2 の変形例 1]

使用する 2 つのロスレス 2 点変換は図 5 (A) と図 6 (A) の組み合わせであってもよい。図 6 (A) の回転方向は図 5 (A) の逆方向になるので、これに対応して図 6 (A) 側に入力する 2 つのデータを交換する。この構成を図 8 に示す。

【 0 0 5 0 】

図 8 は、本発明の実施の形態 2 の変形例 1 に係るロスレス 4 点直交変換を説明するブロック図である。 40

【 0 0 5 1 】

この変形例が意味することは、回転方向が逆方向のもの 2 つからでもロスレス 4 点直交変換を構成できるということである。

【 0 0 5 2 】

図 8 の構成から得られる 4 点直交変換の変換式を以下に示す。但し、丸め処理は無視して、線形変換として見た時の変換式である。上述した図 7 の変換式 (数 1) と較べると 3 目と 4 目の式が入れ替わっていることが分かる。これは入力データの交換と回転処理の方向が逆になっているのに対応している。

[数 2]

$$Y0 = (X0 - a X1 - a X2 + a^2 X3) / (1 + a^2)$$

$$Y1 = (a \times X0 - a^2 \times X1 + X2 - a \times X3) / (1 + a^2)$$

$$Y2 = (a^2 \times X0 + a \times X1 + a \times X2 + X3) / (1 + a^2)$$

$$Y3 = (a \times X0 + X1 - a^2 \times X2 - a \times X3) / (1 + a^2)$$

【実施の形態 2 の変形例 2】

更に図 7 の構成を図 9 の構成に変形すると、丸め処理の回数を少なくでき、更に変換誤差を減らすことが出来る。

【0053】

図 9 は、本発明の実施の形態 2 の変形例 2 に係るロスレス 4 点直交変換を説明するブロック図である。

【0054】

図 9 では、図 7 におけるロスレス変換 501 の 2 段目の梯子演算の丸め処理とロスレス変換 504 の 1 段目の梯子演算の丸め処理を一体化する。即ち、両者の乗算結果を加算した後、まとめて 1 回丸め処理をして、その結果を加算対象となるデータに加算してもロスレス性は保たれる。

【0055】

また図 7 の変換部 503 の 2 段目の梯子演算の丸め処理と変換部 502 の 1 段目の梯子演算の丸め処理も同様に、1 つの丸め処理で一体化できる。

【0056】

次に、このように一体化した丸め処理を梯子演算における 3 つめの処理である加算処理の後に移動する。こうして移動した丸め処理部を 801, 803 で示している。このような移動が可能なのは、 $\text{round}()$ を丸め関数、 R を実数、 N を整数とした時に以下の関係が成り立つからである。

【数 3】

$$\text{round}(R) + N = \text{round}(R + N)$$

ここで左辺は、移動前の丸め処理、右辺が移動後の丸め処理に対応している。この数式(3)は、実数に整数を加算してから丸め処理しても、実数を丸め処理した後、その丸め処理した整数を加算しても同じ結果になることを示している。この実数は、図 9 の新たな丸め処理部 801, 803 より手前の梯子演算 2 段及び 3 段の各乗算結果の和に対応する。尚、本発明の実施の形態に係る丸め処理としては、最も一般的な四捨五入でもよいし、切り捨て或いは切り上げ処理でもよい。

【0057】

【実施の形態 2 の変形例 3】

図 7 の構成は図 10 の構成のようにも変形できる。

【0058】

図 10 は、本発明の実施の形態 2 の変形例 3 に係るロスレス 4 点直交変換を説明するブロック図である。

【0059】

図 10 は、図 7 の構成における乗算係数 $\{a / (1 + a \times a)\}$ の乗算を共通化したもので、当業者がみれば容易に分かる変形である。901 は共通化した乗算処理部である。903 は減算処理で、乗算を共通化するためにデータを一本化している。905 は丸め処理部で、乗算処理部 901 の乗算結果を整数化している。907 と 909 は加算処理部で、それぞれ整数化したデータを別のデータに加算している。その他の処理部はこれまでに出てきているものである。

【0060】

この図 10 の構成の特徴は、ロスレス 2 点直交変換 2 個分の演算規模よりも、このロスレス 4 点直交変換の演算規模の方が小さいということである(減算処理が 1 つ増えているが、それよりも複雑な演算である乗算が 1 つ減っている。ハードウェアではこの差は大きい)。

【0061】

但し、図 10 のように変形してしまうと、梯子演算のみで全て処理しているとはいえない

10

20

30

40

50

くなってしまう。しかし、梯子演算を次のように拡張することで、図 10 の構成も全て梯子演算で処理していると解釈することができる。

【 0 0 6 2 】

通常の梯子演算は、1 入力 1 出力形式であるが、図 10 の 9 0 1 , 9 0 3 , 9 0 5 , 9 0 7 , 9 0 9 を含む構成を 2 入力 2 出力形式の梯子演算と考える。これを更に一般化すると、n 入力 m 出力の梯子演算も考えられる。但し、乗算処理部の数は 1 つに限定する。また、1 つの乗算処理部に対して複数入力のデータを一本化するための加減算処理部が、拡張梯子演算には必要となる。

【 0 0 6 3 】

このような拡張梯子演算を導入することによって、図 10 の構成は、1 入力 1 出力形式の梯子演算 4 つと 2 入力 2 出力形式の梯子演算 1 つから構成されているとすることができる。

10

【 0 0 6 4 】

図 10 の構成から丸め処理を取り除くと、線形の 4 点直交変換を少ない演算量で実現することができる。その構成は図 10 から 5 つの丸め処理部を取り除いただけである。その構成を図 11 に示す。

【 0 0 6 5 】

この図 11 の構成は図 10 に似ているので、この変形例に含めているが、ロスレス変換よりも汎用性のある線形直交変換の高速演算法として有効である。さらに図 11 の構成は図 18 のように変形でき、梯子演算の乗算処理は最終的に 4 つまで減らすことができる。この図 18 の構成においても注意深く丸め処理を導入すれば、ロスレス変換が可能であることは言うまでもない。尚、図 18 において、1 8 0 1 は係数 a を乗算する乗算器、1 8 0 3 は加算器、1 8 0 5 は減算器である。

20

【 0 0 6 6 】

[実施の形態 2 の変形例 4]

更に図 7 において、 $a = \tan(\quad) = 1$ の時、図 7 に示す 4 点直交変換はロスレス 4 点アダマール変換となる。

【 0 0 6 7 】

通常、アダマール変換を行なう際は、入力データを並び替えるのが一般的である（例えば、X0 と X3 の間でバタフライ演算を行なう）が、ここでは、入力データの並び替えは行なわず、その代わりに出力データの並び替えを行なった。

30

【 0 0 6 8 】

図 7 で $a = 1$ にして、出力の並び替えを行なった構成を図 12 に示す。

【 0 0 6 9 】

図 12 は、本発明の実施の形態 2 の変形例 4 に係るロスレス 4 点直交変換を説明するブロック図である。

【 0 0 7 0 】

梯子演算における乗算係数が整数値の場合、小数点以下の値は「0」のままなので丸め処理は不要となり、その分、丸め処理が少なくなる。また、乗算係数 (1 / 2) はビットシフトのみで実現できる処理なので、乗算器も不要となる。

40

【 0 0 7 1 】

図 12 の構成は当然のことながら、前述の変形例 2 (図 9)、変形例 3 (図 10) のように変形できる。図 10 のように変形した場合の構成は重要な意味があるので、その構成を図 13 に示す。

【 0 0 7 2 】

図 13 は、図 10 で $a = 1$ とした場合のロスレス 4 点直交変換を説明するブロック図である。

【 0 0 7 3 】

図 13 の構成をみると、ビットシフト (1 / 2) 1 3 0 0 と 1 回の丸め処理 1 3 0 1 と、7 回の加減算処理 1 3 0 2 ~ 1 3 0 8 で演算できる。この演算量は、線形のアダマール

50

変換では高速演算として当たり前のように用いられるバタフライ演算で実現した時の演算量（加減算 8 回）よりも少ない。

【 0 0 7 4 】

一方、ロスレス 4 点アダマール変換の構成は、下記の文献 2 に示されている。この文献 2 では、該ロスレス変換を実現するために、4 点アダマール行列を三角行列に分解して、それを梯子演算に置き換えるといった複雑な手順で実現しているが、その構成は、本実施の形態 2 の変形例 4 から得られた図 1 2 の構成より加算処理が 1 つ多く、8 回の加減算処理を要している。本実施の形態 2 を用いた場合、一般化したロスレス 4 点直交変換の特殊解として得られる上に、僅かな変形で加減算器の数が最小となる構成が得られる。

【 0 0 7 5 】

（文献 2）福岡慎治、大山公一、岩橋政宏、神林紀、「ロスレスアダマール変換を応用したロスレス 8 点高速離散コサイン変換」、信学技報、I E 9 9 - 6 5、p p . 3 7 - 4 4、1 9 9 9 年 1 0 月

〔実施の形態 2 の適用例〕

図 2 に示す 4 点 D C T 演算では、 $(3/8)$ の回転処理が必要である。この回転角 $(3/8)$ は、変換空間の軸の交換や正負反転などの操作により、 $(/8)$ の回転処理に変えることもできるが、ここでは $(3/8)$ の回転処理のまま扱う。この 4 点 D C T を 2 次元化して、水平方向処理と垂直方向の処理の一部の順序を交換すると、途中の処理として局所的に以下のような演算が現われる。

【 0 0 7 6 】

【数 4】

$$\begin{bmatrix} \cos \frac{3\pi}{8} & \sin \frac{3\pi}{8} \\ -\sin \frac{3\pi}{8} & \cos \frac{3\pi}{8} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} \cos \frac{3\pi}{8} & -\sin \frac{3\pi}{8} \\ \sin \frac{3\pi}{8} & \cos \frac{3\pi}{8} \end{bmatrix}$$

【 0 0 7 7 】

この数式（ 4 ）において、 X_{11} 、 X_{12} 、 X_{21} 、 X_{22} は演算途中のデータであり、このデータの左側の変換行列を水平方向処理に対応付けると、このデータの右側の変換行列は垂直方向処理に対応する。どちらの変換行列も $(3/8)$ の回転処理を表わす。線形変換ではどちらの変換処理から先に行なってもよい（この時点ではまだロスレス化のための丸め処理が入っていないのでロスレス変換ではなく、線形変換である）が、ここでは左側の変換行列から先に処理を行なうものとする。

【 0 0 7 8 】

この変換処理を細かく見ていくと、 (X_{11}, X_{21}) 、 (X_{12}, X_{22}) という 2 個 2 組のデータに対して各々に $(3/8)$ の回転処理を行なった後、この変換結果の転置、即ち、一部のデータを交換して再度 $(3/8)$ の回転処理を行なうものである。この処理をロスレス変換として実現するのが、前述の図 5 ～ 図 9 に示す構成において、 $= 3/8$ とした場合である。

【 0 0 7 9 】

〔実施の形態 3〕

実施の形態 3 では、前述の実施の形態 1 で説明した図 5（ A ）と（ B ）、それに、データ切り替え部を用いて、2 点直交変換と 4 点直交変換の切り替えが可能な直交変換処理を提供する。この構成を図 1 4 に示す。

【 0 0 8 0 】

図 1 4 は、本発明の実施の形態 3 に係るロスレス 4 点直交変換を説明するブロック図で

10

20

30

40

50

ある。

【 0 0 8 1 】

この構成で新たに登場する構成要素は、データ切り替え部 1 2 0 1 である。不図示の制御信号に基づいて、このデータ切り替え部 1 2 0 1 によりデータの流を切り替えればロスレス 4 点直交変換となり、データを切り替えなければ 2 組のロスレス 2 点直交変換となる。

【 0 0 8 2 】

[実施の形態 3 の変形例]

前述の実施の形態 2 では、図 7 に示す構成を図 1 0 に示す構成まで簡略化できたが、この実施の形態 3 では 2 種類の機能を実現するため、そこまで簡略化することはできない。しかし、図 1 5 に示す構成に変形することができる。

10

【 0 0 8 3 】

図 1 5 は、本発明の実施の形態 3 の変形例に係るロスレス 4 点直交変換を説明するブロック図である。

【 0 0 8 4 】

この図 1 5 では、図 1 4 における係数 $\{ a / (1 + a \times a) \}$ を乗算する乗算器と、係数 $\{ - a / (1 + a \times a) \}$ を乗算する乗算器とそれぞれ 1 つにまとめることにより、乗算の回数をロスレス 2 点直交変換 2 個分と同じ数である 6 回に減らしている。

【 0 0 8 5 】

[実施の形態 4]

20

実施の形態 4 では、これまで説明した梯子演算を適用した場合のロスレス 2 次元 D C T 変換部により変換して得られた D C T 係数を量子化及びハフマン符号化して画像データ等を符号化する場合で説明する。

【 0 0 8 6 】

一般的に 2 次元 D C T と言えば、J P E G など用いられている 8×8 のブロックサイズが使用されるが、ここでは 4×4 のロスレス 2 次元 D C T 変換の場合で示す。 4×4 の 2 次元 D C T を 8×8 の 2 次元 D C T へ拡張するのは、公知の技術で対応できる。

【 0 0 8 7 】

4 点 D C T の変換行列 M d c t を次のように表わし、

【 0 0 8 8 】

30

【 数 5 】

$$M_{dct} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ C1 & C3 & -C3 & -C1 \\ 1 & -1 & -1 & 1 \\ C3 & -C1 & C1 & -C3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta \\ 0 & 0 & 1 & 0 \\ 0 & -\beta & 0 & \alpha \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$Ci = \sqrt{2} \cos \frac{i\pi}{8}, \quad \alpha = \cos \frac{\pi}{8}, \quad \beta = \sin \frac{\pi}{8}$$

40

【 0 0 8 9 】

元の 4×4 データを、 $d_{00}, d_{01}, d_{02}, \dots, d_{32}, d_{33}$ で表わすと、 4×4 の 2 次元 D C T は以下のように数式で表現される。

【 0 0 9 0 】

【数 6】

$$M_{dct} \begin{bmatrix} d_{00} & d_{01} & d_{02} & d_{03} \\ d_{10} & d_{11} & d_{12} & d_{13} \\ d_{20} & d_{21} & d_{22} & d_{23} \\ d_{30} & d_{31} & d_{32} & d_{33} \end{bmatrix} M_{dct}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \beta \\ 0 & 0 & 1 & 0 \\ 0 & -\beta & 0 & \alpha \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & -\beta \\ 0 & 0 & 1 & 0 \\ 0 & \beta & 0 & \alpha \end{bmatrix}$$

【 0 0 9 1 】

10

上式において、 x_{00} 、 x_{01} 、 x_{02} 、 \dots 、 x_{32} 、 x_{33} は、元のデータを2次元アダマール変換したデータを示している。丸め処理を入れてロスレス化を図ったロスレス2次元アダマール変換には、特願2002-193298に記載の変換方法を用いることができる。

【 0 0 9 2 】

ロスレス2次元アダマール変換後のデータに対して、水平方向のロスレス回転変換と垂直方向のロスレス回転変換を行うと、ロスレス2次元DCT変換となる。水平方向のロスレス回転変換は、 x_{01} と x_{03} 、 x_{11} と x_{13} 、 x_{21} と x_{23} 、 x_{31} と x_{33} の4組のデータに対して行い、垂直方向のロスレス回転変換は、水平変換の結果である、 x_{10} と x_{30} 、 x_{11} と x_{31} 、 x_{12} と x_{32} 、 x_{13} と x_{33} の4組のデータに対して行う。

20

【 0 0 9 3 】

図16は、本発明の実施の形態4に係る4×4ロスレス2次元DCT変換を説明するブロック図である。

【 0 0 9 4 】

図16において、 x_{01} と x_{03} 、 x_{21} と x_{23} の2組のデータに対しては水平方向のみのロスレス回転変換1601、1602、 x_{10} と x_{30} 、 x_{12} と x_{32} の2組のデータに対しては垂直方向のみのロスレス回転変換1603、1604を行い、 x_{11} と x_{13} 、 x_{31} 、 x_{33} の1組のデータに対しては、水平方向と垂直方向のロスレス2次元回転変換1605を行っている。

【 0 0 9 5 】

30

水平方向又は垂直方向のみのロスレス回転変換1601～1604のそれぞれは、図3に示す従来の3段の梯子演算で行い、ロスレス2次元回転変換1605は、図9又は図10に示す構成の梯子演算を行うことで実現できる。回転変換を一切行わない他のデータ x_{00} と x_{02} 、 x_{20} 、 x_{22} は、ロスレス2次元アダマール変換係数がそのままロスレス2次元DCT変換係数となる。

【 0 0 9 6 】

図17は、本発明の実施の形態4に係るロスレス符号化が可能な符号化処理を説明するブロック図である。

【 0 0 9 7 】

図16に示すロスレス2次元DCT変換処理1701を最初に行い、その後、量子化処理1702、ハフマン符号化処理1703を行うことにより符号化データが得られる。ここで量子化ステップが全て「1」の時にロスレス符号化を行うことができる。即ち、図16のロスレス2次元DCT変換1605とロスレス逆変換の関係にあるロスレス2次元逆DCT変換を復号処理で行えば、量子化ステップが全て「1」の時に、元のデータを完全に復号できる。

40

【 0 0 9 8 】

よって、符号化処理時の量子化ステップの設定により、ロスレス符号化から非ロスレスで劣化のある高圧縮符号化まで連続的に圧縮及び伸張画像の画質を制御できる。

【 0 0 9 9 】

[他の実施の形態]

50

本発明の目的は前述したように、実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体をシステム或は装置に提供し、そのシステム或は装置のコンピュータ（又はCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。このようなプログラムコードを供給するための記憶媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0100】

10

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOS（オペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれている。

【0101】

更に、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書きこまれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含む。

20

【0102】

以上説明したように本実施の形態によれば、少ない演算処理で変換精度の良い、ロスレス4点直交変換処理方法および装置を提供できるようになった。具体的には、最適化した構成で5回の乗算と5回の丸め処理でロスレス4点直交変換を実現できる。

【0103】

また従来は、12回の乗算と12回の丸め処理、又は15回の乗算と5回の丸め処理を要していたのに対して、変換精度がほぼ同じもの（丸め処理回数が同じもの）と較べても乗算回数を1/3に減らすことができた。

【図面の簡単な説明】

30

【0104】

【図1】一般的な8点DCTの1つの演算方法を説明するブロック図である。

【図2】一般的な4点DCTの1つの演算方法を説明するブロック図である。

【図3】従来のロスレス2点直交回転変換処理部の構成を示すブロック図である。

【図4】従来のロスレス4点直交変換処理部の構成を示すブロック図である。

【図5】本発明の実施の形態1に係るロスレス4点直交変換を説明するブロック図である。

。

【図6】本発明の実施の形態1に係るロスレス4点直交変換を説明するブロック図である。

。

【図7】本発明の実施の形態2に係るロスレス4点直交変換を説明するブロック図である。

40

。

【図8】本発明の実施の形態2の変形例1に係るロスレス4点直交変換を説明するブロック図である。

【図9】本発明の実施の形態2の変形例2に係るロスレス4点直交変換を説明するブロック図である。

【図10】本発明の実施の形態2の変形例3に係るロスレス4点直交変換を説明するブロック図である。

【図11】実施の形態2の変形例3から丸め処理部を取り除いた構成で高速線形4点直交変換処理を実現する構成を示す図である。

【図12】本発明の実施の形態2の変形例4に係るロスレス4点直交変換を説明するブロ

50

ック図である。

【図 1 3】本発明の実施の形態 2 の変形例 5 に係るロスレス 4 点直交変換（ロスレスアダマール変換）を説明するブロック図である。

【図 1 4】本発明の実施の形態 3 に係るロスレス 4 点直交変換を説明するブロック図である。

【図 1 5】本発明の実施の形態 3 の変形例に係るロスレス 4 点直交変換を説明するブロック図である。

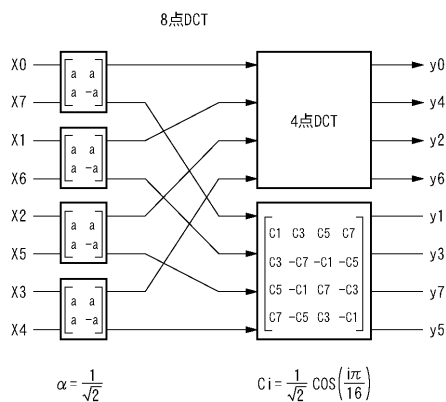
【図 1 6】本発明の実施の形態 4 に係る 4×4 ロスレス 2 次元 DCT 変換を説明するブロック図である。

【図 1 7】本発明の実施の形態 4 に係るロスレス符号化が可能な符号化処理を説明するブロック図である。

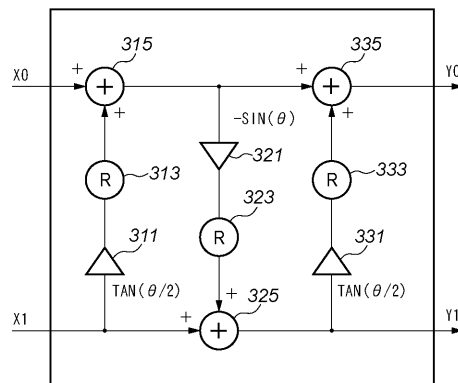
【図 1 8】実施の形態 2 の変形例 3 から丸め処理部を取り除いた構成に変更を加えて線形 4 点直交変換処理を実現する構成を示す図である。

10

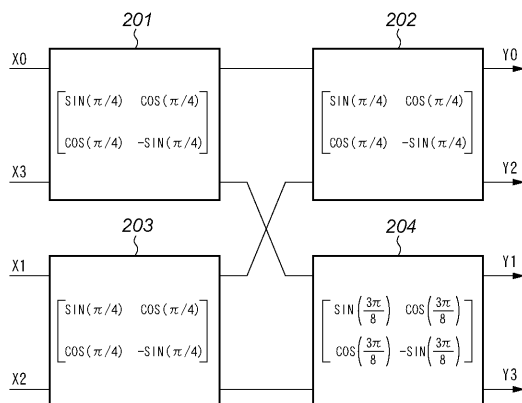
【図 1】



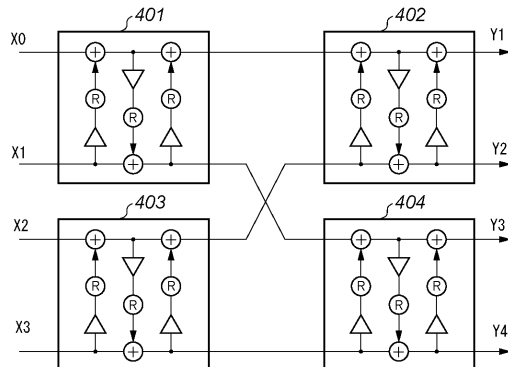
【図 3】



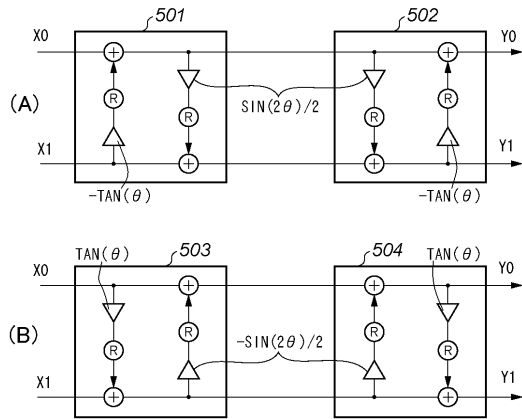
【図 2】



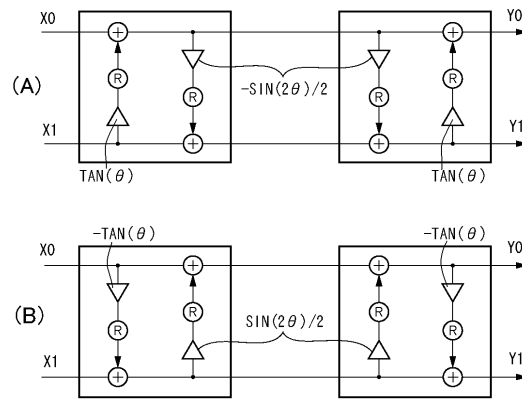
【図 4】



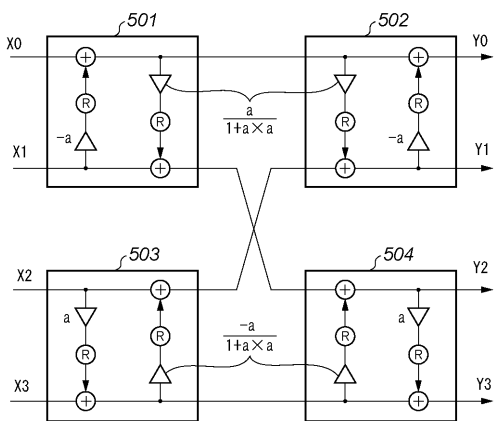
【図 5】



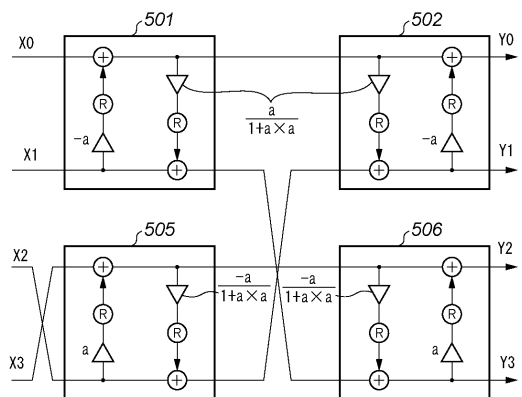
【図 6】



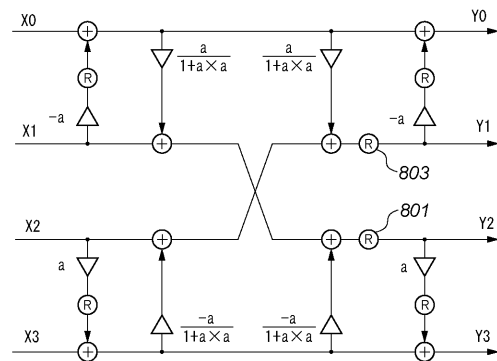
【図 7】



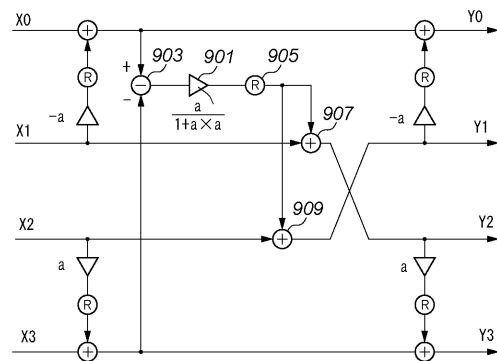
【図 8】



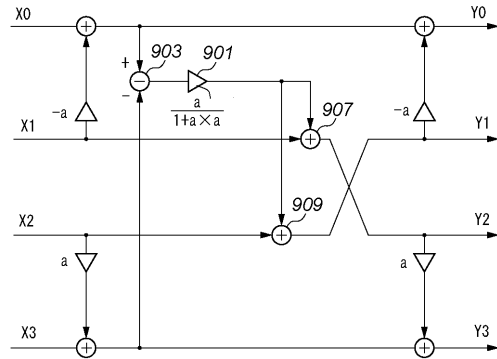
【図 9】



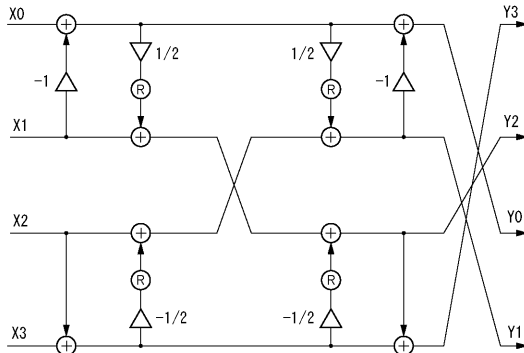
【図 10】



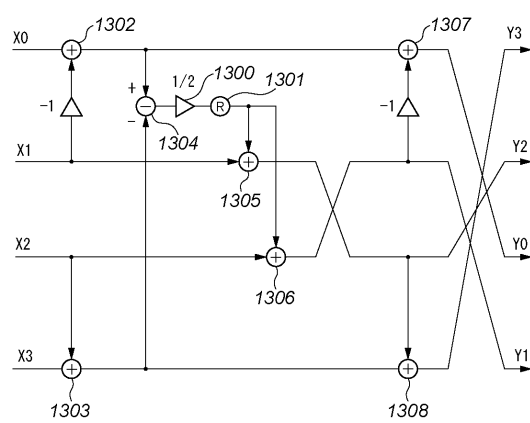
【図 1 1】



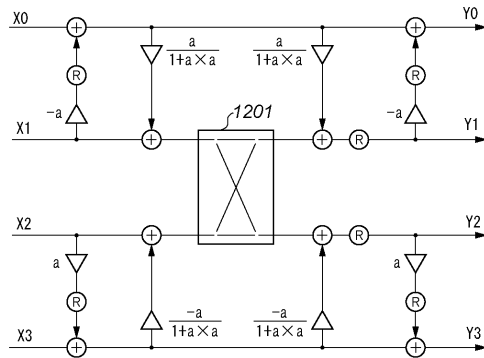
【図 1 2】



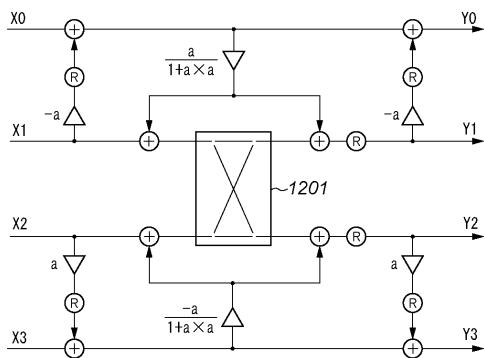
【図 1 3】



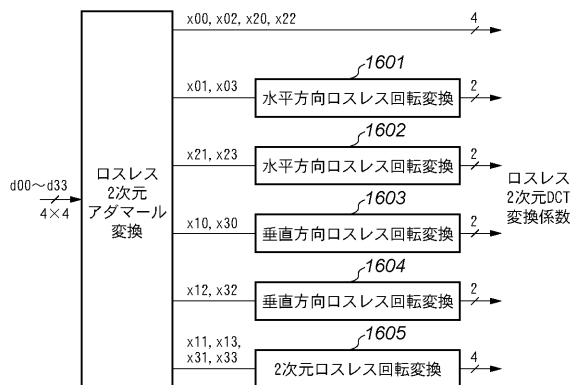
【図 1 4】



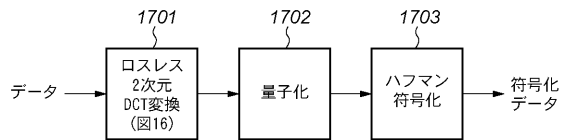
【図 1 5】



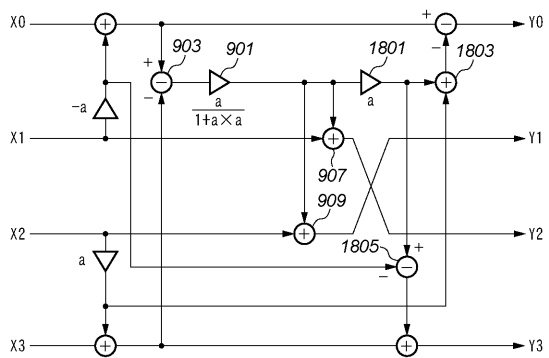
【図 1 6】



【図 1 7】



【図 1 8】



フロントページの続き

審査官 北村 智彦

- (56)参考文献 特開平10-322219(JP,A)
特開2001-128174(JP,A)
特開2003-258645(JP,A)
特開2004-038451(JP,A)
小林邦紀, 瀬崎薫, 2次元ロスレス変換とその画像符号化への応用, 電子情報通信学会技術研究報告, 2000年10月17日, Vol.100, No.385, DSP2000-120, p.19-26
福岡慎治, 大山公一, 岩橋政宏, 神林紀嘉, ロスレスアダマール変換ベースロスレス高速DCTの性能評価, 2000年電子情報通信学会総合大会講演論文集, 2000年 3月 7日, 情報・システム2, p.47
Kunitoshi KOMATSU, Kaoru SEZAKI, DESIGN OF LOSSLESS LOT AND ITS PERFORMANCE EVALUATION, Acoustics, Speech, and Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, 2000年, Vol.6, p.2119-2122

(58)調査した分野(Int.Cl., DB名)

H03M3/00-11/00
H04N 1/41
H04N 7/30