

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-178865

(P2012-178865A)

(43) 公開日 平成24年9月13日(2012.9.13)

(51) Int.Cl. F I テーマコード(参考)
HO4N 7/32 (2006.01) HO4N 7/137 Z 5C159

審査請求 有 請求項の数 1 O L (全 22 頁)

<p>(21) 出願番号 特願2012-110176 (P2012-110176) (22) 出願日 平成24年5月14日(2012.5.14) (62) 分割の表示 特願2011-252210 (P2011-252210)の分割 原出願日 平成10年6月9日(1998.6.9) (31) 優先権主張番号 特願平9-150656 (32) 優先日 平成9年6月9日(1997.6.9) (33) 優先権主張国 日本国(JP)</p>	<p>(71) 出願人 000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号 (74) 代理人 100100310 弁理士 井上 学 (72) 発明者 中屋 雄一郎 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内 (72) 発明者 彌養 義人 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内 Fターム(参考) 5C159 LB15 MA00 MA04 MA05 MA23 NN14 PP05 PP06 PP07 PP16 RC12 UA02 UA05</p>
---	---

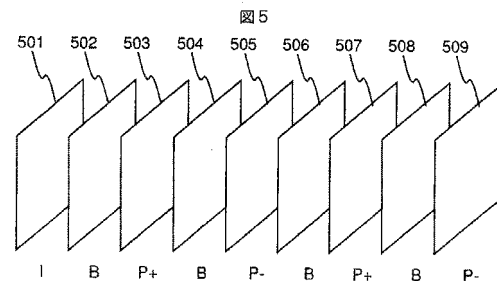
(54) 【発明の名称】 動画像の符号化方法および復号化方法

(57) 【要約】

【課題】 輝度または色差の強度値が整数に量子化される画像符号化方法において、動き補償において行われる共1次内挿の処理によって発生する丸め込み誤差の蓄積を防ぐ方法を提供する。

【解決手段】 片方向の動き補償予測を行うPフレームに関し、整数に0.5を加えた値を切り上げるP+フレーム503、507と、整数に0.5を加えた値を切り捨てるP-フレーム505、509の2種類を用意する。

【選択図】 図5



【特許請求の範囲】

【請求項 1】

動き補償を用いて予測画像を合成し、前記予測画像と入力画像との差分に関する情報と

、
前記動き補償によって求められた動きベクトルに関する情報とを多重化する動画像の符号化方法において、

前記予測画像は、プラスの丸め込みを用いる動き補償によって合成される場合と、マイナスの丸め込みを用いる動き補償によって合成される場合とを有することを特徴とする動画像の符号化方法。

【発明の詳細な説明】

10

【技術分野】

【0001】

本発明は、フレーム間予測を行い、輝度または色の強度が量子化された数値として表される動画像符号化および復号化方法、ならびに動画像の符号化装置および復号化装置に関するものである。

【背景技術】

【0002】

動画像の高能率符号化において、時間的に近接するフレーム間の類似性を活用するフレーム間予測（動き補償）は情報圧縮に大きな効果を示すことが知られている。現在の画像符号化技術の主流となっている動き補償方式は、動画像符号化方式の国際標準である H. 263、MPEG1、MPEG2 に採用されている半画素精度のブロックマッチングである。この方式では、符号化しようとする画像を多数のブロックに分割し、ブロックごとにその動きベクトルを水平・垂直方向に隣接画素間距離の半分の長さを最小単位として求める。

20

【0003】

この処理を数式を用いて表現すると以下ようになる。符号化しようとするフレーム（現フレーム）の予測画像 P の座標 (x, y) におけるサンプル値（輝度または色差の強度のサンプル値）を $P(x, y)$ 、参照画像 R（P と時間的に近接しており、既に符号化が完了しているフレームの復号画像）の座標 (x, y) におけるサンプル値を $R(x, y)$ とする。また、 x と y は整数であるとして、P と R では座標値が整数である点に画素が存在すると仮定する。また、画素のサンプル値は負ではない整数として量子化されているとする。このとき、P と R の関係は、

30

【0004】

【数 1】

$$P(x,y) = R(x+u_i, y+v_i) \quad (x, y) \in B_i, 0 \leq i < N \quad \dots \text{ (数 1)}$$

【0005】

で表される。ただし、画像は N 個のブロックに分割されるとして、 B_i は画像の i 番目のブロックに含まれる画素、 (u_i, v_i) は i 番目のブロックの動きベクトルを表している。

40

【0006】

u_i と v_i の値が整数ではないときには、参照画像において実際には画素が存在しない点の強度値を求めることが必要となる。この際の処理としては、周辺 4 画素を用いた共 1 次内挿が使われることが多い。この内挿方式を数式で記述すると、 d を正の整数、 $0 \leq p, q < d$ として、 $R(x + p/d, y + q/d)$ は、

【0007】

【数 2】

$$R\left(x+\frac{p}{d}, y+\frac{q}{d}\right) = ((d-q) \left((d-p) R(x, y) + p R(x+1, y) \right) + q \left((d-p) R(x, y+1) + p R(x+1, y+1) \right)) // d^2 \quad \dots \text{ (数 2)}$$

【0008】

で表される。ただし「//」は除算の一種で、通常の除算（実数演算による除算）の結果を近隣の整数に丸め込むことを特徴としている。

10

【0009】

図1にH.263の符号化器の構成例100を示す。H.263は、符号化方式として、ブロックマッチングとDCT（離散コサイン変換）を組み合わせたハイブリッド符号化方式（フレーム間/フレーム内適応符号化方式）を採用している。

【0010】

減算器102は入力画像（現フレームの原画像）101とフレーム間/フレーム内符号化切り換えスイッチ119の出力画像113（後述）との差を計算し、誤差画像103を出力する。この誤差画像は、DCT変換器104でDCT係数に変換された後に量子化器105で量子化され、量子化DCT係数106となる。この量子化DCT計数は伝送情報として通信路に出力されると同時に、符号化器内でもフレーム間予測画像を合成するために使用される。

20

【0011】

以下に予測画像合成の手順を説明する。上述の量子化DCT係数106は、逆量子化器108と逆DCT変換器109を経て復号誤差画像110（受信側で再生される誤差画像と同じ画像）となる。これに、加算器111においてフレーム間/フレーム内符号化切り換えスイッチ119の出力画像113（後述）が加えられ、現フレームの復号画像112（受信側で再生される現フレームの復号画像と同じ画像）を得る。この画像は一旦フレームメモリ114に蓄えられ、1フレーム分の時間だけ遅延される。したがって、現時点では、フレームメモリ114は前フレームの復号画像115を出力している。この前フレームの復号画像と現フレームの入力画像101がブロックマッチング部116に入力され、

30

ブロックマッチングの処理が行われる。

【0012】

ブロックマッチングでは、画像を複数のブロックに分割し、各ブロックごとに現フレームの原画像に最も似た部分を前フレームの復号画像から取り出すことにより、現フレームの予測画像117が合成される。このときに、各ブロックが前フレームと現フレームの間でどれだけ移動したかを検出する処理（動き推定処理）を行う必要がある。動き推定処理によって検出された各ブロックごとの動きベクトルは、動きベクトル情報120として受信側へ伝送される。

【0013】

受信側は、この動きベクトル情報と前フレームの復号画像から、独自に送信側で得られるものと同じ予測画像を合成することができる。予測画像117は、「0」信号118と共にフレーム間/フレーム内符号化切り換えスイッチ119に入力される。このスイッチは、両入力のいずれかを選択することにより、フレーム間符号化とフレーム内符号化を切り換える。予測画像117が選択された場合（図2はこの場合を表している）には、フレーム間符号化が行われる。一方、「0」信号が選択された場合には、入力画像がそのままDCT符号化されて通信路に出力されるため、フレーム内符号化が行われることになる。受信側が正しく復号化画像を得るためには、送信側でフレーム間符号化が行われたかフレーム内符号化が行われたかを知る必要がある。このため、識別フラグ121が通信路へ出力される。最終的なH.263符号化ビットストリーム123は多重化器122で量子化DCT係数、動きベクトル、フレーム内/フレーム間識別フラグの情報を多重化すること

40

50

によって得られる。

【 0 0 1 4 】

図 2 に図 1 の符号化器が出力した符号化ビットストリームを受信する復号化器 2 0 0 の構成例を示す。受信した H . 2 6 3 ビットストリーム 2 1 7 は、分離器 2 1 6 で量子化 D C T 係数 2 0 1、動きベクトル情報 2 0 2、フレーム内 / フレーム間識別フラグ 2 0 3 に分離される。量子化 D C T 係数 2 0 1 は逆量子化器 2 0 4 と逆 D C T 変換器 2 0 5 を経て復号化された誤差画像 2 0 6 となる。この誤差画像は加算器 2 0 7 でフレーム間 / フレーム内符号化切り換えスイッチ 2 1 4 の出力画像 2 1 5 を加算され、復号化画像 2 0 8 として出力される。フレーム間 / フレーム内符号化切り換えスイッチはフレーム間 / フレーム内符号化識別フラグ 2 0 3 に従って、出力を切り換える。フレーム間符号化を行う場合に用いる予測画像 2 1 2 は、予測画像合成部 2 1 1 において合成される。ここでは、フレームメモリ 2 0 9 に蓄えられている前フレームの復号画像 2 1 0 に対して、受信した動きベクトル情報 2 0 2 に従ってブロックごとに位置を移動させる処理が行われる。一方フレーム内符号化の場合、フレーム間 / フレーム内符号化切り換えスイッチは、「 0 」信号 2 1 3 をそのまま出力する。

10

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 5 】

H . 2 6 3 が符号化する画像は、輝度情報を持つ 1 枚の輝度プレーン (Y プレーン) と色情報 (色差情報とも言う) を持つ 2 枚の色差プレーン (U プレーンと V プレーン) で構成されている。このとき、画像が水平方向に $2m$ 画素、垂直方向に $2n$ 画素持っている場合に (m と n は正の整数とする)、Y プレーンは水平方向に $2m$ 、垂直方向に $2n$ 個の画素を持ち、U および V プレーンは水平方向に m 、垂直方向に n 個の画素を持つことを特徴としている。このように色差プレーンの解像度が低いのは、人間の視覚が色差の空間的な変化に比較的鈍感であるという特徴を持つためである。このような画像を入力として、H . 2 6 3 ではマクロブロックと呼ばれるブロックを単位として符号化・復号化の処理が行われる。

20

【 0 0 1 6 】

図 3 にマクロブロックの構成を示す。マクロブロックは Y ブロック、U ブロック、V ブロックの 3 個のブロックで構成され、輝度値情報を持つ Y ブロック 3 0 1 の大きさは 16×16 画素、色差情報をもつ U ブロック 3 0 2 および V ブロック 3 0 3 の大きさは 8×8 画素となっている。

30

【 0 0 1 7 】

H . 2 6 3 では、各マクロブロックに対して半画素精度のブロックマッチングが適用される。したがって、推定された動きベクトルを (u, v) とすると、 u と v はそれぞれ画素間距離の半分、つまり $1/2$ を最小単位として求められることになる。このときの強度値 (以下では、「輝度値」と色差の強度値を総称して「強度値」と呼ぶ) の内挿処理の様子を図 4 に示す。H . 2 6 3 では、数 2 の内挿を行う際に、除算の結果は最も近い整数に丸め込まれ、かつ除算の結果が整数に 0.5 を加えた値となるときには、これを 0 から遠ざける方向に切り上げる処理が行われる。

40

【 0 0 1 8 】

つまり、図 4 において、画素 4 0 1、4 0 2、4 0 3、4 0 4 の強度値をそれぞれ L_a 、 L_b 、 L_c 、 L_d とすると (L_a 、 L_b 、 L_c 、 L_d は負ではない整数)、内挿により強度値を求めたい位置 4 0 5、4 0 6、4 0 7、4 0 8 の強度値 I_a 、 I_b 、 I_c 、 I_d は (I_a 、 I_b 、 I_c 、 I_d は負ではない整数)、以下の式によって表される。

【 0 0 1 9 】

【数3】

$$I_a = L_a$$

$$I_b = [(L_a + L_b + 1) / 2]$$

$$I_c = [(L_a + L_c + 1) / 2]$$

$$I_d = [(L_a + L_b + L_c + L_d + 2) / 4]$$

... (数3)

10

【0020】

ただし、「[]」は小数部分を切り捨てる処理を表している。

【0021】

このとき、除算の結果を整数値に丸め込む処理によって発生する誤差の期待値を計算することを考える。内挿により強度値を求めたい位置が、図4の位置405、406、407、408となる確率をそれぞれ1/4とする。このとき、位置405の強度値 I_a を求める際の誤差は明らかに0である。また、位置406の強度値 I_b を求める際の誤差は、 $L_a + L_b$ が偶数の場合は0、奇数の場合は切り上げが行われるので1/2となる。 $L_a + L_b$ が偶数になる確率と奇数になる確率は共に1/2であるとすれば、誤差の期待値は、
 $0 \cdot 1/2 + 1/2 \cdot 1/2 = 1/4$ となる。位置407の強度値 I_c を求める際も誤差の期待値は I_b の場合と同様に1/4となる。位置408の強度値 I_d を求める際には、 $L_a + L_b + L_c + L_d$ を4で割った際のあまりが0、1、2、3である場合の誤差はそれぞれ0、 $-1/4$ 、 $1/2$ 、 $1/4$ となり、あまりが0から3になる確率をそれぞれ等確率とすれば、誤差の期待値は
 $0 \cdot 1/4 - 1/4 \cdot 1/4 + 1/2 \cdot 1/4 + 1/4 \cdot 1/4 = 1/8$ となる。上で述べた通り、位置405～408における強度値が計算される確率は等確率であるとすれば、最終的な誤差の期待値は、
 $0 \cdot 1/4 + 1/4 \cdot 1/4 + 1/4 \cdot 1/4 + 1/8 \cdot 1/4 = 5/32$ となる。これは、一回ブロックマッチングによる動き補償を行う度に、画素の強度値に5/32の誤差が発生することを意味している。

20

【0022】

一般的に低レート符号化の場合には、フレーム間予測誤差を符号化するためのビット数を十分に確保することができないため、DCT係数の量子化ステップサイズを大きくする傾向がある。したがって、動き補償で発生した誤差を誤差符号化によって修正しにくくなる。このようなときにフレーム内符号化を行わずにフレーム間符号化をずっと続けた場合には、上記誤差が蓄積し、再生画像が赤色化するなどの悪い影響を与える場合がある。

30

【0023】

上で説明した通り、色差プレーンの画素数は縦方向、横方向共に画素数が半分となっている。したがって、UブロックとVブロックに対しては、Yブロックの動きベクトルの水平・垂直成分をそれぞれ2で割った値が使用される。このとき、もとのYブロックの動きベクトルの水平・垂直成分である u と v が1/2の整数倍の値であるため、通常の割り算
 を実行した場合には、動きベクトルは1/4の整数倍の値が出現することになる。しかし、座標値が1/4の整数倍をとるときの強度値の内挿演算が複雑となるため、H.263ではUブロックとVブロックの動きベクトルも半画素精度に丸め込まれる。このときの丸め込みの方法は以下の通りである。

40

【0024】

いま、 $u/2 = r + s/4$ であるとする。このとき、 r と s は整数であり、さらに s は0以上3以下の値をとるとする。 s が0または2のときは $u/2$ は1/2の整数倍であるため、丸め込みを行う必要がない。しかし、 s が1または3のときは、これを2に丸め込む操作が行われる。これは、 s が2となる確率を高くすることにより、強度値の内挿が行われる回数を増やし、動き補償処理にフィルタリングの作用を持たせるためである。

50

【 0 0 2 5 】

丸め込みが行われる前の s の値が 0 ~ 3 の値をとる確率をそれぞれ $1/4$ とした場合、丸め込みが終わったあとに s が 0、2 となる確率はそれぞれ $1/4$ と $3/4$ となる。以上は動きベクトルの水平成分 u に関する議論であったが、垂直成分である v に関しても全く同じ議論が適用できる。

【 0 0 2 6 】

したがって、UブロックおよびVブロックにおいて、401の位置の強度値が求められる確率は $1/4 \cdot 1/4 = 1/16$ 、402および403の位置の強度値が求められる確率は共に $1/4 \cdot 3/4 = 3/16$ 、404の位置の強度値が求められる確率は $3/4 \cdot 3/4 = 9/16$ となる。これを用いて上と同様の手法により、強度値の誤差の期待値を求めると、 $0 \cdot 1/16 + 1/4 \cdot 3/16 + 1/4 \cdot 3/16 + 1/8 \cdot 9/16 = 21/128$ となり、上で説明したYブロックの場合と同様にフレーム内符号化を続けた場合の誤差の蓄積の問題が発生する。

10

【 0 0 2 7 】

フレーム間予測を行い、輝度または色の強度が量子化された数値として表される動画像符号化および復号化方法では、フレーム間予測において輝度または色の強度を量子化する際の誤差が蓄積する場合がある。本発明の目的は、上記誤差の蓄積を防ぐことにより、再生画像の画質を向上させることにある。

【課題を解決するための手段】

【 0 0 2 8 】

誤差の発生を抑えるか、発生した誤差を打ち消す操作を行うことにより、誤差の蓄積を防ぐ。

20

【発明の効果】

【 0 0 2 9 】

本発明により、フレーム間予測画像における丸め込み誤差の蓄積を抑えることが可能となり、再生画像の画質を向上させることが可能となる。

【図面の簡単な説明】

【 0 0 3 0 】

【図1】H.263の画像符号化器の構成例を示した図である。

【図2】H.263の画像復号化器の構成例を示した図である。

30

【図3】H.263におけるマクロブロックの構成を示した図である。

【図4】半画素成度のブロックマッチングにおける輝度値の内挿処理の様子を示した図である。

【図5】符号化された画像系列の様子を示した図である。

【図6】ソフトウェア画像符号化装置の構成例を示した図である。

【図7】ソフトウェア画像復号化装置の構成例を示した図である。

【図8】ソフトウェア画像符号化装置における処理のフローチャートの例を示した図である。

【図9】ソフトウェア画像符号化装置における符号化モード決定処理のフローチャートの例を示した図である。

40

【図10】ソフトウェア画像符号化装置における動き推定・動き補償処理のフローチャートの例を示した図である。

【図11】ソフトウェア画像復号化装置における処理のフローチャートの例を示した図である。

【図12】ソフトウェア画像復号化装置における動き補償処理のフローチャートの例を示した図である。

【図13】IフレームとP+フレームとP-フレームを混在させる符号化方法により符号化されたビットストリームを記録した蓄積メディアの例を示した図である。

【図14】P+フレームとP-フレームを混在させる符号化方法を使用する装置の具体例を示した図である。

50

【図15】IフレームとBフレームとP+フレームとP-フレームを混在させる符号化方法により符号化されたビットストリームを記録した蓄積メディアの例を示した図である。

【図16】P+フレームとP-フレームを混在させる符号化方法を使用する装置に含まれるブロックマッチング部の例を示した図である。

【図17】P+フレームとP-フレームを混在させる符号化方法により符号化されたビットストリームを復号化する装置に含まれる予測画像合成部の例を示した図である。

【発明を実施するための形態】

【0031】

まず、「従来の技術」で述べた丸め込み誤差の蓄積がどのような場合に発生するかについて考える。

【0032】

図5にMPEG1、MPEG2、H.263などの双方向予測と片方向予測の両方を実行することができる符号化方法により符号化された動画像の例を示す。画像501はフレーム内符号化によって符号化されたフレームであり、Iフレームと呼ばれる。これに対し、画像503、505、507、509はPフレームと呼ばれ、直前のIまたはPフレームを参照画像とする片方向のフレーム間符号化により符号化される。したがって、例えば画像505を符号化するには画像503を参照画像とするフレーム間予測が行われる。画像502、504、506、508はBフレームと呼ばれ、直前と直後のIまたはPフレームを用いた双方向のフレーム間予測が行われる。Bフレームは、他のフレームがフレーム間予測を行う際に参照画像として利用されないという特徴も持っている。

【0033】

まず、Iフレームでは動き補償が行われなため、動き補償が原因となる丸め込み誤差は発生しない。これに対し、Pフレームでは動き補償が行われる上に、他のPまたはBフレームの参照画像としても使用されるため、丸め込み誤差の蓄積を引き起こす原因となる。一方、Bフレームは動き補償は行われるために丸め込み誤差の蓄積の影響は現れるが、参照画像としては使用されないために丸め込み誤差の蓄積の原因とはならない。このことから、Pフレームにおける丸め込み誤差の蓄積を防げば、動画像全体で丸め込み誤差の悪影響を緩和することができる。なお、H.263ではPフレームとBフレームをまとめて符号化するPBフレームと呼ばれるフレームが存在するが（例えばフレーム503と504をPBフレームとしてまとめて符号化することができる）、組み合わせられた2枚のフレームを別々の物として考えれば、上と同じ議論を適用することができる。つまり、PBフレームの中でPフレームに相当する部分に対して丸め込み誤差への対策を施せば、誤差の蓄積を防ぐことができる。

【0034】

丸め込み誤差は、強度値の内挿を行う際に、通常除算（演算結果が実数になる除算）の結果として整数値に0.5を加えた値が出るような場合に、これを0から遠ざける方向に切り上げているために発生している。例えば内挿された強度値を求めるために4で割る操作を行うような場合、あまりが1である場合と3である場合は発生する誤差の絶対値が等しくかつ符号が逆になるため、誤差の期待値を計算する際に互いに打ち消し合う働きをする（より一般的には、正の整数 d' で割る場合には、あまりが t である場合と $d' - t$ である場合が打ち消し合う）。しかし、あまりが2である場合、つまり通常除算の結果が整数に0.5を加えた値が出る場合には、これを打ち消すことができず、誤差の蓄積につながる。

【0035】

そこで、このように通常除算の結果、整数に0.5を加えた値が出た際に切り上げを行う丸め込み方法と切り捨てを行う丸め込み方法の両者を選択可能とし、これらをうまく組み合わせることより、発生した誤差を打ち消すことを考える。以下では、通常除算の結果を最も近い整数に丸め込み、かつ整数に0.5を加えた値は0から遠ざける方向に切り上げる丸め込み方法を「プラスの丸め込み」と呼ぶ。また、通常除算の結果を最も近い整数に丸め込み、かつ整数に0.5を加えた値は0に近づける方向に切り捨てる丸め込み方

10

20

30

40

50

法を「マイナスの丸め込み」と呼ぶこととする。数3は、半画素精度のブロックマッチングにおいてプラスの丸め込みを行う場合の処理を示しているが、マイナスの丸め込みを行う場合には、これは以下のように書き換えることができる。

【0036】

【数4】

$$\begin{aligned} I_a &= L_a \\ I_b &= [(L_a + L_b) / 2] \\ I_c &= [(L_a + L_c) / 2] \\ I_d &= [(L_a + L_b + L_c + L_d + 1) / 4] \end{aligned} \quad \dots \text{ (数4)}$$

10

【0037】

いま、予測画像の合成における強度値の内挿の際にプラスの丸め込みを行う動き補償を、プラスの丸め込みを用いる動き補償、マイナスの丸め込みを行う動き補償をマイナスの丸め込みを用いる動き補償とする。また、半画素精度のブロックマッチングを行い、かつプラスの丸め込みを用いる動き補償が適用されるPフレームをP+フレーム、逆にマイナスの丸め込みを用いる動き補償が適用されるPフレームをP-フレームと呼ぶこととする（この場合、H.263のPフレームはすべてP+フレームということになる）。P-フレームにおける丸め込み誤差の期待値は、P+フレームのそれと絶対値が等しく、符号が逆となる。したがって、時間軸に対し、P+フレームとP-フレームが交互に現れるようにすれば、丸め込み誤差の蓄積を防ぐことができる。

20

【0038】

図5の例では、フレーム503、507をP+フレーム、フレーム505、509をP-フレームとすれば、この処理を実現することができる。また、P+フレームとP-フレームが交互に発生することは、Bフレームにおいて双方向の予測を行う際にP+フレームとP-フレームが一枚ずつ参照画像として使用されることを意味している。一般的にBフレームにおいては順方向の予測画像（例えば図5のフレーム504を符号化する際に、フレーム503を参照画像として合成される予測画像）と逆方向の予測画像（例えば図5のフレーム504を符号化する際に、フレーム505を参照画像として合成される予測画像）の平均が予測画像として使用できる場合が多い。したがって、ここでP+フレームとP-フレームから合成した画像を平均化することは、誤差の影響を打ち消す意味で有効である。

30

【0039】

なお、上で述べた通り、Bフレームにおける丸め込み処理は誤差の蓄積の原因とはならない。したがって、すべてのBフレームに対して同じ丸め込み方法を適用しても問題は発生しない。例えば、図5のBフレーム502、504、506、508のすべてが正の丸め込みに基づく動き補償を行ったとしても、特に画質の劣化の原因とはならない。Bフレームの復号化処理を簡略化する意味では、Bフレームに関しては1種類の丸め込み方法のみを用いることが望ましい。

40

【0040】

図16に、上で述べた複数の丸め込み方法に対応した画像符号化器のブロックマッチング部1600の例を示す。他の図と同じ番号は、同じものを指している。図1のブロックマッチング部116を1600に入れ換えることにより、複数の丸め込み方法に対応することができる。動き推定器1601において、入力画像101と前フレームの復号画像112との間で動き推定の処理が行われる。この結果、動き情報120が出力される。この動き情報は、予測画像合成器1603において予測画像を合成する際に利用される。

50

【 0 0 4 1 】

丸め込み方法決定器 1 6 0 2 は、現在符号化を行っているフレームにおいて使用する丸め込み方法を正の丸め込みとするか、負の丸め込みとするかを判定する。決定した丸め込み方法に関する情報 1 6 0 4 は、予測画像合成器 1 6 0 3 に入力される。この予測画像合成器では、1 6 0 4 によって指定された丸め込み方法に基づいて予測画像 1 1 7 が合成、出力される。なお、図 1 のブロックマッチング部 1 1 6 には、図 1 6 の 1 6 0 2、1 6 0 4 に相当する部分が無く、予測画像は、正の丸め込みによってのみ合成される。また、ブロックマッチング部から決定した丸め込み方法 1 6 0 5 を出力し、この情報をさらに多重化して伝送ビットストリームに組み込んで伝送しても良い。

【 0 0 4 2 】

図 1 7 に、複数の丸め込み方法に対応した画像復号化器の予測画像合成部 1 7 0 0 の例を示す。他の図と同じ番号は、同じものを指している。図 2 の予測画像合成部 2 1 1 を 1 7 0 0 に入れ換えることにより、複数の丸め込み方法に対応することが可能となる。丸め込み方法決定器 1 7 0 1 では、復号化を行う際の予測画像合成処理に適用される丸め込み方法が決定される。

【 0 0 4 3 】

なお、正しい復号化を行うためには、ここで決定される丸め込み方法は、符号化の際に適用された丸め込み方法と同じものでなければならない。例えば、最後に符号化された I フレームから数えて奇数番目の P フレームには正の丸め込み、偶数番目の P フレームに対しては負の丸め込みが適用されることを原則とし、符号化側の丸め込み方法決定器（例えば、図 1 6 の 1 6 0 2）と復号化側の丸め込み方法決定器 1 7 0 1 の両者がこの原則に従えば、正しい復号化を行うことが可能となる。このようにして決定された丸め込み方法に関する情報 1 7 0 2 と、前フレームの復号画像 2 1 0、動き情報 2 0 2 から、予測画像合成器 1 7 0 3 では、予測画像が合成される。この予測画像 2 1 2 は出力され、復号画像の合成に活用される。

【 0 0 4 4 】

なお、ビットストリーム内に丸め込み方法に関する情報が組み込まれる場合（図 1 6 の符号化器で、丸め込み方法に関する情報 1 6 0 5 が出力されるような場合）も考えることができる。この場合、丸め込み方法決定器 1 7 0 1 は使用されず、符号化ビットストリームから抽出された丸め込み方法に関する情報 1 7 0 4 が予測画像合成器 1 7 0 3 に入力される。

【 0 0 4 5 】

本発明は、図 1、2 に示されている従来型の専用回路・専用チップを用いる画像符号化装置、画像復号化装置の他に、汎用プロセッサを用いるソフトウェア画像符号化装置、ソフトウェア画像復号化装置にも適用することができる。図 6 と 7 にこのソフトウェア画像符号化装置 6 0 0 とソフトウェア画像復号化装置 7 0 0 の例を示す。ソフトウェア符号化器 6 0 0 では、まず入力画像 6 0 1 は入力フレームメモリ 6 0 2 に蓄えられ、汎用プロセッサ 6 0 3 はここから情報を読み込んで符号化の処理を行う。この汎用プロセッサを駆動するためのプログラムはハードディスクやフロッピー（登録商標）ディスクなどによる蓄積デバイス 6 0 8 から読み出されてプログラム用メモリ 6 0 4 に蓄えられる。また、汎用プロセッサは処理用メモリ 6 0 5 を活用して符号化の処理を行う。汎用プロセッサが出力する符号化情報は一旦出力バッファ 6 0 6 に蓄えられた後に符号化ビットストリーム 6 0 7 として出力される。

【 0 0 4 6 】

図 6 に示したソフトウェア符号化器上で動作する符号化ソフトウェア（コンピュータ読み取り可能な記録媒体）のフローチャートの例を図 8 に示す。まず 8 0 1 で処理が開始され、8 0 2 で変数 N に 0 が代入される。続いて 8 0 3、8 0 4 で N の値が 1 0 0 である場合には、0 が代入される。N はフレーム数のカウンタであり、1 枚のフレームの処理が終了する度に 1 が加算され、符号化を行う際には 0 ~ 9 9 の値をとることが許される。N の値が 0 であるときには符号化中のフレームは I フレームであり、奇数のときには P + フレ

10

20

30

40

50

ーム、0以外の偶数のときにはP - フレームとなる。Nの値の上限が99であることは、Pフレーム（P + またはP - フレーム）が99枚符号化された後にIフレームが1枚符号化されることを意味している。

【0047】

このように、何枚かのフレームの中に必ず1枚Iフレームを入れることにより、(a)符号化器と復号化器の処理の不一致（例えば、DC Tの演算結果の不一致）による誤差の蓄積を防止する、(b)符号化データから任意のフレームの再生画像を得る処理（ランダムアクセス）の処理量を減少させる、などの効果を得ることができる。Nの最適な値は符号化器の性能や符号化器が使用される環境により変化する。この例では100という値を使用した、これはNの値が必ず100でなければならないことを意味しているわけではない。

10

【0048】

フレームごとの符号化モード、丸め込み方法を決定する処理は805で行われるが、その処理の詳細を表すフローチャートの例を図9に示す。まず、901でNは0であるか否かが判定され、0である場合には902で予測モードの識別情報として' I 'が出力バッファに出力され、これから符号化処理を行うフレームはIフレームとなる。なお、ここで「出力バッファに出力される」とは、出力バッファに蓄えられた後に符号化ビットストリームの一部として符号化装置から外部に出力されることを意味している。Nが0ではない場合には、903で予測モードの識別情報として' P 'が出力される。Nが0ではない場合には、さらに904でNが奇数か偶数であるかが判定される。Nが奇数の場合には905で丸め込み方法の識別情報として' + 'が出力され、これから符号化処理を行うフレームはP + フレームとなる。一方、Nが偶数の場合には906で丸め込み方法の識別情報として' - 'が出力され、これから符号化処理を行うフレームはP - フレームとなる。

20

【0049】

再び図8に戻る。805で符号化モードを決定した後、806で入力画像はフレームメモリAに蓄えられる。なお、ここで述べたフレームメモリAとは、ソフトウェア符号化器のメモリ領域（例えば、図6の605のメモリ内にこのメモリ領域が確保される）の一部を意味している。807では、現在符号化中のフレームがIフレームであるか否かが判定される。そして、Iフレームではない場合には808で動き推定・動き補償処理が行われる。

30

【0050】

この808における処理の詳細を表すフローチャートの例を図10に示す。まず、1001でフレームメモリAとB（本段落の最後に書かれている通り、フレームメモリBには前フレームの復号画像が格納されている）に蓄えられた画像の間でブロックごとに動き推定の処理が行われ、各ブロックの動きベクトルが求められ、その動きベクトルは出力バッファに出力される。続いて1002で現フレームがP + フレームであるか否かが判定され、P + フレームである場合には1003で正の丸め込みを用いて予測画像が合成され、この予測画像はフレームメモリCに蓄えられる。一方、現フレームがP - フレームである場合には1004で負の丸め込みを用いて予測画像が合成され、この予測画像がフレームメモリCに蓄えられる。そして1005ではフレームメモリAとCの差分画像が求められ、これがフレームメモリAに蓄えられる。

40

【0051】

ここで再び図8に戻る。809における処理が開始される直前、フレームメモリAには、現フレームがIフレームである場合には入力画像が、現フレームがPフレーム（P + またはP - フレーム）である場合には入力画像と予測画像の差分画像が蓄えられている。809では、このフレームメモリAに蓄えられた画像に対してDC Tが適用され、ここで計算されたDC T係数は量子化された後に出力バッファに出力される。そしてさらに810で、この量子化DC T係数には逆量子化され、逆DC Tが適用され、この結果得られた画像はフレームメモリBに格納される。続いて811では、再び現フレームがIフレームであるか否かが判定され、Iフレームではない場合には812でフレームメモリBとCの画

50

像が加算され、この結果がフレームメモリBに格納される。ここで、1フレーム分の符号化処理が終了することになる。

【0052】

そして、813の処理が行われる直前にフレームメモリBに格納されている画像は、符号化処理が終了したばかりのフレームの再生画像（復号側で得られるものと同じ）である。813では、符号化が終了したフレームが最後のフレームであるか否かが判定され、最後のフレームであれば、符号化処理が終了する。最後のフレームではない場合には、814でNに1が加算され、再び803に戻って次のフレームの符号化処理が開始される。

【0053】

図7にソフトウェア復号化器700の例を示す。入力された符号化ビットストリーム701は一旦入力バッファ702に蓄えられた後に汎用プロセッサ703に読み込まれる。汎用プロセッサはハードディスクやフロッピー（登録商標）ディスクなどによる蓄積デバイス708から読み出されたプログラムを蓄えるプログラム用メモリ704、および処理用メモリ705を活用して復号化処理を行う。この結果得られた復号化画像は一旦出力フレームメモリ706に蓄えられた後に出力画像707として出力される。

10

【0054】

図7に示したソフトウェア復号化器上で動作する復号化ソフトウェアのフローチャートの例を図11に示す。1101で処理が開始され、まず1102で入力情報があるか否かが判定される。ここで入力情報が無ければ1103で復号化の処理を終了する。入力情報がある場合には、まず、1104で符号化識別情報が入力される。なお、この「入力される」とは、入力バッファ（例えば、図7の702）に蓄えられた情報を読み込むことを意味している。1105では、読み込んだ符号化モード識別情報が「I」であるか否かが判定される。そして、「I」ではない場合には、1106で丸め込み方法の識別情報が入力され、続いて1107で動き補償処理が行われる。

20

【0055】

この1107で行われる処理の詳細を表したフローチャートの例を図12に示す。まず、1201でブロックごとの動きベクトル情報が入力される。そして、1202で1106で読み込まれた丸め込み方法の識別情報が「+」であるか否かが判定される。これが「+」である場合には、現在復号化中のフレームがP+フレームである。このとき1203で正の丸め込みにより予測画像が合成され、この予測画像はフレームメモリDに格納される。

30

【0056】

なお、ここで述べたフレームメモリDとは、ソフトウェア復号化器のメモリ領域（例えば、図7の705のメモリ内にこのメモリ領域が確保される）の一部を意味している。一方、丸め込み方法の識別情報が「+」ではない場合には、現在復号化中のフレームがP-フレームであり、1204で負の丸め込みにより予測画像が合成され、この予測画像はフレームメモリDに格納される。このとき、もし何らかの誤りにより、P+フレームがP-フレームとして復号化されたり、逆にP-フレームがP+フレームとして復号化された場合には、符号化器が意図したものと異なる予測画像が復号化器において合成されることになり、正しい復号化が行われずに画質が劣化する。

40

【0057】

ここで図11に戻る。1108では量子化DCT係数が入力され、これに逆量子化、逆DCTを適用して得られた画像がフレームメモリEに格納される。1109では、再び現在復号化中のフレームがIフレームであるか否かが判定される。そして、Iフレームではない場合には、1110でフレームメモリDとEに格納された画像が加算され、この結果の画像がフレームメモリEに格納される。1111の処理を行う直前にフレームメモリEに格納されている画像が、再生画像となる。1111では、このフレームメモリEに格納された画像が出力フレームメモリ（例えば、図7の706）に出力され、そのまま出力画像として復号化器から出力される。こうして1フレーム分の復号化処理が終了し、処理は再び1102に戻る。

50

【 0 0 5 8 】

図 6 と 7 に示したソフトウェア画像符号化器、ソフトウェア画像復号化器に図 8 ~ 1 2 に示したフローチャートに基づくプログラムを実行させると、専用回路・専用チップを用いる装置を使用した場合と同様の効果を得ることができる。

【 0 0 5 9 】

図 6 のソフトウェア符号化器 6 0 1 が図 8 ~ 1 0 のフローチャートに示した処理を行うことにより生成されたビットストリームを記録した蓄積メディア(記録媒体)の例を図 1 3 に示す。デジタル情報を記録することができる記録ディスク(例えば磁気、光ディスクなど) 1 3 0 1 には、同心円上にデジタル情報が記録されている。このディスクに記録されているデジタル情報の一部 1 3 0 2 を取り出すと、符号化されたフレームの符号化モード識別情報 1 3 0 3、1 3 0 5、1 3 0 8、1 3 1 1、1 3 1 4、丸め込み方法の識別情報 1 3 0 6、1 3 0 9、1 3 1 2、1 3 1 5、動きベクトルや D C T 係数等の情報 1 3 0 4、1 3 0 7、1 3 1 0、1 3 1 3、1 3 1 6 が記録されている。図 8 ~ 1 0 に示した方法に従えば、1 3 0 3 には ' I '、1 3 0 5、1 3 0 8、1 3 1 1、1 3 1 4 には ' P '、1 3 0 6、1 3 1 2 には ' + '、1 3 0 9、1 3 1 5 には ' - ' を意味する情報が記録されることとなる。この場合、例えば ' I ' と ' + ' は 1 ビットの 0、' P ' と ' - ' は 1 ビットの 1 で表せば、復号化器は正しく記録された情報を解釈し、再生画像を得ることが可能となる。このようにして蓄積メディアに符号化ビットストリームを蓄積することにより、このビットストリームを読み出して復号化した場合に丸め込み誤差の蓄積が発生することを防ぐことができる。

【 0 0 6 0 】

図 5 に示した P + フレーム、P - フレーム、B フレームが存在する画像系列に関する符号化ビットストリームを記録した蓄積メディアの例を図 1 5 に示す。図 1 3 の 1 3 0 1 と同様に、デジタル情報を記録することができる記録ディスク(例えば磁気、光ディスクなど) 1 5 0 1 には、同心円上にデジタル情報が記録されている。このディスクに記録されているデジタル情報の一部 1 5 0 2 を取り出すと、符号化されたフレームの符号化モード識別情報 1 5 0 3、1 5 0 5、1 5 0 8、1 5 1 0、1 5 1 3、丸め込み方法の識別情報 1 5 0 6、1 5 1 2、動きベクトルや D C T 係数等の情報 1 5 0 4、1 5 0 7、1 5 0 9、1 5 1 1、1 5 1 4 が記録されている。

【 0 0 6 1 】

このとき、1 5 0 3 には ' I '、1 5 0 5、1 5 1 0 には ' P '、1 5 0 8、1 5 1 3 には ' B '、1 5 0 5 には ' + '、1 5 1 1 には ' - ' を意味する情報が記録されている。例えば ' I '、' P '、' B ' をそれぞれ 2 ビットの 0 0、0 1、1 0、' + ' と ' - ' はそれぞれ 1 ビットの 0 と 1 で表せば、復号化器は正しく記録された情報を解釈し、再生画像を得ることが可能となる。

【 0 0 6 2 】

このとき図 5 の 5 0 1 (I フレーム) に関する情報が 1 5 0 3 と 1 5 0 4、5 0 2 (B フレーム) に関する情報が 1 5 0 8 と 1 5 0 9、フレーム 5 0 3 (P + フレーム) に関する情報が 1 5 0 5 ~ 1 5 0 7、フレーム 5 0 4 (B フレーム) に関する情報が 1 5 1 3 と 1 5 1 4、フレーム 5 0 5 (P - フレーム) に関する情報が 1 5 1 0 ~ 1 5 1 2 である。このように動画像を B フレームを含む形で符号化場合、一般的にフレームに関する情報を伝送する順番と、再生する順番は異なる。これは、ある B フレームを復号化する前に、この B フレームが予測画像を合成する際に使用する前後の参照画像を復号化しておかなければならないためである。このため、フレーム 5 0 2 はフレーム 5 0 3 の前に再生されるにもかかわらず、フレーム 5 0 2 が参照画像として使用するフレーム 5 0 3 に関する情報がフレーム 5 0 2 に関する情報の前に伝送されるのである。

【 0 0 6 3 】

上述の通り、B フレームは丸め込み誤差の蓄積を引き起こす要員とはならないため、P フレームのように複数の丸め込み方法を適用する必要はない。このため、ここに示した例では、B フレームに関しては丸め込み方法を指定する ' + ' や ' - ' のような情報は伝

10

20

30

40

50

送されていない。こうすることにより、例えばBフレームに関しては常に正の丸め込みのみが適用されるようにしたとしても、誤差の蓄積の問題は発生しない。このようにして、蓄積メディアにBフレームに関する情報を含む符号化ビットストリームを蓄積することにより、このビットストリームを読み出して復号化した場合に丸め込み誤差の蓄積が発生することを防ぐことができる。

【0064】

図14に、本明細書で示したP+フレームとP-フレームが混在する符号化方法に基づく符号化・復号化装置の具体例を示す。パソコン1401に画像符号化・復号化用のソフトウェアを組み込むことにより、画像符号化・復号化装置として活用することが可能である。このソフトウェアはコンピュータ読み取り可能な記録媒体である何らかの蓄積メディア(CD-ROM、フロッピー(登録商標)ディスク、ハードディスクなど)1412に記録されており、これをパソコンが読み込んで使用する。また、さらに何らかの通信回線にこのパソコンを接続することにより、映像通信端末として活用することも可能となる。

10

【0065】

記録媒体である蓄積メディア1402に記録した符号化ビットストリームを読み取り、復号化する再生装置1403にも本明細書に示した復号化方法を実装することが可能である。この場合、再生された映像信号はテレビモニタ1404に表示される。また、1403の装置は符号化ビットストリームを読み取るだけであり、テレビモニタ1404内に復号化装置が組み込まれている場合も考えられる。

20

【0066】

最近では衛星、地上波によるデジタル放送が話題となっているが、デジタル放送用のテレビ受信機1405にも復号化装置を組み込むことができる。

【0067】

また、ケーブルテレビ用のケーブル1408または衛星/地上波放送のアンテナに接続されたセットトップボックス1409内に復号化装置を実装し、これをテレビモニタ1410で再生する構成も考えられる。このときも1404の場合と同様に、セットトップボックスではなく、テレビモニタ内に符号化装置を組み込んで良い。

【0068】

1413、1414、1415は、デジタル衛星放送システムの構成例を示したものである。放送局1413では映像情報の符号化ビットストリームが電波を介して通信または放送衛星1414に伝送される。これを受けた衛星は、放送用の電波を発信し、この電波を衛星放送受信設備をもつ家庭1415が受信し、テレビ受信機またはセットトップボックスなどの装置により符号化ビットストリームを復号化してこれを再生する。

30

【0069】

低い伝送レートでの符号化が可能となったことにより、最近ではデジタル携帯端末1406によるデジタル動画像通信も注目されるようになっている。デジタル携帯端末の場合、符号器・復号器を両方持つ送受信型の端末の他に、符号化器のみの送信端末、復号器のみの受信端末の3通りの実装形式が考えられる。

【0070】

動画像撮影用のカメラ1407の中に符号化装置を組み込むことも可能である。この場合撮影用カメラは符号化装置と該符号化装置からの出力を記録媒体に記録する記録装置とを持ち、符号化装置から出力された符号化ビットストリームを記録媒体に記録する。また、カメラは映像信号を取り込むのみであり、これを専用の符号化装置1411に組み込む構成も考えられる。

40

【0071】

この図に示したいずれの装置・システムに関しても、本明細書に示した方法を実装することにより、従来技術を活用した場合と比較して、より画質の高い画像情報を扱うことが可能となる。

【0072】

なお、以下の変形も本発明に含まれることは明らかである。

50

(1) 上の議論では、動き補償方式としてブロックマッチングが使用されることが前提となっていた。しかし、本発明は動きベクトルの水平・垂直成分が水平・垂直方向の画素のサンプリング間隔の整数倍以外値をとることができ、サンプル値の存在しない位置における強度値を共1次内挿によって求める動き補償方式を採用する画像符号化方式および画像復号化方式すべてに対して適用することができる。たとえば特願平08-060572に記載されているグローバル動き補償や、特願平08-249601に記載されているワーピング予測に対しても、本発明は適用可能である。

(2) これまでの議論では、動きベクトルの水平・垂直成分が1/2の整数倍の値をとる場合のみについて議論してきた。しかし、議論を一般化すれば、本発明は動きベクトルの水平・垂直成分が1/dの整数倍(dは正の整数、かつ偶数)をとる方式に対して適用可能である。しかし、dが大きくなった場合には、共1次内挿の除算の除数(dの2乗、数2参照)が大きくなるため、相対的に通常の除算の結果が整数に0.5を足した値となる確率が低くなる。したがって、プラスの丸め込みのみを行った場合の、丸め込み誤算の期待値の絶対値が小さくなり、誤差の蓄積による悪影響が目立ちにくくなる。そこで、例えばdの値が可変である動き補償方式などにおいては、dがある一定値より小さい場合にはプラスの丸め込みとマイナスの丸め込みの両方を使用し、dが上記一定値以上の場合にはプラスまたはマイナスの丸め込みのみを用いるという方法も有効である。

(3) 従来技術で述べた通り、DCTを誤差符号化方式として利用した場合、丸め込み誤差の蓄積による悪影響はDCT係数の量子化ステップサイズが大きい場合に現れやすい。そこで、DCT係数の量子化ステップサイズがある一定値より大きい場合にはプラスの丸め込みとマイナスの丸め込みの両方を使用し、DCT係数の量子化ステップサイズが上記一定値以下の場合にはプラスまたはマイナスの丸め込みのみを用いるという方法も有効である。

(4) 輝度プレーンで丸め込み誤差の蓄積が起こった場合と色差プレーンで丸め込み誤差の蓄積が起こった場合では、一般的に色差プレーンで発生した場合の方が再生画像に与える影響が深刻である。これは、画像が全体的にわずかに明るくなったり暗くなったりすることよりも、画像の色が全体的に変化した場合の方が目立ちやすいためである。そこで、色差信号に対してはプラスの丸め込みとマイナスの丸め込みの両方を使用し、輝度信号に対してはプラスまたはマイナスの丸め込みのみを用いるという方法も有効である。

【0073】

また、従来技術でH.263における1/4画素精度の動きベクトルの1/2画素精度の動きベクトルへの丸め込み方法に関して述べたが、この方法に多少の変更を加えることにより、丸め込み誤差の期待値の絶対値を小さくすることが可能である。従来技術でとりあげたH.263では、輝度プレーンの動きベクトルの水平成分または垂直成分を半分にした値が $r + s/4$ (rは整数、sは0以上4未満の整数)で表されるとして、sが1または3であるときに、これを2に丸め込む操作がおこなわれる。これをsが1のときにはこれを0とし、sが3であるときにはrに1を加えてsを0とする丸め込みを行うように変更すればよい。こうすることにより、図4の406~408の位置の強度値を計算する回数が相対的に減少する(動きベクトルの水平・垂直成分が整数となる確率が高くなる)ため、丸め込み誤差の期待値の絶対値が小さくなる。しかし、この方法では発生する誤差の大きさを抑えることはできても、誤差が蓄積することを防ぐことはできない。

(5) Pフレームに対して、2種類の動き補償方式によるフレーム間予測画像の平均を最終的なフレーム間予測画像とする方式がある。例えば特願平8-3616では、縦横16画素のブロックに対して一個の動きベクトルを割り当てるブロックマッチングと、縦横16画素のブロックを4個の縦横8画素のブロックに分割して、それぞれに対して動きベクトルを割り当てるブロックマッチングの2種類の方法によって得た2種類のフレーム間予測画像を用意し、これらのフレーム間予測画像の強度値の平均を求めたものを最終的なフレーム間予測画像とする方法が述べられている。この方法において2種類の画像の平均値を求める際にも丸め込みが行われる。この平均化の操作でプラスの丸め込みのみを行い続けると、新たな丸め込み誤差の蓄積の原因を作ることになる。この方式では、ブロックマッ

10

20

30

40

50

グにおいてプラスの丸め込みを行う P + フレームに対しては、平均化の操作ではマイナスの丸め込みを行い、P - フレームに対しては平均化の操作ではプラスの丸め込みを行うようにすれば、同一フレーム内でブロックマッチングによる丸め込み誤差と平均化による丸め込み誤差が打ち消し合う効果を得ることができる。(6) P + フレームと P - フレームを交互に配置する方法を用いた場合、符号化装置と復号化装置は現在符号化している P フレームが P + フレームである P - フレームであるかを判定するために、例えば以下の処理を行なうことが考えられる。現在符号化または復号化している P フレームが、最も最近に符号化または復号化された I フレームの後の何番目の P フレームであるかを数え、これが奇数であるときには P + フレーム、偶数であるときは P - フレームとすれば良い(これを暗示的方法と呼ぶ)。また、符号化装置側が現在符号化している P フレームが P + フレームであるか、P - フレームであるかを識別する情報を、例えばフレーム情報のヘッダ部分に書き込むという方法もある(これを明示的方法と呼ぶ)。この方法の方が、伝送誤りに対する耐性は強い。

10

20

30

40

50

【0074】

また、P + フレームと、P - フレームを識別する情報をフレーム情報のヘッダ部分に書き込む方法には、以下の長所がある。「従来の技術」で述べた通り、過去の符号化標準(例えば MPEG - 1 や MPEG - 2)では、P フレームにおいて正の丸め込みのみが行われる。したがって、例えば既に市場に存在している MPEG - 1 / 2 用の動き推定・動き補償装置(例えば、図1の106に相当する部分)は、P + フレームと P - フレームが混在する符号化には対応できないことになる。いま、P + フレームと P - フレームが混在する符号化に対応した復号化器があるとすると、この場合に、もしこの復号化器が上記暗示的方法に基づくものであれば、MPEG - 1 / 2 用の動き推定・動き補償装置を用いて、この暗示的方法に基づく復号化器が正しく復号化できるようなビットストリームを生成する符号化器を作ることは困難である。

【0075】

しかし、復号化器が上記明示的方法に基づくものである場合には、この問題を解決することができる。MPEG - 1 / 2 用の動き推定・動き補償装置を使用した符号化器は、常に P + フレームを送り続け、これを示す識別情報をフレーム情報のヘッダに書き込み続ければ良い。こうすれば、明示的方法に基づく復号化器は、この符号化器が生成したビットストリームを正しく再生することができる。

【0076】

もちろん、この場合には P + フレームのみが存在するため、丸め込み誤差の蓄積は発生しやすくなる。しかし、この符号化器が DCT 係数の量子化ステップサイズとして小さい値のみを用いるもの(高レート符号化専用の符号化器)であるような場合には、誤差の蓄積は大きな問題とはならない。

【0077】

この過去の方式との互換性の問題以外にも、明示的方法にはさらに、(a) 高レート符号化専用の符号化器や、頻繁に I フレームを挿入することにより丸め込み誤差が発生しにくい符号化器は、正か負のどちらかの丸め込み方法のみを実装すれば良く、装置のコストを抑えることができる、(b) 上記の丸め込み誤差が発生しにくい符号化器は、P + または P - フレームのどちらか一方のみを送り続ければ良いため、現在符号化を行っているフレームを P + フレームとするか、P - フレームとするかの判定を行う必要がなく、処理を簡略化できる、といった長所がある。

(7) 本発明は、フレーム間予測画像に対し、丸め込み処理を伴うフィルタリングを行う場合にも適用することができる。例えば、動画像符号化の国際標準である H. 261 では、フレーム間予測画像において動きベクトルが 0 ではなかったブロック内の信号に対しては、低域通過型フィルタ(これをループフィルタと呼ぶ)が適用される。また、H. 263 では、ブロックの境界部に発生する不連続(いわゆるブロック歪み)を平滑化するためのフィルタを使用することができる。これらのフィルタでは、画素の強度値に対して重み付け平均化の処理が行われ、フィルタリング後の強度値に対して整数への丸め込みの操作

が行われる。ここでもプラスの丸め込みとマイナスの丸め込みを使い分けることにより、誤差の蓄積を防ぐことが可能である。

(8) $IP + P - P + P - \dots$ の他に、 $IP + P + P - P - P + P + \dots$ や、 $IP + P - P - P + P + \dots$ など、 $P +$ フレームと $P -$ フレームの混在の仕方には様々な方法が考えられる。例えば、それぞれ $1/2$ の確率で 0 と 1 が発生する乱数発生器を使用し、 0 が出れば $P +$ 、 1 が出れば $P -$ としても良い。いずれにせよ、一般的に $P +$ と $P -$ フレームが混在し、かつ一定時間内のそれぞれの存在確率の差が小さいほど、丸め込み誤差の蓄積は発生しにくくなる。また、符号化器に対し、任意の $P +$ フレームと $P -$ フレームの混在の仕方を許すような場合、符号化器と復号化器は(6)で示した暗示的方法に基づくものではなく、明示的方法に基づくものでなければならない。したがって、符号化器と復号化器に関してより柔軟な実装形態を許すという観点からは、明示的方法の方が有利となる。

(9) 本発明は、画素の存在しない点の強度値を求める方法を共1次内挿に限定するものではない。強度値の内挿方法は一般化すると、以下の式のように表すことができる。

【0078】

【数5】

$$R(x+r, y+s) = T \left(\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(r-j, s-k) R(x+j, y+k) \right) \quad \dots \text{ (数5)}$$

10

20

【0079】

ここで、 r 、 s は実数、 $h(r, s)$ は内挿のための実数の関数、 $T(z)$ は実数 z を整数に丸め込む関数であり、 $R(x, y)$ 、 x 、 y の定義は数4と同じである。 $T(z)$ が、プラスの丸め込みを表す関数である場合にはプラスの丸め込みを用いる動き補償、マイナスの丸め込みを表す関数である場合にはマイナスの丸め込みを用いる動き補償が行われる。この数5の形式で表すことのできる内挿方法に対しては、本発明を適用することが可能である。例えば $h(r, s)$ を、

【0080】

【数6】

$$h(r, s) = \begin{cases} (1 - |r|)(1 - |s|), & 0 \leq |r| \leq 1, 0 \leq |s| \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad \dots \text{ (数6)}$$

30

【0081】

のように定義すれば共1次内挿が行われる。しかし、例えば $h(r, s)$ を

【0082】

40

【数 7】

$$\begin{aligned}
 h(r, s) &= 1 - |r| - |s|, & 0 \leq |r| + |s| \leq 1, rs < 0, \\
 &1 - |r|, & |r| \geq |s|, |r| \leq 1, rs \geq 0, \\
 &1 - |s|, & |s| > |r|, |s| \leq 1, rs > 0, \\
 &0, & \text{otherwise.}
 \end{aligned}
 \quad \dots \text{ (数 7)}$$

10

【0083】

のように定義すれば、共1次内挿とは異なる内挿方法が実施されるが、この場合も本発明を適用することは可能である。

(10) 本発明は、誤差画像の符号化方法をDCTに限定するものではない、例えば、DCTではなく、ウェーブレット変換(例えば、M. Antonioni, et. al, "Image Coding Using Wavelet Transform", IEEE Trans. Image Processing, vol. 1, no.2, April 1992)や、ウォルシュアダマール変換(Walsh-Hadamard Transform)(例えば、A. N. Netravalli and B. G. Haskell, "Digital Pictures", Plenum Press, 1998)を使用した場合でも本発明は適用可能である。

20

【符号の説明】

【0084】

100...画像符号化器、101...入力画像、102...減算器、103...誤差画像、104...DCT変換器、105...DCT係数量子化器、106、201...量子化DCT係数、108、204...DCT係数逆量子化器、109、205...逆DCT変換器、110、206...復号誤差画像、111、207...加算器、112...現フレームの復号画像、113、215...フレーム間/フレーム内符号化切り換えスイッチの出力画像、114、209...フレームメモリ、115、210...前フレームの復号画像、116、1600...ブロックマッチング部、117、212...現フレームの予測画像、118、213...「0」信号、119、214...フレーム間/フレーム内符号化切り換えスイッチ、120、202...動きベクトル情報、121、203...フレーム間/フレーム内識別フラグ、122...多重化器、123...伝送ビットストリーム、200...画像復号化器、208...出力画像、211、1700...予測画像合成部、216...分離器、301...Yブロック、302...Uブロック、303...Vブロック、401~404...画素、405~408...共1次内挿により強度値を求める位置、501...Iフレーム、503、505、507、509...Pフレーム、502、504、506、508...Bフレーム、600...ソフトウェア画像符号化器、602...入力画像用フレームメモリ、603、703...汎用プロセッサ、604、704...プログラム用メモリ、605、705...処理用メモリ、606...出力バッファ、607、701...符号化ビットストリーム、608、708...蓄積デバイス、700...ソフトウェア画像復号化器、702...入力バッファ、706...出力画像用フレームメモリ。801~815、901~906、1001~1005、1101~1111、1201~1204...フローチャートの処理項目、1301、1402、1501...蓄積メディア、1302、1502...デジタル情報を記録したトラック、1303~1316、1503~1514...デジタル情報、1401...パソコン、1403...蓄積メディアの再生装置、1404、1410...テレビモニタ、1405...テレビ放送受信機、1406...無線携帯端末、1407...テレビカメラ、1408...ケーブルテレビ用のケーブル、1409...セットトップボックス、1411...画像符号化装置、1412...ソフトウェア情報を記録した蓄積メディア、1413...放送局、1414...通信または放送衛星、1415...衛星放送受信設備を持つ家庭、1601...動き推定器、1602、1701...丸め込み方法決定器、1604、1605、1702、1704...丸め込み方法に関する情報、1603、

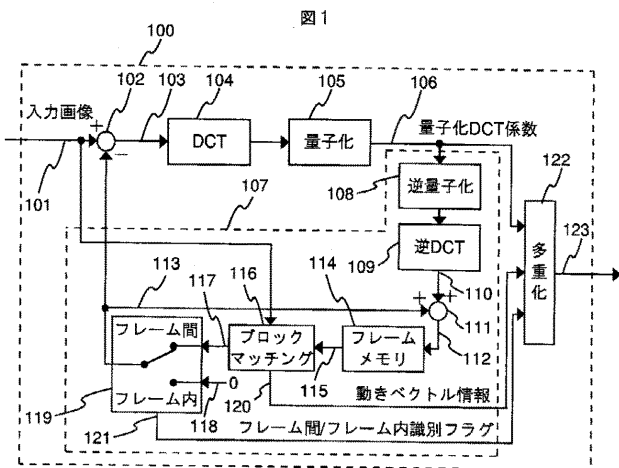
30

40

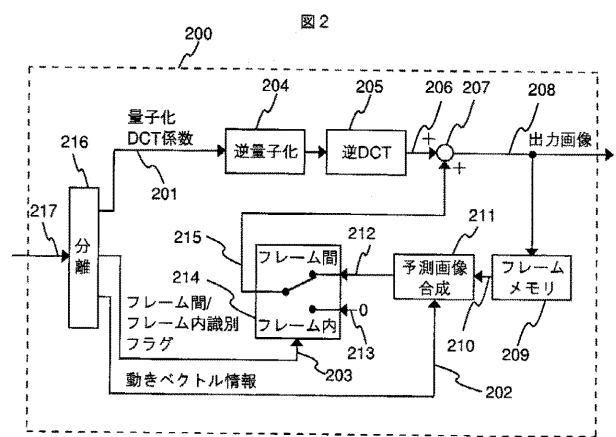
50

1703...予測画像合成器。

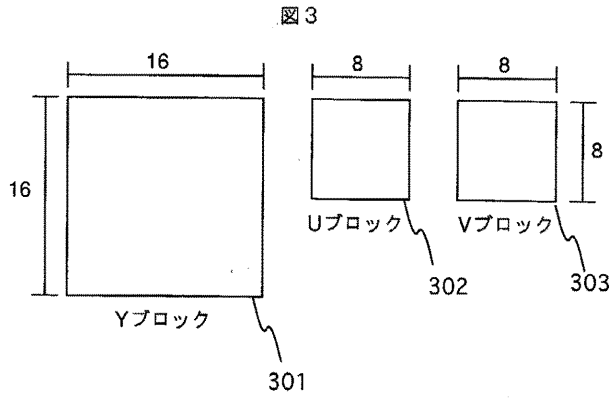
【図1】



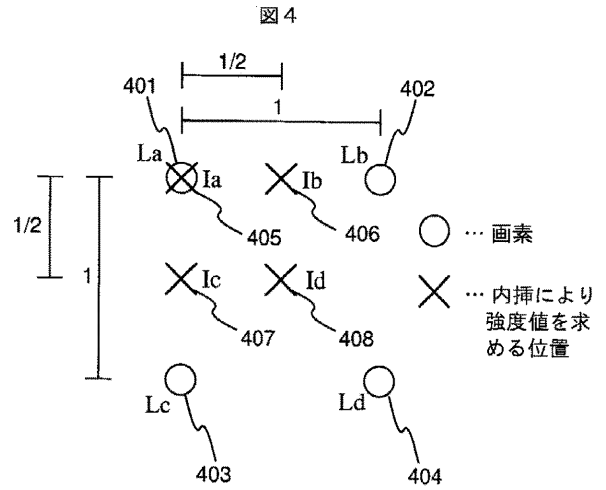
【図2】



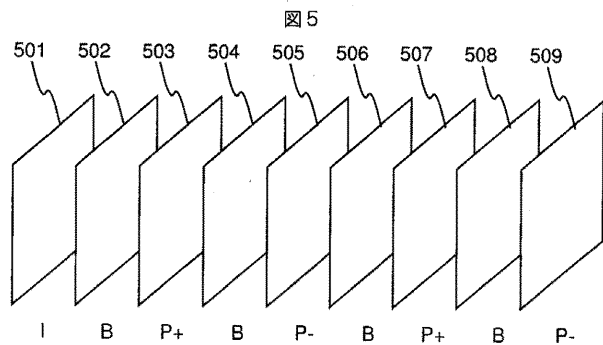
【 図 3 】



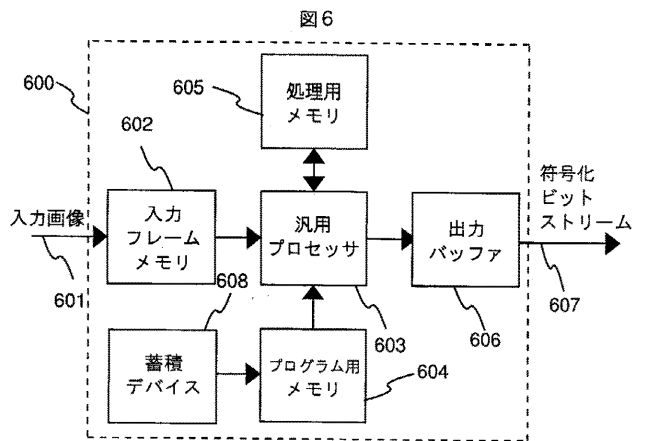
【 図 4 】



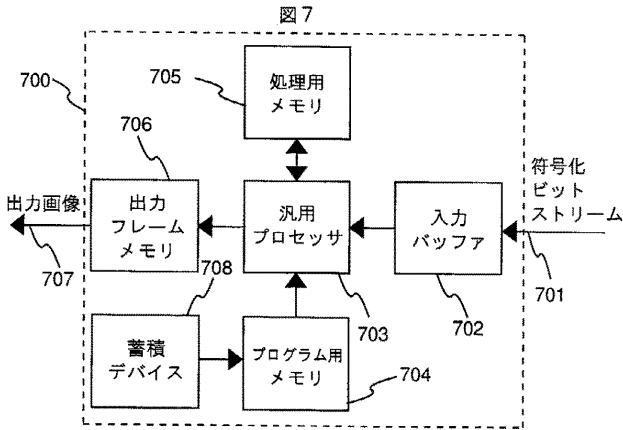
【 図 5 】



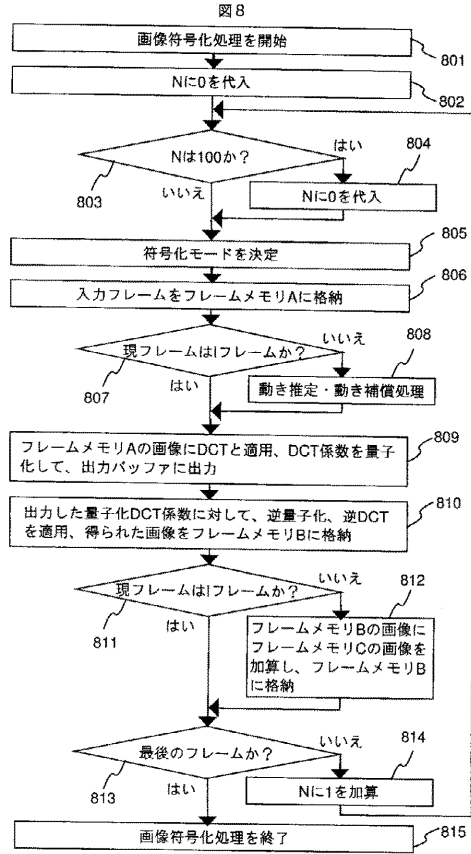
【 図 6 】



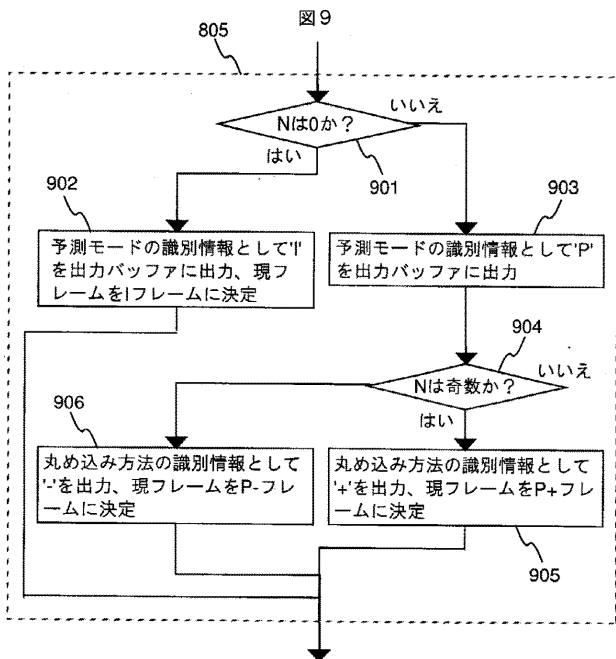
【 図 7 】



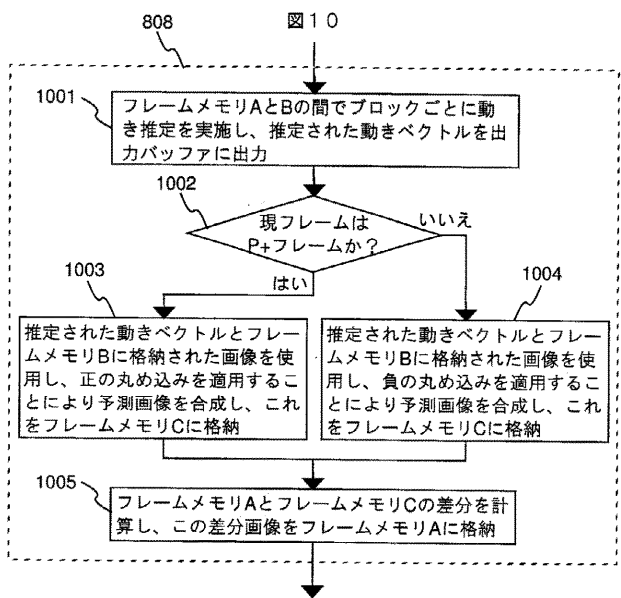
【 図 8 】



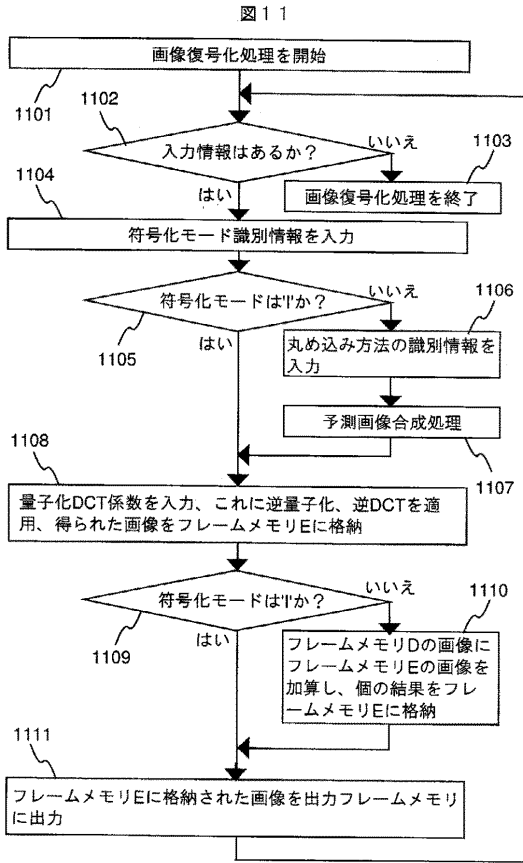
【 図 9 】



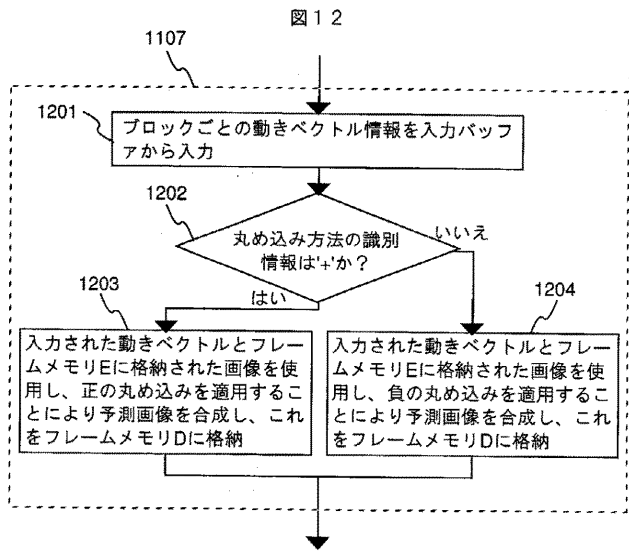
【 図 10 】



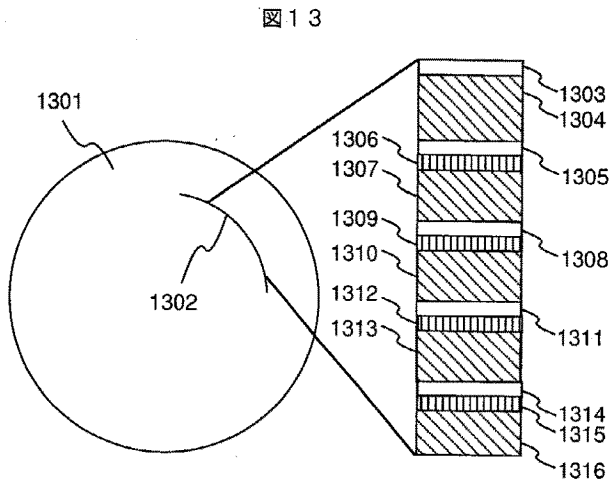
【 図 1 1 】



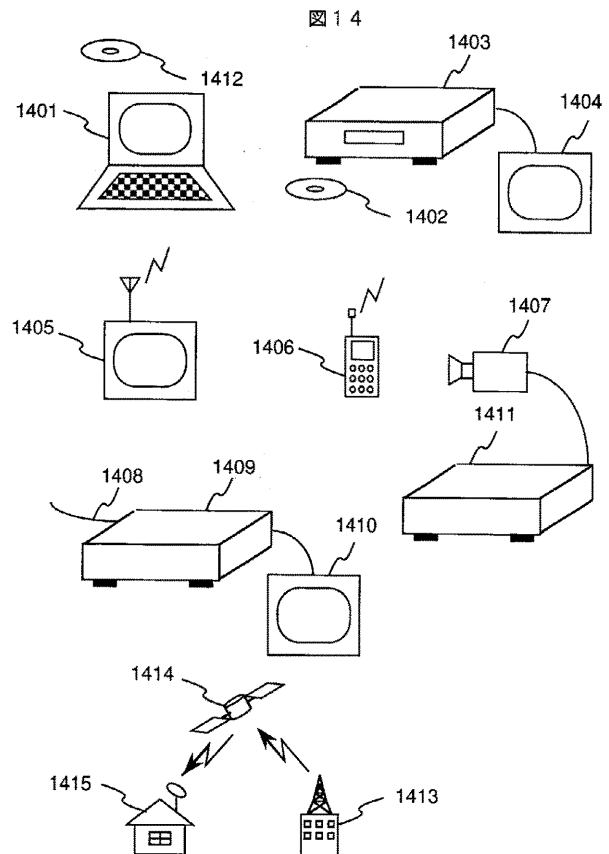
【 図 1 2 】



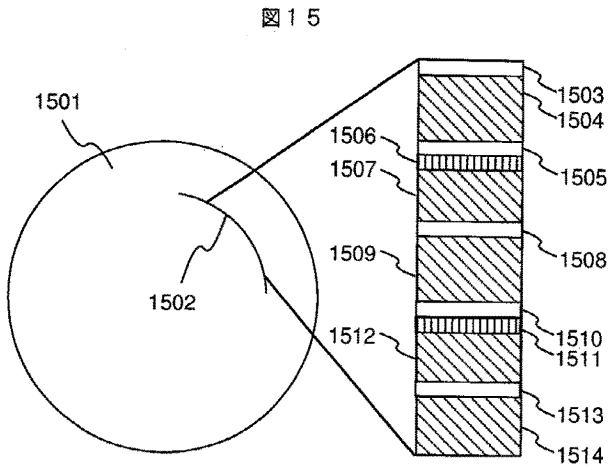
【 図 1 3 】



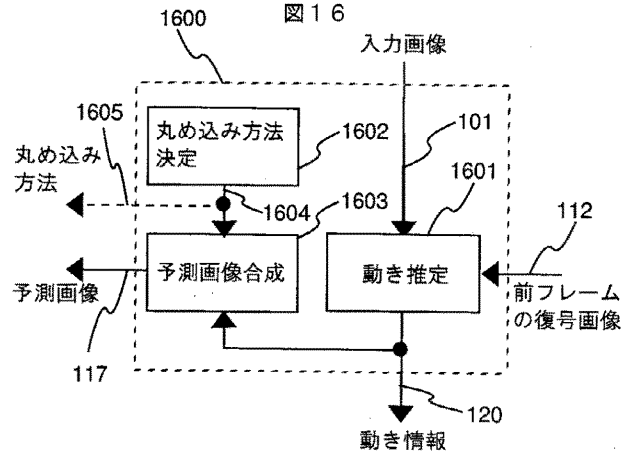
【 図 1 4 】



【図15】



【図16】



【図17】

