

(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2013-201773

(P2013-201773A)

(43) 公開日 平成25年10月3日(2013.10.3)

(51) Int.Cl. F I テーマコード (参考)  
 H04N 7/32 (2006.01) H04N 7/137 Z 5C159

審査請求 有 請求項の数 9 O L (全 27 頁)

<p>(21) 出願番号 特願2013-117274 (P2013-117274)</p> <p>(22) 出願日 平成25年6月3日(2013.6.3)</p> <p>(62) 分割の表示 特願2011-33474 (P2011-33474) の分割</p> <p>原出願日 平成17年11月29日(2005.11.29)</p> <p>(31) 優先権主張番号 60/631,826</p> <p>(32) 優先日 平成16年11月29日(2004.11.29)</p> <p>(33) 優先権主張国 米国 (US)</p> <p>(31) 優先権主張番号 11/288,822</p> <p>(32) 優先日 平成17年11月28日(2005.11.28)</p> <p>(33) 優先権主張国 米国 (US)</p>	<p>(71) 出願人 392026693 株式会社エヌ・ティ・ティ・ドコモ 東京都千代田区永田町二丁目11番1号</p> <p>(74) 代理人 100088155 弁理士 長谷川 芳樹</p> <p>(74) 代理人 100113435 弁理士 黒木 義樹</p> <p>(74) 代理人 100121980 弁理士 沖山 隆</p> <p>(74) 代理人 100128107 弁理士 深石 賢治</p> <p>(72) 発明者 レレスクウ, ダン アメリカ合衆国, カリフォルニア州, サン ホゼ, スイート 300, メト ロ ドライヴ 181</p> <p style="text-align: right;">最終頁に続く</p>
--	--

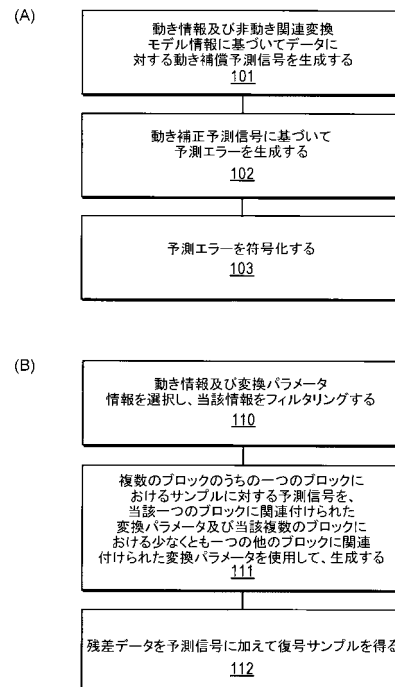
(54) 【発明の名称】 データを符号化するための方法、装置、およびコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 データを符号化するための優れた方法、装置、およびコンピュータ読み取り可能な記録媒体を得る。

【解決手段】 データを符号化するための方法は、1つのフレームにおける1つのブロック内のデータに対する動き補償予測信号を、当該フレーム内の複数のブロックに関する動き情報及び非動き関連変換モデル情報に基づいて、生成するステップと、上記動き補償予測信号に基づいて予測エラーを生成するステップと、上記予測エラーを符号化するステップと、を含む。同方法は、複数の予測サンプルをフィルタリングするステップと、上記フレームの現ブロックに対する動き補償予測信号を形成するために、フィルタリングされた予測サンプルを組み合わせるステップと、を更に含んでもよい。

【選択図】 図1



**【特許請求の範囲】****【請求項 1】**

データを符号化するための方法であって、

1つのフレームにおける1つのブロック内のデータに対する動き補償予測信号を、当該フレーム内の複数のブロックに関する動き情報及び非動き関連変換モデル情報に基づいて、生成するステップと、

前記動き補償予測信号に基づいて予測エラーを生成するステップと、

前記予測エラーを符号化するステップと、

を含む方法。

**【請求項 2】**

前記動き補償予測信号の予測が、参照フレームにおける複数のブロックからのサンプルに基づいている、請求項 1 に記載の方法。

**【請求項 3】**

前記複数の予測サンプルをフィルタリングするステップと、

前記フレームの現ブロックに対する動き補償予測信号を形成するために、フィルタリングされた予測サンプルを組み合わせるステップと、

を更に備える、請求項 1 に記載の方法。

**【請求項 4】**

前記予測エラーは、前記フレームの現ブロックと最終の予測信号との間の差である、請求項 1 に記載の方法。

**【請求項 5】**

データを符号化するための装置であって、

1つのフレームにおける1つのブロック内のデータに対する動き補償予測信号を、当該フレーム内の複数のブロックに関する動き情報及び非動き関連変換モデル情報に基づいて、生成するための動き補償予測モジュールと、

前記動き補償予測信号に基づいて予測エラーを生成するための予測エラー計算モジュールと、

前記予測エラーを符号化するための符号化モジュールと、

を有する、装置。

**【請求項 6】**

前記予測モジュールは、参照フレームにおける複数のブロックのデータからのサンプルに基づいて動き補償予測信号を生成する、請求項 5 に記載の装置。

**【請求項 7】**

選択動き補償は、前記動き補償予測信号における前記複数の予測サンプルをフィルタリングし、前記予測モジュールは、フィルタリングされた予測サンプルを組み合わせる前記フレームの現ブロックに対する動き補償予測信号を形成する、請求項 5 に記載の装置。

**【請求項 8】**

前記予測エラーは、前記フレームの現ブロックと最終の予測信号との間の差である、請求項 5 に記載の装置。

**【請求項 9】**

コンピュータに、

1つのフレームにおける1つのブロック内のデータに対する動き補償予測信号を、当該フレーム内の複数のブロックに関する動き情報及び非動き関連変換モデル情報に基づいて、生成するステップと、

前記動き補償予測信号に基づいて予測エラーを生成するステップと、

前記予測エラーを符号化するステップと、

を実行させるためのプログラム、を記録したコンピュータ読み取り可能な記録媒体。

**【発明の詳細な説明】****【技術分野】****【0001】**

10

20

30

40

50

本発明は、データを符号化するための方法、装置、およびコンピュータ読み取り可能な記録媒体に関する。

【背景技術】

【0002】

時間予測は、ビデオ情報の効率的な符号化のために一般的に使用される技術である。ビデオシーケンスにおける画像の実際の輝度値又は色値を符号化し送信するのではなく、この画像の予測信号は、前に符号化され再構成されて記憶された画像データを参照信号として使用することによって形成され、次いで、符号化すべき画像とその予測信号との間の差が符号化される。予測信号形成のプロセスが良好であればある程、符号化される予測エラーが減少するので、ビデオ符号化はより効率的なものとなる。従って、従来の予測ビデオ符号化は、ビデオシーケンスのフレームにおけるデータを、既に符号化され再構成されたフレームからの情報を参照フレームと称されるシーケンスから使用することによって予測するプロセスを含んでいる。このプロセスは、画像におけるブロックレベルにて規定通りに作動する。予測ブロック符号化においては、ブロック自体のデータを符号化するのではなく、対応する予測ブロックが符号化されるべきブロックから減じられ、これによって得られる予測エラーが符号化される。

10

【0003】

時間予測は、二つの主コンポーネント、即ち、動き推定プロセス及び動き補償プロセスから成るプロセスである。ビデオシーケンスからの現画像における現ブロックを符号化する場合には、動き推定によって、そのビデオシーケンスのうちの一以上の参照フレームにおける最良のマッチングブロックを見つけ出す試みが行なわれる。この動き推定プロセスの出力は、動きベクトルからなり、当該動きベクトルは、現フレーム内のブロックの、一以上の参照フレームにおける最良の対応のマッチングブロック（予測値）に対する変位を示す。

20

【0004】

動き推定プロセスは、輝度又は色情報からなる参照フレームデータを何らかの方法で変換して、当該参照フレームデータを現フレームデータとより密接に相関させた場合に、より良好に実行することができる。この変換は、現データブロックと参照データブロックとの間の差を記述するモデルを反映したものであり、それらの差を補償するための一つの手段として見ることができる。動き推定プロセスに連動するこのような補償変換によって得られる潜在的な利点は、動き推定がより良好なものとなり、また、動き補償プロセスによって現ブロックに対するより良好な予測信号を形成できることに起因して、最終的には符号化効率を向上することができるというものである。

30

【0005】

動き補償プロセスは、動き推定プロセスによって生成された動きベクトルを利用して、符号化すべき現フレームと参照フレームとの間に存在する動きを補償する。このような従来の動き補償の出力は、一以上の予測ブロックであり、当該予測ブロックは、一以上の参照フレームから得られるものであって、符号化すべき現ブロックに対応している。良好な予測信号を形成するために、参照画像データは、そのまま使用されるか、または、予測信号を形成する前に特定のモデルに従って変換される。現ブロックに対して実行される動き推定プロセスに関連して参照ブロックに適用するに適切なものとして決定され得る任意の変換が、予測誤差を計算する前に対応する動き補償ブロックへ適用される。予測のための参照画像データのフレームレベル変換は、一般的には、十分に改善された予測信号を求めるとは不十分なものである。参照データのブロックレベル変換は、符号化すべき画像におけるブロックに対するより良好なブロックレベルの予測信号を生成することができるが、より高いオーバーヘッドを伴ってしまう。

40

【0006】

この動き及び変換 - 補償の結合予測プロセスを使用して、現ブロックに対する予測信号が形成される。より良好なブロック予測信号によって、ビデオ符号化器にて符号化する必要のある予測エラーが小さくなり、その結果、符号化利得が得られる。しかしながら、従

50

来の方法は、参照フレームにおけるデータの変換及び動き補償の結合を用いて符号化すべきフレームの予測信号を生成するものであり、当該方法によって生成されるレート歪み符号化利得の量に限界がある。これは、参照フレームにて利用できる予測データの使用が不十分なためである。また、従来の方法では、従来のビデオ符号化における独立したブロックレベルの処理を使用することに伴うブロッキングアーチファクトが目立ってしまう。

【 0 0 0 7 】

ビデオ符号化における予測信号の質を改善するために参照画像に適用される補償変換のパラメータを決定する方法については、2002年9月に発行されたNTT(JP)、K.Kamikura氏等による「Brightness-Variation Compensation Method and Coding/Decoding Apparatus for Moving Pictures」と題する米国特許第6,456,658号、1996年8月に発行されたインテルコーポレーション、S.J.Golin氏等による「Methods and Apparatus for Improving Motion Analysis of Fades」と題する米国特許第5,544,239号、2004年1月に出版されたJ.M.Boyce氏による「Adaptive Weighting of Reference Pictures in Video Encoding」と題する米国特許出願公開第US2004/0008786号、N.M.M.Rodrigues氏等による「Hierarchical Motion Compensation with Spatial Luminance Transformations」と題する2001年の文献、1998年12月のIEEE Transactions on Circuits and Systems for Video Technology Vol.8, No.8におけるK.Kamikura氏等による「Global Brightness-Variation Compensation for Video Coding」と題する論文、及び2003年4月のIEEE Transactions on Circuits and Systems for Video Technology Vol.13, No.4におけるS.H.Kim氏等による「Fast Local Motion Compensation Algorithm for Video Sequences with Brightness Variations」と題する論文に開示されている。この変換は、参照フレームのデータに対してフレームレベルにて決定され、当該データに適用されるものであり、この場合には、動き推定プロセスは、変換された参照フレームに対して動作する。或いは、ブロックレベルの変換を、動き推定プロセス中に決定することが可能である。これらの変換は、ブロックレベルで適用されて、通常のビデオ符号化器における予測エラーの計算に使用される予測ブロックが形成される。第1のステージで決定されたブロックレベルの変換パラメータに基づいて、実際には、パラメータのサブセットのみが、所与のフレームに対して使用するために保持される。このサブセットは、例えば、フレームにおいて最も頻繁に使用されるブロックレベルモデルを含むものである。関連技術において使用されている変換モデルとしては、単一パラメータ(オフセットティング、又はスケーリング)モデル、2-パラメータリニアモデル(スケーリング及びオフセットティング)及び6-パラメータアフィンモデルがある。別の手法として、初期フレームレベルの補償を、変換の後のブロックレベルの改善と結合させることによる階層型補償が適用される。

【 0 0 0 8 】

予測ビデオ符号化を補償するために使用すべき単一の変換モデル又は少数の変換モデルを決定するのでは、符号化効率の利得の点で限界がある。これは、符号化すべき現フレームと参照用のフレームとの間に存在する差を記述するための能力が制限されるからである。ブロックレベル変換モデルを決定し使用する方法は、このような制限をいくらか緩和するが、これら方法は、依然として、予測信号の質、ひいては、目標とする符号化効率を向上することに関して、限界がある。また、それらは個々のブロックに対して個別に動作するので、これら方法は、復号された画像の主観的品質に対して有害なブロッキングアーチファクトを保持してしまうか、又はより目立つようにしてしまうものである。

【 先行技術文献 】

【 特許文献 】

【 0 0 0 9 】

【 特許文献 1 】 米国特許第 6 , 4 5 6 , 6 5 8 号

【 特許文献 2 】 米国特許第 5 , 5 4 4 , 2 3 9 号

【 特許文献 3 】 米国特許出願公開第 US 2 0 0 4 / 0 0 0 8 7 8 6 号

10

20

30

40

50

## 【非特許文献】

【0010】

【非特許文献1】N.M.M.Rodrigues氏等による「Hierarchical Motion Compensation with Spatial Luminance Transformations」と題する2001年の文献

【非特許文献2】1998年12月のIEEE Transactions on Circuits and Systems for Video Technology Vol.8, No.8におけるK.Kamikura氏等による「Global Brightness-Variation Compensation for Video Coding」と題する論文

【非特許文献3】2003年4月のIEEE Transactions on Circuits and Systems for Video Technology Vol.13, No.4におけるS.H.Kim氏等による「Fast Local Motion Compensation Algorithm for Video Sequences with Brightness Variations」と題する論文

10

## 【発明の概要】

【0011】

時間予測を行うための方法、装置及び製品について説明する。ある実施の形態では、復号方法が、複数のブロックのうちの一つにおけるサンプルに対する予測信号を、当該一つのブロックに関連した変換パラメータ及びブロックのグループにおける少なくとも一つの他のブロックに関連した変換パラメータを使用して生成することと、残差データを当該予測信号に加えて、復号したサンプルを得ることと、を含む。

【0012】

本発明は、以下の詳細な説明及び本発明の種々な実施の形態を示す添付図面からより完全に理解されるであろう。しかしながら、これらの詳細な説明及び実施の形態は、本発明をそれらの特定の実施の形態に限定するものとして理解されるべきでなく、これらは、説明及び理解のためのものに過ぎない。

20

## 【図面の簡単な説明】

【0013】

【図1】復号プロセスおよび符号化プロセスの一実施の形態のフロー図である。

【図2】ビデオ符号化器の機能ブロックの一部を示す図である。

【図3】ビデオ符号化器の機能ブロックにおける図2以外の部分を示す図である。

【図4】ビデオ符号化プロセスの一つの実施の形態のフロー図である。

【図5】動き補償予測モジュールの一つの実施の形態のブロック図である。

30

【図6】動き補償予測モジュールプロセスの一つの実施の形態のフロー図である。

【図7】条件付き選択モジュールにおける条件付き選択プロセスの一つの実施の形態のフロー図である。

【図8】空間位置生成モジュールの一つの実施の形態によって生成された空間位置に対応するブロックのブロック図である。

【図9】予測モジュールプロセス-1の一つの実施の形態のフロー図である。

【図10】予測モジュールプロセス-2の一つの実施の形態のフロー図である。

【図11】予測モジュールプロセス-3の一つの実施の形態のフロー図である。

【図12】予測モジュールプロセス-4の一つの実施の形態のフロー図である。

【図13】ビデオ復号器の一つの実施の形態のブロック図である。

40

【図14】予測フレームのためのビデオ復号プロセスの一つの実施の形態のフロー図である。

【図15】コンピュータシステムの一つの実施の形態のブロック図である。

## 【発明を実施するための形態】

【0014】

ビデオ符号化における予測フィルタリングプロセスのための装置及び方法について説明する。予測フィルタリングは、複数の画像ブロックに対応する動き情報及び非動き関連変換モデルの組み合わせに基づいており、ビデオシーケンスのフレームにおいて符号化すべきブロックに対する予測信号を求める。

【0015】

50

ある実施の形態では、ブロックレベル変換モデルが動き推定プロセス中に決定される。条件付き動き補償予測決定及びフィルタリングが、符号化すべきフレームの予測されるブロックに対する予測信号を形成するために、なされる。条件付き選択プロセスは、予測データ（例えば、予測サンプル）及びフィルタリングプロセスによって使用される初期ブロックレベルの変換パラメータセットを決定する。そして、予測フィルタリングプロセスは、予測サンプル及び変換パラメータのセットを使用して、データサンプルに対するフィルタリングされたサンプルレベルの予測信号を形成する。ある実施の形態では、予測フィルタリング方法は、フレームにおけるブロックに関連した強度補償パラメータ及び時間予測サンプル候補を処理して、より良好な予測信号を生成する。このように、現フレームにおけるブロックを対応の参照ブロックに関連づける変換モデル及び動き情報を組み合わせて、予測符号化する現ブロックに対する最終の予測信号を得る。

10

**【 0 0 1 6 】**

本明細書に説明する処理は、ブロッキングアーチファクトが画像における個々のブロックを個別に処理することによって生じてしまう関連技術のビデオ符号化方法に比較して、符号化効率を向上させ、ブロッキングアーチファクトを減少させる。本明細書に説明する予測決定及びフィルタリングプロセスの設計は、ビデオ符号化プロセスのレート歪み性能について、また、ブロッキングアーチファクトを減少させ得る程度を決定する上でも重要である。

**【 0 0 1 7 】**

図 1 ( A ) は、復号プロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

20

**【 0 0 1 8 】**

以下、図 1 ( A ) を参照する。プロセスは、処理ロジックが動き情報及び非動き関連変換モデル情報に基づいてデータに対する動き補償予測信号を生成することによって始まる（処理ブロック 1 0 1 ）。ある実施の形態では、動き及び非動き関連変換モデル情報は、フレームにおける複数のブロックに関連した強度補償パラメータ及び時間予測サンプルのセットを含んでいる。ある実施の形態では、動き補償予測信号は、参照フレームにおける複数のブロックのデータからのサンプルに基づいている。

30

**【 0 0 1 9 】**

ある実施の形態では、処理ロジックは、予測サンプルをフィルタリングし、フィルタリングした予測サンプルを組み合わせることによって動き補償予測信号を生成して、フレームの現ブロックに対する動き補償予測信号を形成する。

**【 0 0 2 0 】**

予測信号を生成した後、処理ロジックは、動き補償予測信号に基づいて予測エラーを生成する（処理ブロック 1 0 2 ）。ある実施の形態では、予測エラーは、フレームの現ブロックと最終予測信号との間の差から構成される。

**【 0 0 2 1 】**

予測エラーを生成すると、処理ロジックは、その予測エラーを符号化する（処理ブロック 1 0 3 ）。

40

**【 0 0 2 2 】**

図 1 ( B ) は、符号化プロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

**【 0 0 2 3 】**

以下、図 1 ( B ) を参照する。プロセスは、処理ロジックが動き情報及び変換パラメータ情報を条件に従って選択し、当該情報をフィルタリングすることによって始まる（処理ブロック 1 1 0 ）。ある実施の形態では、動き情報は、動きベクトルを含む。また、ある

50

実施の形態では、変換パラメータは、強度補償情報を含む。

【0024】

ある実施の形態では、動き情報及び強度補償情報のフィルタリングは、予測サンプルをフィルタリングすることを含み、ここでは、予測サンプルの各々が、ブロック強度補償パラメータを用いて補償される。別の実施の形態では、動き情報及び強度補償情報のフィルタリングは、ブロック強度補償パラメータをフィルタリングして、サンプルレベル強度補償パラメータを得ることを含む。更に別の実施の形態では、動き情報及び強度補償情報のフィルタリングは、予測サンプルをフィルタリングすることを含み、ここでは、予測サンプルの各々が、サンプルレベル強度補償パラメータを用いて補償される。

【0025】

フィルタリングの後、処理ロジックは、複数のブロックのうちの一つのブロックにおけるサンプルに対する予測信号を、当該一つのブロックに関連した変換パラメータ及び当該複数のブロックのうち少なくとも一つの他のブロックに関連した変換パラメータを用いて、生成する（処理ブロック111）。ある実施の形態では、処理ロジックは、選択された動き及び強度補償情報に基づいて、予測信号を生成する。

【0026】

予測信号を生成した後、処理ロジックは、残差データを予測信号に加えて、復号サンプルを得る（処理ブロック112）。ある実施の形態では、予測信号は、強度補償情報及び動き情報に基づいて生成される。

【0027】

これらのプロセスについて、以下より詳細に説明する。

【0028】

以下の説明において、多数の詳細について述べ、本発明のより完全に説明する。しかしながら、これらの特定の細部無くとも本発明を実施できることは、当業者には明らかであろう。他の例では、よく知られた構造及び装置については、本発明を不明瞭なものとするのを避けるために、ブロック図の形式にて示し、細部については示していない。

【0029】

以下の詳細な説明の幾つかの部分は、コンピュータメモリ内のデータビットについてのオペレーションの記号的表現及びアルゴリズムの形で、説明がなされている。これらアルゴリズムの記述及び表現は、データ処理技術の分野における当業者が他の当業者へ自分の研究の内容を最も効果的に知らせるのに使用される手段である。アルゴリズムとは、本明細書において、また、一般的にも、所望の結果に至るステップの自己矛盾のないシーケンスであると考えられるものである。これらステップは、物理量の物理的操作を必要とするものである。通常、必ずしもそうではないが、これらの量は、記憶されたり、転送されたり、結合されたり、比較されたり、その他の方法で操作され得るような電氣的又は磁氣的信号の形態をとる。主に慣用されていることを理由として、これらの信号を、ビット、値、要素、記号、文字、用語、数等として言及することが時には便利であることがわかっている。

【0030】

しかしながら、これらの用語及び類似の用語の全ては、適当な物理量に関連付けられるべきものであり、これらの量に適用される単なる便宜的なラベルであるということに留意されたい。以下の説明から明らかでないように特に具体的に言及しない限り、本明細書の記載を通して、「処理する」又は「演算する」又は「計算する」又は「決定する」又は「表示する」等の如き用語を使用している説明は、コンピュータシステム又は同様の電子的計算装置の動作及びプロセス、即ち、コンピュータシステムのレジスタ及びメモリ内の物理的（電子的）量として表されたデータを、当該コンピュータシステムのメモリ又はレジスタ、若しくは他の同様の情報記憶装置、送信装置又は表示装置内の物理的量として同様に表される他のデータへと操作・変換する動作及びプロセスを指すものであることを理解されたい。

【0031】

本発明はまた、本明細書におけるオペレーションを行うための装置にも関するものである。この装置は、必要とされる目的に対して特別に構成されてもよいし、又は、コンピュータに記憶されたコンピュータプログラムによって選択的に作動され又は再構成されるような汎用コンピュータであってもよい。このようなコンピュータプログラムは、これに限定されるものではないが、フレキシブルディスク、光ディスク、CD-ROM、及び磁気・光ディスクを含む任意のタイプのディスク、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、EPROM、EEPROM、磁気又は光カード、又は電子的命令を記憶するのに適し且つ各々がコンピュータシステムバスに結合されるような任意のタイプの媒体のようなコンピュータ読取り可能な記録媒体に記憶される。

#### 【0032】

本明細書に示すアルゴリズム及びディスプレイは、本来的には、如何なる特定のコンピュータ又は他の装置にも関連付けられているものではない。種々の汎用システムを本明細書の教示に従うプログラムと共に使用してもよく、又は、必要とされる方法ステップを行うためより特化した装置を構成することも便利であることがわかっている。様々なシステムのために必要とされる構成は、以下の説明から明らかとなろう。また、本発明は、特定のプログラミング言語に関連して説明していない。種々のプログラミング言語を使用して、本明細書に説明する本発明の教示を実施し得ることが理解されよう。

#### 【0033】

機械読取り可能な媒体は、マシン(例えば、コンピュータ)によって読取り可能な形で情報を記憶又は送信する任意の機構を含む。機械読取り可能な媒体としては、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、磁気ディスク記録媒体、光記録媒体、フラッシュメモリデバイス、電氣的、音響的又はその他の形式の伝搬信号(例えば、搬送波、赤外線信号、デジタル信号等)等がある。

#### 【0034】

##### <ビデオ符号化器の実施例>

図2および図3は、ビデオ符号化器(VC)の一つの実施の形態のブロック図である。以下、図2および図3を参照する。ビデオ符号化器は、動き推定モジュール(MEM)215、動き補償予測モジュール(MCPM)213、変換モデル決定モジュール(TMDM)216、メモリ214、変換符号化モジュール(TCM)230、動きデータ処理モジュール(MDPM)206、フレームストア(FS)(メモリ)212、並びに、スイッチSW207~209及び217~220を備えている。変換符号化器(CM)230は、変換モジュール201、量子化器モジュール202、及びエントロピー符号化モジュール203を有している。TCM230の入力でのフレームデータは、ビデオフレーム又はDFD(Displaced Frame Difference)フレームから構成される。DFDフレームは、ビデオ符号化器において、ビデオフレームにおけるデータとMCPM213の出力に生成されたその予測信号との間の差をとることによって、得られる。MCPM213は、FS212に記憶された動き補償再構成ビデオフレームからなるデータに基づいて予測信号を生成する。動き補償は、MEM215によって生成された動き情報と、TMDM216によって生成された変換モデル情報とを使用して行なわれる。このプロセスは、以下に説明する図2および図3のスイッチの構成によって可能となる。或いは、ビデオ符号化器が、予測信号を使用せずに、従来の符号化器のイントラ符号化モードに対応するビデオフレームを直接的に符号化するときには、MEM215、TMDM216、MCPM213及びMDPM213は使用されない。ある実施の形態では、ビデオ符号化器は、ブロックにおける各位置に輝度値又は色値を有するブロック分割フレームデータについて順次作用する。

#### 【0035】

図4は、図2および図3に示すVCによって実行されるプロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア(回路、専用ロジック等)、ソフトウェア(汎用コンピュータシステム又は専用マシンにて実行されるような)、又は両者の組み合わせを備え得る処理ロジックによって実行される。

10

20

30

40

50

## 【 0 0 3 6 】

以下、図 4 を参照する。オリジナルの現フレームからのブロックを読み取って、当該現フレームの全てのブロックを処理するまで、説明するように以下のプロセスによって順次処理する。処理ロジックは、現フレームにおけるブロックに対するカウンタ変数（インデックス） $i$  を 0 に初期化する（処理ブロック 3 0 1）。この V C オペレーションの第一段階において、図 2 および図 3 におけるスイッチは、次のようにコンフィギュレーションされる。即ち、S W 1 は、開、S W 2 は、閉、S W 3 は、開、S W 4 は、開、S W 5 は、開、S W 6 は、開である。そして、処理ロジックは、現フレームにおける現在の非圧縮ブロック  $c(i)$  を読み取る（処理ブロック 3 0 2）。

## 【 0 0 3 7 】

次に、動き推定プロセスを実行して、これによって、F S 2 1 2 からの参照フレームにおけるサーチエリアを探索して、現ブロックに最もマッチングする参照ブロックを特定する。このサーチエリアにおける位置の数は、 $K$  である。ここで、現ブロック  $c(i)$  に対応するサーチエリアからの位置を  $k$  が指すものとする。処理ロジックは、インデックス  $k$  を 1 に初期化する（処理ブロック 3 0 3）。この位置は、現ブロック  $c(i)$  を原点とし参照フレームにおける参照ブロック  $b(k)$  へ向いた動きベクトル  $m v(k)$  によって特定される。処理ロジックは、サーチエリアにおける位置  $k$  に対応する動きベクトル  $m v(k)$  をフェッチし、T M D M 2 1 6 へ送り（処理ブロック 3 0 4）、当該 T M D M 2 1 6 が、 $b(k)$  に適用されるべき変換  $T$  のパラメータ  $a(k)$  を決定する（処理ブロック 3 0 5）。ある実施の形態では、これは、現ブロック  $c(i)$  と変換ブロックデータ  $T(b(k))$  との間のエラー  $d(i, k)$  の量を最小とされるようにされる。このエラーは、例えば、 $c(i)$  及び  $T(b(k))$  との間の二乗差の和によって定量化される。 $a(k)$  を用いて変換された  $b(k)$ 、即ち  $T(b(k))$  と  $c(i)$  との間のエラー  $d(i, k)$ 、対応する動きベクトル  $m v(k)$ 、及びその変換  $T$  を特徴づける変換パラメータ  $a(k)$  は、メモリ 2 1 4 に記憶される（処理ブロック 3 0 6 及び 3 0 7）。

## 【 0 0 3 8 】

次いで、処理ロジックは、参照フレームにおけるサーチエリア内に候補位置があるか（ $k \leq K$ ）をテストする（処理ブロック 3 0 9）。処理ロジックは、インデックス  $k$  をインクリメントする（処理ブロック 3 0 8）。参照フレームにおけるサーチエリア内に候補位置が残っている場合には（即ち、 $k < K$ ）、処理は処理ブロック 3 0 4 へ移行し、参照フレームにおける別のブロック  $b(k)$  を選択して、前述したようなプロセスを繰り返す。参照フレームにおけるサーチエリア内に候補位置が残っていない場合には、処理は処理ブロック 3 1 0 へ移行する。

## 【 0 0 3 9 】

現ブロック  $c(i)$  に対応する  $K$  のサーチエリアの全ての位置をアクセスすると、T M D M 2 1 6 における処理ロジックは、メモリ 2 1 4 に記憶された値  $d(i, k)$  のアレイの中で最も小さいエラー  $d_{min}$  を探し出す（処理ブロック 3 1 0）。このエラー  $d_{min}$  は、 $b_{min}$  によって示される特定の参照ブロックに対応するものである。処理ロジック M E M 2 1 5 は、サンプルをイントラ符号化することによる現ブロック  $c(i)$  の符号化コストと、ブロック  $b_{min}$  に対するブロック  $c(i)$  の予測符号化のコストとを比較することによって、現ブロックの符号化モード  $mode(i)$  を決定する（処理ブロック 3 1 1）。前者のコストがより小さいならば、ブロック  $c(i)$  のモードは、I N T R A としてマークされる。

## 【 0 0 4 0 】

処理ロジックは、ブロック  $c(i)$  のモードを I N T R A としてマークするかをテストする（処理ブロック 3 1 2）。I N T R A としない場合には、T M D M 2 1 6 における処理ロジックは、 $d_{min}$  及び  $b_{min}$  に関連した変換（ $T$ ）パラメータ  $a$  をメモリ 2 1 4 にセーブして（処理ブロック 3 1 3）、処理ブロック 3 1 4 へ移行する。I N T R A とする場合には、処理ロジックは、処理ブロック 3 1 4 へ直接的に移行する。処理ブロック 3 1 4 にて、処理ロジックは、現ブロックのモードをメモリ 2 1 4 へ書き込み（処理ブロッ

10

20

30

40

50

ク 3 1 4 )、プロセスは、現フレームにおける次のブロック  $c(i)$  へ移行する。MEM における処理ロジックは、また、 $dmin$  及び  $bmin$  に対応する動きベクトルをメモリ 2 1 4 に保存する。

【 0 0 4 1 】

処理ロジックは、次いで、インデックス  $i$  をインクリメントし (処理ブロック 3 1 5 )、現フレームに更なるブロックがある ( $i < N$ ) かをテストする (処理ブロック 3 1 6 )。更にブロックがある場合には、処理ロジックは、処理ブロック 3 0 2 に移行し、処理を繰り返す。更なるブロックがない場合には、処理は、処理ブロック 3 1 7 へ移行する。

【 0 0 4 2 】

現フレームにおける全てのブロックを前述したように処理すると、新しいプロセスを開始して、現フレームからのブロック  $c(i)$  を順次読み取る。この部分のビデオ符号化器のオペレーションでは、図 2 および図 3 におけるスイッチは、以下の初期コンフィギュレーションを有している。即ち、 $SW1$  は、閉、 $SW2$  は、開、 $SW3$  は、閉、 $SW4$ 、 $SW5$ 、 $SW6$  は、現ブロックのモードが  $INTRA$  の場合には、開、当該モードが  $INTRA$  でない場合には、閉である。 $SW0$  は、現ブロックのモードが  $INTRA$  である場合には、位置 1 にあり、 $INTRA$  でない場合には、位置 2 にある。

【 0 0 4 3 】

処理ブロック 3 1 7 にて、処理ロジックは、インデックス  $i$  をリセットする。次いで、 $MCPM213$  における処理ロジックは、現フレームにおけるブロックのセットに対応する現フレームにおける空間位置のセット  $S$  を生成する (処理ブロック 3 1 8 )。ある実施の形態では、この空間位置のセットは、現フレームにおいて符号化されるべき現ブロックの近傍ブロックを次のように特定する。即ち、現ブロック  $c(i)$  の左側、上側、右側、及び下側の近傍の位置が、図 8 に示すように、セット  $S$  に含まれる。

【 0 0 4 4 】

処理ロジックは、非圧縮の現フレームから現ブロック  $c(i)$  を読み取る (処理ブロック 3 1 9 )。処理ロジックは、メモリ 2 1 4 から現ブロック  $c(i)$  のモード  $mode(i)$  を読み取り、当該モードが  $INTRA$  であるかをテストする (処理ブロック 3 2 0 )。

【 0 0 4 5 】

このモードが  $INTRA$  である場合には、処理ロジックは、処理ブロック 3 2 5 へ直接移行する。読み取られた現ブロックのモードが  $INTRA$  でない場合には、処理ロジックは、現ブロックに対応する動きベクトルデータを  $MCPM213$  へ送り、当該  $MCPM213$  において、動きベクトルデータを処理して、符号化用のエントロピー符号化モジュールへ送る。より詳細には、処理ロジックは、 $MCPM213$  において現ブロック  $c(i)$  に対応する予測ブロック  $p(i)$  を決定することによって、 $VC$  における予測エラー  $e(i)$  を決定し (処理ブロック 3 2 1)、現ブロック  $c(i)$  と  $MCPM213$  によって生成される予測信号  $p(i)$  との差を取り (処理ブロック 3 2 2)、動きベクトルを  $MCPM213$  へ送り (処理ブロック 3 2 3)、動き情報をエントロピー符号化する (処理ブロック 3 2 4)。

【 0 0 4 6 】

処理ブロック 3 2 5 にて、処理ロジックは、符号化  $mode(i)$  をエントロピー符号化する。その後、処理ロジックは、現ブロックのデータを  $TCM230$  へ送り、当該  $TCM230$  において、当該データを変換符号化する (処理ブロック 3 2 6)。変換符号化は、 $mode(i)$  に応じて、ブロックデータ  $c(i)$ 、又は予測エラーデータ  $e(i)$  に適用され、その結果、符号化されたブロックデータが得られる。変換符号化の後、処理ロジックは、現ブロックを再構成し (処理ブロック 3 2 7)、再構成した現ブロックを、従来のビデオ符号化器におけるように、 $FS212$  に記憶する (処理ブロック 3 2 8)。

【 0 0 4 7 】

次いで、処理ロジックは、インデックス  $i$  をインクリメントし (処理ブロック 3 2 9)、参照フレームに処理すべき更なるブロックがあるかをテストする (処理ブロック 3 3 0)

10

20

30

40

50

)。更なるブロックがある場合には、プロセスは、処理ブロック 3 1 9 に移行し、処理が繰り返される。更なるブロックがない場合には、プロセスは、終了する。

【 0 0 4 8 】

ある実施の形態では、TMDM 2 1 6 は、以下の入力、即ち、MEM 2 1 4 によって生成される動き情報（動きベクトルの形態）、現ビデオフレームデータ（ブロック）、及びフレームストアからの再構成フレームデータを有する。従来の方法におけるように、所与の変換モデルのオーダーに対して、これら入力に基づき且つ最適性基準（例えば、最小二乗基準）に従って、TMDM 2 1 6 は、変換モデル T のパラメータ a を決定し、当該パラメータをメモリ 2 1 4 に記憶する。これは、現ビデオフレームからの現ブロック c と、MEM 2 1 5 から受け取られた動きベクトル情報によって与えられる位置にあり、且つ、パラメータ a を用いて変換された参照ブロックとの間の差に関する最適性基準を満足するようになされる。TMDM 2 1 6 の出力は、参照ブロックに関連する変換モデルのパラメータを含む。これらパラメータは、メモリモジュール 2 1 4 に保存される。

10

【 0 0 4 9 】

図 5 は、MCPM の一つの実施の形態のブロック図である。以下、図 5 を参照する。MCPM 4 0 0 は、空間位置生成モジュール（SPGM）4 0 1、条件付き選択モジュール（CSM）4 0 2、メモリ 4 0 3、選択動き補償モジュール（SMCM）4 0 4、メモリ 4 0 5、及び予測モジュール（PM）4 0 6 を備えている。MCPM 4 0 0 は、二つの入力、即ち、現フレームにおける所与のブロック用の動きベクトル、変換モデルパラメータ、及び符号化モードからなる入力（4 1 0）と、FS 2 1 2 からの再構成参照フレームデータからなる入力（4 1 1）とを有する。MCPM 4 0 0 は、現フレームにおけるブロックに対する予測データ 4 1 2 を出力する。

20

【 0 0 5 0 】

SPGM 4 0 1 は、現フレームにおけるブロック位置に関する情報を CSM 4 0 2 へ送る。CSM 4 0 2 は、また、現フレームにおけるブロックに対応する動きベクトル、変換パラメータ、及び符号化モード（4 1 0）を受け取る。これら入力から、CSM 4 0 2 は、動きベクトル、変換パラメータ、及び符号化モードのサブセットを選択し、これらをメモリ 4 0 3 に記憶させる。SMCM 4 0 4 は、フレームストア 2 1 2 からの参照フレームデータ 4 1 1 を、メモリ 4 0 3 からのモデルパラメータと共に、受け取る。これらのモデルパラメータは、変換パラメータである。これら入力を使用して、SMCM 4 0 4 は、予測サンプルの形態で、動き補償データを生成し、当該データを、メモリ 4 0 5 に記憶させる。予測モジュール 4 0 6 は、メモリ 4 0 3 からのモデル（変換）パラメータを、メモリ 4 0 5 からの動き補償データ（予測サンプル）と共に受け取り、これに応答して、フィルタリングされた予測信号 4 1 2 を生成する。

30

【 0 0 5 1 】

図 6 は、MCPM によって実行される動き補償予測プロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

【 0 0 5 2 】

以下、図 6 を参照する。MCPM における処理ロジックは、現フレームにおける全ブロック用の符号化モード、動きベクトル、及び変換モデルパラメータを、メモリ 2 1 4 から取り出す（処理ブロック 5 0 1）。次いで、処理ロジックは、各ブロックについて、当該ブロック、及び SPGM 4 0 1 によって生成されたセット S において位置が特定される現ビデオフレームにおける他のブロックに対応する動きベクトルと変換モデルパラメータとを条件に従って選択し、それらをメモリ 4 0 3 に記憶させる（処理ブロック 5 0 2）。次いで、現フレームにおいて予測されるブロックの各サンプルについて、SMC 4 0 4 における処理ロジックは、動き補償サンプルを、参照フレームからメモリ 4 0 3 に記憶された動きベクトルに従って取り出し、それら予測サンプルをメモリ 4 0 5 に記憶させる（処理ブロック 5 0 3）。PM 4 0 6 における処理ロジックは、メモリ 4 0 5 に記憶された予

40

50

測サンプルを、メモリ403に記憶された変換モデルパラメータと共に取り出し、現ブロックの各サンプルについて予測信号Pを算出する(処理ブロック504)。

【0053】

ある実施の形態では、CSM402は、現ブロックcに対応する符号化モード、動きベクトル、及び変換パラメータを、メモリ214から、それぞれMEM215及びTMDM216によって生成された状態で、取り出す。CSM402は、SPGM401から現フレームにおける空間位置のセットSからなる別の入力を受け取る。CSM402は、選択された動きベクトル、変換パラメータを出力し、MCPM400のメモリ403に書き込む。

【0054】

図7は、CSMによって実行される条件付き選択プロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア(回路、専用ロジック等)、ソフトウェア(汎用コンピュータシステム又は専用マシンにて実行されるような)、又は両者の組み合わせを備え得る処理ロジックによって実行される。

【0055】

以下、図7を参照する。CSM402は、現ブロックcに対応するブロック符号化モードを、メモリ214から取り出す(処理ブロック601)。処理ロジックは、次いで、現ブロックのモードがINTRAであるかをテストする(処理ブロック602)。現ブロックcの符号化モードがINTRAである場合には、それ以上の当該ブロックに関する動作はCSM402によって行われず、プロセスは終了する。別の場合、即ち、現ブロックのモードがINTRAでない場合には、CSM402は、メモリ214から、現ブロックc用の動きベクトル $mv\_c$ 及び対応の変換パラメータ $a\_c$ を取り出し、それらをメモリ403に書き込む(処理ブロック603)。

【0056】

そして、処理ロジックは、インデックスkを1へリセットする(処理ブロック604)。次に、CSM402は、メモリ214から、現フレームにおいてセットBを形成する他のブロックcの符号化モードを順次に取り出す。このセットBのブロックの空間位置は、SPGM401によって生成されるセットSによって決定されるものである。セットBにおけるブロックcの各々には、メモリ403におけるcandidate\_flagsによって示されるフラグの配列におけるCSM維持値が、対応している。より詳細には、処理ロジックは、セットSにおける位置kのブロックc(k)の符号化モードを取り出す(処理ブロック605)。次いで、処理ロジックは、ブロックc(k)のモードがINTRAであるかをテストする(処理ブロック606)。セットSにおけるk番目の空間位置に対応するセットBにおけるk番目のブロックについて、その符号化モードがINTRAである場合には(メモリ214から取り出されるように)、処理ロジックは、対応の値candidate\_flags(k)を0にセットし(処理ブロック607)、当該値をメモリ403に書き込み、動きベクトル $v(k)$ を現ブロックの $mv\_c$ に等しくセットし、変換パラメータ $m(k)$ を現ブロックの $a\_c$ に等しくセットし、また、これらをメモリ403に書き込み、その配列candidate\_flagsにおける現在の位置と関連付ける。その後、処理は、処理ブロック617へ移行し、Bにおける次のブロックの処理を継続する。

【0057】

別の場合、即ち、k番目のブロックの符号化モードがINTRAでない場合には、処理ロジックは、candidate\_flags(k)の位置を1にセットし(処理ブロック609)、CSM402における処理ロジックが、Bにおけるブロックc(k)に関連した対応の動きベクトル $mv(k)$ 及び変換モデルパラメータ $a(k)$ を、メモリ214から取り出す(処理ブロック610)。次に、処理ロジックは、現ブロックcの動きベクトルと、セットBからの試験すべきブロックに関連した動きベクトルとの間の $d(mv\_c, mv(k))$ によって示される距離メトリックを算出する(処理ブロック611)。ある実施の形態では、この距離メトリックは、動きベクトルの差のL1-ノルムによって表される。

【0058】

10

20

30

40

50

次いで、処理ロジックは、距離メトリックが閾値  $t$  より小さいかをテストする（処理ブロック 6 1 2）。 $d(mv\_c, mv(k))$  が閾値  $t$  より小さい場合には、処理ロジックは、 $candidate\_flags(k)$  を 1 に等しくセットし（処理ブロック 6 1 5）、動きベクトル  $v(k)$  を  $mv(k)$  に等しくセットし、パラメータ  $m(k)$  を  $a(k)$  に等しくセットする（処理ブロック 6 1 6）。これらは、メモリ 4 0 3 に書き込まれ、アレイ  $candidate\_flags$  における現在位置に関連付けられる。その後、処理は、処理ブロック 6 1 7 に引き続く。他の場合、即ち、距離  $d(mv\_c, mv(k))$  が閾値  $t$  を越える場合、CSM 4 0 2 における処理ロジックは、対応の値  $candidate\_flags(k)$  をメモリ 4 0 3 において 0 にセットし（処理ブロック 6 1 3）、動きベクトル  $v(k)$  を  $mv\_c$  に等しくセットし、変換モデルパラメータ  $m(k)$  を  $a\_c$  に等しくセットし、それらをメモリ 4 0 3 に保存する（処理ブロック 6 1 4）。その後、その処理は、処理ブロック 6 1 7 に引き続く。

10

【 0 0 5 9 】

処理ブロック 6 1 7 にて、処理ロジックは、インデックス  $k$  をインクリメントする。そして、処理ロジックは、 $k$  がセット  $S$  の要素の数より小さいかをテストする（処理ブロック 6 1 8）。 $k$  がセット  $S$  の要素の数より小さい場合には、前述のプロセスが、SPGM 4 0 1 から受け取られるセット  $S$  に含まれる全ての空間位置に対応するブロックについて繰り返される。こうして、セット  $B$  からの処理する現ブロック  $c$  の各々について、CSM 4 0 2 は、前述したように、アレイ  $candidate\_flags$  における対応の値をメモリ 4 0 3 に書き込む。

20

【 0 0 6 0 】

セット  $S$  における全ての空間位置に対応するブロックの処理を終えると、このアレイにおける各値について、処理ロジックは、対応する動きベクトル  $v(k)$  及び変換モデルパラメータ  $a(k)$  をメモリ 4 0 3 に書き込んでいる。例えば、CSM 4 0 2 による現ブロック  $c$  の処理の完了時には、メモリ 4 0 3 は、以下の表に示すようなデータを含む（単なる一例である）。

【表 1】

表 1 : CSM 4 0 2 によってメモリ 4 0 3 に書き込まれたサンプルデータ

S における $k$ 番目の位置	1	2	3	4	5	6	...
$candidate\_flags(k)$	1	1	0	1	0	1	...
動きベクトル $v(k)$	$mv(1)$	$mv(2)$	$mv\_c$	$mv(4)$	$mv\_c$	$mv(6)$	...
model_params $m(k)$	$a(1)$	$a(2)$	$a\_c$	$a(4)$	$a\_c$	$a(6)$	...

30

【 0 0 6 1 】

これらデータに加えて、CSM 4 0 2 における処理ロジックは、また、メモリ 4 0 3 に、現ブロック  $c$  に対応するベクトル  $mv\_c$  及びパラメータ  $a\_c$  を、前述したように書き込む。これらデータは、二つの動きベクトル及びパラメータアレイ  $v$  及び  $m$  における 0 番目のインデックスに書き込まれる。即ち、 $v(0)$  が  $mv\_c$  に等しく、 $m(0)$  が  $a\_c$  に等しくされる（処理ブロック 6 1 9）。値  $candidate\_flags(0)$  は 1 に等しく、処理されている現ブロック  $c$  に対応している。従って、表 1 の例におけるメモリ 4 0 3 に記憶されるデータは、次の表 2 のように増補される。

40

【表 2】

表 2 : CSM 4 0 2 によってメモリ 4 0 3 に書き込まれた増補サンプルデータ

インデックス $j$	0	1	2	3	4	5	6	...
$Candidate\_flags(j)$	1	1	1	0	1	0	1	...
モーションベクトル $v(j)$	$mv\_c$	$mv(1)$	$mv(2)$	$mv\_c$	$mv(4)$	$mv\_c$	$mv(6)$	...
model_params. $m(j)$	$a\_c$	$a(1)$	$a(2)$	$a\_c$	$a(4)$	$a\_c$	$a(6)$	...

【 0 0 6 2 】

現フレームにおいて空間位置  $(i_0, j_0)$  を有する処理すべき非イントラ現ブロック  $c$  について、SMC モジュール 4 0 4 は、メモリ 4 0 3 から、CSM 4 0 2 によって前述

50

したように生成された対応のアレイcandidate\_flagsを読み取る。現ブロックc内のデータサンプルの二次元空間位置は、 $i, j$ によってインデックスされる。したがって、ブロックcにおけるサンプルのフレーム位置は、 $(i_0 + i, j_0 + j)$ によって与えられる。現ブロックcにおける各サンプル $c(i, j)$ について、SMC404は、動き補償予測サンプルのセットを、参照フレームから、次のように取り出す。参照フレームにおける予測サンプルの位置は、CSMプロセスによって生成されたようなメモリ403に記憶された動きベクトルによって決定される。ある実施の形態では、SPGM401によって生成されたセットSに含まれる空間位置は、現ブロックの左側、上側、右側及び下側の隣接ブロックを含む現ブロックの $K = 4$ の近傍ブロックに対応する。この場合には、図8に示すように番号付けされた四つの近傍ブロックがある。従って、 $k = 1 \dots 4$ によってインデックスされるS内の四つの位置がある。CSMプロセスによって生成されるように、 $k = 1 \dots 4$ として四つの対応する動きベクトル $v(k)$ 及び四つの変換モデルパラメータ $m(k)$ が、メモリ403に記憶されている。さらに、メモリ403は、現フレームにおける位置 $(i_0, j_0)$ での現ブロック用の動きベクトル値 $v(0)$ 及びモデルパラメータ $m(0)$ を含んでいる。この場合には、ブロックcにおける各サンプル $c(i, j)$ について、SMC404は、三つの動き補償データサンプルを、参照フレームから、次のように取り出す。予測サンプル $p(1)$ は、参照フレームRにおける位置 $(i_0 + i + v_x(0), j_0 + j + v_y(0))$ にて動き補償サンプルを取り出すことによって得られる。ここで、 $v_x(0), v_y(0)$ は、動きベクトル $v(0)$ の水平及び垂直成分を表している。

$$p(1) = R(i_0 + i + v_x(0), j_0 + j + v_y(0))$$

ある実施の形態では、ブロックのサイズは、 $M \times M$ である。従って、カウンタ変数 $i$ 及び $j$ は、1からMまでの値をとることによって、ブロックのサンプルを二次元的にインデックスする。ある実施の形態では、残りの二つの予測サンプル $p(2)$ 及び $p(3)$ は、参照フレームRから、次の論理的条件に従って、 $k = 1 \dots 4$ の動きベクトル $v(k)$ を使用することによって、取り出される。

【数1】

If ( $i > 1$  AND  $i \leq M/2$ ) AND ( $j > 1$  AND  $j \leq M/2$ )

$$p(2) = R(i_0 + i + v_x(1), j_0 + j + v_y(1))$$

$$p(3) = R(i_0 + i + v_x(2), j_0 + j + v_y(2))$$

Else if ( $i > 1$  AND  $i \leq M/2$ ) AND ( $j > M/2$  AND  $j \leq M$ )

$$p(2) = R(i_0 + i + v_x(2), j_0 + j + v_y(2))$$

$$p(3) = R(i_0 + i + v_x(3), j_0 + j + v_y(3))$$

Else if ( $i > M/2$  AND  $i \leq M$ ) AND ( $j > 1$  AND  $j \leq M/2$ )

$$p(2) = R(i_0 + i + v_x(1), j_0 + j + v_y(1))$$

$$p(3) = R(i_0 + i + v_x(4), j_0 + j + v_y(4))$$

Else if ( $i > M/2$  AND  $i \leq M$ ) AND ( $j > M/2$  AND  $j \leq M$ )

$$p(2) = R(i_0 + i + v_x(3), j_0 + j + v_y(3))$$

$$p(3) = R(i_0 + i + v_x(4), j_0 + j + v_y(4))$$

ここで、 $v_x(k)$ 及び $v_y(k)$  ( $k = 1 \dots 4$ )は、動きベクトル $v(k)$ の水平及び垂直成分である。参照フレームRから検索され、且つ、現ブロックcにおける各サンプル $c(i, j)$ に対応する予測サンプル $p(l)$  ( $l = 1 \dots 3$ )は、メモリ405に記憶される。別の実施の形態では、セットSにおける値によって特定されるような位置を有する異なる数のブロックの動きベクトルをSMC404によって使用して、現ブロッ

ク c における各サンプルに対する動き補償予測サンプル p ( 1 ) を取り出す。

【 0 0 6 3 】

M C P M 2 1 3 の入力で処理される現ブロック c の符号化モードが I N T R A でない場合には、予測モジュール 4 0 6 は、現ブロック c において座標 ( i , j ) を有するデータサンプルの各々に対する予測信号を、次のように生成する。各サンプル c ( i , j ) に対する予測信号は、S M C 4 0 4 について前述したようなプロセスに従って S M C 4 0 4 によって生成されてメモリ 4 0 5 に記憶された対応の予測サンプル p ( 1 ) を使用することによって、算出される。現ブロック用にメモリ 4 0 3 に記憶されたモデルパラメータ m ( 0 ) 及び現ブロックに関連するアレイ candidate\_flags に対応のパラメータ m ( k ) ( k = 1 . . . 4 ) もまた、予測モジュール 4 0 6 によってメモリ 4 0 3 から取り出される。ある実施の形態では、予測モジュール 4 0 6 は、次いで、次の論理的条件に基づいて、且つ、そのために予測信号 P を形成するブロック c におけるデータサンプルの位置 ( i , j ) に応じて、メモリ 4 0 3 から取り出したモデルパラメータを使用して、値 m f ( n ) ( n = 1 . . . 3 ) のセットを形成する。ある実施の形態では、この予測信号 P の算出に使用されるモデルパラメータは、次のように決定される。

10

【 数 2 】

mf(1)=m(0)

If (i>=1 AND i<=M/2) AND (j>=1 AND j<=M/2)

mf(2) = m(1)

20

mf(3) = m(2)

Else if (i>=1 AND i<=M/2) AND (j>M/2 AND j<=M)

mf(2) = m(2)

mf(3) = m(3)

Else if (i>M/2 AND i<=M) AND (j>=1 AND j<=M/2)

mf(2) = m(1)

mf(3) = m(4)

30

Else if (i>M/2 AND i<=M) AND (j>M/2 AND j<=M)

mf(2) = m(3)

mf(3) = m(4)

【 0 0 6 4 】

変換モデルパラメータ m f ( l ) ( l = 1 . . . 3 ) の各セットには、変換 T <sub>1</sub> ( . ) ( l = 1 . . . 3 ) が対応する。このとき、現サンプル c ( i , j ) に対する予測信号 P は、次のように、予測サンプルの線形結合として形成される。

【 数 3 】

$P(i,j)=(w1(i,j)*T_1(p(1))+w2(i,j)*T_2(p(2))+w3(i,j)*T_3(p(3)))/W,$

40

ここで、w 1 ( i , j ) 、 w 2 ( i , j ) 、 w 3 ( i , j ) は、重みであり、これらの値は、サイズ M × M の 2 次元アレイ w 1 、 w 2 、 w 3 に記憶されており、  
w 1 ( i , j ) + w 2 ( i , j ) + w 3 ( i , j ) = W  
である。

【 0 0 6 5 】

アレイ w 1 、 w 2 、 w 3 は、予測モジュール 4 0 6 によって、それらが記憶されているメモリ 4 0 3 から取り出される。前述したようなプロセスを、図 9 のフロー図に示す。

50

## 【 0 0 6 6 】

図 9 のフロー図において説明するプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

## 【 0 0 6 7 】

以下、図 9 を参照する。このプロセスは、処理ロジックが変数  $n$  を 1 にセットすることで始まる（処理ブロック 8 0 1）。次に、処理ロジックは、現ブロックにおける  $n$  番目の現サンプルのブロック位置  $(i, j)$  を決定する（処理ロジック 8 0 2）。ブロック位置を決定した後に、処理ロジックは、位置  $(i, j)$  に応じた予測サンプル  $p(i)$  をメモリ 4 0 5 から取り出す（処理ブロック 8 0 3）。処理ロジックは、また、モデルパラメータ  $m(i)$  を、位置  $(i, j)$  に基づいてメモリ 4 0 3 から取り出す（処理ブロック 8 0 4）。この情報を使用して、処理ロジックは、各予測信号  $p(i)$  に、 $m(i)$  のパラメータによって表される変換  $T$  を適用し（処理ブロック 8 0 5）、変換予測信号  $T(p(i))$  の線形結合を使用して、予測信号  $P$  を算出する（処理ブロック 8 0 6）。

10

## 【 0 0 6 8 】

予測信号を算出した後、処理ロジックは、変数  $n$  を 1 だけインクリメントし（処理ブロック 8 0 7）、 $n$  がそのブロックにおけるサンプルの数より小さいかをテストする（処理ブロック 8 0 8）。 $n$  がサンプルの数より小さい場合には、処理ロジックは処理ブロック 8 0 2 へ移行し、プロセスが現ブロックにおける次のサンプルに対して繰り返される。 $n$  がサンプルの数以上の場合には、プロセスは終了する。

20

## 【 0 0 6 9 】

別の実施の形態では、現フレームにおける各ブロックについて、サンプルレベルのモデルパラメータ  $m_p(i, j)$  ( $i, j = 1 \dots M$ ) が、次のように、メモリ 4 0 5 に記憶されたブロックモデルパラメータ  $m(l)$  から決定される（現ブロックの近傍ブロックは、図 7 に示した通りにラベル付けされる）。

## 【 数 4 】

If  $(i \geq 1 \text{ AND } i \leq M/2) \text{ AND } (j \geq 1 \text{ AND } j \leq M/2)$

$$mp(i,j) = (h1(i,j)*m(0) + h2(i,j)*m(1) + h3(i,j)*m(2))/H$$

30

Else if  $(i \geq 1 \text{ AND } i \leq M/2) \text{ AND } (j > M/2 \text{ AND } j \leq M)$

$$mp(i,j) = (h1(i,j)*m(0) + h2(i,j)*m(2) + h3(i,j)*m(3))/H$$

Else if  $(i > M/2 \text{ AND } i \leq M) \text{ AND } (j \geq 1 \text{ AND } j \leq M/2)$

$$mp(i,j) = (h1(i,j)*m(0) + h2(i,j)*m(1) + h3(i,j)*m(4))/H$$

Else if  $(i > M/2 \text{ AND } i \leq M) \text{ AND } (j > M/2 \text{ AND } j \leq M)$

$$mp(i,j) = (h1(i,j)*m(0) + h2(i,j)*m(3) + h3(i,j)*m(4))/H$$

40

ここで、 $h_1$ 、 $h_2$ 、 $h_3$  の値は、メモリ 4 0 5 における  $M \times M$  のアレイに記憶されており、

$$h_1(i, j) + h_2(i, j) + h_3(i, j) = H$$

である。

## 【 0 0 7 0 】

現フレームのブロック  $k$  におけるサンプル  $c(i, j)$  用の変換モデルパラメータ  $m_p(i, j)$  ( $i, j = 1 \dots M$ ) のセットには、変換のセット  $S^k_{i, j}(\cdot)$  ( $i, j = 1 \dots M$ ) が対応する。

50

## 【 0 0 7 1 】

現フレームにおける現ブロックを、図 8 におけるような近傍ブロックのラベリングを有するものと考え、ある実施の形態では、ブロック c における現サンプル c ( i , j ) に対する予測信号 P ( i , j ) は、次のように形成される。

## 【 数 5 】

If (i>=1 AND i<=M/2) AND (j>=1 AND j<=M/2)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(1))+w2(i,j)* S^1_{ij}(p(2))+w3(i,j)* S^2_{ij}(p(3)))/W,$$

Else if (i>=1 AND i<=M/2) AND (j>M/2 AND j<=M)

10

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(1))+w2(i,j)* S^2_{ij}(p(2))+w3(i,j)* S^3_{ij}(p(3)))/W,$$

Else if (i>M/2 AND i<=M) AND (j>=1 AND j<=M/2)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(1))+w2(i,j)* S^1_{ij}(p(2))+w3(i,j)* S^4_{ij}(p(3)))/W,$$

Else if (i>M/2 AND i<=M) AND (j>M/2 AND j<=M)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(1)) + w2(i,j) * S^3_{ij}(p(2)) + w3(i,j)* S^4_{ij}(p(3)))/W,$$

20

ここで、 $S^k_{i,j}(\cdot)$  は、現フレームにおける所与の現ブロックに対して図 8 におけるようにラベル付けされたブロック k に対応する変換を表している。前述したプロセスを、図 10 のフロー図に示す。

## 【 0 0 7 2 】

図 10 のプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

## 【 0 0 7 3 】

以下、図 10 を参照する。このプロセスは、処理ロジックがインデックス変数 k を 1 に等しくセットすることで始まる（処理ブロック 901）。処理ロジックは、次いで、ブロック k における全ての位置 ( i , j ) に対するサンプルレベル変換パラメータ  $m_p(i, j)$  を算出する（処理ブロック 902）。次に、処理ロジックは、変数 k を 1 だけインクリメントし（処理ブロック 903）、k の値が現フレームにおけるブロックの数より小さいかをテストする（処理ブロック 904）。k がブロックの数より小さい場合には、処理ロジックは、処理ブロック 902 へ移行し、プロセスは、サンプルレベル変換パラメータを全てのブロックについて算出するまで、繰り返される。k がブロックの数以上である場合には、プロセスは、処理ブロック 905 へ移行し、当該ブロック 905 において、変数 n を 1 に初期化する。

30

## 【 0 0 7 4 】

次いで、処理ロジックは、現ブロックにおける n 番目の現サンプルのブロック位置 ( i , j ) を決定する（処理ブロック 906）。ブロック位置を決定した後に、処理ロジックは、位置 ( i , j ) の予測サンプル p ( i ) をメモリ 405 から取り出し（処理ブロック 907）、位置 ( i , j ) のモデルパラメータ  $m_p(i, j)$  をメモリ 403 から取り出す（処理ブロック 908）。これらのメモリから取り出されたデータを使用して、処理ロジックは、 $m_p(i, j)$  によるパラメータ化された変換  $S^l_{i,j}(\cdot)$  ( l が 0 . . . 4 ) を各予測信号に適用し（処理ブロック 909）、変換予測信号  $S^l_{i,j}(p(i))$  の線形結合を使用して予測信号 P ( i , j ) を算出する（処理ブロック 910）。

40

## 【 0 0 7 5 】

予測信号を算出した後、処理ロジックは、変数 n を 1 だけインクリメントし（処理プロ

50

ック 9 1 1)、n の値が現ブロックにおけるサンプルの数より小さいかをテストする(処理ブロック 9 1 2)。n がサンプルの数より小さい場合には、処理は、処理ブロック 9 0 6 へ移行し、プロセスが、現ブロックにおける全てのサンプルに対して繰り返される。n がサンプルの数以上の場合には、プロセスは終了する。

【 0 0 7 6 】

別の実施の形態では、現ブロック c におけるサンプル c ( i , j ) に対応する予測信号 P ( i , j ) は、次のように算出される ( S <sup>k</sup> <sub>i , j</sub> ( . ) は、前述したものと同一手法で算出される )。

【 数 6 】

If (i>=1 AND i<=M/2) AND (j>=1 AND j<=M/2) 10

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(l))+w2(i,j)* S^1_{ij}(P(l))+w3(i,j)* S^2_{ij}(p(l)))/W,$$

Else if (i>=1 AND i<=M/2) AND (j>M/2 AND j<=M)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(l))+w2(i,j)* S^2_{ij}(P(l))+w3(i,j)* S^3_{ij}(p(l)))/W,$$

Else if (i>M/2 AND i<=M) AND (j>=1 AND j<=M/2)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(l))+w2(i,j)* S^1_{ij}(p(l))+w3(i,j)* S^4_{ij}(p(l)))/W,$$

Else if (i>M/2 AND i<=M) AND (j>M/2 AND j<=M)

$$P(i,j)=(w1(i,j)* S^0_{ij}(p(l))+w2(i,j)* S^3_{ij}(p(l))+w3(i,j)* S^4_{ij}(p(l)))/W, 20$$

ここで、S <sup>k</sup> <sub>i , j</sub> ( . ) は、現フレームにおける所与の現ブロックに対して図 8 におけるようにラベル付けされたブロック k に対応する変換を表している。前述したプロセスを、図 1 1 のフロー図に示す。

【 0 0 7 7 】

図 1 1 のプロセスは、ハードウェア(回路、専用ロジック等)、ソフトウェア(汎用コンピュータシステム又は専用マシンにて実行されるような)、又は両者の組み合わせを備え得る処理ロジックによって実行される。

【 0 0 7 8 】 30

以下、図 1 1 を参照する。このプロセスは、処理ロジックが、インデックス変数 k を 1 に等しくセットすることで始まる(処理ブロック 1 0 0 1)。処理ロジックは、次いで、ブロック k における全ての位置 ( i , j ) に対するサンプルレベル変換パラメータ m p ( i , j ) を算出する(処理ブロック 1 0 0 2)。次に、処理ロジックは、変数 k を 1 だけインクリメントし(処理ブロック 1 0 0 3)、k の値が現フレームにおけるブロックの数より小さいかをテストする(処理ブロック 1 0 0 4)。k がブロックの数より小さい場合には、処理ロジックは、処理ブロック 1 0 0 2 へ移行し、プロセスは、全てのブロックに対してサンプルレベル変換パラメータを算出するまで、繰り返される。k がブロックの数以上である場合には、プロセスは、処理ブロック 1 0 0 5 へ移行し、当該ブロック 1 0 0 5 において、変数 n を 1 に初期化する。 40

【 0 0 7 9 】

変数 n を初期化した後に、処理ロジックは、現ブロックにおける n 番目の現サンプルのブロック位置 ( i , j ) を決定する(処理ブロック 1 0 0 6)。ブロック位置を決定した後に、処理ロジックは、現ブロックにおける現サンプル c ( i , j ) に対応する予測サンプル p をメモリ 4 0 5 から取り出し(処理ブロック 1 0 0 7)、位置 ( i , j ) 用のモデルパラメータ m p ( i , j ) をメモリ 4 0 3 から取り出す(処理ブロック 1 0 0 8)。これらメモリから取り出したデータを使用して、処理ロジックは、m p ( i , j ) によってパラメータ化された変換 S <sup>1</sup> <sub>i , j</sub> ( . ) を各予測信号 p に適用し(処理ブロック 1 0 0 9)、変換予測信号 S <sup>1</sup> <sub>i , j</sub> ( p ) の線形結合を使用して、予測信号 P ( i , j ) を算出する(処理ブロック 1 0 1 0)。

## 【 0 0 8 0 】

予測信号を算出した後に、処理ロジックは、変数  $n$  を 1 だけインクリメントし（処理ブロック 1 0 1 1）、 $n$  の値が現ブロックにおけるサンプルの数より小さいかをテストする（処理ブロック 1 0 1 2）。 $n$  がサンプルの数より小さい場合には、プロセスは、処理ブロック 1 0 0 6 へ移行して、現ブロックにおける全てのサンプルに対して繰り返される。 $n$  がサンプルの数以上である場合には、プロセスは、終了する。

## 【 0 0 8 1 】

別の実施の形態では、変換  $S^k_{i,j}(\cdot)$  は、前述したように決定される。しかしながら、現ブロックにおけるサンプル  $c(i,j)$  に対する予測信号  $P(i,j)$  は、次のように決定される。

## 【 数 7 】

$$P(i,j) = S^0_{ij}(p(l)).$$

前述したプロセスを、図 1 2 のフロー図によって示す。

## 【 0 0 8 2 】

図 1 2 のプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

## 【 0 0 8 3 】

以下、図 1 2 を参照する。プロセスは、処理ロジックが、インデックス変数  $k$  を 1 に等しくセットすることで始まる（処理ブロック 1 1 0 1）。処理ロジックは、次いで、ブロック  $k$  における全ての位置  $(i,j)$  に対するサンプルレベル変換パラメータ  $m_p(i,j)$  を算出する（処理ブロック 1 1 0 2）。次に、処理ロジックは、変数  $k$  を 1 だけインクリメントし（処理ブロック 1 1 0 3）、 $k$  の値が現フレームにおけるブロックの数より小さいかをテストする（処理ブロック 1 1 0 4）。 $k$  がブロックの数より小さい場合には、プロセスは、処理ブロック 1 1 0 2 に移行して、全てのブロックに対してサンプルレベル変換パラメータを算出するまで、繰り返される。 $k$  がブロックの数以上である場合には、プロセスは、処理ブロック 1 1 0 5 に移行し、当該ブロック 1 1 0 5 において、変数  $n$  を 1 に初期化する。

## 【 0 0 8 4 】

変数  $n$  を初期化した後に、処理ロジックは、現ブロックにおける  $n$  番目の現サンプルのブロック位置  $(i,j)$  を決定する（処理ブロック 1 1 0 6）。ブロック位置を決定した後に、処理ロジックは、現ブロックにおける現サンプル  $c(i,j)$  に対応する予測サンプル  $p$  をメモリ 4 0 5 から取り出し（処理ブロック 1 1 0 7）、現サンプル  $c(i,j)$  に対応するモデルパラメータ  $m_p(i,j)$  をメモリ 4 0 3 から取り出す（処理ブロック 1 1 0 8）。これらのメモリから取り出されたデータを使用して、処理ロジックは、 $m_p(i,j)$  によってパラメータ化された変換  $S^1_{i,j}(\cdot)$  を各予測信号に適用し（処理ブロック 1 1 0 9）、変換予測信号  $S^1_{i,j}(p)$  の線形結合を使用して、予測信号  $P(i,j)$  を算出する（処理ブロック 1 1 1 0）。

## 【 0 0 8 5 】

予測信号を演算した後、処理ロジックは、変数  $n$  を 1 だけインクリメントし（処理ブロック 1 1 1 1）、 $n$  の値が現ブロックにおけるサンプルの数より小さいかをテストする（処理ブロック 1 1 1 2）。 $n$  がサンプルの数より小さい場合には、プロセスは、処理ブロック 1 1 0 6 に移行し、当該ブロック 1 1 0 6 において、プロセスが、現ブロックにおける全てのサンプルに対して繰り返される。 $n$  がサンプルの数以上である場合には、プロセスは、終了する。

## 【 0 0 8 6 】

図 1 3 は、ビデオ復号器の一つの実施の形態のブロック図である。以下、図 1 3 を参照する。ビデオ復号器 1 2 0 0 は、変換復号器（TD）1 2 0 1、フレームストア FS 1 2

10

20

30

40

50

02 (メモリ)、動き補償及び予測モジュール(MCPM)1203、動きデータ処理モジュールMDPM1204、メモリ1205、及びスイッチSW7、SW8、SW9、SW10を備えている。変換復号器1201は、エントロピー復号器1210、逆量子化器1211、及び逆変換モジュール1212を有している。変換復号器1201は、符号化されたフレームデータにおける全てのブロックを復号する。より詳細には、変換復号器1201は、符号化データを受け取り、そのフレームがビデオ符号化器によってINTRAモードにて符号化されている場合には、従来のイントラフレーム復号プロセスに従って再構成されたビデオフレームを生成する。復号されるべき現フレームがINTRAでない場合には、ビデオ復号器は、以下に説明するようなプロセスに従ってビデオフレームデータを再構成する。

10

#### 【0087】

エントロピー復号器1210は、符号化データ1220を受け取り、復号する。エントロピー復号器1210の出力は、量子化変換係数1221である。SW8が閉じられているときに、量子化変換係数1221は、逆量子化装置1211によって量子化されて、再構成変換係数1222が生成される。逆変換1212は、再構成変換係数1222に逆変換を実行し、再構成フレームデータを生成する。再構成フレームデータ1223は、コンパイナ1230へ出力され、当該コンパイナ1230は、再構成フレームデータ1223を、SW7が閉じられている場合に、MCPM1203からのフィルタリングされた予測信号1225と組み合わせる。コンパイナ1230及びビデオ復号器1200の出力は、再構成ビデオフレーム1224である。また、再構成ビデオフレーム1224は、フレームストア1202に記憶される。

20

#### 【0088】

MCPM1203は、SW1が閉じられているときに、フレームストア1202からの一以上の参照フレーム、及びメモリ1205からのデータに基づいてフィルタリングされた予測信号1225を生成する。このデータは、SW10が閉じられているときにエントロピー復号器1210からメモリ1205へ与えられるモデル(変換)パラメータ、及びMDPM1204からの動きベクトルを含む。MDPM1204は、SW9が閉じられているときに、エントロピー復号器1210から受け取った動きベクトル情報に応答して動きベクトルを生成する。

30

#### 【0089】

図14は、図13のビデオ復号器によって行われるビデオ復号処理の一つの実施の形態のフロー図である。VDプロセスを、符号化フレームタイプがINTRAでない場合について示している。初期的には、SW7、SW8、SW9は開であり、SW10は閉である。エントロピー復号器は、復号されるビデオフレームにおける現ブロックのブロック符号化モードを復号する。ブロックモードがINTRAである場合には、モード情報は、メモリ1205へ送られる。そして、現フレームにおける次のブロックの符号化モードが復号される。現ブロックの符号化モードがINTRAでない場合には、SW7が開、SW8が開、SW9が閉、SW10が閉、SW11が開とされる。エントロピー復号器1210は、現ブロックに対応する動きベクトル及び変換パラメータ情報を復号し、次いで、それらを現ブロックの符号化モードと共に、メモリ1205へ送る。次いで、現フレームにおける次のブロックが処理される。前述したようなプロセスは、復号されるべき現フレームの全てのブロックに対して繰り返される。次に、現フレームにおける全てのブロックを次のように処理するプロセスが開始される。この場合には、SW8は、閉、SW9は、開、SW10は、開、SW11は、閉であり、SW7は、現ブロックの符号化モードがINTRAである場合には、開であり、他の場合には、閉とされる。フレームにおける現ブロックの符号化モードがINTRAである場合には、変換復号器は、従来の手法でそのINTRAブロックを復号し、再構成ブロックデータをフレームストア1202へ送る。復号される現ブロックの符号化モードがINTRAでない場合には、変換復号器1201は、現フレームにおける現ブロック位置に対応する予測エラーブロックを復号する。また、MCPM1203は、メモリ1205に記憶されたブロック符号化モード、動きベクトル、及び

40

50

変換パラメータを取り出すことによって、ビデオ符号化器において説明したMCPMプロセスに従って現ブロックに対応する予測信号を算出する。次いで、再構成現ブロックデータが、変換復号器1201によって復号された予測エラーブロックを、MCPM1203によって生成された予測信号に加えることによって、得られる。この再構成現ブロックデータは、フレームストア1202に書き込まれる。前述したようなプロセスは、復号されるフレームにおける全てのブロックに対して繰り返される。

【0090】

図14は、予測されるフレームに対するビデオ復号プロセスの一つの実施の形態のフロー図である。このプロセスは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシンにて実行されるような）、又は両者の組み合わせを備え得る処理ロジックによって実行される。

10

【0091】

以下、図14を参照する。このプロセスは、処理ロジックが、インデックス*i*を1に等しく初期化することで始まる（処理ブロック1301）。インデックス*i*を使用して、処理ロジックは、現ブロック*c*(*i*)の符号化モードを復号し（処理ブロック1302）、その符号化モードがINTRAであるかをテストする（処理ブロック1303）。符号化モードがINTRAである場合には、処理は、直接、処理ブロック1306へ移行する。符号化モードがINTRAでない場合には、処理ロジックは、現ブロック*c*(*i*)の動きベクトル及びモデルパラメータを復号し（処理ブロック1304）、符号化モード、動きベクトル、及びモデルパラメータをメモリ1205へ送り、次いで、処理ブロック1306へ進む。

20

【0092】

処理ブロック1306において、処理ロジックは、インデックス*i*を1だけインクリメントし、インデックスがブロック(*N*)の総数より小さいかをテストする（処理ブロック1307）。インデックスがブロックの総数より小さい場合には、処理ロジックは、処理ブロック1302に移行し、当該ブロック1302において、プロセスが次のブロックに対して繰り返される。インデックスがブロックの総数以上である場合には、処理ロジックは、処理ブロック1308へ移行し、当該ブロック1308において、インデックス*i*が再び1へと初期化される。

【0093】

インデックス*i*が1へ初期化された状態で、処理ロジックは、現ブロック*c*(*i*)の符号化モードをメモリ1205から取り出し（処理ブロック1309）、その符号化モードがINTRAであるかをテストする（処理ブロック1310）。符号化モードがINTRAである場合には、処理ロジックは、現ブロック*c*(*i*)のイントラ復号を実行し（処理ブロック1313）、次いで、処理ブロック1315へ移行する。符号化モードがINTRAでない場合には、処理ロジックは、現ブロックの予測エラー*e*(*i*)を復号し（処理ブロック1311）、MCPM1203において現ブロックの予測信号*p*(*i*)を算出し（処理ブロック1312）、等式 $rc(i) = p(i) + e(i)$ に従って現ブロックを再構成し（処理ブロック1314）、処理ブロック1315へ移行する。

30

【0094】

処理ブロック1315において、処理ロジックは、再構成データをフレームストア1202に書き込む。その後、処理ロジックは、インデックス*i*をインクリメントし（処理ブロック1316）、インデックス*i*がブロックの数(*N*)より小さいかをテストする（処理ブロック1317）。*i* < *N*である場合には、プロセスは、処理ブロック1309へ移行し、残っているブロックに対して繰り返される。*i* < *N*でない場合には、プロセスは終了する。

40

【0095】

本明細書で説明した方法及び装置を使用して、従来技術の対応手法と比較して、優れた性能を有する時間予測信号を決定することができる。

【0096】

50

< コンピュータシステムの実施例 >

図15は、本明細書で説明したオペレーションのうちの一以上のオペレーションを実行し得る典型的なコンピュータシステムのブロック図である。以下、図15を参照する。コンピュータシステム1400は、典型的なクライアント又はサーバコンピュータシステムであってもよい。コンピュータシステム1400は、情報を伝送するための通信機構、即ちバス1411、及びバス1411に結合された情報処理用のプロセッサ1412を備えている。プロセッサ1412は、例えば、Pentium(登録商標)、PowerPC(登録商標)等のようなマイクロプロセッサを含むが、これに限定されるものではない。

【0097】

システム1400は、更に、情報及びプロセッサ1412によって実行されるべき命令を記憶するための、バス1411に結合されたランダムアクセスメモリ(RAM)又は他のダイナミック記憶装置1404を備えている。主メモリ1404もまた、プロセッサ1412による命令の実行中に変数又は他の中間情報を一時的に記憶するために使用することができる。

10

【0098】

コンピュータシステム1400は、また、静的情報及びプロセッサのための命令を記憶するための、バス1411に結合された読取り専用メモリ(ROM)及び/又は他の静的記憶装置1406、磁気ディスク又は光ディスクのようなデータ記憶装置1407、並びに、それに対応するディスクドライブを備えている。データ記憶装置1407は、情報及び命令を記憶するために、バス1411に結合されている。

20

【0099】

さらに、コンピュータシステム1400は、コンピュータユーザに対して情報を表示するための、バス1411に結合された陰極線管(CRT)又は液晶ディスプレイのような表示装置1421に結合されていてもよい。また、プロセッサ1412へ情報及びコマンド選択を伝送するための、英数字及びその他のキーを含む英数字入力装置1422がバス1411に結合されていてもよい。更なるユーザ入力装置として、プロセッサ1412へのディレクション情報及びコマンド選択を伝送し、且つディスプレイ1421上のカーソル移動を制御するための、バス1411に結合されたマウス、トラックボール、トラックパッド、スタイラス、又はカーソルディレクションキーのようなカーソルコントロール1423がある。

30

【0100】

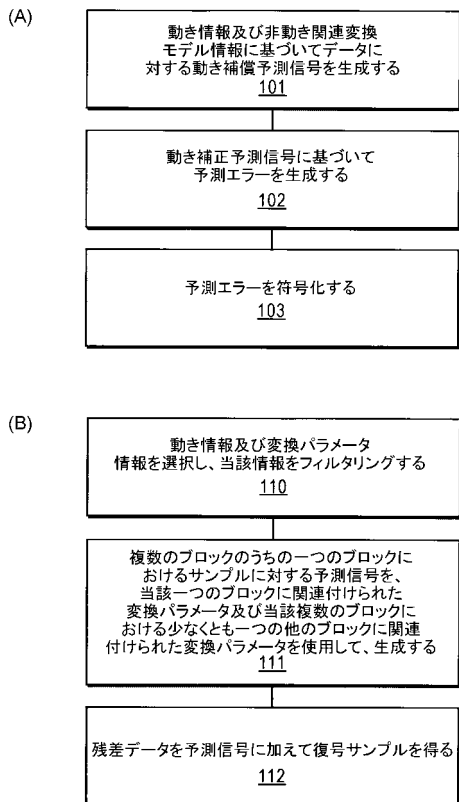
バス1411に結合される別の装置としては、ハードコピー装置1424がある。このハードコピー装置は、紙、フィルム、又は類似の種類媒体のような媒体に情報を記録するために使用される。バス1411に結合される更に別の装置としては、電話又はハンドヘルドパームデバイスと通信するための有線/無線通信機能1425がある。システム1400の構成要素、及びその関連ハードウェアのうちの何れか、又は全てを本発明において使用し得ることに留意されたい。しかしながら、コンピュータシステムの他のコン構成は、それら装置のうちの幾つか又は全てを含むようなものとすることができることも理解されよう。

【0101】

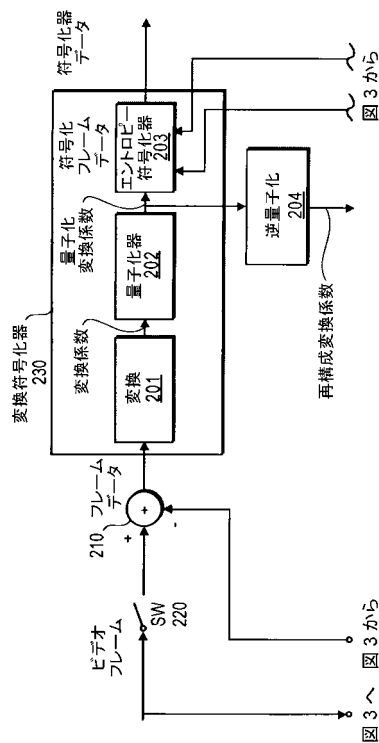
前述した説明を読めば、当業者には、本発明の多くの変更例及び変形例が当然に明らかとなるが、例示のために示し説明した特定の実施の形態は、これに限定することを意図したものではないことを理解されたい。従って、種々の実施の形態について詳細に言及したことは、特許請求の範囲を限定することを意図したものではなく、特許請求の範囲自体が、本発明に必須な特徴のみを列挙している。

40

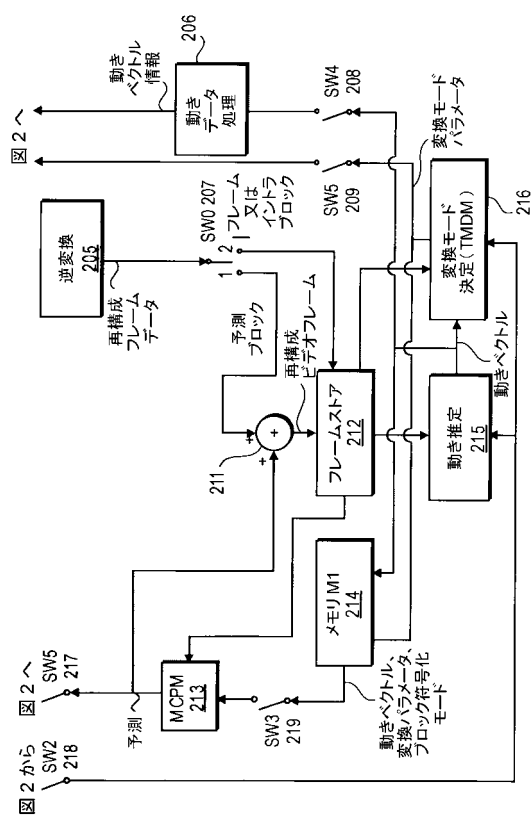
【図1】



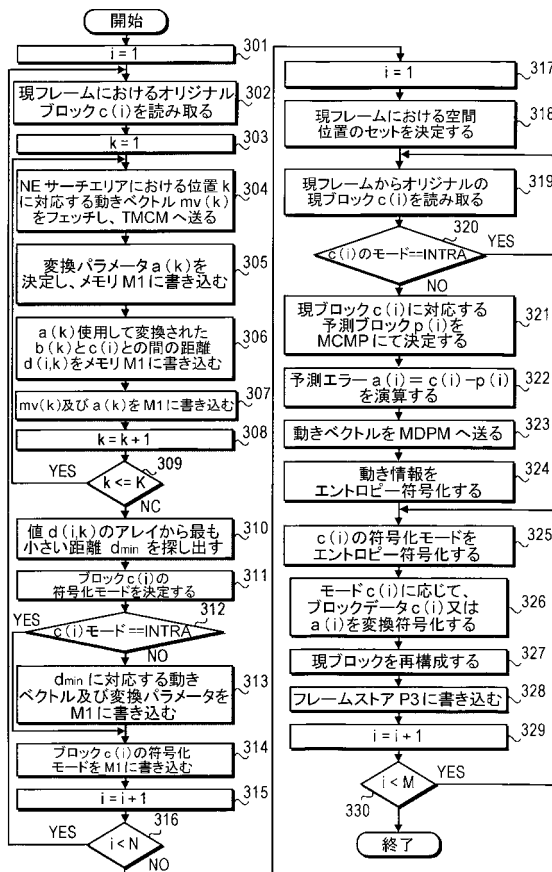
【図2】



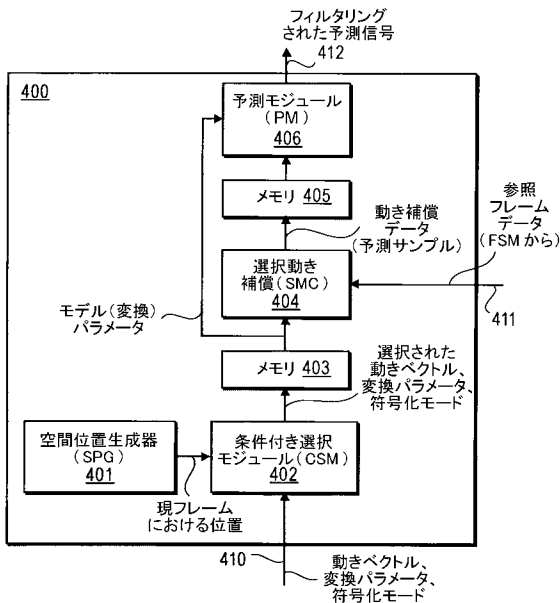
【図3】



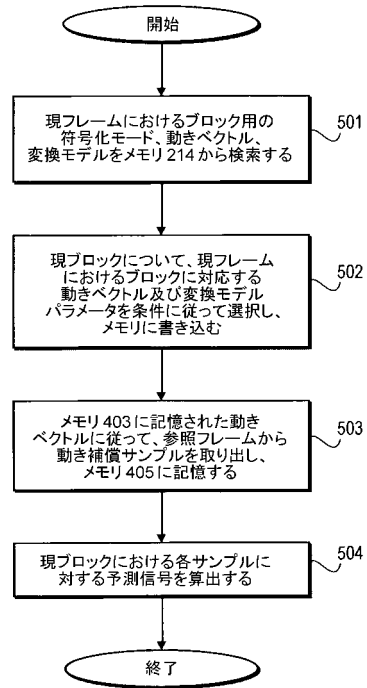
【図4】



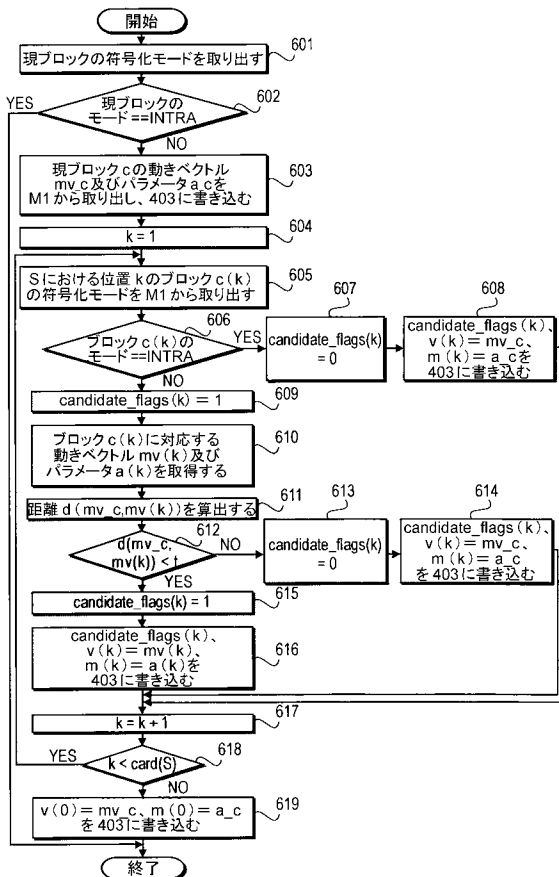
【 図 5 】



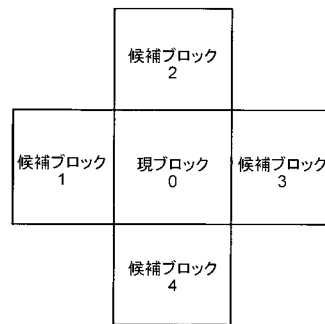
【 図 6 】



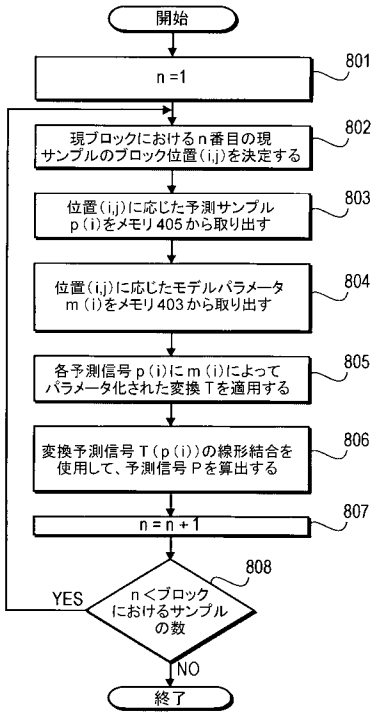
【 図 7 】



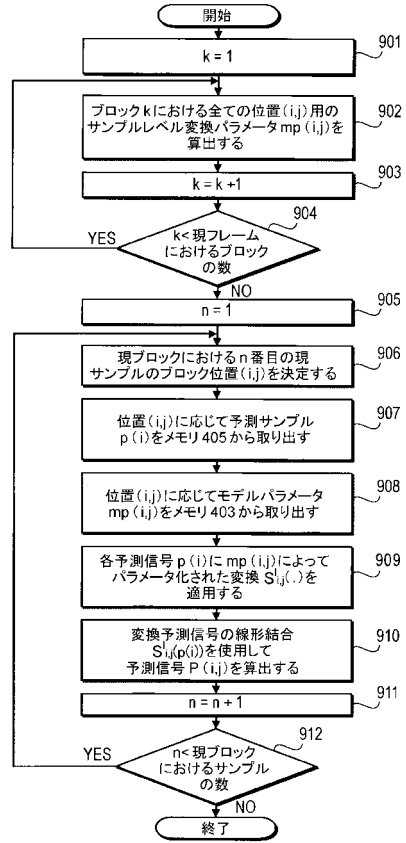
【 図 8 】



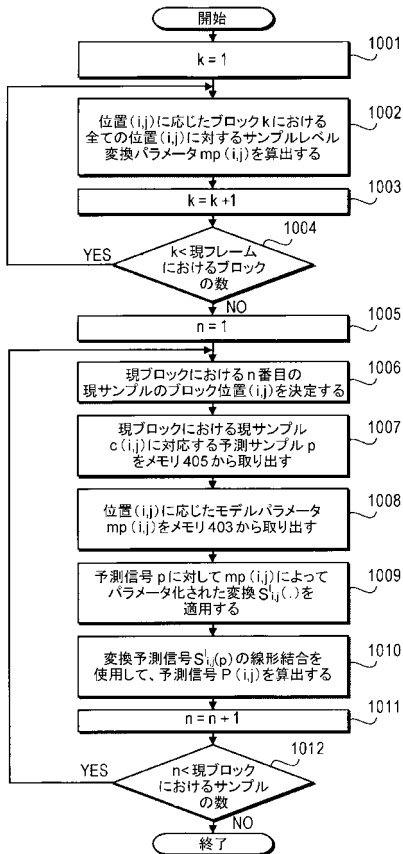
【 図 9 】



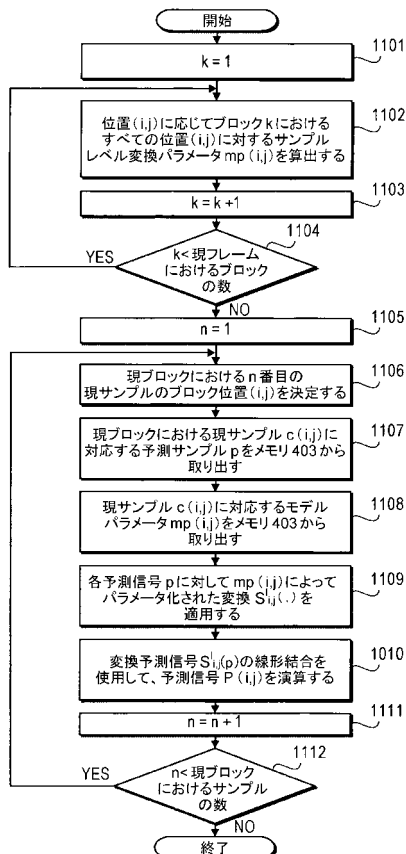
【 図 10 】



【 図 11 】



【 図 12 】





---

フロントページの続き

(72)発明者 ボッセン, フランク, ジャン

アメリカ合衆国, カリフォルニア州, サン ホゼ, スイート 300, メトロ ドライヴ  
181

Fターム(参考) 5C159 MA04 MA05 MA12 ME01 NN01 RC16 RC40 TA62 TB08 TC12  
TC35 TC42 TD11 UA02 UA05 UA38