

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4610195号
(P4610195)

(45) 発行日 平成23年1月12日(2011.1.12)

(24) 登録日 平成22年10月22日(2010.10.22)

(51) Int.Cl. F I
H04N 7/32 (2006.01) H04N 7/137 Z

請求項の数 18 (全 36 頁)

(21) 出願番号	特願2003-553839 (P2003-553839)	(73) 特許権者	500046438
(86) (22) 出願日	平成14年12月16日(2002.12.16)		マイクロソフト コーポレーション
(65) 公表番号	特表2005-513883 (P2005-513883A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成17年5月12日(2005.5.12)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2002/040208		クロソフト ウェイ
(87) 国際公開番号	W02003/053066	(74) 代理人	100077481
(87) 国際公開日	平成15年6月26日(2003.6.26)		弁理士 谷 義一
審査請求日	平成17年12月7日(2005.12.7)	(74) 代理人	100088915
(31) 優先権主張番号	60/341,674		弁理士 阿部 和夫
(32) 優先日	平成13年12月17日(2001.12.17)	(72) 発明者	スリドハー スリニヴァサン
(33) 優先権主張国	米国 (US)		アメリカ合衆国 98109 ワシントン
(31) 優先権主張番号	60/377,712		州 シアトル オーロラ アベニュー ノ
(32) 優先日	平成14年5月3日(2002.5.3)		ース 1504 ナンバー509
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 スキップマクロブロックコード化

(57) 【特許請求の範囲】

【請求項1】

コンピュータシステムにおいて、各々1つの基準ビデオピクチャから予測される複数の予測マクロブロックを含む、ビデオシーケンスの複数のビデオピクチャを処理する方法であって、

エントロピー符号化モード選択構文要素を含む、ビットストリーム内の符号化されたビデオを受信するステップと、

前記エントロピー符号化モード選択構文要素に基づいて、複数のエントロピーコード化モードから1つのエントロピーコード化モードを選択するステップと、

前記複数の予測マクロブロックの間の1つまたは複数のスキップマクロブロックについて、前記ビットストリームにおけるビットストリーム構文の層において、前記選択されたエントロピー符号化モードにしたがって、前記複数のエントロピーコード化モードのうちから選択された1つのエントロピーコード化モードで符号化されたスキップマクロブロック情報を復号するステップと、

前記1つまたは複数のスキップマクロブロックを復号するステップであって、前記1つまたは複数のスキップマクロブロックの1つの復号は、前記スキップマクロブロックの周辺の複数の他の予測マクロブロックの動きに基づき、該スキップマクロブロックについての予測される動きによる動き補償を使用し、および前記1つまたは複数のスキップマクロ

ブロックは各々残留情報を欠いているステップと

を備えたことを特徴とする方法。

10

20

【請求項 2】

前記ビットストリーム構文は、複数の階層を含み、前記スキップマクロブロック情報が示された前記層は、前記ビットストリーム構文におけるマクロブロック層よりも高位であることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記複数の階層は、少なくとも 1 つのピクチャ層およびマクロブロック層を有し、前記高位層は、前記ピクチャ層であることを特徴とする請求項 2 に記載の方法。

【請求項 4】

前記高位層のヘッダは、前記スキップマクロブロック情報を含むことを特徴とする請求項 2 に記載の方法。

【請求項 5】

前記スキップマクロブロックの予測される動きは、前記スキップマクロブロックの周辺の前記複数の他の予測マクロブロックの動きベクトルから導かれる予測動きベクトルに等しい動きベクトルを用いることを特徴とする請求項 1 に記載の方法。

【請求項 6】

コンピュータシステムにおいて、各々 1 つの基準ビデオピクチャから予測される複数の予測マクロブロックを含む、ビデオシーケンスの複数のビデオピクチャを処理する方法であって、

複数のエン트로ピーコード化モードから 1 つのエン트로ピーコード化モードを選択するステップと、

前記複数の予測マクロブロックの間の 1 つまたは複数のスキップマクロブロックを符号化するステップであって、前記 1 つまたは複数のスキップマクロブロックの 1 つの符号化は、前記スキップマクロブロックの周辺の複数の他の予測マクロブロックの動きに基づき、該スキップマクロブロックについての予測される動きによる動き補償を使用し、および前記 1 つまたは複数のスキップマクロブロックは各々残留情報を欠いているステップと、

前記 1 つまたは複数のスキップマクロブロックについて、ビットストリームにおけるビットストリーム構文の層において、前記複数のエン트로ピーコード化モードのうちから選択された 1 つのエン트로ピーコード化モードにしたがって、前記スキップマクロブロック情報を符号化するステップと、

前記選択された 1 つのエン트로ピーコード化モードを示すエン트로ピー符号化モード選択構文要素を含む、ビットストリーム内の符号化されたビデオを出力するステップとを備えたことを特徴とする方法。

【請求項 7】

前記ビットストリーム構文は、複数の階層を含み、前記複数の階層は、少なくとも 1 つのピクチャ層およびマクロブロック層を有し、前記スキップマクロブロック情報が示された前記層は、前記ビットストリーム構文におけるマクロブロック層よりも高位であり、前記高位層は、前記ピクチャ層であることを特徴とする請求項 6 に記載の方法。

【請求項 8】

前記スキップマクロブロック情報が示された前記層は、前記ビットストリーム構文におけるマクロブロック層よりも高位であり、前記高位層のヘッダは、前記スキップマクロブロック情報を含むことを特徴とする請求項 6 に記載の方法。

【請求項 9】

前記スキップマクロブロックについての予測される動きは、前記スキップマクロブロックの周辺の前記複数の別の予測マクロブロックの動きベクトルから導かれる予測動きベクトルに等しいことを特徴とする請求項 6 に記載の方法。

【請求項 10】

請求項 1 ないし 9 のいずれかに記載の方法をコンピュータシステムに実行させるためのコンピュータ実行可能命令を記憶したことを特徴とするコンピュータ読取可能な記録媒体。

【請求項 11】

10

20

30

40

50

複数のマクロブロックを有する1つまたは複数の予測ビデオピクチャを含む、ビデオシーケンスの複数のビデオピクチャを処理して、

複数のエントロピーコード化モードから1つのエントロピーコード化モードを選択する手段と、

前記1つまたは複数の予測ビデオピクチャの少なくとも1つのビデオピクチャの前記複数のマクロブロックの間の1つまたは複数のスキップマクロブロックを符号化する手段であって、前記1つまたは複数の予測ビデオピクチャの少なくとも1つのビデオピクチャは各々1つの基準ビデオピクチャから予測され、1つまたは複数のスキップマクロブロックの1つの符号化は、前記スキップマクロブロックの周辺の複数の他の予測マクロブロックの動きに基づき、該スキップマクロブロックについての予測される動きによる動き補償を使用し、および前記1つまたは複数のスキップマクロブロックは各々残留情報を欠いている手段と、

10

前記1つまたは複数のスキップマクロブロックについて、ビットストリームにおけるビットストリーム構文におけるマクロブロック層よりも高位層において、前記複数のエントロピーコード化モードのうちから選択された1つのエントロピーコード化モードにしたがって、前記スキップマクロブロック情報を符号化する手段と、

前記選択された1つのエントロピーコード化モードを示すエントロピー符号化モード選択構文要素を含む、ビットストリーム内の符号化されたビデオを出力する手段と

を備えたことを特徴とするシステム。

【請求項12】

20

前記スキップマクロブロックについての予測される動きは、前記スキップマクロブロックの周辺の前記複数の別の予測マクロブロックの動きベクトルから導かれる予測動きベクトルに等しいことを特徴とする請求項11に記載のシステム。

【請求項13】

エントロピー符号化モード選択構文要素を含む、ビットストリーム内の符号化されたビデオを受信する手段と、

各々1つの基準ビデオピクチャから予測される複数の予測マクロブロックを含む、ビデオシーケンスの複数のビデオピクチャを処理して、

前記エントロピー符号化モード選択構文要素に基づいて、複数のエントロピーコード化モードから1つのエントロピーコード化モードを選択する手段と、

30

前記複数の予測マクロブロックの間の1つまたは複数のスキップマクロブロックについて、前記ビットストリームにおけるビットストリーム構文の層において、前記選択されたエントロピー符号化モードにしたがって、前記複数のエントロピーコード化モードのうちから選択された1つのエントロピーコード化モードで符号化されたスキップマクロブロック情報を復号する手段と、

前記1つまたは複数のスキップマクロブロックを復号する手段であって、前記1つまたは複数のスキップマクロブロックの1つの復号は、前記スキップマクロブロックの周辺の複数の他の予測マクロブロックの動きに基づき、該スキップマクロブロックについての予測される動きによる動き補償を使用し、および前記1つまたは複数のスキップマクロブロックは各々残留情報を欠いている手段と

40

を備えたことを特徴とするシステム。

【請求項14】

前記ビットストリーム構文は、少なくとも1つのピクチャ層およびマクロブロック層を有する複数の階層を含むことを特徴とする請求項13に記載のシステム。

【請求項15】

前記ビットストリーム構文は、少なくとも1つのピクチャ層およびマクロブロック層を有する複数の階層を含み、前記スキップマクロブロック情報が示された前記層は、前記ビットストリーム構文におけるマクロブロック層よりも高位であり、前記高位層は、前記ピクチャ層であることを特徴とする請求項13に記載のシステム。

【請求項16】

50

前記スキップマクロブロック情報が示された前記層は、前記ビットストリーム構文におけるマクロブロック層よりも高位であり、前記高位層のヘッダは、前記スキップマクロブロック情報を含むことを特徴とする請求項 1 3 に記載のシステム。

【請求項 1 7】

前記 1 つまたは複数のスキップマクロブロックは、複数のスキップマクロブロックを含み、

前記複数のエントロピーコード化モードのうちの第 1 のエントロピーコード化モードにしたがって、前記ビットストリーム内の単一のコードが、前記複数のスキップマクロブロックのための前記スキップマクロブロック情報を表し、および

前記複数のエントロピーコード化モードのうちの第 2 のエントロピーコード化モードにしたがって、前記ビットストリーム内の複数のコードが、前記複数のスキップマクロブロックのための前記スキップマクロブロック情報を表すことを特徴とする請求項 1 に記載の方法。

10

【請求項 1 8】

前記 1 つまたは複数のスキップマクロブロックは、複数のスキップマクロブロックを含み、

前記複数のエントロピーコード化モードのうちの第 1 のエントロピーコード化モードにしたがって、前記ビットストリーム内の単一のコードが、前記複数のスキップマクロブロックのための前記スキップマクロブロック情報を表し、および

前記複数のエントロピーコード化モードのうちの第 2 のエントロピーコード化モードにしたがって、前記ビットストリーム内の複数のコードが、前記複数のスキップマクロブロックのための前記スキップマクロブロック情報を表すことを特徴とする請求項 6 に記載の方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

ビデオ符号化 / 復号応用分野でバイナリ情報を符号化 / 復号するための技法およびツールに関する。たとえば、ビデオエンコーダによって、スキップマクロブロック情報を符号化される。

【背景技術】

30

【0002】

本願は、2001年12月17日に出願された「Techniques and Tools for Video Encoding and Decoding」という名称の米国特許仮出願第60/341,674号を参照により組み込む。また、本願は、2002年5月3日に出願された「Skip Macroblock Coding」という名称の米国特許仮出願第60/377,712号を参照により組み込む。

【0003】

デジタルビデオは、大量の記憶容量と伝送容量を消費する。典型的な非圧縮デジタルビデオシーケンスには、1秒間に15フレームまたは30フレームが含まれる。各フレームは、数万または数十万の（ペルとも呼ばれる）画素を含む可能性がある。各画素は、画面の小さな要素を表す。非圧縮状態では、コンピュータは通常、24ビットで1画素を表す。したがって、典型的な非圧縮デジタルビデオシーケンスの1秒当たりビット数、またはビットレートは、5百万ビット / 秒以上にのぼる可能性がある。

40

【0004】

ほとんどのコンピュータおよびコンピュータネットワークは、非圧縮デジタルビデオを処理するほどの諸資源を有していない。そのため、技術者は、（コード化または符号化とも呼ばれる）圧縮を使用してデジタルビデオのビットレートを削減する。可逆（lossless）が可能であり、この場合ビデオの品質が損なわれないが、ビットレートの削減はビデオが複雑なため制限される。あるいは、非可逆（lossy）圧縮も可能であり、この場合ビデオの品質は損なわれるが、より劇的にビットレートが削減される。伸張は、圧縮の反対である。

50

【 0 0 0 5 】

一般に、ビデオ圧縮技法には、フレーム内圧縮とフレーム間圧縮がある。フレーム内圧縮技法は、一般にIフレームまたはキーフレームと呼ばれる個々のフレームを圧縮する。フレーム間圧縮技法は、一般に、予測フレーム、Pフレーム、またはBフレームと呼ばれる先行するフレームおよび/または次に来るフレームを参照して諸フレームを圧縮する。

【 0 0 0 6 】

Microsoft CorporationのWindows（登録商標）Media Videoバージョン7（WMV7）は、ビデオエンコーダとビデオデコーダを含む。WMV7エンコーダは、フレーム内圧縮とフレーム間圧縮を使用し、WMV7デコーダは、フレーム内伸張とフレーム間伸張を使用する。

10

【 0 0 0 7 】

A. WMV7内のフレーム内圧縮

図1は、WMV7エンコーダにおけるキーフレーム内画素のブロック（105）のブロックベースのフレーム内圧縮（100）を示す。ブロックは、1組の画素、たとえば、8×8配列の画素である。WMV7エンコーダは、キービデオフレームを画素の8×8ブロック群に分割し、8×8離散コサイン変換（DCT）（110）を、ブロック（105）など個々のブロックに適用する。DCTは、画素の8×8ブロック（空間情報）を、周波数情報であるDCT係数（115）の8×8ブロックに変換するある種の周波数変換である。DCT演算自体は可逆またはほぼ可逆である。しかし、元の画素値に比べて、DCT係数は、エンコーダが圧縮するのにより効率的である。というのは、有意な情報の大部分が低周波係数（従来、ブロック（115）の左上）に集中され、高周波係数（従来、ブロック（115）の右下）の多数がゼロまたはゼロに近い値を有するからである。

20

【 0 0 0 8 】

次いで、エンコーダは、DCT係数を量子化し（120）、量子化されたDCT係数の8×8ブロック（125）を生じる。たとえば、エンコーダは、均一なスカラー量子化ステップサイズを各係数に適用し、これは、各係数を同じ値および丸めによって割ることに似ている。たとえば、DCT係数値が163であり、ステップサイズが10である場合、量子化されたDCT係数値は16である。量子化は非可逆である。再構築されるDCT係数値は、163ではなく160となることになる。低周波DCT係数は、より高い値を有する傾向があるため、量子化により精度が失われるが、その係数についての情報が完全に失われることにはならない。一方、高周波DCT係数は、ゼロまたはゼロに近い値を有する傾向があるため、高周波係数を量子化すると、一般にゼロ値の連続領域になる。さらに、場合によっては、高周波DCT係数が低周波DCT係数より粗く量子化され、その高周波DCT係数についてより多くの精度/情報が失われる。

30

【 0 0 0 9 】

次いで、エンコーダは、可逆圧縮の一形態であるエントロピー符号化用に、量子化されたDCT係数（125）の8×8ブロックを用意する。エントロピー符号化の厳密なタイプは、係数がDC係数（低周波）であるか、それとも上端行または左列のAC係数（他の周波数）であるか、それとも別のAC係数であるかに応じて変わる可能性がある。

【 0 0 1 0 】

エンコーダは、符号化されるブロックの予め符号化済みの隣接ブロック（neighbor）（たとえば、上端または左）である隣接8×8ブロックのDC係数（136）との差分としてDC係数（126）を符号化する（図1は、フレーム内で符号化されるブロックの左に位置する隣接ブロック（135）を示す）。エンコーダは、その差分をエントロピー符号化する（140）。

40

【 0 0 1 1 】

エントロピーエンコーダは、AC係数の左列または上端行を、隣接する8×8ブロックの対応する列または行から差分として符号化することができる。図1は、（左に）隣接するブロック（135）の左列（137）から差分（147）として符号化されたAC係数の左列（127）を示す。差分符号化は、差分係数がゼロの値を有する機会を増大する。

50

残りのAC係数は、量子化されたDCT係数のブロック(125)からのものである。

【0012】

エンコーダは、予測され量子化されたAC DCT係数の8×8ブロックを1次元アレイ(155)の形に走査し(150)、次いで、ランレングス符号化(160)の一変形形態を使用して、走査されたAC係数をエントロピー符号化する。エンコーダは、エントロピーコードを1つまたは複数のラン/レベル/ラストテーブル(165)から選択し、エントロピーコードを出力する。

【0013】

キーフレームは、予測フレームよりはるかにビットレートに貢献する。低ビットレートまたは中ビットレートの応用分野では、キーフレームは、しばしば性能において決定的なボトルネックとなり、その結果、キーフレームの効率的な圧縮が重要である。

10

【0014】

図2は、図1に示されているものなどフレーム内圧縮の欠点を示す図である。具体的には、キーフレームのブロック間の冗長な利用は、ブロック(210)の左(220)または上端(230)隣接ブロックからの周波数係数のサブセット(たとえば、DC係数、およびAC係数の左列(または上端行))の予測に限定される。DC係数は、ブロックの平均を表し、AC係数の左列は、ブロックの行の平均を表し、上端行は、列の平均を表す。実際には、WMV7の場合のようにDCおよびAC係数の予測は、外挿を左(または上端)隣接ブロックの行毎(または列毎)平均信号に制限する。左ブロック(220)内の特定の行(221)の場合、左ブロック(220)についての左DCT係数列内のAC係数が、ブロック(210)の対応する行(211)全体を予測するために使用される。

20

【0015】

B. WMV7内のフレーム間圧縮

WMV7エンコーダのフレーム間圧縮は、ブロックベースの動き補償予測コード化と、それに続く残留誤差の変換コード化とを使用する。図3および4は、WMV7エンコーダにおける予測フレームについてのブロックベースのフレーム間圧縮を示す。具体的には、図3は、予測フレーム(310)についての動き推定を示し、図4は、予測フレームの動き推定済みブロックについての予測残差の圧縮を示す。

【0016】

WMV7エンコーダは、予測フレームを画素の8×8ブロック群に分割する。4個の8×8ブロックからなるグループによりマクロブロックが形成される。各マクロブロックについて、動き推定処理が実行される。動き推定により、基準フレーム、たとえば先にコード化された先行フレームに対して、画素のマクロブロックの動きが概算される。図3では、WMV7エンコーダは、予測フレーム(310)内のマクロブロック(315)について動きベクトルを計算する。動きベクトルを計算するために、エンコーダは、基準フレーム(330)の探索領域(335)内で探索する。探索領域(335)内で、エンコーダは、良好に合致する候補マクロブロックを見つけるために、予測フレーム(310)からのマクロブロック(315)を様々な候補マクロブロックと比較する。エンコーダは、エンコーダについて所望の動き推定分解能に応じて、探索領域(335)内で画素毎または1/2画素毎に候補マクロブロックを検査することができる。他のビデオエンコーダは、他の増分、たとえば1/4画素毎で検査する。ある候補マクロブロックについて、エンコーダは、予測フレーム(310)のマクロブロック(315)とその候補マクロブロックとの間の差異、および、そのマクロブロックについて動きベクトルを符号化するコストを検査する。エンコーダが良好に合致するマクロブロックを見つけた後に、ブロック合致処理は終了する。エンコーダは、合致マクロブロックについて(エントロピー符号化された)動きベクトルを出力し、その結果、デコーダは、復号中に合致マクロブロックを見つけることができる。予測フレーム(310)を復号するとき、デコーダはその動きベクトルを使用し、基準フレーム(330)からの情報を使用して、マクロブロック(315)についての予測マクロブロックを計算する。マクロブロックについての予測が完璧であることはまれであり、その結果、エンコーダは通常、予測マクロブロックとマクロブロック(

30

40

50

315) 自体との間の画素差の 8×8 ブロック (誤差または残留ブロックとも呼ばれる) を符号化する。

【0017】

図4は、WMV7エンコーダにおける動き推定済みブロックについての誤差ブロック (435) の計算および符号化を示す。誤差ブロック (435) は、予測ブロック (415) と元の現行ブロック (425) との間の差である。エンコーダは、DCT (440) を誤差ブロック (435) に適用して係数の 8×8 ブロック (445) を生成する。誤差ブロック (435) についての有意な情報は、画素値についてのDCT係数の場合よりもさらに、低周波係数 (従来、ブロック (445) の左上) に集中され、高周波係数の多数がゼロまたはゼロに近い値を有するのである (従来、ブロック (445) の右下)。

10

【0018】

次いで、エンコーダは、DCT係数を量子化し (450)、量子化されたDCT係数の 8×8 ブロック (455) を生成する。量子化ステップのサイズは、調整可能である。この場合も、低周波DCT係数は、より高い値を有する傾向があるため、量子化により精度が失われるが、その係数についての情報が完全に失われることにはならない。一方、高周波DCT係数は、ゼロまたはゼロに近い値を有する傾向があるため、高周波係数を量子化すると、ゼロ値の連続領域になる。さらに、場合によっては、高周波DCT係数が低周波DCT係数より粗く量子化され、その高周波DCT係数についてより多くの精度/情報が失われる。

【0019】

20

さらに、エンコーダは、エントロピー符号化用に、量子化されたDCT係数の 8×8 ブロック (455) を用意する。エンコーダは、 8×8 ブロック (455) を64要素有する1次元アレイ (465) の形に走査し (460)、その結果、係数は概して低周波から高周波に順序付けられ、これにより、典型的には、ゼロ値の長いランが生み出される。

【0020】

エンコーダは、ランレングス符号化 (470) の一変形態態を使用して、走査された係数をエントロピー符号化する。エンコーダは、エントロピーコードを1つまたは複数のラン/レベル/ラストテーブル (475) から選択し、エントロピーコードを出力する。

【0021】

マクロブロックについての動きベクトルがゼロ (すなわち、動きがない) であり、そのマクロブロックについて残留ブロック情報が送信されない場合、エンコーダは、そのマクロブロックについて1ビットのスキップマクロブロックフラグを使用する。これは、多くの種類のビデオコンテンツ (たとえば、動きの少ない、および/またはビットレートの低いビデオ) について、動きベクトルと残留ブロック情報の送信を回避することによってビットレートを削減する。エンコーダは、マクロブロックについての他の情報と共に、出力ビットストリーム内のマクロブロック層に、そのマクロブロックについてのスキップマクロブロックフラグを置く。

30

【0022】

図5は、フレーム間コード化済みブロックについての復号プロセス (500) を示す。DCT係数の量子化により、再構築されたブロック (575) は、対応する元のブロックと同一ではない。圧縮は非可逆となる。

40

【0023】

図5を要約すると、デコーダは、可変長符号化と、1つまたは複数のラン/レベル/ラストテーブル (515) とを使用し、予測残差を表すエントロピー符号化済み情報を復号する (510、520)。デコーダは、エントロピー符号化済み情報を格納する1次元アレイ (525) を2次元ブロック (535) の形に逆走査する (530)。デコーダは、そのデータを逆量子化および逆離散コサイン変換し (共に540)、再構築された誤差ブロック (545) を生成する。別個の経路で、デコーダは、基準フレームからの変位についての動きベクトル情報を使用し、予測ブロック (565) を計算する。デコーダは、予測ブロック (565) を再構築された誤差ブロック (545) と組み合わせ (570)、

50

再構築されたブロック(575)を形成する。

【0024】

デコーダは、あるマクロブロックについてスキップマクロブロックフラグを受信すると、そのマクロブロックについて予測の計算と残留ブロック情報の復号とをスキップする。その代わりに、デコーダは、基準フレーム内のマクロブロックの位置からの対応する画素データを使用する。

【0025】

元のフレームと再構築されたフレームとの間の変化量は歪みと呼ばれ、フレームをコード化するのに必要とされるビット数はレートと呼ばれる。歪みの量は、レートにおおよそ反比例する。換言すれば、より少ないビット(より高い圧縮率)でフレームをコード化すると、歪みがより多くなることになり、逆も同様である。ビデオ圧縮方式の目標の1つは、レート歪みを改善しようと試みることで、換言すれば、より少ないビットを使用して同じ歪み(あるいは、同じビット数で、より少ない歪み)を達成しようと試みることである。

10

【0026】

WMV7でスキップマクロブロックフラグを使用することにより、多くの種類のビデオコンテンツについてビットレートが削減されるが、場合によっては、最適とは言えない。多くの場合において、たとえば、スキップマクロブロックが映像内に集まって発生したとき、マクロブロックからマクロブロックにかけてスキップマクロブロックフラグの冗長性を利用できない。また、WMV7は、予測フレーム内のマクロブロックがスキップされると、そのマクロブロックについての動き予測も無視されるが、これは、場合によっては予測フレームの圧縮の効率を損なう。

20

【0027】

C. ビデオ圧縮および伸長のための標準

WMV7の他に、いくつかの国際標準がビデオ圧縮/伸長に関連して定められている。これらの標準には、MPEG(Motion Picture Experts Group)1、2、4標準、および国際電気通信連合(ITU)のH.261、H.262、H.263標準がある。WMV7と同様に、これらの標準は、フレーム内およびフレーム間圧縮の組合せを使用するが、これらの標準は一般に、使用される圧縮技法の詳細においてWMV7と異なる。

【0028】

いくつかの国際標準は、マクロブロックのスキッピング符号化を、ビデオ圧縮/伸長で使用すべきツールと考えている。各標準でのスキップマクロブロックコード化に関する追加の詳細については、各標準の仕様書自体を参照されたい。

30

【発明の開示】

【発明が解決しようとする課題】

【0029】

上記の標準におけるスキップマクロブロックコード化は、一般に、多くの種類のビデオコンテンツについてビットレートを削減するが、場合によっては、最適とは言えない。多くの場合において、たとえばスキップマクロブロックが所定の映像に固まって発生したとき、マクロブロックからマクロブロックにかけてスキップマクロブロックフラグの冗長性を利用できない。また、WMV7は、予測マクロブロック/映像内のマクロブロックがスキップされると、そのマクロブロックについて動き予測も無視されるが、これは、場合によっては予測マクロブロック/映像の圧縮の効率を損なうこととなる。

40

【0030】

デジタルビデオに対するビデオ圧縮/伸長の決定的な重要性を考えると、ビデオ圧縮/伸長が十分に研究開発されてきている分野であることは驚くべきことではない。しかしながら、従来のビデオ圧縮/伸長技法がどのような利益を有しているとしても、以下の技法およびツールの利点は有していない。

【課題を解決するための手段】

【0031】

50

要約すると、この詳細な説明は、バイナリ情報を（たとえば、ビデオエンコーダ/デコーダ内で）符号化/復号するための様々な技法およびツールを目的とする。バイナリ情報は、ビデオエンコーダまたはデコーダがビデオフレーム内のいくつかのマクロブロックをスキップするかどうかを示すビットを含むことができる。あるいは、バイナリ情報は、マクロブロックの動きベクトル分解能（たとえば、1 - MVまたは4 - MV）、インターレースモード（たとえば、フィールドまたはフレーム）、または何らかの他の情報を示すビットを含むことができる。バイナリ情報は、フレーム毎に、または何らかの他の基準に基づいて符号化することができる。

【0032】

いくつかの実施形態において、バイナリ情報は、ビットプレーン内に配置される。たとえば、ビットプレーンは、ピクチャ/フレーム層でコード化される。またはこれに替えて、バイナリ情報は、何らかの他の方法で構成され、および/または異なる層でコード化される。エンコーダ/デコーダは、このバイナリ情報を処理する。バイナリ情報は、マクロブロックレベルの情報を含むことができる。またはこれに替えて、エンコーダ/デコーダは、ブロックレベル情報、サブブロックレベル情報、または画素レベル情報のビットプレーンを処理する。

10

【0033】

いくつかの実施形態において、エンコーダ/デコーダは、コード化モードを切り換える。たとえば、エンコーダ/デコーダは、ノーマルモード、行スキップモード、列スキップモードを使用する。異なるモードにより、エンコーダ/デコーダは、バイナリ情報の冗長性を利用することができる。またはこれに替えて、エンコーダ/デコーダは、差分モードなど他のモードおよび/または追加のモードを使用する。効率を高めるために、エンコーダ/デコーダは、いくつかのモードでビットプレーン反転技法を使用することができる。

20

【0034】

いくつかの実施形態において、エンコーダ/デコーダは、動きがその因果的に予測された動きに等しい、また、残留誤差がゼロである予測マクロブロックとして、スキップマクロブロックを規定する。またはこれに替えて、エンコーダ/デコーダは、動きゼロ、残留誤差ゼロの予測マクロブロックとして、スキップマクロブロックを規定する。

【0035】

いくつかの実施形態において、エンコーダ/デコーダは、非圧縮コード化モードを使用し、低遅延応用例を可能にする。たとえば、非圧縮コード化モードでは、フレーム/ピクチャ内のマクロブロックすべてが符号化されるまで待つ必要なしに、符号化されたマクロブロックを直ちにデコーダに送信することができる。エンコーダ/デコーダは、非圧縮コード化モードと他のモードとの間で切り換えることができる。

30

【0036】

様々な技法およびツールを、組み合わせて、または独立して使用することができる。具体的には、本願では、スキップマクロブロック符号化/復号の2つの実施について、対応するビットストリーム構文と共に説明する。様々な実施形態において、説明する技法およびツールの1つまたは複数が実施される。

【0037】

追加の特徴および利点は、添付の図面を参照しながら進行する様々な実施形態の以下の詳細な説明から明らかになるであろう。

40

【発明を実施するための最良の形態】

【0038】

以下に説明する実施形態は、バイナリ情報を（たとえば、ビデオエンコーダ/デコーダ内で）符号化し、および復号するための技法およびツールに関する。バイナリ情報は、ビデオエンコーダまたはデコーダがビデオフレーム内のいくつかのマクロブロックをスキップするかどうかを示すビットを含むことができる。あるいは、バイナリ情報は、マクロブロックの動きベクトル分解能（たとえば、1 - MVまたは4 - MV）、インターレースモード（たとえば、フィールドまたはフレーム）、または何らかの他の情報を示すビットを

50

含むことができる。バイナリ情報は、フレーム毎に、または何らかの他の基準に基づいて符号化することができる。

【 0 0 3 9 】

いくつかの実施形態において、バイナリ情報は、ビットプレーンで構成される。ビットプレーンは、ピクチャ/フレーム層でコード化される。またはこれに替えて、バイナリ情報は、何らかの他の方法で構成され、および/または異なる層でコード化される。

【 0 0 4 0 】

いくつかの実施形態において、エンコーダ/デコーダは、コード化モードを切り換える。たとえば、エンコーダ/デコーダは、ノーマルモード、行スキップモード、列スキップモードを使用する。異なるモードにより、エンコーダ/デコーダは、バイナリ情報の冗長性を利用することができる。またはこれに替えて、エンコーダ/デコーダは、他のモードおよび/または追加のモードを使用する。

10

【 0 0 4 1 】

いくつかの実施形態において、エンコーダ/デコーダは、動きがその因果的に予測された動きに等しく、および残留誤差がゼロである予測マクロブロックとして、スキップマクロブロックを規定する。またはこれに替えて、エンコーダ/デコーダは、動きゼロ、残留誤差ゼロの予測マクロブロックとして、スキップマクロブロックを規定する。

【 0 0 4 2 】

いくつかの実施形態において、効率的なフレーム/ピクチャレベルのコード化の代わりに、低遅延の応用例を可能にするように非圧縮コード化モードが許される。非圧縮コード化モードでは、フレーム/ピクチャ内のマクロブロックすべてが符号化されるまで待つことなしに、符号化されたマクロブロックを直ちにデコーダに送信することができる。

20

【 0 0 4 3 】

いくつかの実施形態において、エンコーダ/デコーダは、マクロブロックレベル情報のビットプレーンを処理する。またはこれに替えて、エンコーダ/デコーダは、ブロックレベル情報、サブブロックレベル情報、または画素レベル情報のビットプレーンを処理する。

【 0 0 4 4 】

様々な技法およびツールを、組み合わせて、または独立して使用することができる。具体的には、本願では、スキップマクロブロック符号化/復号の2つの実装については、対応するビットストリーム構文と共に説明している。様々な実施形態において、説明している技法およびツールの1つまたは複数を実装される。

30

【 0 0 4 5 】

説明する実施形態において、ビデオエンコーダ/デコーダは、様々な技法を実装する。これらの技法のための動作は、一般に、提示の目的で特定の順番にそって順次説明するが、これには、特定の順序付けが必要とされない限り、この動作の順序を重要でない範囲で再構成した場合も包含されることを理解されたい。たとえば、順次説明する動作は、場合によっては、再構成し、または同時に実行することができる。さらに、話を簡単にするために、フローチャートには、一般に、特定の技法を他の技法と共に使用することができる様々な方法は示さない。

40

【 0 0 4 6 】

説明する実施形態において、ビデオエンコーダ/デコーダは、ビットストリーム内の様々なフラグおよび信号を使用する。ここで特定のフラグおよび信号について説明するが、これには、そのフラグおよび信号について様々な取決め(たとえば、1でなく0)が包含されることを理解されたい。

【 0 0 4 7 】

I . コンピューティング環境

図6は、説明する実施形態のいくつかを実施することができる好適なコンピューティング環境(600)の一般化された例を示す。この技法およびツールは、様々な汎用または専用コンピューティング環境で実施することができるため、コンピューティング環境(6

50

00)は、使用または機能の範囲についてのどんな制限も示唆もないものとする。

【0048】

図6を参照すると、コンピューティング環境(600)は、少なくとも1つの処理装置(610)およびメモリ(620)を含んでいる。図6では、この最も基本的な構成(630)が破線内に含まれている。処理装置(610)はコンピュータ実行可能命令を実行し、実プロセッサでも仮想プロセッサでもよい。多重処理システムでは、複数の処理装置がコンピュータ実行可能命令を実行し、処理力を高める。メモリ(620)は、揮発性メモリ(たとえば、レジスタ、キャッシュ、RAM)、不揮発性メモリ(たとえば、ROM、EEPROM、フラッシュメモリなど)、またはこれら2つの何らかの組合せとすることができる。メモリ(620)は、ビデオエンコーダまたはデコーダなど、エンコーダまたはデコーダを実施するソフトウェア(680)を記憶する。

10

【0049】

コンピューティング環境は、追加の特徴を有することができる。たとえば、コンピューティング環境(600)は、記憶装置(640)、1つまたは複数の入力デバイス(650)、1つまたは複数の出力デバイス(660)、1つまたは複数の通信接続(670)を含む。バス、コントローラ、ネットワークなど相互接続機構(図示せず)が、コンピューティング環境(600)の構成要素を相互接続する。一般に、オペレーティングシステムソフトウェア(図示せず)が、コンピューティング環境(600)内で実行する他のソフトウェアのための動作環境を提供し、コンピューティング環境(600)の構成要素の活動を調整する。

20

【0050】

記憶装置(640)は、取外し式または固定式とすることができ、磁気ディスク、磁気テープもしくはカセット、CD-ROM、DVD、または、情報を記憶するために使用することができる、また、コンピューティング環境(600)内でアクセスされる可能性がある任意の他の媒体を含む。記憶装置(640)は、エンコーダまたはデコーダを実施するソフトウェア(680)用の命令を記憶する。

【0051】

入力デバイス(650)は、キーボード、マウス、ペン、またはトラックボールなどタッチ入力デバイス、音声入力デバイス、走査デバイス、あるいは、コンピューティング環境(600)に入力を提供する別のデバイスとすることができる。オーディオまたはビデオ符号化の場合、入力デバイス(650)は、サウンドカード、ビデオカード、TV同調器カード、または、オーディオ入力もしくはビデオ入力をアナログ形態もしくはデジタル形態で受け入れる類似のデバイス、あるいは、オーディオサンプルまたはビデオサンプルをコンピューティング環境(600)内に読み取るCD-ROMまたはCD-RWとすることができる。出力デバイス(660)は、ディスプレイ、プリンタ、スピーカ、CDライター、または、コンピューティング環境(600)からの出力を提供する別のデバイスとすることができる。

30

【0052】

通信接続(670)は、通信媒体を介して別のコンピューティングエンティティに対する通信を可能にする。通信媒体は、コンピュータ実行可能命令、オーディオもしくはビデオの入力もしくは出力、または、変調データ信号内の他のデータなど、情報を搬送する。変調データ信号は、情報を信号に符号化するようにその特性の1つまたは複数が設定された、または変化した信号である。限定ではなく例を挙げると、通信媒体には、電気、光、RF、赤外線、音響、または他の搬送波と共に実施される有線技法または無線技法が含まれる。

40

【0053】

この技法およびツールについては、コンピュータ読取可能な媒体の一般的なコンテキストで述べるることができる。コンピュータ読取可能な媒体は、コンピューティング環境内でアクセスされる可能性がある任意の使用可能な媒体である。限定ではなく例を挙げると、コンピューティング環境(600)と共に、コンピュータ読取可能な媒体には、メモリ(

50

620)、記憶装置(640)、通信媒体、および上記のいずれかの組合せが含まれる。

【0054】

この技法およびツールについては、コンピューティング環境内で、ターゲットの実プロセッサまたは仮想プロセッサ上で実行される、プログラムモジュール内に含まれるものなどコンピュータ実行可能命令の一般的な状況で述べることができる。概して、プログラムモジュールは、特定のタスクを実行する、または特定の抽象データタイプを実施するルーチン、プログラム、ライブラリ、オブジェクト、クラス、コンポーネント、データ構造などを含む。プログラムモジュールの機能は、様々な実施形態で望まれるように、組み合わせることも、プログラムモジュール間で分割することもできる。プログラムモジュール用のコンピュータ実行可能命令は、ローカルまたは分散型コンピューティング環境内で実行

10

【0055】

提示の目的で、この詳細な説明では、「決定(判定)する」「選択する」「再構築する」「通知する」のような用語を使用して、コンピューティング環境内のコンピュータの動作について述べる。これらの用語は、コンピュータによって実行される動作について高レベルで抽象化したものであり、人間によって行われる動作と混同すべきでない。これらの用語に対応する実際のコンピュータの動作は、実施形態に応じて変わる。

【0056】

II. 一般化されたビデオエンコーダ/デコーダ

図7は、一般化されたビデオエンコーダ(700)のブロック図であり、図8は、一般化されたビデオデコーダ(800)のブロック図である。

20

【0057】

エンコーダおよびデコーダ内のモジュール間に示されている関係は、エンコーダおよびデコーダ内の情報の主な流れを示し、他の関係については、話を簡単にするため示さない。具体的には、図7および8は、通常、ビデオシーケンス、フレーム、マクロブロック、ブロックなどのために使用されるエンコーダ設定、モード、テーブルなどを示す副次情報を示していない。そのような副次情報は、典型的には、副次情報をエントロピー符号化した後に、出力ビットストリーム内で送られる。出力ビットストリームのフォーマットは、Windows(登録商標)Media Videoバージョン8または別のフォーマットとすることができる。

30

【0058】

エンコーダ(700)およびデコーダ(800)はブロックベースのものであり、4:2:0マクロブロックフォーマットを使用し、各マクロブロックは、4つの8x8輝度ブロック(1つの16x16マクロブロックとして扱うこともある)と、2つの8x8色差ブロックとを含む。またはこれに替えて、エンコーダ(700)およびデコーダ(800)はオブジェクトベースのものであり、8x8ブロックおよび16x16マクロブロックと異なるマクロブロックまたはブロックフォーマットを使用する、あるいは異なるサイズまたは構成の画素の集合に対して演算を実行する。

【0059】

実施形態および所望の圧縮のタイプとに応じて、エンコーダまたはデコーダのモジュールを、追加し、省略し、複数のモジュールに分割し、他のモジュールと組み合わせ、および/または同様なモジュールと置き換えることができる。代替の実施形態において、異なるモジュールおよび/またはモジュールの他の構成を有するエンコーダまたはデコーダによって、説明する技法の1つまたは複数を実装される。

40

【0060】

A. ビデオエンコーダ

図7は、一般的なビデオエンコーダシステム(700)のブロック図である。エンコーダシステム(700)は、現行フレーム(705)を含むビデオフレームのシーケンスを受信し、出力として圧縮済みビデオ情報(795)を生成する。ビデオエンコーダの特定の実施形態では、典型的には、一般化されたエンコーダ(700)を変形し、または補足

50

したバージョンが使用される。

【 0 0 6 1 】

エンコーダシステム (7 0 0) は、予測フレームとキーフレームを圧縮する。提示の目的で、図 7 は、エンコーダシステム (7 0 0) を介したキーフレームのための経路と、順方向予測フレームのための経路とを示す。エンコーダシステム (7 0 0) の構成要素の多くが、キーフレームと予測フレームを共に圧縮するために使用される。これらの構成要素によって実行される厳密な動作は、圧縮される情報のタイプに応じて変わる可能性がある。

【 0 0 6 2 】

予測フレーム (P フレーム、双方向予測用の B フレーム、または *i n t e r - c o d e d* フレームとも呼ばれる) は、1 つまたは複数の他のフレームからの予測 (または差) によって表される。予測残差は、予測されたものと元のフレームとの間の差である。一方、キーフレーム (I フレーム、*i n t r a - c o d e d* フレームとも呼ばれる) は、他のフレームを参照することなしに圧縮される。

【 0 0 6 3 】

現行フレーム (7 0 5) が順方向予測フレームである場合、動き推定器 (7 1 0) は、マクロブロック、または現行フレームの他の画素集合の動きを、基準フレームに対して推定する。基準フレームは、フレームストア (7 2 0) 内にバッファされた再構築済みの先のフレームである。代替の実施形態において、基準フレームとは後のフレームのことであり、あるいは現行フレームは双方向で予測されることとなる。動き推定器 (7 1 0) は、1 画素、1 / 2 画素、1 / 4 画素、またはその他の増分だけ動きを推定することができ、フレーム毎に、または他の基準に基づいて、動き推定の分解能を切り換えることができる。動き推定の分解能は、水平方向と垂直方向で同じ、または異なるものとすることができる。動き推定器 (7 1 0) は、動きベクトルなど動き情報 (7 1 5) を副次情報として出力する。動き補償器 (7 3 0) は、動き情報 (7 1 5) を再構築済みの先のフレーム (7 2 5) に適用し、動き補償済み現行フレーム (7 3 5) を形成する。しかし、予測が完璧であることはまれであり、動き補償済み現行フレーム (7 3 5) と元の現行フレーム (7 0 5) との間の差が予測残差 (7 4 5) である。またはこれに替えて、動き推定器と動き補償器は、別の種類の動き推定 / 補償を適用する。

【 0 0 6 4 】

周波数変換器 (7 6 0) は、空間領域のビデオ情報を周波数領域 (すなわち、スペクトル) データに変換する。ブロックベースのビデオフレームの場合、周波数変換器 (7 6 0) は、離散コサイン変換 (D C T) または D C T の変形形態を画素データまたは予測残差データに適用し、D C T 係数のブロックを生成する。またはこれに替えて、周波数変換器 (7 6 0) は、フーリエ変換など別の従来の周波数変換を適用する、または、ウェーブレットもしくはサブバンド解析を使用する。エンコーダが空間外挿 (図 7 には図示せず) を使用してキーフレームのブロックを符号化する実施形態において、周波数変換器 (7 6 0) は、スキュー D C T (*s k e w e d D C T*) など、新たに方向付けられた周波数変換をキーフレームの予測残差のブロックに適用することができる。他の実施形態において、周波数変換器 (7 6 0) は、8 x 8、8 x 4、4 x 8、または他のサイズの周波数変換 (たとえば、D C T) を予測フレームの予測残差に適用する。

【 0 0 6 5 】

次いで、量子化器 (7 7 0) がスペクトルデータ係数のブロックを量子化する。量子化器は、ステップサイズをフレーム毎に変化させ、または他の基準に基づいて変化させて、均一なスカラー量子化をスペクトルデータに適用する。またはこれに替えて、量子化器は、スペクトルデータ係数に別のタイプの量子化、たとえば非均一な量子化、ベクトル量子化、若しくは非適応量子化を適用し、または周波数変換を使用しないエンコーダシステム内の空間領域データを直接量子化する。適応量子化に加えて、エンコーダ (7 0 0) は、フレームドロップ、適応フィルタリング、またはレート制御を行う他の技法を使用することができる。

10

20

30

40

50

【 0 0 6 6 】

予測フレーム内の所与のマクロブロックがある種のタイプの情報を有していない（たとえば、そのマクロブロックについての動き情報がない、また、残差情報がない）場合、エンコーダ（700）は、そのマクロブロックを、スキップマクロブロックとして符号化することができる。その場合、エンコーダは、圧縮済みビデオ情報（795）の出力ビットストリーム内で、そのスキップマクロブロックを示す。

【 0 0 6 7 】

後続の動き推定／補償のために再構築済みの現行フレームが必要とされるとき、逆量子化器（776）が、量子化されたスペクトルデータ係数に対して逆量子化を実行する。次いで、逆周波数変換器（766）が周波数変換器（760）の動作の逆を実行し、再構築された（予測フレームについての）予測残差、または再構築されたキーフレームを生成する。現行フレーム（705）がキーフレームであったならば、再構築済みキーフレームは、再構築済み現行フレーム（図示せず）として取り上げられる。現行フレーム（705）が予測フレームであった場合、再構築された予測残差が動き補償済み現行フレーム（735）に追加され、再構築済み現行フレームを形成する。フレームストア（720）は、次のフレームを予測する際に使用するために、再構築済み現行フレームをバッファする。いくつかの実施形態において、エンコーダは、デブロッキングフィルタを再構築済みフレームに適用し、フレームのブロック内の不連続を調整して滑らかにする。

【 0 0 6 8 】

エントロピーコーダ（780）は、量子化器（770）の出力、ならびにある種の副次情報（たとえば、動き情報（715）、空間外挿モード、量子化ステップサイズ）を圧縮する。典型的なエントロピー符号化技法には、算術符号化、差分符号化、ハフマン符号化、ランレングス符号化、LZ符号化、辞書型符号化、および上記の組合せが含まれる。エントロピーコーダ（780）は、典型的には、異なる種類の情報（たとえば、DC係数、AC係数、異なる種類の副次情報）について異なるコード化技法を使用し、特定のコード化技法内の複数のコードテーブルの中から選択することができる。

【 0 0 6 9 】

エントロピーコーダ（780）は、圧縮済みビデオ情報（795）をバッファ（790）内に入れる。バッファレベルインジケータがビットレート適応モジュールに返される。

【 0 0 7 0 】

圧縮済みビデオ情報（795）は、一定の、または比較的一定のビットレートでバッファ（790）から送り出され、そのビットレートで後のストリームのために格納される。したがって、バッファ（790）のレベルは、主にフィルタされた量子化済みビデオ情報のエントロピーの関数であり、エントロピー符号化の効率に影響を及ぼす。またはこれに替えて、エンコーダシステム（700）は、圧縮済みビデオ情報を圧縮後すぐにストリームし、バッファ（790）のレベルは、やはり送信するために情報がバッファ（790）から全て送り出せるレートによって決まる。

【 0 0 7 1 】

バッファ（790）の前後には、圧縮済みビデオ情報（795）は、ネットワークを介して送信するためにチャンネル符号化することができる。チャンネル符号化は、誤り検出および補正データを圧縮済みビデオ情報（795）に適用することができる。

【 0 0 7 2 】

B . ビデオデコーダ

図8は、一般的なビデオデコーダシステム（800）のブロック図である。デコーダシステム（800）は、ビデオフレームの圧縮済みシーケンスについて情報を受信し、再構築済みフレーム（805）を含む出力を生成する。ビデオデコーダの特定の実施形態は、典型的には、一般化されたデコーダ（800）の変形形態または補足されたバージョンを使用する。

【 0 0 7 3 】

デコーダシステム（800）は、予測フレームとキーフレームを伸長する。提示の目的

10

20

30

40

50

で、図 8 には、デコーダシステム (800) を介したキーフレームのための経路と、順方向予測フレームのための経路とを示す。デコーダシステム (800) の構成要素の多くは、キーフレームと予測フレームを共に伸張するために使用される。これらの構成要素によって実行される厳密な動作は、伸張されている情報のタイプに応じて変わる可能性がある。

【0074】

バッファ (890) は、圧縮されたビデオシーケンスについての情報 (895) を受信し、受信された情報をエントロピーデコーダ (880) が使用できるようにする。バッファ (890) は、典型的には、経時的にかなり一定のレートで情報を受信し、帯域幅または伝送における短期的な変動を滑らかにするためのジッタバッファを含む。バッファ (890) は、再生バッファや他のバッファをも含むことができる。またはこれに替えて、バッファ (890) は、可変レートで情報を受信する。バッファ (890) の前後には、圧縮済みビデオ情報は、チャンネル復号し、誤り検出および補正のために処理することができる。

10

【0075】

エントロピーデコーダ (880) は、エントロピー符号化された量子化済みデータ、ならびにエントロピー符号化された副次情報 (たとえば、動き情報 (815)、空間外挿モード、量子化ステップサイズ) を復号し、典型的にはエンコーダ内で実行されたエントロピー符号化の逆を行う。エントロピー復号技法には、算術復号、差分復号、ハフマン復号、ランレングス復号、LZ 復号、辞書型復号、および上記の組合せが含まれる。エントロピーデコーダ (880) は、しばしば異なる種類の情報 (たとえば、DC 係数、AC 係数、異なる種類の副次情報) について異なる復号技法を使用し、特定の復号技法内の複数のコードテーブルの中から選択することができる。

20

【0076】

再構築すべきフレーム (805) が順方向予測フレームである場合、動き補償器 (830) は、動き情報 (815) を基準フレーム (825) に適用し、再構築されるフレーム (805) の予測 (835) を形成する。たとえば、動き補償器 (830) は、マクロブロック動きベクトルを使用し、基準フレーム (825) 内のマクロブロックを見つける。フレームバッファ (820) は、基準フレームとして使用するために、先に再構築されたフレームを格納する。動き補償器 (830) は、1 画素、1/2 画素、1/4 画素、または他の増分で動きを補償することができ、フレーム毎に、または他の基準に基づいて、動き補償の分解能を切り換えることができる。動き補償の分解能は、水平方向と垂直方向で同じものとし、または異なるものとすることができる。またはこれに替えて、動き補償器は、別のタイプの動き補償を適用する。動き補償器による予測が完璧であることはまれであり、その結果、デコーダ (800) は、予測残差をも再構築することとなる。

30

【0077】

デコーダが、後続の動き補償のために再構築済みフレームを必要とするときは、フレームストア (820) は、次のフレームを予測する際に使用するため、再構築済み現行フレームをバッファする。いくつかの実施形態において、エンコーダは、デブロッキングフィルタを再構築済みフレームに適用し、フレームのブロック内の不連続を調整して滑らかにする。

40

【0078】

逆量子化器 (870) は、エントロピー符号化済みデータを逆量子化する。一般に、逆量子化器は、ステップサイズをフレーム毎に変化させ、または他の基準に基づいて変化させて、均一なスカラ逆量子化をエントロピー符号化済みデータに適用する。またはこれに替えて、逆量子化器は、データに別のタイプの逆量子化、たとえば非均一な量子化、ベクトル量子化、もしくは非適応逆量子化を適用する、または逆周波数変換を使用しないデコーダシステム内の空間領域データを直接逆量子化する。

【0079】

逆周波数変換器 (860) は、量子化された周波数領域データを空間領域ビデオ情報に

50

変換する。ブロックベースのビデオフレームの場合、逆周波数変換器(860)は、逆DCT(IDCT)またはIDCTの変形変換をDCT係数のブロックに適用して、キーフレームまたは予測フレームについて画素データまたは予測残差データを生成する。またはこれに替えて、逆周波数変換器(860)は、逆フーリエ変換など別の従来の逆周波数変換を適用し、またはウェーブレットもしくはサブバンド合成を使用する。デコーダが空間外挿(図8には図示せず)を使用してキーフレームのブロックを復号する実施形態において、逆周波数変換器(860)は、スキューIDCT(skewed IDCT)など、新たに方向付けられた逆周波数変換をキーフレームの予測残差のブロックに適用することができる。他の実施形態において、逆周波数変換器(860)は、8×8、8×4、4×8、または他のサイズの逆周波数変換(たとえば、IDCT)を予測フレームの予測残差に適用する。

10

【0080】

スキップマクロブロックが、ビデオフレームの圧縮されたシーケンスについての情報(895)のビットストリーム内で示されているときは、デコーダ(800)は、非スキップマクロブロックについてビットストリーム内に通常含まれる情報(たとえば、動き情報および/または残差情報)を使用せずにスキップマクロブロックを再構築する。

【0081】

III. 第1の実施例

第1の実施例において、ビデオエンコーダとデコーダは、それぞれ改善された効率でスキップマクロブロック情報を符号化して復号する。スキップマクロブロック情報は、ビデオビットストリーム内のピクチャ層で示され、エンコーダはスキップマクロブロック情報内の冗長性を利用することが可能となる。また、エンコーダ/デコーダは、スキップマクロブロック情報を符号化/復号するために、複数のコード化モード間で選択する。

20

【0082】

A. スキップマクロブロック情報のピクチャ層コード化

第1の実施例において、圧縮されたビデオシーケンスは、4つの階層の形に構造化されたデータで構成される。これらの層は、上から下に1)シーケンス層、2)ピクチャ層、3)マクロブロック層、4)ブロック層である。ピクチャ層では、各ピクチャについてのデータが、ピクチャヘッダと、それに続くマクロブロック層についてのデータとからなる(同様に、マクロブロック層では、各マクロブロック層についてのデータが、マクロブロックヘッダと、それに続くブロック層とからなる)。IピクチャとPピクチャ用のビットストリーム要素には同一のものもあるが、Pピクチャだけに現れるものもあり、逆も同様である。

30

【0083】

図9は、Pピクチャ層(900)を構成するビットストリーム要素を示す。表1は、Pピクチャ層(900)のビットストリーム要素を簡単に述べている。

【0084】

【表 1】

フィールド	説明
PTYPE (910)	ピクチャタイプ
PQUANT (912)	ピクチャ量子化器スケール
SMBC (920)	スキップマクロブロックコード
SMB (930)	スキップマクロブロックフィールド
CPBTAB (940)	コード化済みブロックパターンテーブル
MVRES (942)	動きベクトル分解能
TTMBF (944)	マクロブロックレベルの変換タイプフラグ
TIFRM (946)	フレームレベルの変換タイプ
DCTACMBF (948)	マクロブロックレベルのDCT ACコード化設定フラグ
DCTACFRM (950)	フレームレベルのDCT ACコード化設定インデックス
DCTDCTAB (952)	Intra DCT DCテーブル
MVTAB (954)	動きベクトルテーブル
MB LAYER (960)	マクロブロック層

表 1 : 第 1 の実施例における P ピクチャ層のビットストリーム要素

【 0 0 8 5 】

具体的には、P ピクチャ層 (9 0 0) は、P ピクチャ内のマクロブロックのためのスキップマクロブロックフィールド (S M B) (9 3 0) と、そのスキップマクロブロックフィールド (9 3 0) についてコード化モードを示すスキップマクロブロックコード (S M B C) フィールド (9 2 0) とを備える。S M B C フィールド (9 2 0) は、P ピクチャヘッダ内だけに存在する。S M B C (9 2 0) は、フレーム内のスキップマクロブロックを示すために使用される 4 つのモードのうち 1 つを示す 2 ビット値である。第 1 の実施例において、スキップマクロブロックコード化モード群用の固定長コード (F L C) は、以下の通りである。

【 0 0 8 6 】

【表 2】

SMBC FLC	スキップビットコード化モード
00	スキップビットコード化なし
01	ノーマルスキップビットコード化
10	行予測 (または「行スキップ」) スキップビットコード化
11	列予測 (または「列スキップ」) スキップビットコード化

表 2 : 第 1 の実施例におけるスキップマクロブロックコード化モードのコードテーブル

【 0 0 8 7 】

コード化モードがノーマル、行予測、または列予測の場合には、ビットストリーム内の次のフィールドは、スキップマクロブロック情報を含む S M B フィールド (9 3 0) となる。したがって、S M B フィールドは、P ピクチャヘッダ内だけに存在し、および S M B C がノーマル、行予測、または列予測スキップマクロブロックコード化を示す場合だけ存在する。S M B C がノーマルコード化を示す場合には、S M B フィールドのサイズがフレーム内のマクロブロックの数に等しい。S M B C が行予測または列予測を示す場合には、以下に説明するように、S M B のサイズが可変である。

【 0 0 8 8 】

スキップマクロブロック情報は、フレーム内のどのマクロブロックがマクロブロック層に存在しないかについてデコーダに伝える。デコーダは、これらのマクロブロックについ

10

20

30

40

50

て、そのマクロブロックを再構築するとき対応するマクロブロック画素データを基準フレームからコピーすることになる。

【 0 0 8 9 】

B . スキップマクロブロック情報についてコード化モードを切り換える

上述のように、SMB Cフィールド(9 2 0)は、スキップマクロブロックフィールド(9 3 0)についてコード化モードを示す。より一般的に、図 1 0 は、複数のスキップマクロブロックコード化モードを有するビデオエンコーダ内でスキップマクロブロック情報を符号化するための技法(1 0 0 0)を示す。図 1 1 は、複数のスキップマクロブロックコード化モードを有するビデオエンコーダによって符号化されたスキップマクロブロック情報を復号するための対応する技法(1 1 0 0)を示す。

10

【 0 0 9 0 】

図 1 0 を参照すると、エンコーダは、スキップマクロブロック情報をコード化するために、スキップマクロブロックコード化モードを選択する(1 0 1 0)。たとえば、第 1 の実施例において、スキップマクロブロックコード化モード群には、どのマクロブロックもスキップされないモード、ノーマルモード、行予測(または「行スキップ」)モード、および列予測(または「列スキップ」)モードが含まれる。コード化モードが選択された後に、エンコーダは、そのスキップマクロブロック情報を符号化する(1 0 2 0)。エンコーダは、ピクチャ毎にコード化モードを選択する。またはこれに替えて、エンコーダは、何らかの他の基準に基づいて(たとえば、シーケンスレベルで)コード化モードを選択する。エンコーダがスキップマクロブロック情報を符号化し終わると(1 0 3 0)、符号化が終了する。

20

【 0 0 9 1 】

図 1 1 を参照すると、デコーダは、スキップマクロブロック情報を符号化するためにエンコーダによって使用されたスキップマクロブロックコード化モードを決定する(1 1 1 0)。次いで、デコーダは、そのスキップマクロブロック情報を復号する(1 1 2 0)。デコーダは、ピクチャ毎にコード化モードを決定する。またはこれに替えて、デコーダは、何らかの他の基準に基づいて(たとえば、シーケンスレベルで)コード化モードを決定する。デコーダがスキップマクロブロック情報を復号し終わると(1 1 3 0)、復号が終了する。

30

【 0 0 9 2 】

C . コード化モード

第 1 の実施例において、スキップマクロブロックコード化モード群には、どのマクロブロックもスキップされないモード、ノーマルモード、行予測(または「行スキップ」)モード、および列予測(または「列スキップ」)モードが含まれる。以下のセクションでは、スキップマクロブロックコード化フレームの例(1 2 0 0)を示す図 1 2 を参照しながら、各モードでどのようにスキップマクロブロック情報が符号化されるかについて説明する。

【 0 0 9 3 】

1 . ノーマルスキップマクロブロックコード化モード

ノーマルモードでは、各マクロブロックのスキップ/非スキップ状況がビットで表される。したがって、ビットユニットのSMBフィールドのサイズは、フレーム内のマクロブロックの数に等しい。SMBフィールド内のビット位置は、左上のマクロブロックで始まって、フレーム内のマクロブロックのラスト走査順に対応する。ビット値 0 は、対応するマクロブロックがスキップされないことを示し、ビット値 1 は、対応するマクロブロックがスキップされることを示す。

40

【 0 0 9 4 】

図 1 3 は、ノーマルスキップマクロブロックコード化モードで符号化するための技法(1 3 0 0)を示す図である。最初に、エンコーダは、マクロブロックのコード化がスキップされることになるかどうか検査する(1 3 1 0)。スキップされる場合、エンコーダは、SMBフィールドにビット値 1 を追加し、対応するマクロブロックがスキップされるこ

50

とを示す(1320)。そうでない場合には、エンコーダは、SMBフィールドにビット値0を追加し、対応するマクロブロックがスキップされないことを示す(1330)。エンコーダがSMBフィールドにビットを追加し終わると(1340)、スキップマクロブロックコード化が終了する。

【0095】

例として、ノーマルモードのコード化を使用すると、図12のフレーム例(1200)についてのSMBフィールドは、01001011111111111111010010と符号化されるはずである。

【0096】

2. 行予測スキップマクロブロックコード化モード

行予測モードでは、各マクロブロック行(上から下)の状況がビットで示される。ビットが1の場合には、その行が、全面的スキップマクロブロックを含み、次の行の状況が続く。ビットが0に等しい場合には、その行内の各マクロブロックについてのスキップ/非スキップ状況がビットで示される。したがって、長さが行内のマクロブロックの数に等しいビットフィールドが続く。ビットフィールド内のビット群は、左から右の順序でマクロブロックを表す。この場合も、値0は、対応するマクロブロックがスキップされないことを示し、値1は対応するマクロブロックがスキップされることを示す。

【0097】

図14は、行予測(または「行スキップ」)マクロブロックコード化モードで符号化するための技法(1400)を示す図である。最初に、エンコーダは、所定の行に全面的スキップマクロブロックが含まれるかどうか検査する(1410)。含まれる場合、エンコーダは、インジケータビット1をSMBフィールドに追加し(1420)、次の行の状況が続く。その行が全面的スキップマクロブロックを含んでいない場合、エンコーダはインジケータビット0をSMBフィールドに追加し、その行内の各マクロブロックについてのスキップ/非スキップ状況がビットで示される(1430)。エンコーダがフレーム内のすべての行を処理すると(1440)、行予測符号化が終了する。

【0098】

復号について説明すると、図15は、スキップマクロブロック情報の行予測復号を示す疑似コード(1500)を示す図である。疑似コード(1500)では、関数get_bits(n)がビットストリームからn個のビットを読み取り、その値を返す。

【0099】

例として、行予測モードのコード化を使用すると、図12のフレーム例(1200)についてのSMBフィールドは、0010010110010010と符号化されるはずである。

【0100】

3. 列予測スキップマクロブロックコード化モード

列予測モードでは、各マクロブロック列(左から右)の状況がビットで示される。ビットが1の場合には、その列は、全面的スキップマクロブロックを含み、次の列の状況が続く。ビットが0に等しい場合には、その列内の各マクロブロックについてのスキップ/非スキップ状況がビットで示される。したがって、長さが列内のマクロブロックの数に等しいビットフィールドが続く。ビットフィールド内のビット群は、上から下の順序でマクロブロックを表す。この場合も、値0は、対応するマクロブロックがスキップされないことを示し、値1は、対応するマクロブロックがスキップされることを示す。

【0101】

図16は、列予測(または「列スキップ」)マクロブロックコード化モードで符号化するための技法(1600)を示す。最初に、エンコーダは、全面的スキップマクロブロックを列が含むかどうか検査する(1610)。含む場合、エンコーダは、インジケータビット1をSMBフィールドに追加し(1620)、次の列の状況が続く。列が全面的スキップマクロブロックを含んでいない場合、エンコーダはインジケータビット0をSMBフィールドに追加し、その列内の各マクロブロックについてのスキップ/非スキップ状況が

10

20

30

40

50

ビットで示される(1630)。エンコーダがフレーム内の列すべてを処理したとき(1640)、列予測符号化が終了する。

【0102】

復号について説明すると、図17は、スキップマクロブロック情報の列予測復号を示す疑似コード(1700)を示す図である。

【0103】

例として、列予測モードのコード化を使用すると、図12のフレーム例(1200)についてのSMBフィールドは、0011010011000110100110と符号化されるはずである。

【0104】

IV. 第2の実施例

第2の実施例において、ビデオエンコーダとデコーダは、それぞれ改善された効率でスキップマクロブロック情報および/または他の2Dバイナリデータを符号化し、復号する。エンコーダ/デコーダは、(動きゼロとは限らない)デフォルトの動きを有するものとしてスキップマクロブロックを規定し、これにより、エンコーダ/デコーダは、多くの場合において、より多くのマクロブロックをスキップすることができる。ビットプレーンの効率的なフレームレベルコード化は、スキップマクロブロック情報および/または他の2Dバイナリデータを示す。また、エンコーダ/デコーダは、低遅延応用例のためにスキップマクロブロックの非圧縮(MBレベル)コード化オプションを使用することができる。

【0105】

A. スキップビット定義(スキップマクロブロックの定義)

第2の実施例は、スキップマクロブロックの概念の新しい定義を含む。「スキップ」は、そこまでの細分レベルで情報をさらに送信する必要がないビットストリームの場合の条件に関連する。スキップマクロブロック(ブロック)は、デフォルトのタイプ、デフォルトの動き、デフォルトの残留誤差を有するマクロブロック(ブロック)である(これに比較して、他の実施例および標準では、スキップマクロブロックとは、動きゼロ、残差ゼロの予測マクロブロックである)。

【0106】

スキップマクロブロックの新しい定義は、動きがその因果的に予測された動きに等しく、および残留誤差がゼロである予測マクロブロックであるということである(他の定義との相違点は、デフォルトの動きが動きプレディクタ(predictor)に等しく、これは必ずしもゼロでない可能性があることである)。

【0107】

たとえば、いくつかの実施形態において、現在のマクロブロックについての予測動きベクトルは、現在のマクロブロックの直接上方に、または直接左に向けられる。あるいは、プレディクタの水平成分および垂直成分は、現在のマクロブロックの左、上端、右上のマクロブロックの平行/水平成分毎の中央値から生成される。

【0108】

4つの動きベクトル(4MV)を有するスキップマクロブロックの動きベクトルは、自然な走査順で連続的に実行されたそれらのプレディクタによって与えられる。1つの動きベクトル(1MV)の場合と同様に、誤差残差はゼロである。

【0109】

図18は、スキップマクロブロックの新定義に従って、ビデオエンコーダ内で特定のマクロブロックのコード化をスキップするかどうか判定するための技法(1800)を示すフローチャートである。最初に、エンコーダは、現行フレームがIフレームであるか、それともPフレームであるか検査する(1810)。現行フレームがIフレームである場合、現行フレーム内のどのマクロブロックもスキップされず(1820)、そのフレームについてのスキップマクロブロックコード化が終了する。

【0110】

一方、現行フレームがPフレームである場合、エンコーダは、スキップすることができ

10

20

30

40

50

る現行フレーム内マクロブロックがないか検査する。所与のマクロブロックについて、エンコーダは、そのマクロブロックについての動きベクトルが、そのマクロブロックについて因果的に予測された動きベクトルに等しいかどうか（たとえば、そのマクロブロックについての差分動きベクトルがゼロに等しいかどうか）検査する（1830）。マクロブロックについての動きが因果的に予想された動きに等しくない場合、エンコーダは、そのマクロブロックをスキップしない（1840）。そうでない場合には、エンコーダは、そのマクロブロックについて符号化すべき残差があるかどうか検査する（1850）。コード化すべき残差がある場合、エンコーダは、そのマクロブロックをスキップしない（1860）。しかし、そのマクロブロックについて残差がない場合、エンコーダは、そのマクロブロックをスキップする（1870）。エンコーダは、符号化を終えるまで、引き続きマクロブロックを符号化またはスキップする（1880）。

10

【0111】

B. ビットプレーンコード化

第2の実施例において、（スキップマクロブロックを示すことを含む）ある種のマクロブロック特有の情報を、マクロブロック当たり1ビットで符号化することができる。フレーム内のマクロブロックすべてについての状況を、ビットプレーンとして一緒にコード化し、フレームヘッダ内で送信することができる。

【0112】

第2の実施例において、エンコーダは、フレーム内のマクロブロックについての情報を示すために、3つの場合でビットプレーンコード化を使用する。3つの場合とは、1) スキップマクロブロックを示すこと、2) フィールドまたはフレームマクロブロックモードを示すこと、および3) 各マクロブロックについて1MVまたは4MV動きベクトルモードを示すことである。このセクションは、3つの場合のいずれかについてのビットプレーンコード化、およびそれに対応する復号について述べる。

20

【0113】

フレームレベルのビットプレーンコード化は、2次元バイナリ配列を符号化するために使用される。各配列のサイズは、rowMB x colMBであり、rowMBおよびcolMBは、それぞれマクロブロックの行と列の数である。各配列は、ビットストリーム内で、1組の連続したビットとしてコード化される。表3に列挙するが、以下に説明するように、各配列を符号化するために7つのモードの1つが使用される。

30

【0114】

【表3】

コード化モード	説明
Raw	シンボル当たり1ビットとしてコード化される
ノーマル2	2つのシンボルを一緒にコード化
Diff-2	ビットプレーンの差分符号化と、それに続いて2つの残留シンボルを一緒にコード化すること
ノーマル6	6つのシンボルを一緒にコード化
Diff-6	ビットプレーンの差分符号化と、それに続いて6つの残留シンボルを一緒にコード化すること
行スキップ	設定ビットのない行を示すための1ビットスキップ
列スキップ	設定ビットのない列を示すための1ビットスキップ

40

表3：第2の実施例におけるコード化モード

【0115】

第2の実施例において、エンコーダは、ビットプレーン内に情報を埋め込むために、MODE、INVERT、DATABITSという3つの構文要素を使用する。

【0116】

MODEフィールドは、ビットプレーンについてのコード化を符号化する可変長符号（

50

VLC)である。たとえば、MODEフィールド内のVLCは、表3に列挙されている7つのコード化モードのいずれかを表す。ビットを節約するために、より可能性の高いコード化モードにより短いコードを、あまり可能性の高くないコード化モードにより長いコードを割り当てることができる。上記で指摘したように、MODEフィールドは、フレームヘッダ内で送信することができる。

【0117】

エンコーダ/デコーダは、フレーム毎にコード化モード間で切り換える。たとえば、エンコーダ/デコーダは、第1の実施例のエンコーダ/デコーダがそれぞれ図10および11において、スキップマクロブロックコード化モード間で切り換えたように、コード化モード間で切り換える。またはこれに替えて、エンコーダ/デコーダは、他の技法を使用して、および/または何らかの他の基準に基づいて切り換える。

10

【0118】

モードが非圧縮モードでない場合、1ビットのINVERTフィールドが送信される。条件反転を実施することができるいくつかのコード化モードでは、INVERTフィールドは、エンコーダ内で符号化が行われる前にビットプレーン内のビットを反転すべきかどうか、およびデコーダ内の復号の出力を反転すべきかどうかを示す。ビットプレーン内の大部分のビットが1に等しいときINVERTフィールドは1であり、ビットプレーン内の大部分のビットが0に等しいとき0である。エンコーダは、0がより多く存在するとき消費するビットがより少ないいくつかのコード化モード(ノーマル2およびノーマル6など)を使用する。符号化すべきビットプレーンで0より1が多い場合、エンコーダは、ビットプレーンを反転し、ビットプレーン内の0の割合を増やし、より多くのビットを節約する可能性が高くなる。他のモード(Diff-2およびDiff-6など)は、INVERTの値を使用し、プレディクタビットプレーンを計算する。したがって、いくつかのコード化モードでは、デコーダ部で再構築される最終ビットプレーンがINVERTによって決まる。

20

【0119】

MODEフィールドおよびINVERTフィールドが与えられれば、DATABITSフィールドは、ビットプレーンを再構築するのに必要な情報を含むVLCシンボルのエントロピー符号化済みストリームである。

【0120】

C. コード化モード

第2の実施例において、エンコーダは、行スキップモード、列スキップモード、ノーマル2モード、ノーマル6モード、Diff-2モード、Diff-6モード、非圧縮モードという7つの異なるコード化モードのいずれかでバイナリ情報(たとえば、スキップマクロブロック情報)を符号化する。デコーダは、7つのコード化モードのいずれかについて対応する復号を実行する。各モードについて以下で詳しく述べる。

30

【0121】

またはこれに替えて、エンコーダ/デコーダは、他の、および/または追加のコード化モードを使用する。

【0122】

1. 行スキップモードおよび列スキップモード

行スキップコード化モードは、行の各バイナリシンボルがある種の値のものである場合に、ビットプレーン内のビットを単一のビットで表すことによってビットを節約する。たとえば、エンコーダは、スキップモードをビットプレーン内の0で表し、全面的に0の行を単一のビットで表す行スキップコード化モードを使用する。したがって、エンコーダは、マクロブロックの行全体がスキップされるときビットを節約する。デコーダは、対応する復号を実行する。

40

【0123】

第2の実施例において、全面的にゼロの行は、0に設定された1ビットを使用して示される。行が全面的にゼロでないとき、1ビットインジケータは、1に設定され、この後に

50

、ビットプレーンを含む c o l M B ビットが順序正しく続く。行は自然な順序で走査される。

【 0 1 2 4 】

同様に、列スキップモードについては、列全体がゼロである場合、0ビットが送信される。そうでない場合は、1が送信され、その後、列全体を含む r o w M B ビットが順序正しく続く。列は自然な順序で走査される。

【 0 1 2 5 】

D i f f - 6 モードおよびノーマル6モード（後述）での残りの行および/または列のコード化については、同じ論理が適用される。1ビットフラグは、行または列が全面的ゼロであるかどうかを示す。全面的にゼロでない場合、シンボル当たり1つのビットを使用して行または列全体が送信される。

10

【 0 1 2 6 】

エンコーダが、主に1からなるビットプレーンを符号化するときは、行スキップおよび列スキップコード化は、通常、行/列が全体的に0からなる可能性がより低いため、あまり効率的でない。しかし、そのような場合に、エンコーダはビットプレーンに対して反転を実行して、0の割合を増やし、および潜在的にビット節約を高めることができる。したがって、I N V E R T ビットによって条件反転が示されているとき、エンコーダは、ビットプレーンがタイル化およびコード化される前に、ビットプレーンを予め反転する。デコーダ側では、最終出力の逆を取ることによって条件反転が実行される（これは、D i f f - 2 および D i f f - 6 モードについては実行されない）。

20

【 0 1 2 7 】

図19は、行スキップコード化モードでビットプレーン内のバイナリ情報を符号化するための技法（1900）を示す。最初に、エンコーダは、ビットプレーンの反転が適切かどうか検査し、適切である場合は反転を実行する（1910）。次いで、エンコーダは、ビットプレーン内の行を検査し、その行内の各ビットが0に等しいかどうか確認する（1920）。0に等しい場合、エンコーダは、その行についてのインジケータビットを0に設定する（1930）。行内のビットのいずれかが0でない場合、エンコーダは、その行についてのインジケータビットを1に設定し、その行内の各ビットを1つのビットで符号化する（1940）。エンコーダがビットプレーン内の行すべてを符号化し終わると（1950）、ビットプレーンの符号化が終了する。

30

【 0 1 2 8 】

デコーダは、行スキップコード化モードについて対応する復号を実行する。

【 0 1 2 9 】

図20は、列スキップコード化モードでバイナリ情報を符号化するための技法を示す。最初に、エンコーダは、ビットプレーンの反転が適切かどうか検査し、適切である場合は反転を実行する（2010）。次いで、エンコーダは、ビットプレーン内の列を検査し、その列内の各ビットが0に等しいかどうか確認する（2020）。0に等しい場合、エンコーダは、その列についてのインジケータビットを0に設定する（2030）。列内のビットのいずれかが0でない場合、エンコーダは、その列についてのインジケータビットを1に設定し、その列内の各ビットを1つのビットで符号化する（2040）。エンコーダがビットプレーン内の列すべてを符号化し終わると（2050）、ビットプレーンの符号化が終了する。

40

【 0 1 3 0 】

デコーダは、列スキップコード化モードについて対応する復号を実行する。

【 0 1 3 1 】

2. ノーマル2モード

エンコーダは、ノーマル2モードを使用して（たとえば、ベクトルハフマンまたは他の可変長符号化方式を使用して）ビットプレーン内で複数のバイナリシンボルを一緒に符号化する。エンコーダは、可変長符号で1対のバイナリシンボルを符号化する。デコーダは、対応する復号を実行する。

50

【0132】

rowMB × colMB が奇数である場合、第1のシンボルが単一のビットとして符号化される。後続のシンボルは、自然な走査順で対毎に符号化される。シンボル対を符号化するためにVLCテーブルが使用され、全体的なエントロピーを削減する。

【0133】

INVERTビットによって条件反転が示されているときは、エンコーダは、ビットプレーンが対毎にコード化される前に、ビットプレーンを予め反転する。デコーダ側では、最終出力の逆を取ることで条件反転が実行される(Diff-2モードが使用されているとき、このステップで条件反転は実行されない)。

【0134】

図21は、ノーマル2モードでバイナリ情報を符号化するための技法(2100)を示す。エンコーダは、初期検査を実行し、コード化効率を改善するためにビットプレーンの反転が適切であるかどうか判定し、適切である場合は反転を実行する(2110)。次いで、エンコーダは、コード化されるビットプレーンが奇数のバイナリシンボルを有するかどうか判定する(2120)。奇数である場合、エンコーダは、単一のビットで第1のシンボルを符号化する(2130)。次いで、エンコーダは、より可能性の高い対を表すためにより短いコードを使用し、あまり可能性の高くない対を表すためにはより長いコードを使用して可変長符号でシンボル対を符号化する(2140)。シンボル対の符号化を終えたと(2150)符号化が終了する。

【0135】

デコーダは、ノーマル2コード化モードについて対応する復号を実行する。

【0136】

3. ノーマル6モード

エンコーダはまた、ノーマル6モードを使用して(たとえば、ベクトルハフマンまたは他の可変長符号化方式を使用して)ビットプレーン内で複数のバイナリシンボルを一緒に符号化する。エンコーダは、6つのバイナリシンボルのグループをタイル化し、各グループを可変長符号で表す。デコーダは、対応する復号を実行する。

【0137】

ノーマル6モード(およびDiff-6モード)では、ビットプレーンが6つの画素のグループで符号化される。これらの画素は、2×3または3×2のタイルの形にグループ化される。ビットプレーンは、1組の規則を使用して最大限にタイル化され、残りの画素は、行スキップモードおよび列スキップモードの変形モードを使用して符号化される。

【0138】

第2の実施例において、rowMBが3の倍数であり、colMBが3の倍数でない場合のみ、3×2の「垂直」タイルが使用される。そうでない場合は、2×3の「水平」タイルが使用される。図22、23、24は、ノーマル6コード化モードでタイル化されたフレームの諸例を示す図である。図22は、3×2垂直タイル群と、列スキップモードでコード化される1シンボル幅残部(陰影付き領域として示される)とを有するフレーム(2200)を示す図である。図23は、2×3水平タイル群と、行スキップモードでコード化される1シンボル幅残部とを有するフレーム(2300)を示す図である。図24は、2×3水平タイルと、行スキップモードおよび列スキップモードでコード化される1シンボル幅残部群とを有するフレーム(2400)を示す図である。

【0139】

この例では3×2と2×3のタイルが使用されているが、他の実施形態では、異なる構成のタイルおよび/または異なるタイル化規則が使用される。

【0140】

最初に6個要素タイルが符号化され、その後列スキップおよび行スキップ符号化された線形タイルが続く。配列サイズが3×2または2×3の倍数である場合、後者の線形タイルは存在せず、ビットプレーンが完璧にタイル化される。6個要素の矩形タイルは、VLCテーブルを使用して符号化される。

10

20

30

40

50

【 0 1 4 1 】

I N V E R Tビットによって条件反転が示されているとき、エンコーダは、ビットプレーンがタイル化およびコード化される前に、ビットプレーンを予め反転する。デコーダ側では、最終出力の逆を取ることにによって条件反転が実施される（D i f f - 6モードが使用されているとき、このステップで条件反転は実行されない）。

【 0 1 4 2 】

図 2 5 は、ノーマル 6 モードでバイナリ情報を符号化するための技法（2 5 0 0）を示すフローチャートである。エンコーダは、初期検査を実行し、コード化効率を改善するためにビットプレーンの反転が適切であるかどうか判定し、適切である場合は、反転を実行する（2 5 1 0）。次いで、エンコーダは、ビットプレーン内の行の数が 3 の倍数であるかどうか検査する（2 5 2 0）。行の数が 3 の倍数でない場合、エンコーダは、ビットプレーン内のシンボルを 2 × 3 水平タイルの形にグループ化する（2 5 3 0）。

10

【 0 1 4 3 】

行の数が 3 の倍数である場合、エンコーダは、ビットプレーン内の列の数が 3 の倍数であるかどうか検査する（2 5 4 0）。列の数が 3 の倍数である場合、エンコーダは、ビットプレーン内のシンボルを 2 × 3 水平タイルの形にグループ化する（2 5 3 0）。列の数が 3 の倍数でない場合、エンコーダは、シンボルを 3 × 2 垂直タイルの形にグループ化する（2 5 5 0）。

【 0 1 4 4 】

シンボルを 3 × 2 または 2 × 3 のタイルの形にグループ化した後に、エンコーダは、6 次元ベクトル Huffman 符号化または何らかの他のコード化技法などの技法を使用して、6 個タイル化済みシンボルのグループを符号化する。エンコーダは、残りのタイル化されていないシンボルを、上述の行スキップおよび/または列スキップコード化技法を使用して符号化する（2 5 7 0）。

20

【 0 1 4 5 】

デコーダは、ノーマル 6 コード化モードについて対応する復号を実行する。

【 0 1 4 6 】

他の実施形態において、エンコーダは、他の技法を使用し、タイル化されたシンボル、およびタイル化されていないシンボルをコード化する。

【 0 1 4 7 】

4 . D i f f - 2 および D i f f - 6 モード

D i f f - 2 モードまたは D i f f - 6 モードなど差分符号化モードは、最初に、コード化すべきビットプレーンについてのプレディクタに基づいて、コード化すべきビットプレーンについて差分（または残留）ビットのビットプレーンを生成することによってビットプレーンを符号化する。次いで、残留ビットプレーンは、条件反転なしで、たとえばノーマル 2 またはノーマル 6 コード化モードを使用して符号化される。

30

【 0 1 4 8 】

第 2 の実施例において、D i f f - 2 モードまたは D i f f - 6 モードは、演算 d i f f によって示される差分符号化を使用する。どちらかの差分モードが使用される場合、最初に、ビットプレーン $b(i, j)$ のプレディクタ

40

【 0 1 4 9 】

【 数 1 】

$$\hat{b}(i, j)$$

【 0 1 5 0 】

すなわち、因果演算

【 0 1 5 1 】

【数 2】

$$\hat{b}(i, j) = \begin{cases} \text{INVERT} & i = j = 0, \text{ または } b(i, j-1) \neq b(i-1, j) \\ b(i, j-1) & i = 0 \\ b(i-1, j) & \text{ そうでない場合} \end{cases} \quad (1)$$

【0152】

として定義されるものを調べることによって、差分ビットのビットプレーンが生成される。

【0153】

換言すれば、所与のバイナリシンボル $b(i, j)$ のプレディクタ

10

【0154】

【数 3】

$$\hat{b}(i, j)$$

【0155】

は、すぐ左側のバイナリシンボル $b(i-1, j)$ となるが、以下の特別な場合を除く。
1) $b(i, j)$ がビットプレーンの左上隅にある場合、または上方のバイナリシンボル $b(i, j-1)$ が左側のバイナリシンボル $b(i-1, j)$ に等しくない場合、プレディクタ

【0156】

20

【数 4】

$$\hat{b}(i, j)$$

【0157】

は、INVERT の値に等しい、または、

2) 1) が当てはまらず、 $b(i, j)$ が左列にある場合 ($i = 0$)、プレディクタ

【0158】

【数 5】

$$\hat{b}(i, j)$$

30

【0159】

は、上方のバイナリシンボル $b(i, j-1)$ になる。

【0160】

エンコーダ側では、

【0161】

【数 6】

$$r(i, j) = b(i, j) \oplus \hat{b}(i, j) \quad (2)$$

【0162】

に従って、diff 演算により残留ビットプレーン r が計算される。上式で、

40

【0163】

【数 7】

\oplus

【0164】

は、排他的論理和演算である。残留ビットプレーンは、条件反転なしで、ノーマル 2 モードまたはノーマル 6 モードを使用して符号化される。

【0165】

デコーダ側では、適切なノーマルモードを使用して残留ビットプレーンが再生成される。その後、残留ビットプレーンが使用され、バイナリ 2 D 差

50

【 0 1 6 6 】

【 数 8 】

$$b(i, j) = r(i, j) \oplus \hat{b}(i, j) \quad (3)$$

【 0 1 6 7 】

として元のビットプレーンを再生成する。

【 0 1 6 8 】

図 2 6 は、差分符号化モードでバイナリ情報を符号化するための技法 (2 6 0 0) を示す。エンコーダは、たとえば式 1 に示されているように、ビットプレーンについてプレディクタを計算する (2 6 1 0)。次いで、エンコーダは、たとえばビットプレーンとそのプレディクタに対して XOR 演算を実行することによって残留ビットプレーンを計算する (2 6 2 0)。次いで、エンコーダは、残留ビットプレーンを (たとえば、ノーマル 2 またはノーマル 6 モードで) 符号化する (2 6 3 0)。

10

【 0 1 6 9 】

図 2 7 は、差分符号化モードで符号化されたバイナリ情報を復号するための技法 (2 7 0 0) を示す。デコーダは、残留ビットプレーンを符号化するために使用されたモード (たとえば、ノーマル 2 またはノーマル 6 モード) に基づいて、適切な復号技法を使用して残留ビットプレーンを復号する (2 7 1 0)。また、デコーダは、エンコーダ内で使用され同じ技法を使用して、ビットプレーンについてプレディクタを計算する (2 7 2 0)。次いで、デコーダは、たとえば復号された残留ビットプレーンとプレディクタビットプレーンに対して XOR 演算を実行することによって、元のビットプレーンを再構築する (2 7 3 0)。

20

【 0 1 7 0 】

5 . 非圧縮モード

非圧縮モードを除くすべてのモードがビットプレーンをフレームレベルで符号化するが、これは、符号化中にフレームを介した第 2 の経路を必要とする。しかし、低遅延 (low - latency) の環境下では、第 2 の経路によって受け入れることができない遅延 (latency) が付加される可能性がある (というのは、たとえば、ビットプレーンを符号化して費やされる時間のために、フレーム内の最後のマクロブロックに達するまでフレームヘッダおよびマクロブロック層情報の送信が遅れるからである)。

30

【 0 1 7 1 】

非圧縮モードは、残りのマクロブロックレベル情報と同じビットストリーム内の位置で、バイナリシンボル当たり 1 つのビットでビットプレーンを符号化する従来の方法を使用する。シンボルのマクロブロックレベルのコード化は、それ自体新しい概念ではないが、シンボルのコード化をフレームレベルからマクロブロックレベルに切り換えることにより、フレームレベルのコード化に対して低遅延とすることができる。

【 0 1 7 2 】

図 2 8 は、低遅延の応用例のために、マクロブロックについてのバイナリ情報を非圧縮コード化モードで選択的に符号化するための技法 (2 8 0 0) を示す。最初に、エンコーダは、バイナリ情報を符号化するために非圧縮モードを使用するかどうか検査する (2 8 1 0)。使用する場合、エンコーダは、マクロブロックについてマクロブロックレベルでビットを符号化し (2 8 2 0)、そのマクロブロックがフレーム内の最後のマクロブロックかどうか検査する (2 8 3 0)。マクロブロックがフレーム内の最後のマクロブロックでない場合、エンコーダは、マクロブロックレベルで次のマクロブロックについてビットを符号化することにより続行する (2 8 2 0)。

40

【 0 1 7 3 】

エンコーダが非圧縮コード化モードを使用しない場合、エンコーダは、フレーム内のマクロブロックについて、フレームレベルでビットプレーンを符号化する (2 8 4 0)。フレーム内のマクロブロックの符号化を終えると (2 8 5 0) そのフレームについて符号化が終了する。

50

【0174】

この技法(2800)は、フレーム毎にモードを切り換えることを示しているが、またはこれに替えて、エンコーダは、何らかの他の基準に基づいて切り換えることとなる。

【0175】

以上、様々な実施形態を参照しながら本発明の原理について述べ、また例示したが、この様々な実施形態は、そのような原理から逸脱することなしに、構成および詳細に修正を加えることができることを理解されたい。本明細書に説明するプログラム、プロセス、または方法は、別段の表示がない限り、どの特定のタイプのコンピューティング環境にも関連しない、あるいは制限されないことを理解されたい。様々なタイプの汎用または専用のコンピューティング環境は、本明細書に説明する技法と共に使用することができ、あるいは、それに従って動作を実行することができる。ソフトウェアで示されている実施形態の要素は、ハードウェアで実施することができ、逆も同様である。

10

【0176】

本発明の原理を適用することができる多数の可能な実施形態に鑑みて、本発明者等は、以下の特許請求の範囲とその等価物の範囲および精神内に入るそのような実施形態すべてを本発明として主張する。

【図面の簡単な説明】

【0177】

【図1】従来技術による、画素の8×8ブロックのブロックベースのフレーム内圧縮を示す図である。

20

【図2】従来技術による、周波数係数の予測を示す図である。

【図3】従来技術による、ビデオエンコーダ内の動き予測を示す図である。

【図4】従来技術による、ビデオエンコーダ内の予測残差の8×8ブロックに対するブロックベースのフレーム間圧縮を示す図である。

【図5】従来技術による、予測残差の8×8ブロックに対するブロックベースのフレーム間伸長を示す図である。

【図6】説明するいくつかの実施形態を実施することができる好適なコンピューティング環境のブロック図である。

【図7】説明するいくつかの実施形態で使用される一般化されたビデオエンコーダシステムのブロック図である。

30

【図8】説明するいくつかの実施形態で使用される一般化されたビデオデコーダシステムのブロック図である。

【図9】第1の実施例による、Pピクチャ層を構成するビットストリーム要素を示すチャートである。

【図10】複数のスキップマクロブロック符号化モードを有するビデオエンコーダ内でスキップマクロブロック情報を符号化するための技法を示すフローチャートである。

【図11】複数のスキップマクロブロック符号化モードを有するビデオエンコーダ内で符号化されたスキップマクロブロック情報を復号するための技法を示すフローチャートである。

【図12】スキップマクロブロック符号化フレームの例を示す図である。

40

【図13】ノーマルスキップマクロブロック符号化モードで符号化するための技法を示すフローチャートである。

【図14】行予測スキップマクロブロック符号化モードで符号化するための技法を示すフローチャートである。

【図15】スキップマクロブロック情報の行予測復号のための擬似コードを示すコードリストである。

【図16】列予測スキップマクロブロック符号化モードで符号化するための技法を示すフローチャートである。

【図17】スキップマクロブロック情報の列予測復号のための擬似コードを示すコードリストである。

50

【図18】ビデオエンコーダ内でいくつかのマクロブロックのコード化をスキップするかどうか判定するための技法を示すフローチャートである。

【図19】行スキップコード化モードでビットプレーン内のバイナリ情報を符号化するための技法を示すフローチャートである。

【図20】列スキップコード化モードでビットプレーン内のバイナリ情報を符号化するための技法を示すフローチャートである。

【図21】ノーマル2コード化モードでビットプレーン内のバイナリ情報を符号化するための技法を示すフローチャートである。

【図22】ノーマル6モードでタイル化されたバイナリ情報のフレームの例を示す図である。

【図23】ノーマル6モードでタイル化されたバイナリ情報のフレームの例を示す図である。

【図24】ノーマル6モードでタイル化されたバイナリ情報のフレームの例を示す図である。

【図25】ノーマル6コード化モードでビットプレーン内のバイナリ情報を符号化するための技法を示すフローチャートである。

【図26】差分符号化モードでバイナリ情報を符号化するための技法を示すフローチャートである。

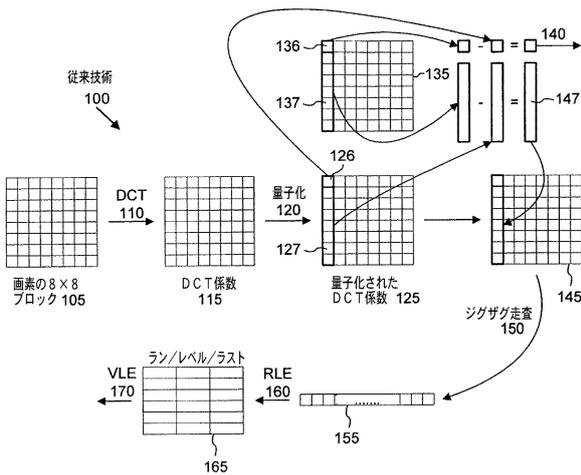
【図27】差分符号化モードで符号化されたバイナリ情報を復号するための技法を示すフローチャートである。

【図28】低遅延応用例のために行コード化モードでバイナリ情報を選択的に符号化するための技法を示すフローチャートである。

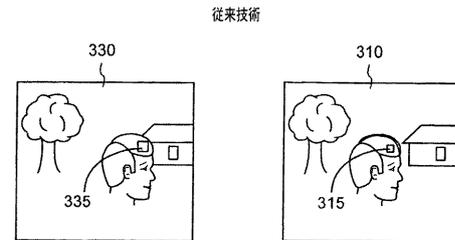
10

20

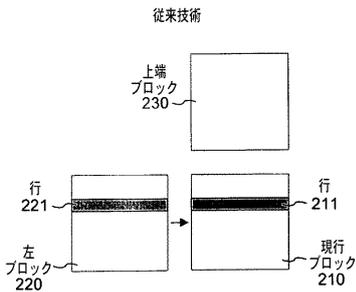
【図1】



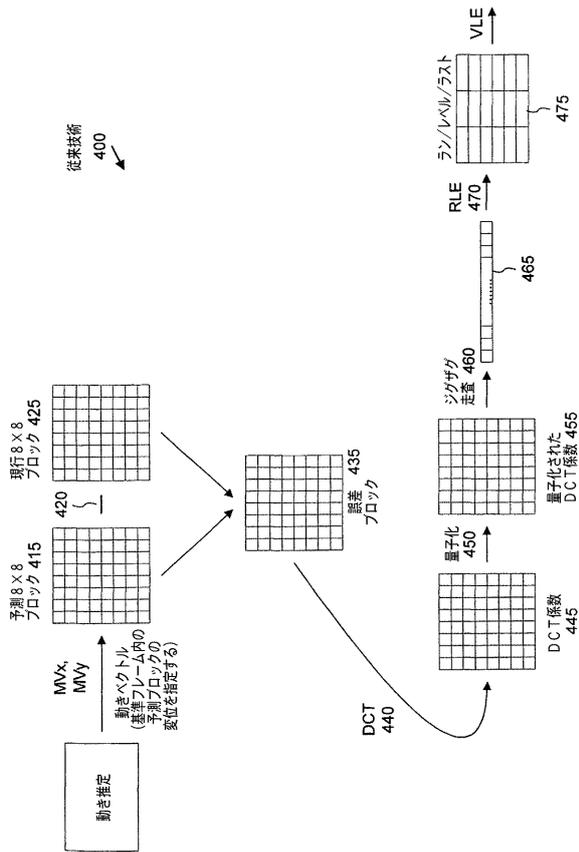
【図3】



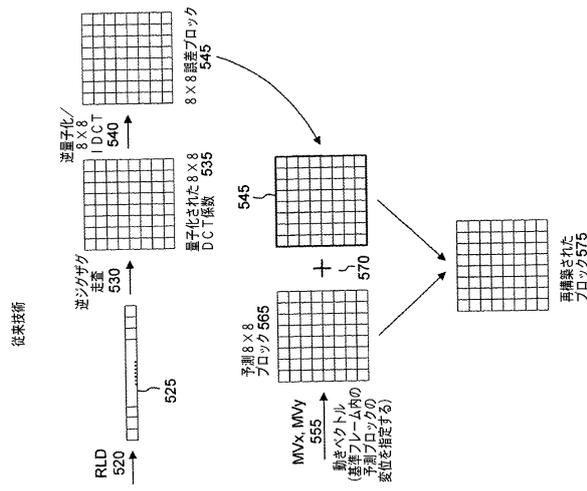
【図2】



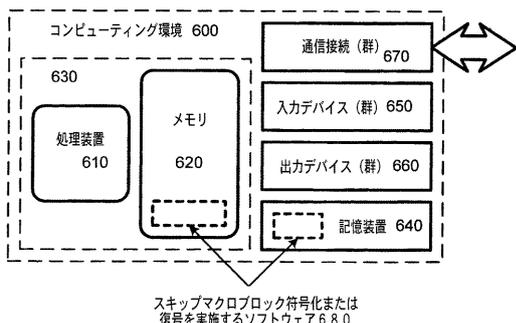
【図4】



【図5】

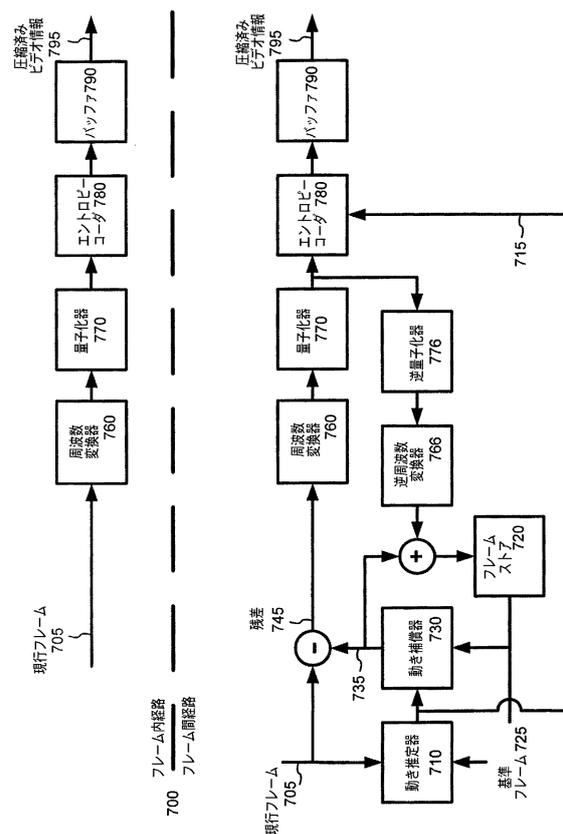


【図6】

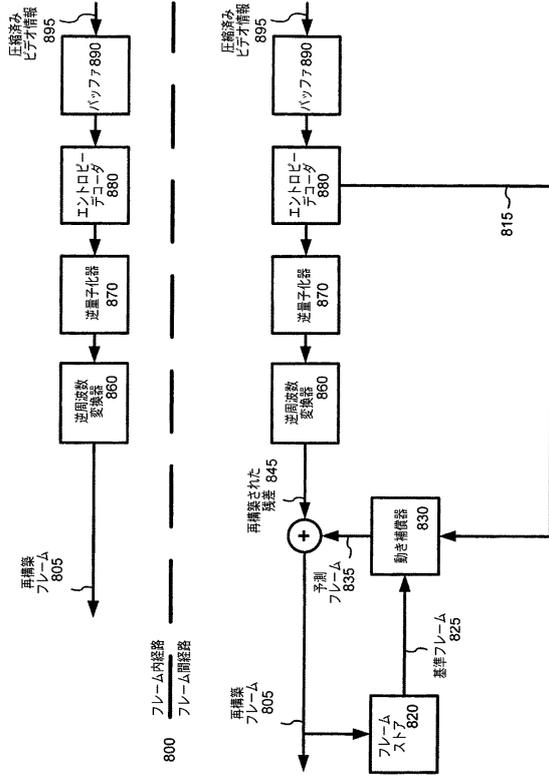


スキップマクロブロック符号化または復号を実施するソフトウェア680

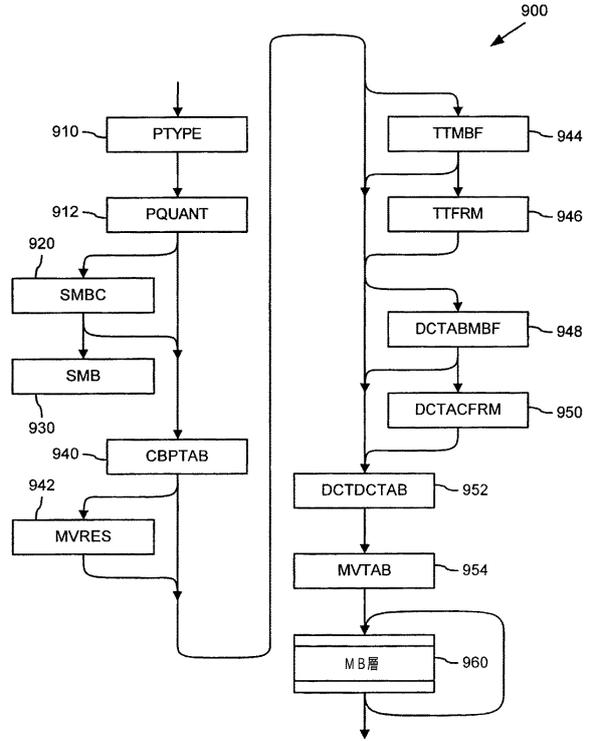
【図7】



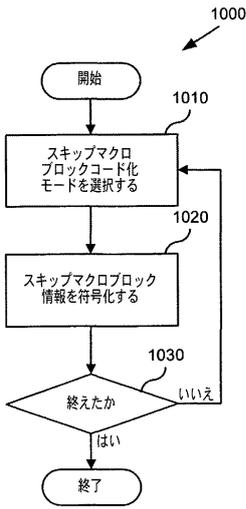
【図8】



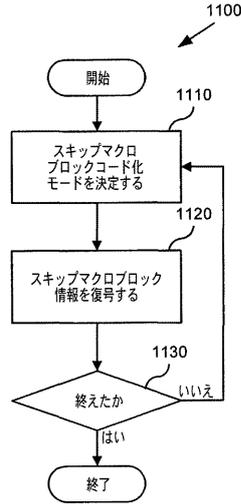
【図9】



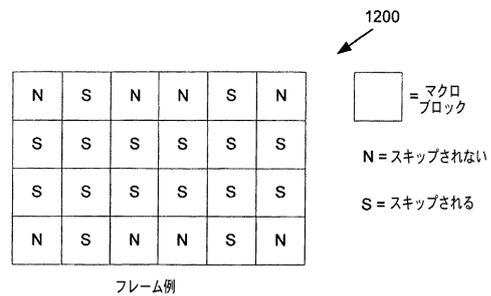
【図10】



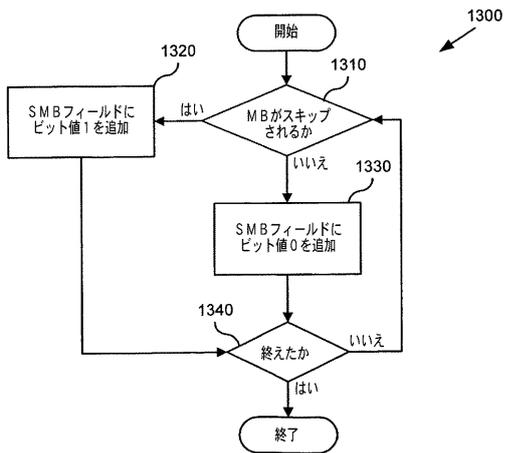
【図11】



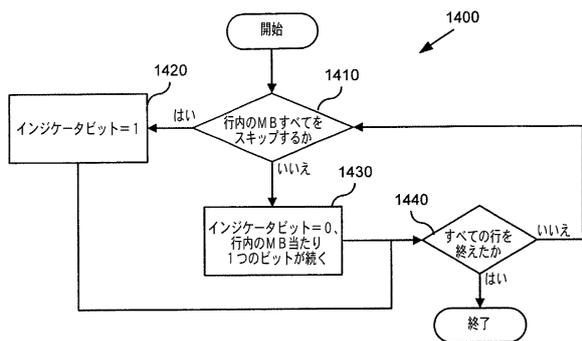
【図12】



【図13】



【図14】



【図17】

```

for (mbcol = 0; mbcol < NumMBColumns; mbcol++)
{
  if (get_bits(1) == 1)
    ## このマクロブロック列内のマクロブロックすべてがスキップされる
  else
  {
    ## この列の少なくとも1つのマクロブロックがスキップされない
    ## 上から下に始めて、列内の各マクロブロックの状況をコード化する
    for (mb = 0; mb < NumMBSPerColumn; mb++)
      k = get_bits(1); ## kはマクロブロックの状況に等しい
    ## スキップされない場合k=0、スキップされる場合k==1
  }
}

```

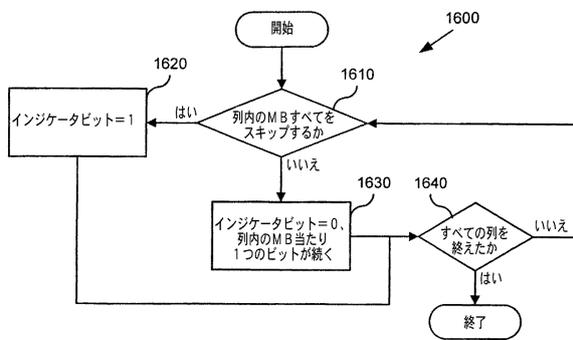
【図15】

```

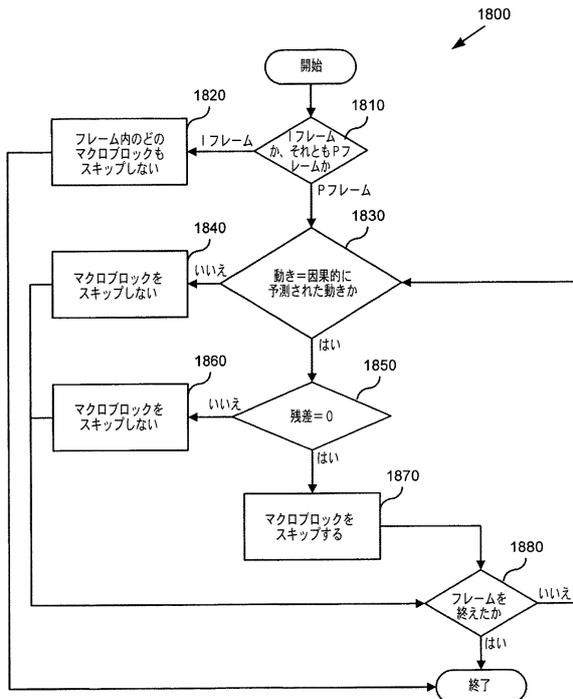
for (mbrow = 0; mbrow < NumMBRows; mbrow++)
{
  if (get_bits(1) == 1)
    ## このマクロブロック行内のマクロブロックすべてがスキップされる
  else
  {
    ## この行の少なくとも1つのマクロブロックがスキップされない
    ## 左から右に始めて、行内の各マクロブロックの状況をコード化する
    for (mb = 0; mb < NumMBSPerRow; mb++)
      k = get_bits(1); ## kはマクロブロックの状況に等しい
    ## スキップされない場合k=0、スキップされる場合k==1
  }
}

```

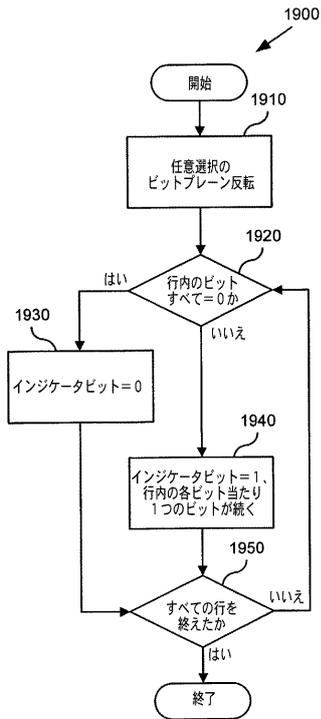
【図16】



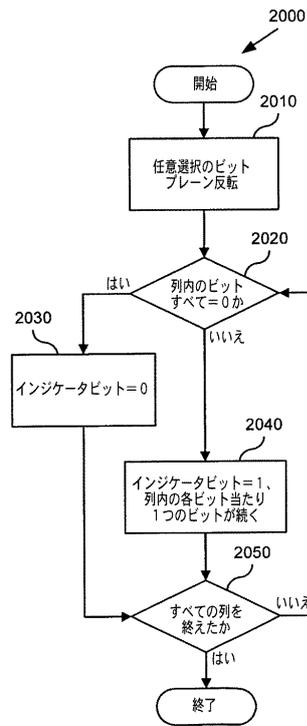
【図18】



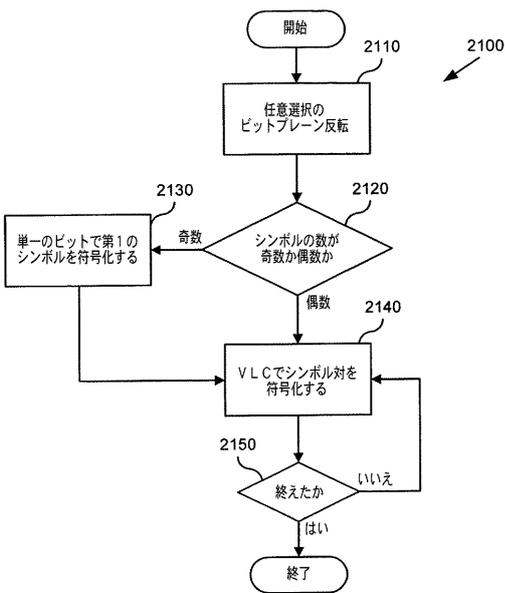
【図19】



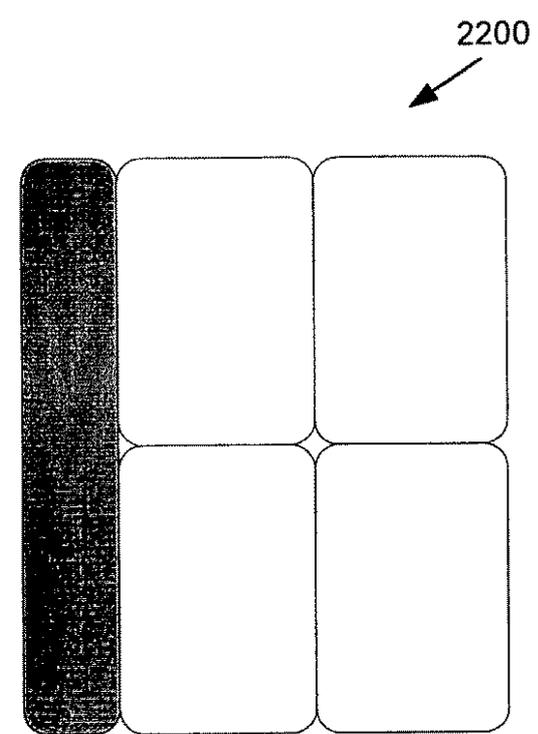
【図20】



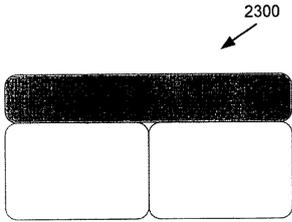
【図21】



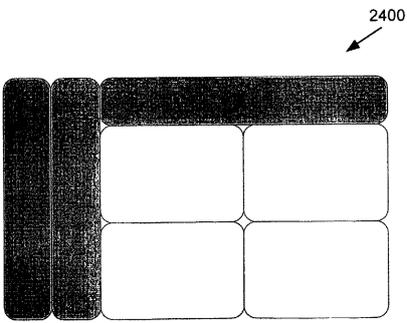
【図22】



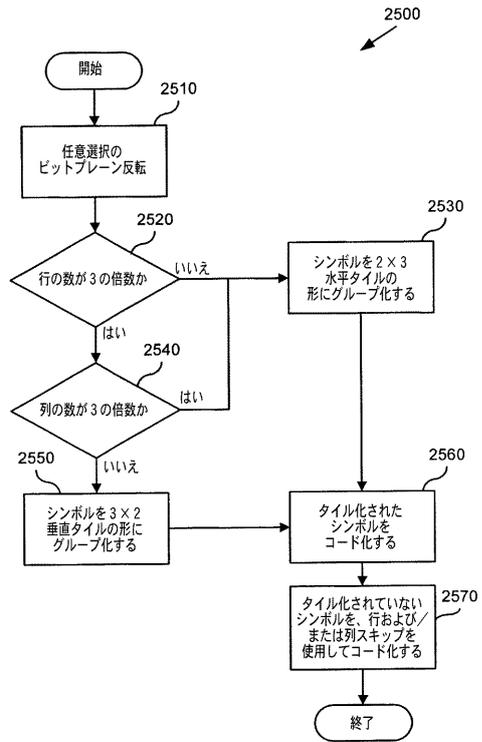
【図 23】



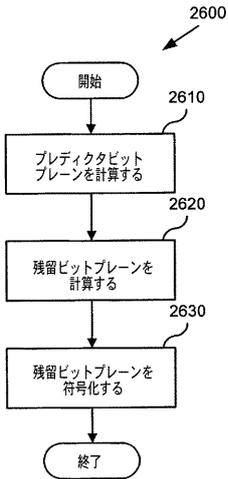
【図 24】



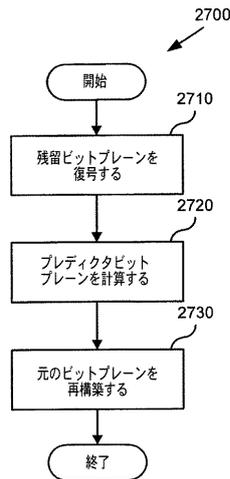
【図 25】



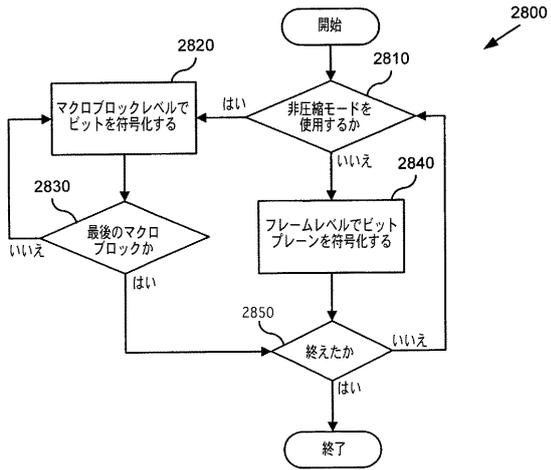
【図 26】



【図 27】



【図 28】



フロントページの続き

(72)発明者 ポシャング スウ

アメリカ合衆国 98052 ワシントン州 レドモンド 156 アベニュー ノース イース
ト 4850 ユニット 93

審査官 坂本 聡生

(56)参考文献 特開平05 - 199422 (JP, A)

特開2001 - 036908 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N 7/24 - 7/68