

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-135007

(P2012-135007A)

(43) 公開日 平成24年7月12日(2012.7.12)

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

審査請求有 請求項の数 8 O L (全 25 頁)

(21) 出願番号 特願2012-24357(P2012-24357)
 (22) 出願日 平成24年2月7日(2012.2.7)
 (62) 分割の表示 特願2006-526223(P2006-526223)
 の分割
 原出願日 平成16年9月3日(2004.9.3)
 (31) 優先権主張番号 60/501,081
 (32) 優先日 平成15年9月7日(2003.9.7)
 (33) 優先権主張国 米国(US)
 (31) 優先権主張番号 10/933,960
 (32) 優先日 平成16年9月3日(2004.9.3)
 (33) 優先権主張国 米国(US)

(71) 出願人 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 110001243
 特許業務法人 谷・阿部特許事務所
 (72) 発明者 シャンカー レグナサン
 アメリカ合衆国 98052 ワシントン
 州 レッドモンド ワン マイクロソフト
 ウェイ マイクロソフト コーポレーシ
 ョン内

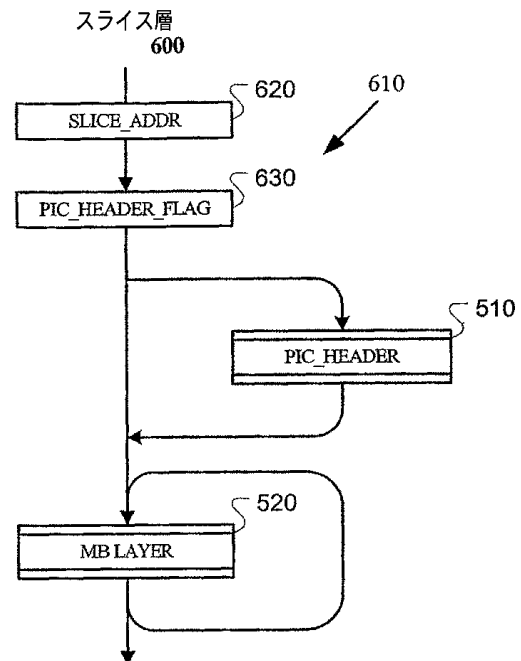
最終頁に続く

(54) 【発明の名称】映像コーデックにおけるスライス層

(57) 【要約】 (修正有)

【課題】映像エンコーダ/デコーダを提供する。

【解決手段】映像エンコーダ/デコーダは、1行または連続した複数行のマクロブロック(スライスと呼ぶ)を含む単位の形とすることができる、独立に復号化可能な部分ピクチャ単位を提供するビットストリーム構文を使用する。このスライス層により、エラー回復力と圧縮効率の柔軟な組合せが得られる。スライス層は、効率的なアドレッシング機構(たとえば、スライス層の開始マクロブロック行を指定する構文要素)、ならびにピクチャヘッダ情報を最適に再送信するための効率的な機構を符号化する。スライス層は、スライス境界をまたぐすべての形の予測、オーバーラップ、およびループフィルタリングを使用不能にすることによって、復号化および再構築の独立性をもたらす。これにより、スライス内モードで符号化されたスライスを、ピクチャの他の領域内のエラーとは無関係に、エラーなしで再構築することができる。



【選択図】図6

【特許請求の範囲】**【請求項 1】**

画像または映像を符号化／復号化する方法であって、

少なくとも1つの独立に復号化可能な部分ピクチャ単位内のピクチャを符号化／復号化することであって、前記少なくとも1つの独立に復号化可能な部分ピクチャ単位のうちの所与の部分ピクチャ単位のピクチャ内容情報の符号化／復号化の開始時に、予測符号化パラメータをリセットすることであって、前記リセットされる予測符号化パラメータが、動きベクトル予測子、AC係数予測の予測子、DC係数予測の予測子、および量子化パラメータの予測子を含むことと、

前記所与の部分ピクチャ単位の外部にあるピクチャ内容とは独立に、前記所与の部分ピクチャ単位のピクチャ内容を再構築することと、

前記それぞれの部分ピクチャ単位の境界エッジを除く、前記それぞれの部分ピクチャ単位の前記ピクチャ内容中のブロックエッジのデブロッキングを実施することと

を備えることを特徴とする方法。

10

【請求項 2】

部分ピクチャ単位符号化が前記ピクチャに対して実施されると、通常はピクチャ層ビットプレーン符号化を使用して通常は通知されるマクロブロックの構文要素が、代わりに各マクロブロックに対してマクロブロック層において通知されることを特徴とする請求項1に記載の方法。

20

【請求項 3】

画像または映像を符号化／復号化する方法であって、

少なくとも1つの独立に復号化可能な部分ピクチャ単位内のピクチャを符号化／復号化することであって、前記少なくとも1つの独立に復号化可能な部分ピクチャ単位のうちの所与の部分ピクチャ単位のピクチャ内容情報の符号化／復号化の開始時に、予測符号化パラメータをリセットすることであって、前記リセットされる予測符号化パラメータが、動きベクトル予測子、AC係数予測の予測子、DC係数予測の予測子、および量子化パラメータの予測子を含むことと、

前記所与の部分ピクチャ単位の外部にあるピクチャ内容とは独立に、前記所与の部分ピクチャ単位のピクチャ内容を再構築することと、

前記それぞれの部分ピクチャ単位の境界エッジを除く、前記それぞれの部分ピクチャ単位の前記ピクチャ内容中のブロックエッジのオーバーラップフィルタリングを実施することと

を備えることを特徴とする方法。

30

【請求項 4】

部分ピクチャ単位符号化が前記ピクチャに対して実施されると、通常はピクチャ層ビットプレーン符号化を使用して通知されるマクロブロックの構文要素が、代わりに各マクロブロックに対してマクロブロック層において通知されることを特徴とする請求項3に記載の方法。

【請求項 5】

画像または映像を符号化／復号化する方法を実施するための、処理装置によって実行可能なソフトウェアモジュールを有する、少なくとも1つのコンピュータ読み取り可能な記録媒体であって、前記方法が、

40

少なくとも1つの独立に復号化可能な部分ピクチャ単位内のピクチャを符号化／復号化することであって、前記少なくとも1つの独立に復号化可能な部分ピクチャ単位のうちの所与の部分ピクチャ単位のピクチャ内容情報の符号化／復号化の開始時に、予測符号化パラメータをリセットすることであって、前記リセットされる予測符号化パラメータが、動きベクトル予測子、AC係数予測の予測子、DC係数予測の予測子、および量子化パラメータの予測子を含むことと、

前記所与の部分ピクチャ単位の外部にあるピクチャ内容とは独立に、前記所与の部分ピクチャ単位のピクチャ内容を再構築することと、

50

前記それぞれの部分ピクチャ単位の境界エッジを除く、前記それぞれの部分ピクチャ単位の
前記ピクチャ内容中のブロックエッジのデブロッキングを実施することと

を備えることを特徴とする記録媒体。

【請求項 6】

部分ピクチャ単位符号化が前記ピクチャに対して実施されると、通常はピクチャ層ビット
プレーン符号化を使用して通知されるマクロブロックの構文要素が、代わりに各マクロ
ブロックに対してマクロブロック層において通知されることを特徴とする請求項 5 に記載
の記録媒体。

【請求項 7】

画像または映像を符号化／復号化する方法を実施するための、処理装置によって実行可
能なソフトウェアモジュールを有する、少なくとも 1 つのコンピュータ読み取り可能な記
録媒体であって、前記方法が、

少なくとも 1 つの独立に復号化可能な部分ピクチャ単位内のピクチャを符号化／復号化
することであって、前記少なくとも 1 つの独立に復号化可能な部分ピクチャ単位のうちの
所与の部分ピクチャ単位のピクチャ内容情報の符号化／復号化の開始時に、予測符号化パ
ラメータをリセットすることであって、前記リセットされる予測符号化パラメータが、動
きベクトル予測子、AC 係数予測の予測子、DC 係数予測の予測子、および量子化パラメ
ータの予測子を含むことと、

前記所与の部分ピクチャ単位の外部にあるピクチャ内容とは独立に、前記所与の部分ピ
クチャ単位のピクチャ内容を再構築することと、

前記それぞれの部分ピクチャ単位の境界エッジを除く、前記それぞれの部分ピクチャ単
位の前記ピクチャ内容中のブロックエッジのオーバーラップフィルタリングを実施すること
と

を備えることを特徴とする記録媒体。

【請求項 8】

部分ピクチャ単位符号化が前記ピクチャに対して実施されると、通常はピクチャ層ビット
プレーン符号化を使用して通知されるマクロブロックの構文要素が、代わりに各マクロ
ブロックに対してマクロブロック層において通知されることを特徴とする請求項 7 に記載
の記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、映像、画像、およびその他のデジタル媒体の内容をデジタル的に符号化し、
復号化し、処理する技術に関する。

【背景技術】

【0002】

デジタル映像は、大量の記憶容量と伝送容量を消費する。典型的な現信号のデジタル映
像シーケンスは、毎秒 15 または 30 フレームを含む。各フレームは、数万または数十万
の画素（ペル（p e l）とも呼ばれる）を含み得る。各画素は、ピクチャのごく小さな要
素を表す。原信号の形式では、コンピュータは一般に、3 つのサンプルで 1 組の合計 24
ビットで画素を表す。たとえば、1 つの画素は、その画素のグレースケール成分を規定す
る 1 つの 8 ビット輝度サンプル（ルマ（l u m a）サンプルとも呼ばれる）と、その画素
の色成分を規定する 2 つの 8 ビットクロミナンスサンプル値（クロマ（c h r o m a）サ
ンプルとも呼ばれる）とを含み得る。したがって、通常の前信号のデジタル映像シーケ
ンスの毎秒ビット数、すなわちビットレートは、毎秒 500 万ビット以上となり得る。

【0003】

多くのコンピュータおよびコンピュータネットワークは、生デジタル映像を処理するた
めのリソースが不足している。この理由で、技術者は圧縮（コード化または符号化とも呼
ばれる）を使用し、デジタル映像のビットレートを低減している。圧縮では、映像を低ビ
ットレート形式に変換することにより、映像を記憶し伝送するコストが低減される。伸張

10

20

30

40

50

(復号化とも呼ばれる)では、圧縮形式から元の映像のバージョンを再構築する。「コーデック」とは、エンコーダ/デコーダシステムのことである。圧縮は、映像の品質が低下しないが、映像データの変動の固有量(エントロピーと呼ばれることもある)によってビットレートの低減が制限される、可逆式(lossless)とすることが可能である。あるいは、圧縮は、映像の品質が低下するが、実現可能なビットレートの低減はより劇的となる、非可逆式(lossy)とすることも可能である。非可逆圧縮で情報の近似を確立、可逆圧縮技術を適用してその近似が表されるシステム設計において、非可逆圧縮はしばしば可逆圧縮と共に使用される。

【0004】

一般に、映像圧縮技術には、「ピクチャ内」圧縮および「ピクチャ間」圧縮が含まれ、その際、ピクチャは、たとえばプログレッシブ走査映像フレーム、インターレース映像フレーム(それぞれの映像フィールドの線を交互に有する)、またはインターレース映像フィールドである。プログレッシブフレームの場合、ピクチャ内圧縮技術では、個々のフレーム(一般にIフレームまたはキーフレームと呼ばれる)を圧縮し、ピクチャ間圧縮技術では、前および/または後のフレーム(一般に参照フレームまたはアンカーフレームと呼ばれる)を参照して、フレーム(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)を圧縮する。

【0005】

予測フレームは、マクロブロックと呼ばれる領域に分割することができる。参照フレーム内のマクロブロックに対するマッチング領域は、そのマクロブロックに対する動きベクトル情報を送信することによって指定される。動きベクトルは、画素の現マクロブロックに対する予測子としてその画素が使用される参照フレーム内の領域の位置を示す。現マクロブロック(またはそのブロック)とマクロ予測子の間には、しばしば誤差信号または残差と呼ばれる、画素間の差分が生じる。通常、この誤差信号は、元の信号よりも小さなエントロピーを有する。したがって、その情報を、より低いレートで符号化することが可能である。エンコーダは、予測子として使用するために、1つまたは複数の参照フレーム内のマッチング領域を探索することにより、フレームのある領域に対する動きベクトルを決定することによって、動き推定を実施する。エンコーダまたはデコーダは、動きベクトルを適用して、1つまたは複数の参照フレーム内で予測子を見つけることによって、動き補償を実施する。

【0006】

マクロブロックに対する動きベクトル値は、しばしば空間的に周りを囲んでいるマクロブロックに対する動きベクトルと相関関係がある。したがって、動きベクトル情報を送信するのに使用されるデータの圧縮は、ある動きベクトルとそれと隣接する動きベクトルから形成された動きベクトル予測子との間の差分を符号化することによって行うことが可能である。

【0007】

映像圧縮技術ではよく、画素のブロックまたは残差など他の空間領域映像データが、変換領域データに変換される。この変換領域データは、しばしば周波数領域(すなわちスペクトル)データである。得られるスペクトルデータ係数のブロックを、量子化し、その後、エントロピー符号化することができる。

【0008】

得られた映像が表示される前にデータを伸張するとき、デコーダは一般に、圧縮動作の逆を行う。たとえば、デコーダは、データを伸張しながら、エントロピー復号化、逆量子化、および逆変換を実行することができる。動き補償を使用すると、デコーダ(およびエンコーダ)は、前に再構築された1つまたは複数のフレーム(これらはここで、参照フレームとして使用される)からフレームを再構築し、後のフレームを動き補償するために、新規に再構築されたフレームを参照フレームとして使用することができる。

【0009】

デジタル的に符号化された映像に対する多くの典型的な使用シナリオでは、装置間で、

10

20

30

40

50

また多くの場合、地理的に離れた場所間で符号化済みの映像を伝送する。さらに、一般に使用される多くのデータ伝送システムでは、パケットベースの伝送プロトコルを使用する。このプロトコルでは、データ伝送は、「パケット」と呼ばれる別々に経路指定される単位に分割される。デジタル映像を運搬するこれら様々な伝送システムでは、しばしばノイズその他の伝送エラー源があり、「パケット損失」が起こり得る。こうしたエラーおよびパケット損失により、個々のフレーム、あるいは映像シーケンスの関連した複数のフレームの復号に失敗する恐れがある。

【0010】

したがって、映像シーケンスにおけるピクチャの部分領域を、独立に復号可能な単位として符号化するのが望ましいことがある。これにより、映像ストリームのパケット化が可能になる。さらに、このために、圧縮映像ビットストリームにおける冗長性が増大し、伝送エラーおよびパケット損失に対する回復力が向上する。たとえば、伝送エラーまたはパケット損失による復号化損失を、映像シーケンスのピクチャ全体ではなく、部分領域に制限することが可能である。しかし、この回復力は、圧縮効率のコストを犠牲にして実現される。

10

【0011】

非常に多くの会社が映像コーデックを生み出してきた。たとえば、マイクロソフト（登録商標）社は、Windows（登録商標）Media Video 8向けにリリースされている映像エンコーダおよびデコーダを生み出した。これらの製品以外に、多くの国際規格で、圧縮映像情報向けの映像デコーダおよびフォーマットの側面を指定している。これらの規格には、H.261、MPEG-1、H.262、H.263、およびMPEG-4の各規格が含まれる。これらの規格では、直接または暗示的に、ある種のエンコーダの詳細も指定されているが、他のエンコーダの詳細は指定されていない。これらの製品および規格は、上述の圧縮および伸張技術の様々な組合せを使用している（または使用をサポートしている）。具体的には、これらの製品および規格は、一部のピクチャを単位とする符号化のための様々な技術を提供している。

20

【0012】

そのような技術の1つでは、映像シーケンス内のあるフレームを複数のスライスに分割する。1つのスライスは、1行または連続した複数行のマクロブロックを、元の左から右の順で含むものと定義される。スライスは、行の最初のマクロブロックから始まり、同じまたは別の行の最後のマクロブロックで終わる。

30

【0013】

様々な標準規格、たとえば、MPEG-1、MPEG-2、H.263（ほぼスライスと同等であるGOBを有する、またはAnnex Kのスライス構造符号化モードを有する）、MPEG-4パート2、およびH.264/JVT/MPEG-4パート10は、すべてその構文の一部としてスライスを有している。とりわけ、これらのすべてが、エラー/損失ロバストネスの理由で、フレーム内予測および動きベクトル予測およびスライス境界をまたぐ他の大部分の形式の予測を無効にしている。特に、H.263（Annex J）およびH.264/JVTだけが、ループフィルタを含む。H.263のインターレースの処理は、かなり原始的（フィールド符号化でAnnex Wの増補の拡張指示だけを使用する）である。H.264は、エラーに対してより強靱なヘッダ構造を有し、ループフィルタリングがスライス境界をまたいで適用されるかどうかを、エンコーダが選択できるようにする。

40

【発明の概要】

【発明が解決しようとする課題】

【0014】

これら様々な映像復号化規格によるスライスの実装ではそれぞれ、回復力と符号化効率のバランスが異なる。

【課題を解決するための手段】

【0015】

50

本明細書に記載される映像コーデックおよびビットストリーム構文は、スライス層を含んでおり、スライス層は、柔軟性のあるように設計され、エラー回復力と圧縮効率を有効に両立させる。このスライス層は、以下の主な機能を提供する。

a) プログレッシブ、インターレースフレーム、およびインターレースフィールド符号化方法で動作する効率的なスライスアドレッシング機構

b) スライス層中のピクチャヘッダを再送信するための柔軟で効率的な機構

c) スライス内モードで符号化されたスライスが、ピクチャの他の領域内のエラーとは無関係に、エラーなしで再構築できるように、スライス境界をまたぐすべての形の予測、オーバーラップ、およびループフィルタリングを無効にすることによる復号化独立性

本発明のさらなる特徴および利点は、実施形態についての以下の詳細な説明を添付の図面と併せ読めば明らかになる。

【図面の簡単な説明】

【0016】

【図1】本明細書に記載のスライス層符号化を用いた映像エンコーダのブロック図である。

【図2】本明細書に記載のスライス層符号化を用いた映像デコーダのブロック図である。

【図3】スライス層符号化を用いた映像エンコーダ/デコーダによって使用される圧縮ビットストリームで表される映像シーケンスの要素の階層構成を示す図である。

【図4】スライス層符号化を用いた映像エンコーダ/デコーダによって使用される圧縮ビットストリームの符号化構文のシーケンスレベルの構文図である。

【図5】スライス層符号化を用いた映像エンコーダ/デコーダによって使用される圧縮ビットストリームの符号化構文のフレームレベルの構文図である。

【図6】スライス層符号化を用いた映像エンコーダ/デコーダによって使用される圧縮ビットストリームの符号化構文のスライス層レベルの構文図である。

【図7】ブロック境界で実施されるオーバーラップ平滑化の一例を示す図である。

【図8】ループ内デブロックフィルタリングが実施されるIピクチャ内の水平ブロック境界画素の例を示す図である。

【図9】ループ内デブロックフィルタリングが実施されるIピクチャ内の垂直ブロック境界画素の例を示す図である。

【図10】図1および2の映像エンコーダ/デコーダに適したコンピューティング環境のブロック図である。

【図11】ループフィルタリングで使用される、4つで1組の画素セグメントを示す図である。

【図12】フィルタリング動作で使用される画素を示す図である。

【図13】あるセグメントの第3の画素対に対するフィルタリング動作の擬似コードリストである。

【図14】あるセグメントの第1、第2、および第4の画素対に対するフィルタリング動作の、擬似コードリストである。

【発明を実施するための形態】

【0017】

以下に、柔軟性のあるように設計され、エラー回復力と圧縮効率の効果的組合せをもたらす、映像コーデックおよびビットストリーム構文におけるスライス層の実装形態について説明する。例として、スライス層符号化を、画像または映像のエンコーダおよびデコーダ内で適用する。したがって、スライス層符号化を、汎用の画像または映像のエンコーダおよびデコーダの文脈で説明するが、その代わりに、以下に述べるこの例示的なビットストリーム構文とは細部が変わり得る、他の様々な画像および映像コーデックのビットストリーム構文にそれを組み込むことも可能である。

(1. 汎用映像エンコーダおよびデコーダ)

図1は汎用映像エンコーダ(100)のブロック図で、図2は汎用映像デコーダ(200)のブロック図であり、これらのエンコーダおよびデコーダには、WMV9/V9

10

20

30

40

50

変換を組み込むことが可能である。

【0018】

図示されているエンコーダ内およびデコーダ内のモジュール間の関係は、エンコーダおよびデコーダにおける情報の主なフローを示している。わかりやすいように、他の関係は示していない。特に、図1および図2は、映像シーケンス、フレーム、マクロブロック、ブロックなど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的情報は通常示さない。そのような副次的情報は、一般に副次的情報のエントロピー符号化の後に、出力ビットストリームで送信される。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フォーマットまたは他のフォーマットとすることができる。

10

【0019】

エンコーダ（100）およびデコーダ（200）はブロックベースであり、それぞれのマクロブロックが4つの8×8輝度ブロック（時々、1つの16×16マクロブロックとして扱われる）および2つの8×8クロミナンスブロックを含む、4:2:0のマクロブロックフォーマットを使用する。あるいは、エンコーダ（100）およびデコーダ（200）はオブジェクトベースであり、8×8ブロックおよび16×16マクロブロックとは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し、あるいは異なるサイズまたは構成の複数組の画素に対して動作を行う。

【0020】

所望の圧縮の実装および種類に応じて、エンコーダまたはデコーダのモジュールを追加し、または省略し、または複数モジュールに分割し、または他のモジュールと組み合わせ、かつ/または同様のモジュールで置き換えることが可能である。代替の実施形態では、互いに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用いるエンコーダまたはデコーダで、上述の技術の1つまたは複数を実施する。

20

【0021】

（A．映像エンコーダ）

図1は、一般的な映像エンコーダシステム（100）である。このエンコーダシステム（100）は、現フレーム（105）を含む一連の映像フレームを受け取り、圧縮映像情報（195）を出力として生成する。映像エンコーダの特定の実施形態では通常、変形形態または増補バージョンの汎用エンコーダ（100）を使用する。

30

【0022】

エンコーダシステム（100）は、予測フレームおよびキーフレームを圧縮する。説明のために、図1に、エンコーダシステム（100）を通るキーフレーム用の経路および順方向予測フレーム用の経路を示す。エンコーダシステム（100）の構成要素の多くは、キーフレームおよび予測フレームの両方を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施される具体的な動作は、圧縮される情報の種類に応じて変わり得る。

【0023】

予測フレーム（Pフレーム、両方向予測のためのBフレーム、またはフレーム間符号化フレームとも呼ばれる）は、他の1つまたは複数のフレームからの予測（または差分）で表される。予測残差は、予測したフレームと元のフレームとの差である。これとは対照的に、キーフレーム（Iフレーム、フレーム内符号化フレームとも呼ばれる）は、他のフレームを参照せずに圧縮される。

40

【0024】

現フレーム（105）が順方向予測フレームの場合、動き推定部（110）が、参照フレームに対する現フレーム（105）のマクロブロックまたはその他の複数組の画素の動きを推定する。その参照フレームは、フレーム記憶部（120）にバッファされている、再構築された、以前のフレーム（125）である。代替の実施形態では、参照フレームがより後のフレームであり、あるいは現フレームが両方向に予測される。動き推定部（110）は、動きベクトルなどの動き情報（115）を副次的情報として出力する。動き補償部（130）が、再構築された前のフレーム（125）に動き情報（115）を適用して

50

、動き補償済み現フレーム(135)を形成する。ただし、予測が完全であることはめったになく、動き補償済み現フレーム(135)と元の現フレーム(105)との差が予測残差(145)となる。あるいは、動き推定部および動き補償部が、別の種類の動き推定/補償を適用する。

【0025】

周波数変換部(160)が、空間領域映像情報を周波数領域(すなわちスペクトル)データに変換する。ブロックベースの映像フレームの場合、周波数変換部(160)は、以下のセクションで説明する、離散コサイン変換(「DCT」)に類似する性質の変換を適用する。いくつかの実施形態では、キーフレームの場合、周波数変換部(160)は、複数ブロックの空間予測残差に周波数変換を適用する。周波数変換部(160)は、 8×8 、 8×4 、 4×8 、またはその他のサイズの周波数変換を適用することが可能である。

10

【0026】

次に量子化部(170)が、複数ブロックのスペクトルデータの係数を量子化する。量子化部は、フレームごとに、または他のものに基づいて変わるステップサイズを用いて、スペクトルデータに均一なスカラー量子化を適用する。あるいは、量子化部は、スペクトルデータの係数に別の種類の量子化、たとえば、均一でない、ベクトルの、または非適応型の量子化を適用し、あるいは周波数変換を使用しないエンコーダシステムで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(100)は、コマ落とし(frame dropping)、適応フィルタリング、またはレート制御のためのその他の技術を使用することが可能である。

20

【0027】

後続の動き予測/補償のために再構築された現フレームが必要なとき、逆量子化部(176)が、量子化されたスペクトルデータの係数に対して逆量子化を実施する。次いで逆周波数変換部(166)が、周波数変換部(160)の動作の逆を行い、(予測フレームのための)再構築予測残差または再構築キーフレームを生成する。現フレーム(105)がキーフレームであった場合、再構築キーフレームを、再構築現フレーム(図示せず)として扱う。現フレーム(105)が予測フレームであった場合、再構築予測残差を動き補償済み現フレーム(135)に加えて、再構築現フレームを形成する。フレーム記憶部(120)は、次のフレームの予測に使用するために、再構築現フレームをバッファする。いくつかの実施形態では、エンコーダは、再構築フレームにデブロッキングフィルタを適用し、フレームの各ブロックにおける不連続を適応的に平滑化する。

30

【0028】

エントロピー符号化部(180)が、量子化部(170)の出力、ならびにある種の副次的情報(たとえば、動き情報(115)、量子化ステップサイズ)を圧縮する。一般的なエントロピー符号化技術には、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、および以上の組合せが含まれる。エントロピー符号化部(180)は、一般に異なる種類の情報(たとえば、DC係数、AC係数、異なる種類の副次的情報)には異なる符号化技術を使用しており、特定の符号化技術の範囲内で複数のコードテーブルのうちから選択することが可能である。

【0029】

エントロピー符号化部(180)は、圧縮映像情報(195)をバッファ(190)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フィードバックされる。圧縮映像情報(195)は、一定の、あるいは比較的一定のビットレートでバッファ(190)からすべて読み出され(deplete)、後続のストリーミングに備えてそのビットレートで記憶される。あるいは、エンコーダシステム(100)は、圧縮後すぐに圧縮映像情報を流す。

40

【0030】

バッファ(190)の前または後で、圧縮映像情報(195)を、ネットワークを介して伝送するために、チャンネル符号化することが可能である。チャンネル符号化では、圧縮映像情報(195)に対してエラー検出およびエラー訂正を適用することが可能である。

50

【 0 0 3 1 】

(B . 映像デコーダ)

図 2 は、一般的な映像デコーダシステム (2 0 0) のブロック図である。このデコーダシステム (2 0 0) は、圧縮された一連の映像フレームに関する情報 (2 9 5) を受け取り、再構築フレーム (2 0 5) を含む出力を生成する。映像デコーダの特定の実施形態は一般に、汎用デコーダ (2 0 0) の一変形または増補バージョンを使用する。

【 0 0 3 2 】

デコーダシステム (2 0 0) は、予測フレームおよびキーフレームを伸張する。提示の目的で、図 2 に、デコーダシステム (2 0 0) を通る、キーフレーム用の経路、および順方向予測フレーム用の経路を示す。デコーダシステム (2 0 0) の構成要素の多くは、キーフレームおよび予測フレームの両方を圧縮するのに使用される。これらの構成要素によって実施される具体的な動作は、圧縮される情報の種類に応じて変わり得る。

【 0 0 3 3 】

バッファ (2 9 0) が圧縮映像シーケンスについての情報 (2 9 5) を受け取り、受け取った情報をエントロピーデコーダ (2 8 0) で利用可能にする。バッファ (2 9 0) は、一般に時間が経過してもほぼ一定のレートで情報を受け取り、帯域幅または伝送の短期変動を平滑化するためのジッタバッファを含む。バッファ (2 9 0) は、再生バッファおよびその他のバッファを含むことも可能である。あるいは、バッファ (2 9 0) は、可変レートで情報を受け取る。バッファ (2 9 0) の前または後で、エラーを検出し訂正するために、圧縮映像情報をチャンネル符号化し処理することが可能である。

【 0 0 3 4 】

エントロピー復号化部 (2 8 0) は、一般にエンコーダで実施されるエントロピー符号化の逆を適用して、エントロピー符号化された量子化データならびにエントロピー符号化された副次的情報 (たとえば、動き情報、量子化ステップサイズ) をエントロピー復号化する。エントロピー復号化技術には、算術復号化、差分復号化、ハフマン復号化、ランレングス復号化、LZ 復号化、辞書型復号化、および以上の組合せが含まれる。エントロピー復号化部 (2 8 0) はしばしば、異なる種類の情報 (たとえば、DC 係数、AC 係数、異なる種類の副次的情報) には異なる復号化技術を使用しており、特定の復号化技術内で複数のコードテーブルのうちから選択することが可能である。

【 0 0 3 5 】

再構築すべきフレーム (2 0 5) が順方向予測フレームの場合、動き補償部 (2 3 0) が参照フレーム (2 2 5) に動き情報 (2 1 5) を適用して、再構築されるフレーム (2 0 5) の予測 (2 3 5) を形成する。たとえば、動き補償部 (2 3 0) は、マクロブロック動きベクトルを使用して、参照フレーム (2 2 5) 内でマクロブロックを探す。フレームバッファ (2 2 0) は、参照フレームとして使用するために、前の再構築フレームを記憶する。あるいは、動き補償部は、別の種類の動き補償を適用する。動き補償部による予測が完全であることはめったになく、したがってデコーダ (2 0 0) は予測残差も再構築する。

【 0 0 3 6 】

デコーダが、後続の動き補償のために再構築フレームを必要とするときは、次のフレームを予測するのに使用するために、フレーム記憶部 (2 2 0) が再構築フレームをバッファする。いくつかの実施形態では、エンコーダは、再構築フレームにデブロッキングフィルタを適用して、フレームの各ブロックにおける不連続を適応的に平滑化する。

【 0 0 3 7 】

逆量子化部 (2 7 0) が、エントロピー符号化データを逆量子化する。一般に、逆量子化部は、フレームごとに、またはその他に基づいて変わるステップサイズを用いて、エントロピー符号化データに、一様なスカラー逆量子化を適用する。あるいは、逆量子化部は、そのデータに別の種類の逆量子化、たとえば、非一様な、ベクトルの、または非適応型の量子化を適用し、あるいは逆周波数変換を使用しないデコーダシステムで空間領域データを直接逆量子化する。

10

20

30

40

50

【 0 0 3 8 】

逆周波数変換部 (2 6 0) が、量子化された周波数領域データを、空間領域映像情報に変換する。ブロックベースの映像フレームの場合、逆周波数変換部 (2 6 0) は、以下のセクションで説明する逆変換を適用する。いくつかの実施形態では、キーフレームの場合、逆周波数変換部 (2 6 0) は、複数ブロックの空間予測残差に逆周波数変換を適用する。逆周波数変換部 (2 6 0) は、 8×8 、 8×4 、 4×8 、またはその他のサイズの逆周波数変換を適用することが可能である。

【 0 0 3 9 】

(2 . スライス層符号化)

上記背景技術のセクションで論じたように、パケット損失および伝送エラーによる復号化失敗を回避または最小限に抑える一技術は、いくつかの従来映像コーデック規格で提供されるスライスなど、独立に復号化可能な一部のピクチャ単位で符号化することによって、冗長性を高めることである。一般に、スライスは、1行または連続した複数行のマクロブロックを含む、ピクチャの一部である。

10

【 0 0 4 0 】

スライスの主な狙いの1つは、コーデックで、エラー回復力と圧縮が適切に両立できるようにすることである。その理由は、いくつかの映像コーデック適用例や利用シナリオは、大量のパケット損失を克服しなければならず、したがってエラー回復力を重要視しなければならないからである。他の適用例では、ごく最小限のエラー回復力しか必要とされず、効率的な圧縮が必要とされる。本明細書に記載される映像コーデックのビットストリーム構文の実装形態では、回復力および効率性を最適に選択可能なように設計されたスライス層またはその他の部分ピクチャ単位層が、構文に組み込まれている。これは、以下の方法で行うことができる。

20

【 0 0 4 1 】

A) スライス内 (i n t r a - s l i c e) の完全再構築 : 図示した構文のスライス層は、ループフィルタリングなどの動作ならびにオーバーラップがスライスをまたいで機能しないように設計されている。したがって、あるスライスのすべてのマクロブロックがスライス内符号化され、かつそのスライスに対応するピクチャヘッダがわかっている場合、そのピクチャの他のスライス (領域) 内のエラーとは無関係に、エラーなしで正確にそのスライスを再構築することが可能である。これにより、スライス内の完全な (エラーなしの) 再構築を行えるようになり、エラー回復能力が大幅に向上する。

30

【 0 0 4 2 】

B) フレームヘッダを繰り返す低コストの方法 : ピクチャヘッダ情報を繰り返すと、ピクチャヘッダがデコーダで受け取られるようになる可能性は増大するが、圧縮効率は低下する。図示したスライス層構文は、1ビットのフラグを使用して、あるスライスにおいてそのピクチャヘッダが送信されるかどうかを通知する。これにより、エンコーダは、スライスの数も、ピクチャヘッダが再送信される特定のスライスも選択することができる。

【 0 0 4 3 】

(A . 構文の階層)

より具体的には、図示した映像コーデック構文は、図3に示すように映像シーケンスの各フレームを3つの基本的階層、すなわちピクチャ310、マクロブロック340、およびブロック350に分解する階層構文構造を用いて映像を表す。ピクチャ310は、輝度 (Y) チャンネル330、およびクロミナンス (C r および C b) チャンネル331 ~ 332を含む。ピクチャ層310は、複数行のマクロブロック340からなる。マクロブロックは一般に、それぞれ6つのブロック、すなわち輝度層の1組 2×2 個のブロック、およびクロミナンスチャンネルそれぞれのブロックを含む。ブロックは一般に、(図示した映像コーデック構文では、 4×8 、 8×4 、 4×4 の変換ブロックを使用することも可能であるが) 8×8 の輝度またはクロミナンスサンプルからなり、変換ベースの符号化ではそれに対して変換が適用される。

40

【 0 0 4 4 】

50

さらに、ピクチャ層 3 1 0 とマクロブロック層 3 4 0 の間に、スライス層 3 2 0 と呼ばれる追加の第 4 層が存在することが可能である。スライスは、ラスト走査順で走査される、1 行または連続した複数行のマクロブロックを含むように定義される。したがって、ピクチャ 3 1 0 をスライス 3 2 0 に分解し、このスライスをマクロブロック 3 4 0 に分解することが可能となる。この図示した映像コーデック構文では、スライスは常に、ある行の最初のマクロブロックから始まり、同じまたは別の行の最後のマクロブロックで終わる。したがって、スライスは、完全な行を整数個含む。さらにピクチャおよびスライスは常に、この図示した映像コーデックビットストリーム構文内でバイト整合されており、以下で述べる IDU (independent decodable unit : 独立復号化可能単位) で送信される。新しいピクチャ、またはスライスは、以下に概要を示す開始コードによって検出される。

【 0 0 4 5 】

(B . スライス層の定義)

1 つのスライスは、ラスト走査順で走査される 1 行または連続した複数行のマクロブロックを表す。図示した構文内のスライス層はオプションであり、ピクチャを 1 つの IDU として符号化することによってスキップすることが可能である。1 つのピクチャを複数の IDU に符号化するとき、スライスを使用する。スライスは常に、ある行の最初のマクロブロックから始まり、同じまたは別の行の最後のマクロブロックで終わることに留意されたい。したがって、スライスは、完全な行を整数個含む。スライスは常にバイト整合されており、各スライスは互いに異なる IDU で送信される。新しいスライスの始まりは、以下に概要を示す開始コードを検索することによって検出される。

【 0 0 4 6 】

新しいスライスが始まるとき、動きベクトル予測子、AC 係数および DC 係数の予測子、および量子化パラメータの予測子がリセットされる。言い換えると、予測に関して、スライス内の最初の行のマクロブロックが、ピクチャ内の最初の行のマクロブロックであると見なされる。これにより、予測子におけるスライス間の依存性がなくなる。さらに、スライスが使用されるとき、すべてのビットプレーン情報は、各マクロブロックがその局所情報を含む生モードで運搬される。

【 0 0 4 7 】

(C . スライス層の構文構造)

図 4 および 5 を参照すると、圧縮映像ビットストリーム 1 9 5 (図 1) が、一連の圧縮プログレッシブ映像フレームまたはその他のピクチャ (たとえば、インターレースフレーム形式ピクチャまたはインターレースフィールド形式ピクチャ) を含む。このビットストリームは複数の階層に編成され、それぞれ図 2 のデコーダ (2 0 0) などのデコーダによって復号化される。最上位層が、フレームのシーケンス全体に関する情報を有するシーケンス層である。さらに (先に要約したように)、各圧縮映像フレームは、3 階層、すなわち (上から下に) ピクチャ層、マクロブロック層、およびブロック層と、オプションでピクチャ層とマクロブロック層の間のスライス層とに組織化されたデータからなる。

【 0 0 4 8 】

図 4 は、シーケンスヘッダ 4 1 0 と、それに続くピクチャ層 5 0 0 (図 5 参照) 用のデータとを含むシーケンス層 4 0 0 の構文図である。シーケンスヘッダ 4 1 0 は、デコーダによって処理され、かつ、シーケンスを復号化するのに使用される、いくつかのシーケンスレベル要素を含む。

【 0 0 4 9 】

図 5 は、インターレースのフレーム内符号化フレーム (「インターレース I フレーム」) 用のピクチャ層 5 0 0 の構文図である。プログレッシブの I フレーム、P ピクチャ、および B フレームなど他のピクチャの構文図は、多くの類似の構文要素を有する。ピクチャ層 5 0 0 は、ピクチャヘッダ 5 1 0 と、それに続くマクロブロック層 5 2 0 用のデータとを含む。ピクチャヘッダ 5 1 0 は、デコーダによって処理され、かつ、対応するフレームを復号化するのに使用される、複数のピクチャレベル要素を含む。これらの要素のうちの

10

20

30

40

50

いくつかは、その存在がシーケンスレベル要素または前のピクチャレベル要素によって通知されあるいは暗示される場合にのみ、存在する。

【 0 0 5 0 】

図 6 は、スライスヘッダ 6 1 0 と、それに続くマクロブロック層 5 2 0 用のデータとを含むスライス層 6 0 0 の構文図である。スライスヘッダ 6 1 0 を構成する要素は、以下の表 1 にも示すように、スライスアドレス (S L I C E _ A D D R) 要素 6 2 0、およびピクチャヘッダ存在フラグ (P I C _ H E A D E R _ F L A G) 要素 6 3 0 を含む。

【 0 0 5 1 】

スライスアドレス要素 6 2 0 は、固定長 9 ビットの構文要素である。この構文要素内では、スライスの最初のマクロブロック行の行アドレスが 2 進符号化されている。図示した実装形態において、この構文要素の範囲は 1 から 5 1 1 であり、最大ピクチャサイズ 8 1 9 2 が、最大マクロブロック行数 5 1 2 に対応する。

10

【 0 0 5 2 】

P I C _ H E A D E R _ F L A G 6 3 0 は、スライスヘッダ内に存在する 1 ビットの構文要素である。P I C _ H E A D E R _ F L A G = 0 の場合、スライスヘッダ内でピクチャヘッダ情報が繰り返されない。P I C _ H E A D E R _ F L A G = 1 の場合は、このスライスを含むピクチャ層中に存在するピクチャヘッダ 5 1 0 (図 5) の情報が、スライスヘッダ内で繰り返される。

【 0 0 5 3 】

【 表 1 】

20

表1:スライス層ビットストリーム

SLICE() {	ビット数	
SLICE_ADDR	9	
PIC_HEADER_FLAG	1	
if(PIC_HEADER_FLAG == 1) {		
PICTURE_LAYER()		
}		
for ('all macroblocks') {		
MB_LAYER ()		
}		
}		

30

【 0 0 5 4 】

(3 . 独立復号化可能単位の開始コード)

40

図示した映像エンコーダ/デコーダのビットストリーム構文では、圧縮映像データの独立復号化可能単位 (I D U) が、開始コード (S C) と呼ぶ識別子で始まる。I D U は、1 つのピクチャ、またはスライス (すなわち、ピクチャ内のマクロブロックのグループ)、または G O P (G r o u p O f P i c t u r e s)、またはシーケンスヘッダを参照し得る。

【 0 0 5 5 】

開始コードは、4 バイトのシーケンスであり、一意の 3 バイトの開始コードプレフィックス (S C P)、および 1 バイトの開始コードサフィックス (S C S) からなる。S C P は、一意の 3 バイト (0 x 0 0 0 0 0 1) のシーケンスである。S C S は、I D U の種類を識別するのに使用され、開始コードの後にくる。たとえば、ピクチャの前にある開始コ

50

ードのサフィックスは、スライスの前にある開始コードのサフィックスとは異なる。開始コードは、常にバイト整合されている。

【0056】

ビットストリーム中の開始コードプレフィックスの模倣を防止するためのカプセル化機構（EM）について説明する。カプセル化前の圧縮データは原信号の独立復号化可能単位（RIDU：Independency Decodable Unit）と呼び、一方、カプセル化IDU（EIDU）はカプセル化後のデータを指す。

【0057】

以下のセクションでは、開始コードおよびカプセル化がどのように動作するかに関するエンコーダ側の全体像を示す。セクションE.2では、デコーダにおける開始コードおよびEIDUの検出について述べる。セクションE.3では、EIDUからのRIDUの抽出を扱う。セクションE.4では、様々なIDU種類用の開始コードサフィックスについて述べる。

【0058】

（A.開始コードおよびカプセル化 - エンコーダ側の視点）

EIDUを得るためのRIDUのカプセル化について以下に説明する。

【0059】

ステップ1：RIDUの最後に後続「1」ビットを追加する。次にEMが、このIDUがバイト整合位置で終わるように、0～7ビットをIDUの最後に付加する。これらの「スタッフィング」ビットの値は「0」である。その結果、このステップの最後では、IDUが整数個のバイトで表され、IDUの最後のバイトはゼロ値のバイトではあり得なくなる。その結果得られるバイト列を、IDUのペイロードバイトと呼ぶ。

【0060】

ステップ2：3バイトの開始コードプレフィックス（0x000001）と、IDU種類を識別する適切な開始コードサフィックスをEIDUの最初に置く。

【0061】

ステップ3：EIDUの残りの部分を、以下の模倣防止処理によりIDUのペイロードバイトを処理することによって形成する。バイト充填によってIDU中の開始コードプレフィックスの模倣を排除する。この模倣防止処理は、以下の動作と等価である。

【0062】

1) 値0x00の連続2バイトと、それに続く6ビットのMSB（LSBの値は関係ない）にゼロ値を含む1バイトとからなる、ペイロード中の各ストリングを、値0x00の2バイトと、それに続く0x03に等しい1バイトと、それに続く元の3バイトのデータストリングの最後のバイトに等しい1バイトとで置き換える。この処理を、表2に示す。

【0063】

【表2】

表2:模倣防止パターン置換

置換対象パターン	置換パターン
0x00, 0x00, 0x00	0x00, 0x00, 0x03, 0x00
0x00, 0x00, 0x01	0x00, 0x00, 0x03, 0x01
0x00, 0x00, 0x02	0x00, 0x00, 0x03, 0x02
0x00, 0x00, 0x03	0x00, 0x00, 0x03, 0x03

【0064】

ステップ3：3バイトの開始コードプレフィックス（0x000001）と、IDU種類を識別する適切な開始コードサフィックスとを、IDUの最初に付ける。その結果得ら

れるペイロードが、カプセル化 I D U である。

【 0 0 6 5 】

エンコーダは、E I D U の最後の後に、任意の数のゼロ値の充填バイトを挿入することもできる。同様に、開始コードプレフィックスの前に、任意の数のゼロ値の充填バイトを挿入することも可能である。開始コードは、たとえこれらゼロ値の充填バイトが存在しても、デコーダによって検出可能な構造になっている。H . 3 2 0 などいくつかの伝送環境では、エンコーダはこの機能を使用して、追加のゼロ値充填バイトを必要に応じて挿入することができ、これにより、デコーダがたとえバイト境界に対するビットストリームの所期の整合を見失っても、開始コードの位置を迅速に回復することが可能となる。さらに、これらのゼロ値充填バイトは、ビットストリームを継ぎ合わせる際や一定のビットレートチャンネルを充填する際などでも有用であり得る。開始コードの前の、または E I D U の最後のゼロ値充填バイトはカプセル化機構によって処理されず、R I D U データだけがそうした処理を必要とする。

10

【 0 0 6 6 】

(B . 開始コードおよび E I D U の検出)

E I D U の検出では、まず開始コードプレフィックスを検索する。

【 0 0 6 7 】

バイト整合位置から始まる開始コードの検出：バイト整合を喪失することがあり得ないデコーダ、または一旦バイト整合が確立されているデコーダでは、開始コード検出は以下のように行う。

20

1 . 値 $0 \times 0 0$ の 2 バイト以上と、それに続く値 $0 \times 0 1$ の 1 バイトとからなるストリングが見つかったときはいつでも、開始コードプレフィックス検出が宣言される。

【 0 0 6 8 】

連続した 2 つの開始コードプレフィックスが検出されたとき、それらの間のペイロードビットストリームが新規 E I D U として宣言される。

【 0 0 6 9 】

デコーダにおけるバイト整合の喪失後の開始コードの検出：(一部の伝送環境で発生し得るように) バイト整合を喪失したデコーダにおいて、開始コードプレフィックス検出およびバイト整合検出が以下のように行われる。値 $0 \times 0 0$ の 3 バイト以上のストリングが見つかり、それに続いて任意の非ゼロバイトが見つかったときはいつでも、開始コードプレフィックス検出が宣言され、バイト整合は、非ゼロバイトの最初の非ゼロビットがバイト整合開始コードの最後のビットであるように回復されているものと見なされる。

30

【 0 0 7 0 】

(C . E I D U からの R I D U の抽出)

カプセル化 I D U からの生 I D U の抽出について以下に述べる。

【 0 0 7 1 】

ステップ 1 : 開始コードサフィックスを使用して、I D U の種類を識別する。

【 0 0 7 2 】

ステップ 2 : 最初のステップで、E I D U の最後にあるゼロ値充填バイトを除去する。このステップの後、I D U の最後のバイトは非ゼロ値を有するはずである。

40

【 0 0 7 3 】

ステップ 3 : 模倣防止用に使用されるバイトが検出され除去される。この処理は以下のとおりである。

【 0 0 7 4 】

値 $0 \times 0 0$ の 2 バイトのストリングに $0 \times 0 3$ と等しい 1 バイトが続く場合は常に、この $0 \times 0 3$ と等しい 1 バイトは模倣防止バイトであると見なされ、破棄される。この処理を表 3 に示す。

【 0 0 7 5 】

【表 3】

表3:模倣防止データのデコーダ除去

置換対象パターン	置換パターン
0x00, 0x00, 0x03, 0x00	0x00, 0x00, 0x00
0x00, 0x00, 0x03, 0x01	0x00, 0x00, 0x01
0x00, 0x00, 0x03, 0x02	0x00, 0x00, 0x02
0x00, 0x00, 0x03, 0x03	0x00, 0x00, 0x03

10

【0076】

(デコーダによる適切なバイト整合の喪失がエラー条件と見なされることに留意すると)以下のバイトパターンがビットストリーム中で見つかった場合、それらはエラー条件を表す。

【0077】

a) 値 0×00 の2バイトと、それに続く 0×02 と等しい1バイトとからなるストリングは、エラー条件を示す。

【0078】

b) 値 0×00 の3バイト以上のストリングの後に 0×01 の1バイトが続いていない場合、そのストリングはエラー条件である(ゼロと等しい2バイト以上に値 0×01 の1バイトが続き、バイト整合が喪失していない場合、後続の開始コードの検出が宣言されることに留意されたい)。

20

【0079】

c) 値 0×00 の2バイトと、それに続く値 0×03 の1バイトと、それに続く 0×00 、 0×01 、 0×02 、 0×03 のどれでもない1バイトからなるストリング。

【0080】

ステップ4: IDUの最後のバイトにおいて、最後の非ゼロビットが識別され、その非ゼロビットと、それに続くすべての「ゼロ」ビットが破棄される。その結果が、生IDUである。

30

【0081】

(D-IDU種類用の開始コードサフィックス)

様々なIDU種類用の開始コードサフィックスを表4に提示する。

【0082】

【表 4】

表4:様々なIDU 種類 用の開始コードサフィックス

開始コードサフィックス	IDU 種類
0x00	SMPTE予約済
0x01-0x09	SMPTE予約済
0x0A	シーケンス終了
0x0B	スライス
0x0C	フィールド
0x0D	フレーム
0x0E	エントリポイントヘッダ
0x0F	シーケンスヘッダ
0x10-0x1A	SMPTE予約済
0x1B	スライスレベルユーザデータ
0x1C	フィールドレベルユーザデータ
0x1D	フレームレベルユーザデータ
0x1E	エントリポイントレベルユーザデータ
0x1F	シーケンスレベルユーザデータ
0x20-0x7F	SMPTE予約済
0x80-0xFF	禁止

10

20

30

40

50

【 0 0 8 3 】

シーケンスヘッダサフィックスが送られ、シーケンスヘッダ 4 1 0 (図 4) を有する I D U が識別される。

【 0 0 8 4 】

エントリポイントヘッダサフィックスが送られ、エントリポイントヘッダを有する I D U が識別される。

【 0 0 8 5 】

ピクチャサフィックスが送られ、ピクチャ 3 2 0 (図 3)、およびピクチャヘッダ 5 1 0 (図 5) を含む I D U が識別される。

【 0 0 8 6 】

フィールドサフィックスが送られ、別々の 2 つのフィールドとして符号化されるピクチャの第 2 フィールドを含む I D U が識別される。

【 0 0 8 7 】

スライスサフィックスが送られ、スライス 3 2 0 (図 3)、およびスライスヘッダ 6 1 0 (図 6) を有する I D U が識別される。

【 0 0 8 8 】

シーケンスレベルユーザデータ、エントリポイントレベルユーザデータ、フレームレベルユーザデータ、フィールドレベルユーザデータ、およびスライスレベルユーザデータの各サフィックスが使用され、それぞれシーケンス、エントリポイント、フレーム、フィールド、およびスライスに関連した任意のユーザ定義データが送信される。

【 0 0 8 9 】

「シーケンス終了」は、現シーケンスが終了していることを示す任意のサフィックスであり、このシーケンスではデータはもう送信されない。「シーケンス終了」が送信されることがあるが、あるシーケンスの終了は次のシーケンスのヘッダから推論されるはずであることに留意されたい。

【 0 0 9 0 】

(4 . スライス層の独立性)

図示したスライス層 3 2 0 (図 3) では、独立した復号化能力および独立した再構築も実現される。これにより、ピクチャ 3 1 0 (図 3) の他の領域における伝送エラーやパケット損失に関係なく、デコーダでエラーなしにスライスを再構築することができる。

10

【 0 0 9 1 】

(A . 独立した復号化能力)

スライス層 3 2 0 の内容は、他のスライス内のピクチャ内容またはピクチャの領域とは独立に復号化される。新しいスライスが始まる時、エンコーダ 1 0 0 およびデコーダ 2 0 0 は、動きベクトル予測子、AC 係数および DC 係数用の予測子、ならびに量子化パラメータ用の予測子をリセットする。言い換えると、予測に関して、スライスのマクロブロックの第 1 行が、ピクチャ内のマクロブロックの第 1 行であるかのように扱われる。これは、予測子におけるスライス間依存性がないようにするのに役立つ。さらに、スライスが使用されると、ピクチャ層で他の方法により (たとえばビットプレーン符号化を使用して) 符号化された (動きベクトルモードや AC 予測用のフラグなどの) マクロブロックレベル情報が、変換係数など他のマクロブロックレベル情報と共に局所的に保持される。これにより、各スライスを独立に (すなわち、ピクチャの他のスライスで復号化されたデータを利用せずに) 復号化することができる。

20

【 0 0 9 2 】

(B . 独立した再構築)

さらに、スライスの再構築の処理は、ピクチャ内の他のスライス (たとえば、隣接するスライス) の再構築とは独立に実施される。したがって、(以下に述べるインループデブロッキングまたはオーバーラップフィルタリングなど) 普通ならピクチャ内の隣接するスライス間の境界をまたいで適用されるような処理はいずれも、許可されない。言い換えると、各スライスの一番上および一番下のマクロブロック行は、そのような境界処理においてピクチャのマクロブロック最上行および最下行のように扱われる。

30

【 0 0 9 3 】

(オーバラップ平滑化)

オーバーラップ変換は、ブロック境界をまたいで情報を変換する、改変されたブロックベースの変換である。うまく設計されたオーバーラップ変換では、ブロックアーティファクトを最小にすることが可能である。フレーム内ブロックの場合、図示した映像コーデックは、8 × 8 ブロック変換を (オーバーラップ平滑化と呼ぶ) フィルタリング動作と組み合わせることによって、オーバーラップ変換をシミュレートする。2つのフレーム内ブロックを分離する、8 × 8 ブロックのエッジが平滑化される。実際にも、オーバーラップ変換がこのインターフェースで実装される。ただし、オーバーラップ平滑化は、どんな場合も、スライス境界をまたいで実施されない。

40

【 0 0 9 4 】

シーケンス層構文要素 OVERLAP 4 2 0 (図 4) が 1 に設定されている場合、輝度チャンネルとクロミナンスチャンネルのどちらについても、フィルタリング動作を、隣接する2つのフレーム内ブロックのエッジをまたいで条件付きで行うことができる。(オーバーラップ平滑化と呼ばれる) このフィルタリング動作は、フレームの復号化後、インループデブロッキングの前に行われる。ただし、オーバーラップ平滑化を、該当するマクロブロックスライスが復号化された後に実施することもできる。これは、フレーム全体を復号化した後に平滑化するのと機能的に同等であるからである。

【 0 0 9 5 】

50

図7に、Iブロックを含むPフレームの一部に対して、オーバーラップ平滑化を実施した例を示す。これは、輝度チャンネルでもクロミナンスチャンネルでもよい。Iブロックはグレー（または網がけ）で、Pブロックは白である。この図では、オーバーラップ平滑化の適用対象のエッジインターフェースは、網がけパターンでマークされている。オーバーラップ平滑化は、分離境界の両側の2画素に適用される。フレームの右下領域を例としてここに示す。画素は個々のセルを占め、ブロックは太線によって分離されている。両方向にフィルタされている2×2画素の隅部サブブロックを、太丸でマークしてある。

【0096】

図7の下部の差込み図 (i n s e t) は、4つのラベル付き画素を示しており、a0およびa1が垂直ブロックエッジの左側で、b1およびb0が右側である。上部の差込み図は、水平エッジをまたぐ、p0、p1、q1、およびq0とマークされた画素を示している。次のセクションでは、これら4つの画素位置に適用されるフィルタについて説明する。

10

【0097】

オーバーラップ平滑化は、制限なしの (u n c l a m p) 16ビット再構築に対して実施される。それが必要なのは、オーバーラップ平滑化に関連した順方向処理により、画素値に対して許容可能な8ビットの範囲を超えて範囲が拡張される恐れがあるからである。オーバーラップ平滑化の結果は、オーバーラップ平滑化によって影響を受けない画素の残存部分に合わせて、8ビットに固定 (c l a m p) される。

【0098】

垂直エッジ（上記の例の画素a0、a1、b1、b0）がまずフィルタされ、その後、水平エッジ（画素p0、p1、q1、q0）がフィルタされる。フィルタリングの第1段階（垂直エッジ平滑化）後の中間結果は、16ビットで記憶される。両エッジにまたがる4画素に適用される中核フィルタを次式で示す。

20

【0099】

【数1】

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 7 & 0 & 0 & 1 \\ -1 & 7 & 1 & 1 \\ 1 & 1 & 7 & -1 \\ 1 & 0 & 0 & 7 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} r_0 \\ r_1 \\ r_0 \\ r_1 \end{pmatrix} \gg 3$$

30

【0100】

フィルタされる元の画素は (x 0 、 x 1 、 x 2 、 x 3) である。r0およびr1は、丸めパラメータであり、3と4の値を交互にとり、統計的に偏りのない丸めを保証する。元の値は、明らかに実装の容易なエントリをもつマトリクスによってフィルタされる。これらの値に、丸め係数を加えた後、3ビットだけビットシフトすると、フィルタ済み出力 (y 0 、 y 1 、 y 2 、 y 3) が得られる。

40

【0101】

水平および垂直エッジフィルタのどちらの場合も、ブロック内の番号付けが1で始まると仮定すれば、丸め値は、奇数インデックスの列および行に対してそれぞれ r 0 = 4 、 r 1 = 3 である。偶数インデックスの列/行の場合、r0 = 3 および r 1 = 4 である。フィルタリングは、インプレース (i n - p l a c e) の16ビット動作として定義され、したがって、元の画素は、平滑化後に上書きされる。垂直エッジフィルタリングの場合、画素 (a 0 、 a 1 、 b 1 、 b 0) は (x 0 、 x 1 、 x 2 、 x 3) に対応し、フィルタされて (y 0 、 y 1 、 y 2 、 y 3) になる。同様に、水平エッジフィルタリングの場合、(p 0 、 p 1 、 q 1 、 q 0) がそれぞれ対応する。

【0102】

50

図7の太線の円で示す隅部の 2×2 画素は、両方向にフィルタされる。フィルタリングの順序によってそれらの最終値が決まるため、垂直エッジフィルタリングとそれに続く水平エッジフィルタリングという順序を、ビットレベルの厳密さで維持することが重要である。概念上、制限は、フィルタされるすべての画素に対して、2方向のフィルタリング段階の後に実行される。ただし、制限をフィルタリングと組み合わせることに、ある種の計算上の利点がある可能性がある。

【0103】

(インルーブデブロックフィルタリング)

ループフィルタリングは、ブロック境界において不連続を平滑化するために、映像エンコーダ/デコーダによって実行される処理である。シーケンス層構文要素LOOPFILTER430(図4)が1に設定されている場合、各再構築済みフレームに対してフィルタリング動作が行われる。このフィルタリング動作は、再構築済みフレームを動き予測符号化の参照として使用する前に行われる。

10

【0104】

ループフィルタリングの目的はブロック境界における不連続を平滑化することであるため、このフィルタリング処理は、隣接ブロックに接する画素に対して働く。

【0105】

Pピクチャの場合、ブロック境界は、 8×8 、 8×4 、または 4×8 のどの逆変換が使用されるかに応じて、第4、第8、第12などの画素行または列ごとに生じ得る。Iピクチャの場合、フィルタリングは、第8、第16、第24などの画素行または列ごとに生じる。

20

【0106】

Iピクチャの場合、デブロックフィルタリングは、(ピクチャエッジと同様に扱われる)スライス境界では実施されないことを除き、すべての 8×8 ブロック境界で実施される。図8および9に、Iピクチャフレームの水平および垂直境界領域に沿ってフィルタされる画素を示す。これらの図は、構成要素(ルマ(Luma)、Cr、またはCb)プレーンの左上隅部を示している。十字印は画素を表し、丸十字印はフィルタされた画素を表している。

【0107】

図に示すように、ピクチャまたはスライスの一番上の水平行および最初の垂直行はフィルタされない。図示しないが、ピクチャまたはスライスの一番下の水平行および最後の垂直行もフィルタされない。より正式にいうと、以下の行がフィルタされる。

30

【0108】

N = 平面内の水平 8×8 ブロックの数 ($N \times 8$ = 水平フレームサイズ)

および

M = フレーム内の垂直 8×8 ブロックの数 ($M \times 8$ = 垂直フレームサイズ)

の場合に、

水平行(7, 8)、(15, 16) . . . ((N - 1) × 8 - 1, (N - 1) × 8)

がフィルタされ、

垂直行(7, 8)、(15, 16) . . . ((M - 1) × 8 - 1, (M - 1) × 8)

40

がフィルタされる。

【0109】

画素がフィルタされる順序は重要である。フレーム内のすべての水平境界行がまずフィルタされ、その後、垂直境界行がフィルタされる。

【0110】

Pピクチャの場合、ブロックは、フレーム内符号化することも、フレーム間符号化することもできる。フレーム内符号化ブロックでは常に 8×8 変換を使用してサンプルが変換され、 8×8 ブロック境界は常にフィルタされる。フレーム間符号化ブロックでは、 8×8 、 8×4 、 4×8 、または 4×4 の逆変換を使用して、残存誤差を表すサンプルを構築することができる。隣接ブロックのステータスに応じて、現ブロックと隣接ブロックの間

50

の境界はフィルタされることもフィルタされないこともある。どちらの場合も、ピクチャまたはスライスの境界エッジはフィルタされない。

【0111】

(フィルタ動作)

このセクションでは、上記で論じたような、IフレームおよびPフレーム内のブロック境界画素に対して行われるフィルタリング動作について説明する。

【0112】

行または列内でフィルタされる連続した画素の最小数が4で、かつ行または列内の画素の合計数が常に4の倍数になるので、フィルタリング動作は4画素のセグメントに対して行われる。

10

【0113】

たとえば、2つのブロック間の垂直境界を構成する8つの画素対がフィルタされる場合、図11に示すように、8画素が、2つの4画素セグメント1100に分割される。各4画素セグメントにおいて、Xで示す第3の画素対がまずフィルタされる。このフィルタ動作の結果により、以下に述べるように、セグメント内の他の3つの画素もフィルタされるかどうかが決まる。

【0114】

図12に、第3の画素対に対して行われるフィルタリング動作で使用される画素1200を示す。画素P4およびP5は、フィルタ動作で変更され得る画素対である。

【0115】

図13の擬似コード1300は、各セグメント内の第3の画素対に対して行われるフィルタリング動作を示している。値 `filter__other__3__pixels` は、セグメント内の残る3つの画素対もフィルタされかどうかを示す。`filter__other__3__pixels = 真` の場合、他の3つの画素対がフィルタされる。`filter__other__3__pixels = 偽` の場合、それらはフィルタされず、フィルタリング動作は次の4画素セグメントに進む。図14の擬似コード1400は、`filter__other__3__pixels = 真` の場合に、第1、第2、第4の画素対に対して行われるフィルタリング動作を示している。

20

【0116】

このセクションでは、例示の目的で、垂直境界を使用した。同じ動作は、水平境界画素をフィルタリングするためにも使用される。

30

【0117】

(5. コンピューティング環境)

スライス層符号化の上述の実装は、とりわけコンピュータ、画像および映像記録、送信および受信機器、ポータブルビデオプレーヤ、ビデオ会議、Webビデオストリーミングアプリケーションなどを含めた、画像および映像信号処理が実行される様々な装置のどれに対しても行うことが可能である。画像および映像符号化技術は、ハードウェア回路内(たとえば、ASIC, FPGAなどの回路内)で実装することも、(中央処理装置(CPU)で実行されるものであれ、あるいは専用グラフィックスプロセッサ、ビデオカードなどで実行されるものであれ)コンピュータまたは図10に示すような他のコンピューティング環境内で実行される画像および映像処理ソフトウェアで実装することもできる。

40

【0118】

図10に、上述のスライス層符号化を実装できる適切なコンピューティング環境(1000)の汎用例を示す。本発明は様々な汎用または専用コンピューティング環境で実装できるので、このコンピューティング環境(1000)は、本発明の使用または機能の範囲に関してどんな制限を示唆するものでもない。

【0119】

図10を参照すると、コンピューティング環境(1000)は、少なくとも1つの処理装置(1010)およびメモリ(1020)を含む。図10では、この最も基本的な構成(1030)を破線で囲んである。処理装置(1010)は、コンピュータ実行可能命令

50

を実行し、実プロセッサでも仮想プロセッサでもよい。マルチプロセッシングシステムでは、処理能力を高めるために、複数の処理装置がコンピュータ実行可能命令を実行する。メモリ(1020)は、揮発性メモリ(たとえば、レジスタ、キャッシュ、RAM)、非揮発性メモリ(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(1020)は、上記スライス層符号化を実装するソフトウェア(1080)を記憶する。

【0120】

コンピューティング環境が他の機能を有することもできる。たとえば、コンピューティング環境(1000)は、記憶装置(1040)、1つまたは複数の入力装置(1050)、1つまたは複数の出力装置(1060)、および1つまたは複数の通信接続(1070)を含む。バス、コントローラ、ネットワークなどの相互接続機構(図示せず)が、コンピューティング環境(1000)の各構成要素を相互接続している。通常、オペレーティングシステムソフトウェア(図示せず)は、コンピューティング環境(1000)で実行され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(1000)の各構成要素のアクティビティを調整する。

10

【0121】

記憶装置(1040)は取外し可能でも取外し不能でもよく、それには磁気ディスク、磁気テープまたはカセット、CD-ROM、CD-RW、DVD、または情報を記憶するのに使用可能でありコンピューティング環境(1000)内でアクセス可能であるその他の媒体が含まれる。記憶装置(1040)は、スライス層符号化を実施する音声エンコーダを実装するソフトウェア(1080)用の命令を記憶する。

20

【0122】

入力装置(1050)は、キーボード、マウス、ペン、トラックボールなどの接触式入力装置、音声入力装置、スキャニング装置、またはコンピューティング環境(1000)に入力を提供する他の装置とすることができる。音声の場合、入力装置(1050)は、アナログまたはデジタル形式で音声入力を受け入れるサウンドカードまたは類似の装置、あるいはコンピューティング環境に音声サンプルを提供するCD-ROM読取り装置とすることができる。出力装置(1060)は、ディスプレイ、プリンタ、スピーカ、CD書込み装置、またはコンピューティング環境(1000)からの出力を提供する他の装置とすることができる。

30

【0123】

通信接続(1070)により、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への通信が可能になる。通信媒体は、コンピュータ実行可能命令、圧縮音声または映像情報、その他のデータなどの情報を変調データ信号で運搬する。変調データ信号とは、その信号特性のうちの1つまたは複数の特性が、信号中で情報が符号化されるような形で設定または変更されている信号である。例を挙げると、それだけには限らないが、通信媒体には、電気、光学、RF、赤外、音響、またはその他の搬送体(carrier)で実装される有線または無線技術が含まれる。

【0124】

本明細書のスライス層符号化/復号化技術は、コンピュータ読み取り可能な媒体の一般的文脈中(context)で説明することが可能である。コンピュータ読み取り可能な媒体は、コンピューティング環境内でアクセスできる任意の利用可能な媒体である。例を挙げると、それだけには限らないが、コンピューティング環境(1000)において、コンピュータ読み取り可能な媒体にはメモリ(1020)、記憶装置(1040)、通信媒体、および上記の任意の組合せが含まれる。

40

【0125】

本明細書のスライス層符号化は、対象の実プロセッサまたは仮想プロセッサ上のコンピューティング環境で実行されるプログラムモジュールに含まれるものなどの、コンピュータ実行可能命令の一般的文脈で説明することが可能である。一般に、プログラムモジュールには、特定のタスクを実行する、または特定の抽象データ種類を実装するルーチン、プ

50

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、データ構造などが含まれる。プログラムモジュールの機能は、様々な実施形態で所望されるように、プログラムモジュール間で結合または分割することができる。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、ローカルコンピューティング環境でも分散コンピューティング環境でも実行することができる。

【0126】

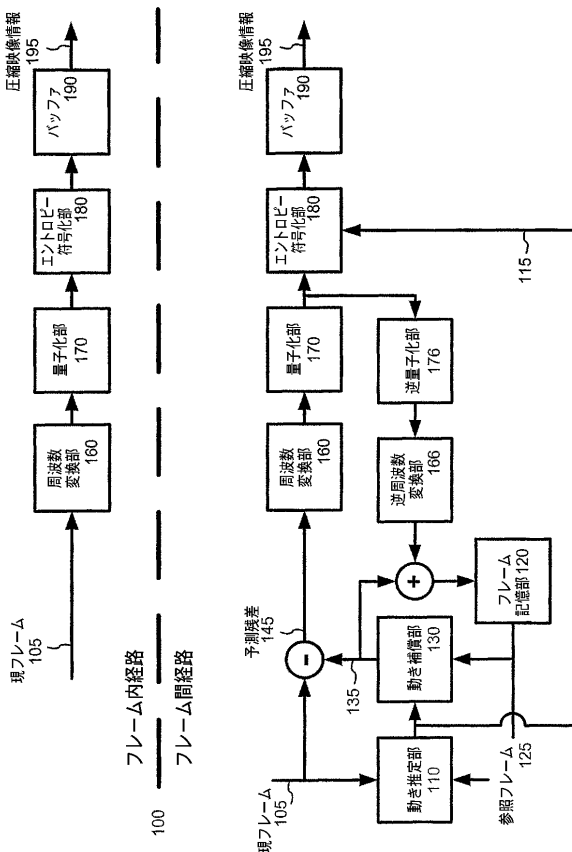
提示の目的で、発明を実施するための最良の形態では、「決定する」、「生成する」、「調整する」、および「適用する」などの用語を使用して、コンピュータ環境内のコンピュータ動作を説明している。これらの用語は、コンピュータによって行われる動作の高レベルの抽象概念であり、人間によって行われる動作と混同すべきではない。これらの用語に対応する実際のコンピュータ動作は、実装に応じて変わってくる。

10

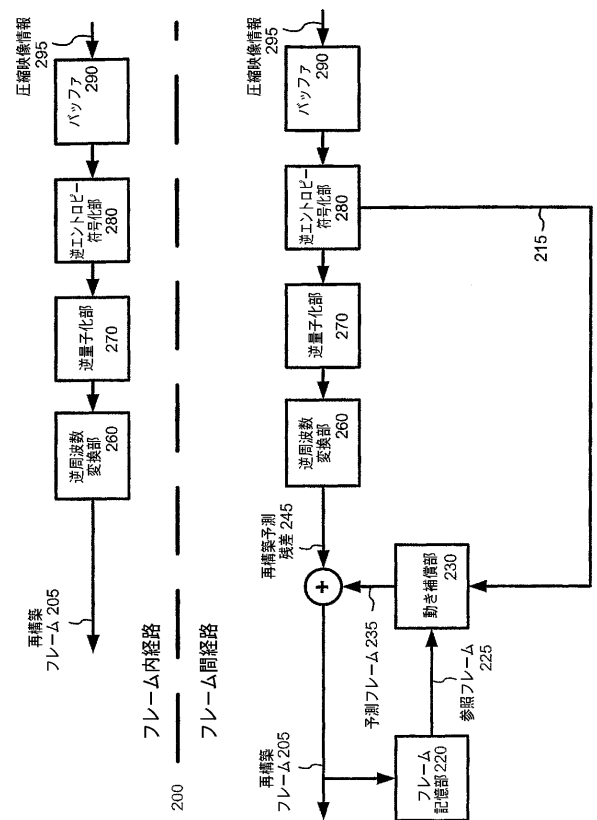
【0127】

本発明の原理を適用できる実施形態が多数考え得ることに鑑みて、本発明者らは、添付の特許請求の範囲およびその等価物の範囲および趣旨に含めることのできるすべての実施形態を本発明者らの発明であると主張するものである。

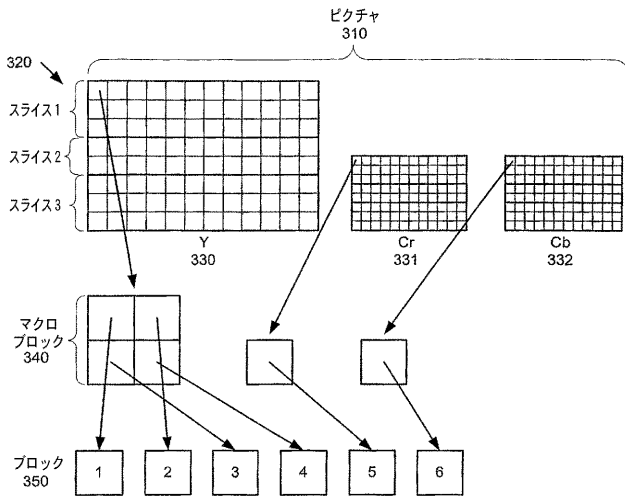
【図1】



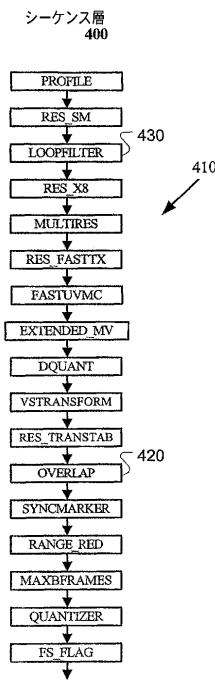
【図2】



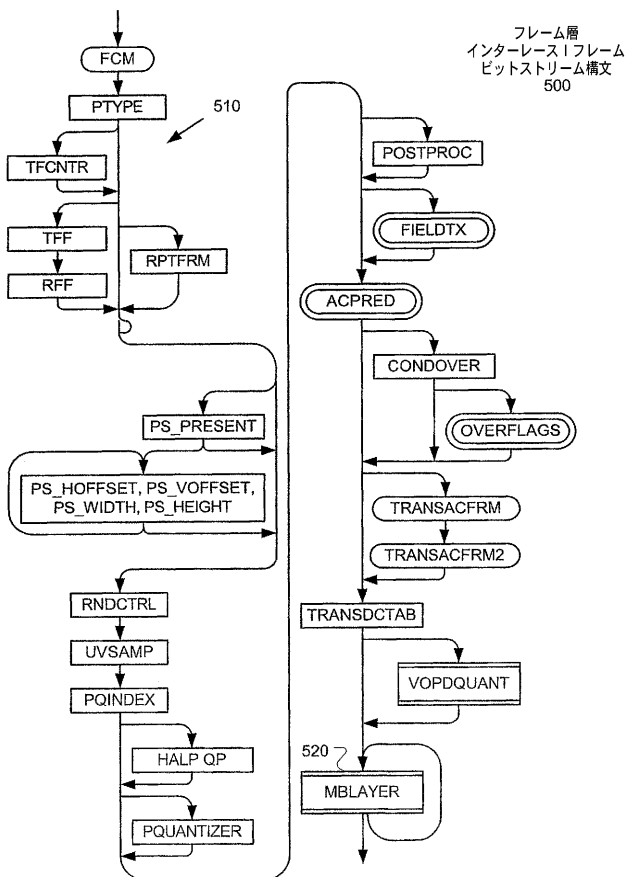
【 図 3 】



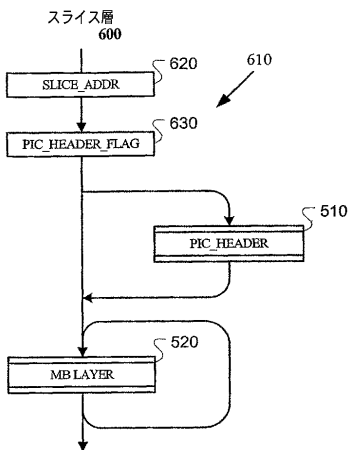
【 図 4 】



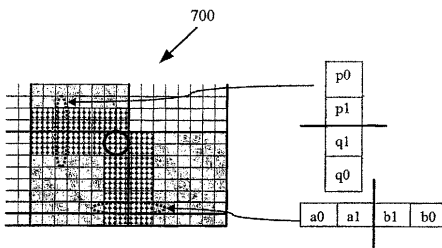
【 図 5 】



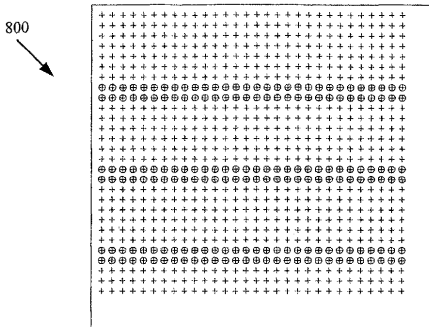
【 図 6 】



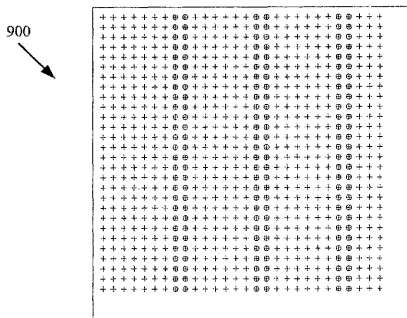
【 図 7 】



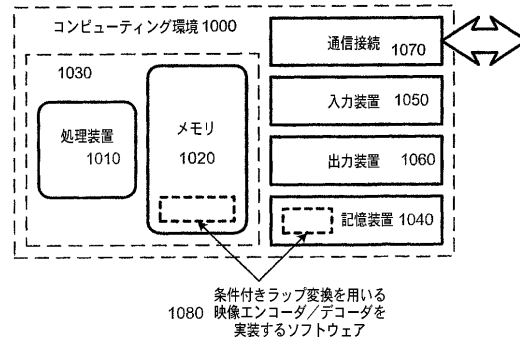
【 図 8 】



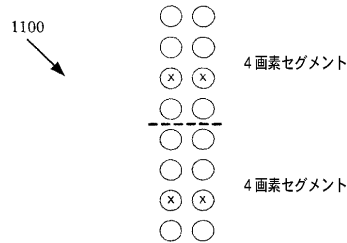
【 図 9 】



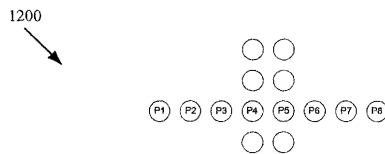
【 図 10 】



【 図 11 】



【 図 12 】



【 図 13 】

```

1300 filter_other_3_pixels = true
      a0 = (2*(P3 - P6) - 5*(P4 - P5) + 4) >> 3
      if (|a0| < PQUANT) {
        a1 = (2*(P1 - P4) - 5*(P2 - P3) + 4) >> 3
        a2 = (2*(P5 - P8) - 5*(P6 - P7) + 4) >> 3
        a3 = min(|a1|, |a2|)
        if (a3 < |a0|)
        {
          d = 5*((sign(a0) * a3) - a0)/8
          clip = (P4 - P5)/2
          if (clip == 0)
            filter_other_3_pixels = false
          else
          {
            if (clip > 0)
            {
              if (d < 0)
                d = 0
              if (d > clip)
                d = clip
            }
            else
            {
              if (d > 0)
                d = 0
              if (d < clip)
                d = clip
            }
            P4 = P4 - d
            P5 = P5 + d
          }
        }
      }
      else
        filter_other_3_pixels = false
    }
    else
      filter_other_3_pixels = false
  
```

【 図 14 】

```

1400 a0 = (2*(P3 - P6) - 5*(P4 - P5) + 4) >> 3
      if (|a0| < PQUANT)
      {
        a1 = (2*(P1 - P4) - 5*(P2 - P3) + 4) >> 3
        a2 = (2*(P5 - P8) - 5*(P6 - P7) + 4) >> 3
        a3 = min(|a1|, |a2|)
        if (a3 < |a0|)
        {
          d = 5*((sign(a0) * a3) - a0)/8
          clip = (P4 - P5)/2

          if (clip > 0)
          {
            if (d < 0)
              d = 0
            if (d > clip)
              d = clip
            P4 = P4 - d
            P5 = P5 + d
          }
          else if (clip < 0)
          {
            if (d > 0)
              d = 0
            if (d < clip)
              d = clip
            P4 = P4 - d
            P5 = P5 + d
          }
        }
      }
    }
  
```

フロントページの続き

- (72)発明者 ボシャン スウ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 セ ワン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 チ - ラン プルース リン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 ジェエ リアン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 スリダー スリニバサン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内

F ターム(参考) 5C159 KK03 LC10 MA04 MA05 MA12 MA23 MA35 MC11 MC32 MC34
ME01 PP05 PP06 PP16 PP25 PP30 RB09 RB13 RB14 RF02
RF05 RF09 RF13 TA69 TA76 TB06 TB07 TC22 TC24 TC33
UA02 UA05 UA16