

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7569413号
(P7569413)

(45)発行日 令和6年10月17日(2024.10.17)

(24)登録日 令和6年10月8日(2024.10.8)

(51)国際特許分類

F I

H 0 4 N 19/11 (2014.01) H 0 4 N 19/11

H 0 4 N 19/136(2014.01) H 0 4 N 19/136

H 0 4 N 19/176(2014.01) H 0 4 N 19/176

請求項の数 28 (全53頁)

(21)出願番号	特願2023-105336(P2023-105336)	(73)特許権者	000001007
(22)出願日	令和5年6月27日(2023.6.27)		キャノン株式会社
(62)分割の表示	特願2021-544700(P2021-544700)の分割		東京都大田区下丸子3丁目30番2号
原出願日	令和2年4月15日(2020.4.15)	(74)代理人	110003281
(65)公開番号	特開2023-123706(P2023-123706 A)		弁理士法人大塚国際特許事務所
(43)公開日	令和5年9月5日(2023.9.5)	(72)発明者	ロゼワーン, クリストファー ジェームズ
審査請求日	令和5年6月27日(2023.6.27)		オーストラリア国 2 1 1 3 ニューサウ
(31)優先権主張番号	2019204437		スウェールズ州 タラベラロード パーク
(32)優先日	令和1年6月24日(2019.6.24)		エステイト5 ビルディング エー キャ
(33)優先権主張国・地域又は機関	オーストラリア(AU)		ノン オーストラリア プロプライエタリ
			ー リミテッド内
		(72)発明者	アフメド, イフテクハ
			オーストラリア国 2 1 1 3 ニューサウ
			スウェールズ州 タラベラロード パーク
			エステイト5 ビルディング エー キャ
			最終頁に続く

(54)【発明の名称】 ビデオサンプルのブロックを符号化及び復号するための方法、装置、及びプログラム

(57)【特許請求の範囲】

【請求項1】

画像フレームのための符号化ツリーユニットにおける対象ブロックに対する予測サンプルを生成する方法であって、

前記対象ブロックに対する行列イントラ予測フラグを復号する工程であって、前記行列イントラ予測フラグは行列イントラ予測が前記対象ブロックに対し用いられるかどうかを示す、工程と、

前記行列イントラ予測フラグが前記対象ブロックに対し行列イントラ予測が用いられることを示す場合、前記対象ブロックに対する行列イントラ予測モードを復号する工程と、
前記行列イントラ予測モードに従って、前記対象ブロックに対する行列イントラ予測のための行列を選択する工程と、

前記対象ブロックに隣接するサンプルに基づく入力サンプルと、前記行列イントラ予測モードに応じて選択された前記行列との乗算を適用することにより、前記予測サンプルを生成する工程と、を含み、

短縮された二値符号が前記行列イントラ予測モードに対し用いられることが可能であり、

前記符号化ツリーユニットにおける或るエリアが、各々が16×8のサイズを有する4つのブロックに分割され、前記対象ブロックが前記4つのブロックのうちの1つである場合において、前記対象ブロックに対する前記行列イントラ予測フラグの復号は、前記対象ブロックに隣接する2つのブロックの行列イントラ予測の使用の状況に依存することを特徴とする方法。

10

【請求項 2】

前記対象ブロックに対する前記行列イントラ予測モードの復号は、前記対象ブロックの幅および前記対象ブロックの高さに依存することを特徴とする請求項 1 に記載の方法。

【請求項 3】

複数のブロックサイズが利用可能であり、前記複数のブロックサイズは、第 1 レベル、第 2 レベル又は第 3 レベルに分類され、

前記第 1 レベルは少なくとも 4×4 サイズを含み、

前記第 1 レベルに対する行列イントラ予測に対し利用可能である行列の数は、前記第 2 レベルに対する行列イントラ予測に対し利用可能である行列の数より大きいことを特徴とする請求項 1 に記載の方法。

10

【請求項 4】

前記第 1 レベルに対する行列イントラ予測に対し利用可能である行列の数は、前記第 3 レベルに対する行列イントラ予測に対し利用可能である行列の数より大きいことを特徴とする請求項 3 に記載の方法。

【請求項 5】

前記第 2 レベルに含まれるブロックサイズは 4×4 サイズより大きいことを特徴とする請求項 3 に記載の方法。

【請求項 6】

前記第 3 レベルに含まれるブロックサイズは 4×4 サイズより大きいことを特徴とする請求項 4 に記載の方法。

20

【請求項 7】

前記符号化ツリーユニット内のブロック分割のための分割フラグを復号することにより、前記符号化ツリーユニット内の前記対象ブロックを含む複数のブロックを決定する工程であって、前記ブロック分割として、水平方向の三分割が用いられることが可能である、工程を更に有することを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記行列イントラ予測フラグの復号は、前記対象ブロックの幅および前記対象ブロックの高さに依存することを特徴とする請求項 1 に記載の方法。

【請求項 9】

前記対象ブロックのサイズが 8×16 のサイズおよび 16×8 のサイズのうちの 1 つである場合において、前記行列イントラ予測モードは、前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な全ての行列イントラ予測モードのうちの 1 つであり、どの行列イントラ予測モードが使用されるかに関わらず、前記行列イントラ予測モードに対し短縮された二値符号が用いられることを特徴とする請求項 1 に記載の方法。

30

【請求項 10】

前記対象ブロックのサイズが 8×16 のサイズおよび 16×8 のサイズのうちの 1 つである場合において、(a) 前記行列イントラ予測モードは、前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な複数の行列イントラ予測モードのうちの 1 つであり、(b) 前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な複数の行列イントラ予測モードの数は n であり (n は 0 より大きい整数値)、(c) 前記行列イントラ予測モードは、0 以上且つ $n - 1$ 以下の整数値により表され、(d) 0 以上且つ所定の整数値以下の整数値により表される前記行列イントラ予測モードに対して m ビット (m は 0 より大きい整数値) の符号が用いられ、(e) 前記所定の整数値より大きく且つ $n - 1$ 以下の整数値により表される前記行列イントラ予測モードに対して $m + 1$ ビットの符号が用いられることを特徴とする請求項 1 に記載の方法。

40

【請求項 11】

n は、2 の m 乗より大きく且つ 2 の $m + 1$ 乗より小さい整数値であることを特徴とする請求項 10 に記載の方法。

【請求項 12】

前記ブロック分割として、垂直方向の三分割を用いることが可能であることを特徴とす

50

る請求項 7 に記載の方法。

【請求項 1 3】

画像フレームのための符号化ツリーユニットにおける対象ブロックに対する予測サンプルを生成する方法であって、

前記対象ブロックに対する行列イントラ予測のための行列を選択する工程と、

前記対象ブロックに隣接するサンプルに基づく入力サンプルと、前記選択された行列との乗算を適用することにより、前記予測サンプルを生成する工程と、

行列イントラ予測が前記対象ブロックに対し用いられるかどうかを示す行列イントラ予測フラグを符号化する工程と、

前記対象ブロックに対し行列イントラ予測が用いられる場合、前記選択された行列を特定するために用いられる行列イントラ予測モードを符号化する工程と、を含み、

短縮された二値符号が前記行列イントラ予測モードに対し用いられることが可能であり、

前記符号化ツリーユニットにおける或るエリアが、各々が 16×8 のサイズを有する 4 つのブロックに分割され、前記対象ブロックが前記 4 つのブロックのうちの 1 つである場合において、前記対象ブロックに対する前記行列イントラ予測フラグの符号化は、前記対象ブロックに隣接する 2 つのブロックの行列イントラ予測の使用の状況に依存することを特徴とする方法。

【請求項 1 4】

前記対象ブロックに対する前記行列イントラ予測モードの符号化は、前記対象ブロックの幅および前記対象ブロックの高さに依存することを特徴とする請求項 1 3 に記載の方法。

【請求項 1 5】

複数のブロックサイズが利用可能であり、前記複数のブロックサイズは、第 1 レベル、第 2 レベル又は第 3 レベルに分類され、

前記第 1 レベルは少なくとも 4×4 サイズを含み、

前記第 1 レベルに対する行列イントラ予測に対し利用可能である行列の数は、前記第 2 レベルに対する行列イントラ予測に対し利用可能である行列の数より大きいことを特徴とする請求項 1 3 に記載の方法。

【請求項 1 6】

前記第 1 レベルに対する行列イントラ予測に対し利用可能である行列の数は、前記第 3 レベルに対する行列イントラ予測に対し利用可能である行列の数より大きいことを特徴とする請求項 1 5 に記載の方法。

【請求項 1 7】

前記第 2 レベルに含まれるブロックサイズは 4×4 サイズより大きいことを特徴とする請求項 1 5 に記載の方法。

【請求項 1 8】

前記第 3 レベルに含まれるブロックサイズは 4×4 サイズより大きいことを特徴とする請求項 1 5 に記載の方法。

【請求項 1 9】

ブロック分割により前記符号化ツリーユニット内の前記対象ブロックを含む複数のブロックを決定する工程であって、前記ブロック分割として、水平方向の三分割が用いられることが可能である、工程を更に有する請求項 1 3 に記載の方法。

【請求項 2 0】

前記行列イントラ予測フラグの符号化は、前記対象ブロックの幅および前記ブロックの高さに依存することを特徴とする請求項 1 3 に記載の方法。

【請求項 2 1】

前記ブロックのサイズが 8×16 のサイズおよび 16×8 のサイズのうちの 1 つである場合において、前記行列イントラ予測モードは、前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な全ての行列イントラ予測モードのうちの 1 つであり、どの行列イントラ予測モードが使用されるかに関わらず、前記行列イントラ予測モードに対し短縮された二値符号が用いられることを特徴とする請求項 1 3 に記載の方法。

10

20

30

40

50

【請求項 2 2】

前記対象ブロックのサイズが 8×16 のサイズおよび 16×8 のサイズのうちの 1 つである場合において、(a) 前記行列イントラ予測モードは、前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な複数の行列イントラ予測モードのうちの 1 つであり、(b) 前記 8×16 のサイズおよび前記 16×8 のサイズに対し利用可能な複数の行列イントラ予測モードの数は n であり(n は 0 より大きい整数値)、(c) 前記行列イントラ予測モードは、0 以上且つ $n - 1$ 以下の整数値により表され、(d) 0 以上且つ所定の整数値以下の整数値により表される前記行列イントラ予測モードに対して m ビット(m は 0 より大きい整数値) の符号が用いられ、(e) 前記所定の整数値より大きく且つ $n - 1$ 以下の整数値により表される前記行列イントラ予測モードに対して $m + 1$ ビットの符号が用いられることを特徴とする請求項 1 に記載の方法。

10

【請求項 2 3】

n は、2 の m 乗より大きく且つ 2 の $m + 1$ 乗より小さい整数値であることを特徴とする請求項 2 2 に記載の方法。

【請求項 2 4】

前記ブロック分割として、垂直方向の三分割を用いることが可能であることを特徴とする請求項 1 9 に記載の方法。

【請求項 2 5】

画像フレームのための符号化ツリーユニットにおける対象ブロックに対する予測サンプルを生成する装置であって、

20

前記対象ブロックに対する行列イントラ予測フラグを復号する手段であって、前記行列イントラ予測フラグは行列イントラ予測が前記対象ブロックに対し用いられるかどうかを示す手段と、

前記行列イントラ予測フラグが前記対象ブロックに対し行列イントラ予測が用いられることを示す場合、前記対象ブロックに対する行列イントラ予測モードを復号する手段と、

前記行列イントラ予測モードに従って、前記対象ブロックに対する行列イントラ予測のための行列を選択する手段と、

前記対象ブロックに隣接するサンプルに基づく入力サンプルと、前記行列イントラ予測モードに応じて選択された前記行列との乗算を適用することにより、前記予測サンプルを生成する手段と、を含み、

30

短縮された二値符号が前記行列イントラ予測モードに対し用いられることが可能であり、

前記符号化ツリーユニットにおける或るエリアが各々が 16×8 のサイズを有する 4 つのブロックに分割され、前記対象ブロックが前記 4 つのブロックのうちの 1 つである場合において、前記対象ブロックに対する前記行列イントラ予測フラグの復号は、前記ブロックに隣接する 2 つのブロックの行列イントラ予測の使用の状況に依存することを特徴とする装置。

【請求項 2 6】

画像フレームのための符号化ツリーユニットにおける対象ブロックに対する予測サンプルを生成する装置であって、

前記対象ブロックに対し行列イントラ予測のための行列を選択する手段と、

40

前記対象ブロックに隣接するサンプルに基づく入力サンプルと、前記選択された行列との乗算を適用することにより、前記予測サンプルを生成する手段と、

行列イントラ予測が前記対象ブロックに対し用いられるかどうかを示す行列イントラ予測フラグを符号化する手段と、

前記対象ブロックに対し行列イントラ予測が用いられる場合、前記選択された行列を特定するために用いられる行列イントラ予測モードを符号化する手段と、を含み、

短縮された二値符号が前記行列イントラ予測モードに対し用いられることが可能であり、

前記符号化ツリーユニットにおける或るエリアが各々が 16×8 のサイズを有する 4 つのブロックに分割され、前記対象ブロックが前記 4 つのブロックのうちの 1 つである場合において、前記対象ブロックに対する前記行列イントラ予測フラグの符号化は、前記対象

50

ブロックに隣接する２つのブロックの行列イントラ予測の使用の状況に依存することを特徴とする装置。

【請求項 ２ ７】

コンピュータに請求項 １ に記載の方法を実行させるためのコンピュータプログラム。

【請求項 ２ ８】

コンピュータに請求項 １ ３ に記載の方法を実行させるためのコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【 ０ ０ ０ １ 】

関連出願への参照

10

本発明は、米国特許法第 １ １ ９ 条に基き、２ ０ １ ９ 年 ６ 月 ２ ４ 日に出願されたオーストラリア特許出願第 ２ ０ １ ９ ２ ０ ４ ４ ３ ７ 号の利益を主張し、これにより、当該発明が本明細書に完全に記載されているものとする。

技術分野

【 ０ ０ ０ ２ 】

本発明は、一般に、デジタルビデオ信号処理に関し、特に、ビデオサンプルのブロックを符号化及び復号するための方法、装置に関する。本発明はまた、ビデオサンプルのブロックを符号化及び復号するためのコンピュータプログラムに関する。

【背景技術】

【 ０ ０ ０ ３ 】

20

現在、ビデオデータの送信及び記憶のためのアプリケーションを含む、ビデオコーディングのための多くのアプリケーションが存在する。多くのビデオコーディング標準が開発され、現在開発中のものも存在する。最近のビデオコーディングの標準化の進展により、「合同ビデオ専門家チーム」(J V E T) と呼ばれるグループが結成された。合同ビデオ専門家チーム(J V E T) には、「ビデオコーディング専門家グループ」(V C E G) として知られている国際電気通信連合(I T U) の電気通信標準化部門(I T U - T) 第 １ ６ 研究委員会第 ６ 諮問(S G １ ６ / Q ６) のメンバーと、「動画専門家グループ」(M P E G) として知られている国際標準化機構/国際電気標準会議の第 １ 合同技術委員会/第 ２ ９ 副委員会/第 １ １ 作業部会(I S O / I E C J T C １ / S C ２ ９ / W G １ １) のメンバーが含まれる。

30

【 ０ ０ ０ ４ 】

合同ビデオ専門家チーム(J V E T) は、提案募集(C f P) を行い、米国サンディエゴにおける第 １ ０ 回会議で応募された提案を分析した。応募された提案は、ビデオ圧縮機能が現在の最先端のビデオ圧縮標準、つまり「高効率ビデオコーディング」(H E V C) を大幅に上回っていることを示した。この分析結果に基づいて、「Versatile Video Coding」(V V C) という名前の新しいビデオ圧縮標準を開発するプロジェクトを開始することが決定された。V V C は、特にビデオ形式の機能が向上するにつれて(例えば、より高い解像度及びより高いフレームレート)、これまで以上に高い圧縮性能に対する継続的な需要に対応すると共に、帯域幅コストが比較的高い W A N を介したサービス配信に対する市場の需要の高まりに対応することが期待されている。同時に、V V C は、従来のシリコン

40

【 ０ ０ ０ ５ 】

ビデオデータは、連続するフレームの画像データを含み、各フレームの画像データは、１以上のカラーチャネルを含む。一般に、１つの原色チャネルと２つの二次色チャネルが必要である。原色チャネルは一般に「ルマ」チャネルと呼ばれ、二次色チャネルは一般に「クロマ」チャネルと呼ばれる。ビデオデータは通常、R G B (赤 - 緑 - 青) 色空間で表示されるが、この色空間は３つの各成分間で高い相関関係を有する。エンコードまたはデコードは、多くの場合、Y C b C r などの色空間を用いてビデオデータを表現する。Y C

50

b C r は、伝達関数に従って「ルマ」にマッピングされた輝度を、Y（一次）チャンネルに集中させ、クロマをC b 及びC r（二次）チャンネルに集中させる。さらに、C b、C r チャンネルは、ルマチャンネルと比較して、例えば、「4：2：0 クロマフォーマット」として知られる、水平方向に半分、垂直方向に半分の、空間的に低いレートでサンプリング（サブサンプリング）され得る。4：2：0 クロマフォーマットは、インターネットビデオストリーミング、テレビ放送、Blu-ray（登録商標）ディスクへの保存などの「消費者向け」アプリケーションにおいて一般的に使用されている。C b、C r チャンネルを水平方向に半分のレートでサブサンプリングし、垂直方向にサブサンプリングしないフォーマットは、「4：2：2 クロマフォーマット」として知られている。4：2：2 クロマフォーマットは、通常、映画制作の映像の撮影などを含む、プロ向けのアプリケーションで使用されている。4：2：2 クロマフォーマットのサンプリングレートが高いほど、結果として得られる映像のカラーグレーディングなどの編集作業に対する柔軟性が高くなる。消費者に提供される前に、4：2：2 クロマフォーマットの素材は、しばしば4：2：0 クロマフォーマットに変換されてから、消費者に提供するために符号化される。クロマフォーマットに加えて、ビデオは解像度とフレームレートによっても特徴付けられている。解像度の例として、解像度が3840×2160の超高精細（UHD）、または、解像度が7680×4320の「8K」があり、フレームレートの例として、60Hzまたは120Hzがある。ルマサンプルレートは、例えば、毎秒約500メガサンプルから毎秒数ギガサンプルの間にある。4：2：0 クロマフォーマットの場合、各クロマチャンネルのサンプルレートは、ルマサンプルレートの4分の1であり、4：2：2 クロマフォーマットの場合、各クロマチャンネルのサンプルレートはルマサンプルレートの半分である。

【0006】

VVC規格は、「ブロックに基づく」符号化／復号であり、フレームは、まず、「符号木の単位(Coding Tree Unit)」（CTU）として知られる正方形の領域に分割される。CTUは通常、128×128のルマサンプルなどの比較的広い領域を占める。しかし、各フレームの右下端のCTUは、面積が小さい場合がある。各CTUには、ルマチャンネルの「符号木」と、クロマチャンネルの追加符号木が関連付けられている。符号木は、CTUの領域を、「符号化ブロック(Coding Block)」（CB）とも呼ばれるブロックの組に分解することを定義している。一つの符号木で、ルマチャンネルとクロマチャンネルの両方のブロックを特定することもでき、その場合、同じ場所にある符号化ブロックは、まとめて「符号化ユニット(Coding Unit)」（CU）と呼ばれる。つまり、各CUは、各カラーチャンネルの符号化ブロックを有する。CBは、特定の順序で符号化処理または復号処理される。4：2：0 クロマフォーマットを使用した場合、128×128のルマサンプル領域用のルマ符号木を有するCTUは、128×128のルマサンプル領域と同じ場所にある、64×64のクロマサンプル領域用の対応するクロマ符号木を有することになる。一つの符号木がルマチャンネルとクロマチャンネルに使用されている場合、ある領域と同じ位置にあるブロックの集合は、例えば、上記CUや、「予測単位(Prediction Unit)」（PU）、及び「変換単位(Transform Unit)」（TU）というように、一般に「単位」と呼ばれる。ある領域について別々の符号木を使用する場合、上記CBや、「予測ブロック(Prediction Block)」（PB）、及び「変換ブロック(Transform Block)」（TB）が用いられる。

【0007】

なお、「単位」と「ブロック」との間に上記の区別があるが、「ブロック」という用語は、すべてのカラーチャンネルに処理が適用されるフレームの領域を表す一般的な用語としても使用される。

【0008】

各CUについて、フレームデータの対応する領域のコンテンツ（サンプル値）の予測単位（PU）が生成される（「予測単位」）。さらに、エンコーダへの入力時に見られる領域の予測とコンテンツとの間の差異（または空間ドメインにおける「残差」）を示すものが形成される。各カラーチャンネルの差は、一連の残差係数として変換及び符号化され、あ

10

20

30

40

50

るCUについて1つ以上のTUを形成する。適用される変換は、離散コサイン変換(DCT)または他の変換であり、各ブロックの残差値に適用される。この変換は個別に適用される。つまり、二次元変換は2つのパスで実行される。ブロックは、まず、ブロック内の各行のサンプルに一次元変換を適用することによって変換される。次に、部分的な結果の各列に一次元変換を適用して当該部分的な結果を変換し、残差サンプルを実質的に無相関化する変換係数の最終的なブロックを生成する。さまざまなサイズの変換がVVC標準でサポートされており、これには、各辺の寸法が2のべき乗となる長方形のブロックの変換が含まれる。変換係数は、ビットストリームへのエントロピー符号化のために量子化される。

【0009】

VVCは、フレーム内予測及びフレーム間予測を特徴とする。フレーム内予測は、フレーム内の現ブロックのサンプルの予測値を生成するために使用されているフレーム内における、以前に処理されたサンプルの使用を含む。フレーム間予測は、以前に復号されたフレームから取得したブロックのサンプルを使用して、フレーム内の現ブロックのサンプルの予測値を生成することを含む。以前に復号されたフレームは、動きベクトルに従って現ブロックの空間位置からオフセットされ、多くの場合、フィルタ処理が適用されている。フレーム内予測ブロックは、均一なサンプル値(「DCイントラ予測」)であっても、オフセット及び水平及び垂直勾配を持つ平面(「平面イントラ予測」)であっても、または特定方向に隣接するサンプルが適用されたブロックの母集団(「角度イントラ予測」)、または、隣接するサンプルと選択された行列係数を使用した行列乗算の結果であってもよい。隣接するサンプルには以前に処理されたブロックからのサンプルが含まれるため、フレーム内予測のフィードバックループは大きく制限されており、サポートされる最高の解像度とフレームレートを満たすために必要なレベルよりも計算の複雑さを低く抑える必要がある。

【発明の概要】

【0010】

本発明の目的は、既存の構成の1つまたは複数の欠点を実質的に克服するか、または少なくとも改善することである。

【0011】

本開示の一つの側面によれば、ビデオビットストリームから画像フレームのための符号木の符号化ユニットを復号する方法を提供し、前記方法は、前記符号木の領域を、各符号化ブロックが予測ブロックを含む、複数の符号化ブロックに分割する工程と、前記各符号化ブロックの前記予測ブロックについて、行列イントラ予測フラグを判断する工程であって、各行列イントラ予測フラグは、前記符号化ブロックのうちの1つの前記予測ブロックに行列イントラ予測が用いられたかを示し、前記判断を(i)前記領域が閾値を満たす場合には面積、または(ii)前記領域の面積が前記閾値を満たさない場合には前記領域の割当量に基づいて行い、前記判断されたフラグに応じて行列イントラ予測を用いると判断された各予測ブロックについて、メモリから行列係数を読み出す工程と、各予測ブロックの参照サンプル及び前記行列係数を用いて前記領域における各符号化ユニットについて生成された予測ブロックを用いて、符号化ユニットを復号する工程と、を含む。

【0012】

別の側面によれば、前記閾値は、512ルマサンプルよりも大きいサイズである。

【0013】

別の側面によれば、前記閾値は、64ルマサンプルよりも大きいサイズである。

【0014】

別の側面によれば、前記割当量では、前記領域について4×4ブロックの40ワードの読み出すことができる。

【0015】

別の側面によれば、行列イントラ予測が用いられた場合にのみ、前記CUの行列イントラ予測フラグは復号される。

10

20

30

40

50

【 0 0 1 6 】

別の側面によれば、行列イントラ予測が用いられたかどうかに関わらず、前記 C U の行列イントラ予測フラグは復号される。

【 0 0 1 7 】

本開示の別の側面によれば、ビデオビットストリームから画像フレームのための符号木の符号化ユニットを復号する方法を提供し、前記方法は、前記符号木の領域を、それぞれが予測ブロックを含む、複数の符号化ブロックに分割する工程と、前記符号化ブロックの予測ブロックについて、各符号化ブロックのサイズに基づいて、行列イントラ予測フラグを判断する工程であって、各行列イントラ予測フラグは、前記対応する符号化ブロックの前記予測ブロックに行列イントラ予測が用いられたかを示し、前記判断されたフラグに応じて行列イントラ予測を用いると判断された各予測ブロックについて、メモリから行列係数を読み出す工程と、各予測ブロックの参照サンプル及び前記行列係数を用いて生成された前記領域における各符号化ユニットの予測ブロックから符号化ユニットを復号する工程と、を含む。

10

【 0 0 1 8 】

別の側面によれば、前記符号化ユニットの前記サイズが 4×4 で無い場合に行列イントラ予測フラグは復号される。

【 0 0 1 9 】

別の側面によれば、前記符号化ユニットの前記サイズが 4×4 、 8×4 、 4×8 のいずれかで無い場合に行列イントラ予測フラグは復号される。

20

【 0 0 2 0 】

別の側面によれば、前記符号化ユニットの前記サイズが 4×4 、 8×4 、 4×8 、 8×8 のいずれかで無い場合に行列イントラ予測フラグは復号される。

【 0 0 2 1 】

別の側面によれば、前記符号化ユニットの前記サイズが 4×4 、 8×4 、 4×8 、 8×8 、 8×16 のいずれかで無い場合に前記行列イントラ予測フラグは復号される。

【 0 0 2 2 】

本開示の別の側面によれば、ビデオビットストリームから画像フレームのための符号木の予測ブロックを生成する方法を提供し、前記方法は、前記ビットストリームから行列イントラ予測モードフラグを復号することにより前記符号化ユニットの予測モードを判断する工程と、予測モードは行列イントラ予測モードを含み、行列イントラ予測モードを判断するために短縮された二値符号語を復号する工程と、前記予測ブロックに隣接する参照サンプル及び前記復号された行列イントラ予測モードに応じて選択された行列に対して、行列乗算を行うことにより、予測ブロックを生成する工程と、を含む。

30

【 0 0 2 3 】

本開示の別の側面によれば、ビデオビットストリームから画像フレームのための符号木の符号化ユニットを復号する方法の方法を実施するためのコンピュータプログラムを格納した非一時的なコンピュータ可読記憶媒体を提供し、前記方法は、前記符号木の領域を、各符号化ブロックが予測ブロックを含む、複数の符号化ブロックに分割する工程と、前記各符号化ブロックの前記予測ブロックについて、行列イントラ予測フラグを判断する工程であって、各行列イントラ予測フラグは、前記符号化ブロックのうちの 1 つの前記予測ブロックに行列イントラ予測が用いられたかを示し、前記判断を (i) 前記領域が閾値を満たす場合には面積、または (i i) 前記領域の面積が前記閾値を満たさない場合には前記領域の割当量に基づいて行い、前記判断されたフラグに応じて行列イントラ予測を用いると判断された各予測ブロックについて、メモリから行列係数を読み出す工程と、各予測ブロックの参照サンプル及び前記行列係数を用いて前記領域における各符号化ユニットについて生成された予測ブロックを用いて、符号化ユニットを復号する工程と、を含む。

40

【 0 0 2 4 】

本開示の別の側面によれば、ビデオデコーダを提供し、ビデオビットストリームから画像フレームのための符号木の符号化ユニットを受信し、前記符号木の領域を、各符号化ブ

50

ロックが予測ブロックを含む、複数の符号化ブロックに分割し、前記各符号化ブロックの前記予測ブロックについて、行列イントラ予測フラグを判断し、各行列イントラ予測フラグは、前記符号化ブロックうちの1つの前記予測ブロックに行列イントラ予測が用いられたかを示し、前記判断を (i) 前記領域が閾値を満たす場合には面積、または (i i) 前記領域の面積が前記閾値を満たさない場合には前記領域の割当量に基づいて行い、前記判断されたフラグに応じて行列イントラ予測を用いると判断された各予測ブロックについて、メモリから行列係数を読み出し、各予測ブロックの参照サンプル及び前記行列係数を用いて前記領域における各符号化ユニットについて生成された予測ブロックを用いて、符号化ユニットを復号する。

【 0 0 2 5 】

10

本開示の別の側面によれば、システムを提供し、メモリと、プロセッサとを含み、前記プロセッサは、ビデオビットストリームから画像フレームのための符号木の符号化ユニットを復号する方法を実現するための、前記メモリに格納されたコードを実行し、前記方法は、前記符号木の領域を、各符号化ブロックが予測ブロックを含む、複数の符号化ブロックに分割する工程と、前記各符号化ブロックの前記予測ブロックについて、行列イントラ予測フラグを判断する工程であって、各行列イントラ予測フラグは、前記符号化ブロックのうちの1つの前記予測ブロックに行列イントラ予測が用いられたかを示し、前記判断を (i) 前記領域が閾値を満たす場合には面積、または (i i) 前記領域の面積が前記閾値を満たさない場合には前記領域の割当量に基づいて行い、前記判断されたフラグに応じて行列イントラ予測を用いると判断された各予測ブロックについて、メモリから行列係数を読み出す工程と、各予測ブロックの参照サンプル及び前記行列係数を用いて前記領域における各符号化ユニットについて生成された予測ブロックを用いて、符号化ユニットを復号する工程と、を含む。

20

【 0 0 2 6 】

他の側面も開示されている。

【 0 0 2 7 】

ここで、本発明の少なくとも1つの実施形態を、以下の図面及び付録を参照して説明する。

【図面の簡単な説明】

【 0 0 2 8 】

30

【図 1】図 1 は、ビデオ符号化及び復号システムを示す概略ブロック図である。

【図 2 A】、

【図 2 B】図 2 A 及び図 2 B は、図 1 のビデオ符号化及び復号システム的一方または両方が実施される汎用コンピュータシステムの概略ブロック図を示す。

【図 3】図 3 は、ビデオエンコーダの機能モジュールを示す概略ブロック図である。

【図 4】図 4 は、ビデオデコーダの機能モジュールを示す概略ブロック図である。

【図 5】図 5 は、1つのブロックから、Versatile Video Codingのツリー構造における1つまたは複数のブロックへの可能な分割を示す概略ブロック図である。

【図 6】図 6 は、1つのブロックから、Versatile Video Codingのツリー構造における1つまたは複数のブロックへの可能な分割を達成するためのデータフローの概略図である。

40

【図 7 A】、

【図 7 B】図 7 A 及び図 7 B は、符号木の単位 (C T U) をいくつかの符号化ユニット (C U) に分割する例を示している。

【図 8】図 8 は、行列イントラ予測モジュールの処理例を示す図である。

【図 9 A】、

【図 9 B】図 9 A 及び図 9 B は、行列イントラ予測モードのためのメモリアクセス帯域幅が適用される領域を有する例示的な C T U を示す。

【図 1 0】図 1 0 は、図 9 の例に対応する符号木を示す。

【図 1 1】図 1 1 は、行列イントラ予測を使用する符号化ユニットを含む、画像フレームの符号化ユニットをビデオビットストリームに符号化する方法を示す。

50

【図 1 2】図 1 2 は、図 1 1 で使用されるような、符号木の単位の複数の符号化ユニットにまたがる行列イントラ予測のためのメモリアクセスの割当量を決定する方法を示す。

【図 1 3】図 1 3 は、画像フレームの一部を、図 1 1 で使用されるようなビデオビットストリームに符号化するための符号化ユニットを決定する方法を示す。

【図 1 4】図 1 4 は、画像フレームの符号化ユニットを、図 1 1 で使用されるようなビデオビットストリームに符号化する方法を示す。

【図 1 5】図 1 5 は、ビデオビットストリームから、画像フレームの符号木を復号する方法を示す。

【図 1 6】図 1 6 は、図 1 5 で使用されるようなビデオビットストリームから、画像フレームの符号化ユニットを復号する方法を示す。

10

【図 1 7】図 1 7 は、画像フレームの一部を、図 1 1 で使用されるようなビデオビットストリームに符号化するための符号化ユニットを決定する別の方法を示す。

【図 1 8】図 1 8 は、画像フレームの符号化ユニットを、図 1 1 で使用されるようなビデオビットストリームに符号化する別の方法を示す。

【図 1 9】図 1 9 は、図 1 5 で使用されるようなビデオビットストリームから、画像フレームの符号化ユニットを復号する別の方法を示す。

【発明を実施するための形態】

【0029】

最良の形態を含む詳細な説明

添付の図面のうちのいずれか 1 つまたは複数において、同じ参照番号を有するステップ及び/または特徴が参照される場合、それらのステップ及び/または特徴は、相反する意図が示されない限り、この明細書の目的のための同じ機能または動作を有する。

20

【0030】

上記のように、フレーム内予測の計算の複雑さは、特に行列イントラ予測 (MIP) の場合に限定されている。MIP はエラーを最小限に抑えるという点で効果的な解決方法を提供できるが、MIP を適用する計算の複雑さは、例えば 4×4 ブロックのみで構成されるフレームの、最悪の場合のブロック処理速度に特に影響を与える。ブロック処理速度は、目的とするアプリケーションの解像度とフレームレートをサポートするのに十分である必要がある。ルマチャネルのみを考えると、毎秒 120 フレームでの「8K」解像度のフレーム (7680×4320) では、毎秒 248.8 $\times 10^6$ 個の 4×4 ブロックを処理する必要がある。フレームまたはビデオシーケンス全体で最悪の場合にならないとしても、最悪の場合に達した部分領域を、ディスプレイに表示用の完全に復号されたフレームの配信を遅らせることなく処理する必要がある。複雑さ理由の 1 つは、行列イントラ予測 (MIP) モードに従って選択された行列係数を取り出すために必要なメモリ帯域幅に因るもので、これは、ブロック毎に制約無く異なり得る。

30

【0031】

図 1 は、ビデオ符号化及び復号システム 100 の機能モジュールを示す概略ブロック図である。実用化、及び/または、MIP モードによって達成される符号化の利点に見合ったものにするために、システム 100 は、行列係数を選択または読み出すための最悪の場合のメモリ帯域幅を確立するために、MIP モードの適用に関わる制限を利用することができる。

40

【0032】

システム 100 は、ソースデバイス 110 とディスティネーションデバイス 130 を含む。通信チャネル 120 は、符号化されたビデオ情報をソースデバイス 110 からディスティネーションデバイス 130 に通信するために用いられる。いくつかの構成では、ソースデバイス 110 とディスティネーションデバイス 130 のいずれかまたは両方は、携帯電話機または「スマートフォン」であってもよく、その場合、通信チャネル 120 は、無線チャネルである。他の構成では、ソースデバイス 110 及びディスティネーションデバイス 130 は、ビデオ会議機器であってもよく、その場合、通信チャネル 120 は、通常、インターネット接続などの有線チャネルである。さらに、ソースデバイス 110 及びデ

50

ィスティネーションデバイス 130 は、無線テレビ放送をサポートするデバイス、ケーブルテレビアプリケーション、インターネットビデオアプリケーション（ストリーミングを含む）、及び、符号化されたビデオデータがファイルサーバーのハードディスクドライブなど、コンピュータで読み取り可能なストレージメディアにキャプチャするアプリケーションを含む、広範囲のデバイスのいずれかであってもよい。

【0033】

図 1 に示すように、ソースデバイス 110 は、ビデオソース 112 と、ビデオエンコーダ 114 と、送信機 116 とを含む。ビデオソース 112 は、典型的には、画像キャプチャセンサ、非一時的な記録媒体に保存された過去にキャプチャされたビデオシーケンス、またはリモート画像キャプチャセンサからの映像といった、キャプチャされたビデオフレームデータ（113 として示される）のソースを含む。ビデオソース 112 はまた、例えば、オペレーティングシステムのビデオ出力や、例えばタブレットコンピュータ等のコンピューティングデバイス上で実行される様々なアプリケーションを表示する、コンピュータグラフィックスカードの出力であってもよい。画像キャプチャセンサをビデオソース 112 として含み得るソースデバイス 110 としては、スマートフォン、ビデオカムコーダー、プロ用ビデオカメラ、及びネットワークビデオカメラが含まれる。

【0034】

図 3 を参照してさらに説明するように、ビデオエンコーダ 114 は、ビデオソース 112 からのキャプチャされたフレームデータ（矢印 113 で示す）を、ビットストリーム（矢印 115 で示す）に変換（または「符号化」）する。ビットストリーム 115 は、符号化されたビデオデータ（または「符号化されたビデオ情報」）として、送信機 116 により通信チャネル 120 を介して送信される。ビットストリーム 115 を、後で通信チャネル 120 を介して送信されるまで、または通信チャネル 120 を介した送信の代わりに、「フラッシュ」メモリまたはハードディスクドライブなどの非一時的な記憶装置 122 に記憶することも可能である。例えば、符号化されたビデオデータは、ビデオストリーミングアプリケーションのために広域ネットワーク（WAN）を介して顧客にオンデマンドで提供することができる。

【0035】

ディスティネーションデバイス 130 は、受信機 132、ビデオデコーダ 134、及び表示装置 136 を含む。受信機 132 は、符号化されたビデオデータを通信チャネル 120 から受信し、受信したビデオデータをビットストリームとしてビデオデコーダ 134 に渡す（矢印 133 で示す）。次に、ビデオデコーダ 134 は、復号したフレームデータ（矢印 135 で示す）を表示装置 136 に出力する。復号されたフレームデータ 135 は、フレームデータ 113 と同じクロマフォーマットを有する。表示装置 136 としては、例えば、陰極線管や、スマートフォン、タブレットコンピュータ、コンピュータモニター、スタンドアロンテレビ機器等の液晶ディスプレイ等を含む。また、ソースデバイス 110 及びディスティネーションデバイス 130 それぞれの機能を、例えば携帯電話やタブレットコンピュータ等の、単一機器で具現化することも可能である。

【0036】

上記例に挙げたデバイスに限られず、ソースデバイス 110 及びディスティネーションデバイス 130 はそれぞれ、汎用コンピューティングシステム内で、典型的にはハードウェアとソフトウェアコンポーネントの組み合わせを介して構成され得る。図 2A は、そのようなコンピュータシステム 200 を示しており、コンピュータモジュール 201 と、ビデオソース 112 として構成され得るキーボード 202、マウスポインタデバイス 203、スキャナ 226 及びカメラ 227 と、マイクロフォン 280 等の入力デバイスと、プリンタ 215、表示装置 136 として構成され得るディスプレイデバイス 214 及びスピーカー 217 を含む出力デバイスを含む。外部変調器 - 復調器（モデム）トランシーバデバイス 216 は、コンピュータモジュール 201 が接続 221 を介して通信ネットワーク 220 と双方向の通信をするために使用され得る。通信チャネル 120 を代表する通信ネットワーク 220 は、インターネット、携帯通信ネットワーク、またはプライベート WAN

10

20

30

40

50

などの（WAN）であってもよい。接続221が電話回線である場合、モデム216は、従来の「ダイヤルアップ」モデムであり得る。あるいは、接続221が大容量（例えば、ケーブルまたは光）接続である場合、モデム216は、ブロードバンドモデムであり得る。無線モデムを、通信ネットワーク220への無線接続のために使用してもよい。トランシーバデバイス216は、送信機116及び受信機132の機能を提供してもよく、通信チャンネル120は、接続221として具現化され得る。

【0037】

コンピュータモジュール201は、典型的には、少なくとも1つのプロセッサユニット205及びメモリユニット206を含む。例えば、メモリユニット206は、半導体ランダムアクセスメモリ（RAM）及び半導体読み取り専用メモリ（ROM）を有し得る。コンピュータモジュール201はまた、ビデオディスプレイ214とスピーカー217とマイクロフォン280と結合するオーディオ・ビデオインターフェース207、キーボード202とマウス203とスキャナ226とカメラ227とオプションでジョイスティックまたは他のヒューマンインターフェースデバイス（図示せず）と結合するI/Oインターフェース213、外部モデム216及びプリンタ215向けのインターフェース208とを含む、多くの入力/出力（I/O）インターフェースを含む。オーディオ・ビデオインターフェース207からコンピュータモニタ214への信号は、一般に、コンピュータグラフィックスカードの出力である。いくつかの実装形態では、モデム216は、コンピュータモジュール201内、例えば、インターフェース208内に組み込まれ得る。コンピュータモジュール201はまた、ローカルネットワークインターフェース211を有し、これにより、コンピュータシステム200を接続223を介してローカルエリアネットワーク（LAN）として知られるローカル通信ネットワーク222に繋げることができる。図2Aに示されるように、ローカル通信ネットワーク222はまた、一般的にいわゆる「ファイアウォール」デバイスまたは同様の機能のデバイスを含む接続224を介して、ワイドネットワーク220に繋げることができる。ローカルネットワークインターフェース211は、Ethernet（登録商標）回路カード、Bluetooth（登録商標）無線構成、またはIEEE 802.11無線構成を含んでもよい。しかしながら、他の多くのタイプのインターフェースをインターフェース211で用いてもよい。ローカルネットワークインターフェース211はまた、送信機116及び受信機132の機能を提供してもよく、通信チャンネル120は、ローカル通信ネットワーク222において具体化され得る。

【0038】

I/Oインターフェース208及び213は、シリアル接続及びパラレル接続のいずれかまたは両方を提供することができ、前者は、通常、ユニバーサルシリアルバス（USB）規格に従って実現され、対応するUSBコネクタ（不図示）を有する。記憶装置209は、通常、ハードディスクドライブ（HDD）210を含む。フロッピーディスクドライブや磁気テープドライブ（不図示）などの他の記憶装置も使用することができる。光ディスクドライブ212は、通常、不揮発性のデータソースとして機能するように提供される。光ディスク（CD-ROM、DVD、Blu-ray Disc（登録商標）等）、USB-RAM、ポータブル外付けハードドライブ、フロッピーディスク等のポータブルメモリデバイスを、コンピュータシステム200の適切なデータソースとして使用することができる。典型的には、HDD210、光学ドライブ212、ネットワーク220及び222はいずれも、ビデオソース112として、またはディスプレイ214での再生用に格納される復号されたビデオデータの送信先として動作するように構成され得る。システム100のソースデバイス110及びディスプレイデバイス130は、コンピュータシステム200により実現してもよい。

【0039】

コンピュータモジュール201の構成要素205から213は、典型的には、相互接続されたバス204を介して、当業者に知られているコンピュータシステム200の従来の動作モードとなる方法で通信する。例えば、プロセッサ205は、接続218を用いてシ

10

20

30

40

50

ステムバス 204 に結合される。同様に、メモリ 206 及び光ディスクドライブ 212 は、接続 219 によってシステムバス 204 に結合される。説明された配置を実施できるコンピュータとしては、例えば、IBM-PC とその互換機、Sun SPARC stations、Apple Mac (登録商標) や同様のコンピュータシステムがある。

【0040】

適切または所望された場合に、ビデオエンコーダ 114 及びビデオデコーダ 134、及び以下に説明する方法は、コンピュータシステム 200 を使用して実現することができる。特に、ビデオエンコーダ 114、ビデオデコーダ 134 及び説明する方法は、コンピュータシステム 200 内で実行可能な 1 以上のソフトウェアアプリケーションプログラム 233 として実現することができる。特に、ビデオエンコーダ 114、ビデオデコーダ 134、及び説明された方法の工程は、コンピュータシステム 200 内で実行されるソフトウェア 233 の命令 231 (図 2B を参照) によって実現できる。ソフトウェア命令 231 は、それぞれが 1 以上の特定のタスクを実行するための 1 以上のコードモジュールとして形成され得る。ソフトウェアはまた、第 1 の部分及び対応するコードモジュールが説明された方法を実行し、第 2 の部分及び対応するコードモジュールが第 1 の部分とユーザとの間のユーザインターフェースを管理する、2 つの独立した部分に分割され得る。

10

【0041】

ソフトウェアは、例えば、以下に説明する記憶装置を含む、コンピュータ可読媒体に記憶され得る。ソフトウェアは、コンピュータ可読媒体からコンピュータシステム 200 に読み出され、コンピュータシステム 200 によって実行される。コンピュータ可読媒体に記録されているそのようなソフトウェアまたはコンピュータプログラムを有するコンピュータ可読媒体は、コンピュータプログラム製品である。コンピュータシステム 200 においてコンピュータプログラム製品を使用することにより、ビデオエンコーダ 114、ビデオデコーダ 134、及び説明された方法を実行するために有利な装置に、有益な効果がもたらされる。

20

【0042】

ソフトウェア 233 は、通常、HDD 210 またはメモリ 206 に格納される。ソフトウェアは、コンピュータ可読媒体からコンピュータシステム 200 に読み出され、コンピュータシステム 200 によって実行される。従って、例えば、ソフトウェア 233 は、光ディスクドライブ 212 によって読み取られる光学的に読み取り可能なディスク記憶媒体 (例えば、CD-ROM) 225 上に格納され得る。

30

【0043】

場合によっては、アプリケーションプログラム 233 は、1 つまたは複数の CD-ROM 225 上で符号化された状態でユーザに供給され、対応するドライブ 212 を介して読み取られてもよいし、あるいは、ネットワーク 220 または 222 からユーザによって読み取られてもよい。さらに、ソフトウェアは、他のコンピュータ可読媒体からコンピュータシステム 200 に読み出すこともできる。コンピュータ可読記憶媒体とは、記録された命令及び/またはデータを、実行及び/または処理するためにコンピュータシステム 200 に提供する、非一時的な有形記憶媒体を指す。このような記憶媒体としては、例えば、フロッピーディスク、磁気テープ、CD-ROM、DVD、Blu-ray Disc (登録商標)、ハードディスクドライブ、ROM または集積回路、USB メモリ、磁気光学ディスク、または PCMCIA カード等のようなコンピュータ可読カードが含まれ、そのようなデバイスがコンピュータモジュール 201 の内部または外部のいずれにあってもよい。ソフトウェア、アプリケーションプログラム、命令及び/またはビデオデータまたは符号化されたビデオデータを、コンピュータモジュール 401 へ提供するために用いることができる一時的または無形のコンピュータ可読伝送媒体としては、例えば、無線または赤外線伝送チャネルや、別のコンピュータまたはネットワークデバイスへのネットワーク接続、及び電子メール送信やウェブサイト等に記録された情報を含むインターネットまたはイントラネットを含む。

40

【0044】

50

アプリケーションプログラム 233 の第 2 の部分及び上述した対応するコードモジュールは、ディスプレイ 214 上に描画または他の方法で表される 1 以上のグラフィカルユーザインターフェース (GUI) を実施するために実行され得る。そして、典型的にはキーボード 202 やマウス 203 の操作により、コンピュータシステム 200 及びアプリケーションのユーザは、機能的に適応可能な方法でインターフェースを操作して、GUI に関連するアプリケーションへの制御コマンド及び/または入力を提供することができる。スピーカ 217 を介して出力されたスピーチプロンプトやマイクロフォン 280 を介して入力されたユーザ音声コマンドを利用する音声インターフェース等、機能的に適応可能な他の形態のユーザインターフェースも実施され得る。

【0045】

10

図 2B は、プロセッサ 205 及び「メモリ」234 の詳細な概略ブロック図である。メモリ 234 は、図 2A のコンピュータモジュール 201 がアクセス可能なすべてのメモリモジュール (HDD 209 及び半導体メモリ 206 を含む) の論理的集合体を表す。

【0046】

コンピュータモジュール 201 が最初に電源投入されると、電源投入時自己診断 (POST) プログラム 250 が実行される。POST プログラム 250 は、通常、図 2A の半導体メモリ 206 の ROM 249 に格納されている。ソフトウェアを格納する ROM 249 等のハードウェアデバイスは、ファームウェアと呼ばれることもある。POST プログラム 250 は、コンピュータモジュール 201 内のハードウェアを検査して、適切に機能しているか確認し、プロセッサ 205、メモリ 234 (209、206)、及び、典型的には ROM 249 に記憶されている基本的な入出力システムソフトウェア (BIOS) モジュール 251 が、正しく動作しているかをチェックする。POST プログラム 250 が正常に実行されると、BIOS 251 は、図 2A のハードディスクドライブ 210 を起動する。ハードディスクドライブ 210 の起動により、ハードディスクドライブ 210 上に常駐するブートストラップローダープログラム 252 がプロセッサ 205 を介して実行される。これにより、オペレーティングシステム 253 は RAM メモリ 206 に読み出され、オペレーティングシステム 253 は動作を開始する。オペレーティングシステム 253 は、プロセッサ管理、メモリ管理、デバイス管理、ストレージ管理、ソフトウェアアプリケーションインターフェース、及び汎用ユーザインターフェースを含む、様々な高レベルな機能を実現するための、プロセッサ 205 によって実行可能なシステムレベルのアプリケーションである。

20

30

【0047】

オペレーティングシステム 253 は、コンピュータモジュール 201 で実行されている各処理及びアプリケーションが、別の処理に割り当てられたメモリと衝突することなく実行に十分なメモリを備えるように、メモリ 234 (209、206) を管理する。さらに、図 2A のコンピュータシステム 200 において利用可能なさまざまなタイプのメモリは、各処理を効果的に実行できるように適切に使用する必要がある。従って、集約メモリ 234 は、メモリの特定のセグメントがどのように割り当てられるかを説明することを意図するのではなく (特に明記しない限り)、コンピュータシステム 200 によってアクセス可能なメモリの概念及びどのように用いられるかを示すことを意図している。

40

【0048】

図 2B に示すように、プロセッサ 205 は、制御ユニット 239、算術論理ユニット (ALU) 240、及びキャッシュメモリと呼ばれることもあるローカルまたは内部メモリ 248 を含む多数の機能モジュールを含む。キャッシュメモリ 248 は、通常、レジスタセクション内に多数のストレージレジスタ 244 - 246 を含む。1 以上の内部バス 241 は、これらの機能モジュールを機能的に相互接続する。プロセッサ 205 はまた、典型的には、接続 218 を使用してシステムバス 204 を介して外部デバイスと通信するための 1 以上のインターフェース 242 を有する。メモリ 234 は、接続 219 を使用してバス 204 に結合される。

【0049】

50

アプリケーションプログラム 233 は、条件付きの分岐及びループ命令を含み得る一連の命令 231 を含む。プログラム 233 は、プログラム 233 の実行に使用されるデータ 232 も含み得る。命令 231 及びデータ 232 は、それぞれ、メモリ位置 228、229、230 及び 235、236、237 に格納される。命令 231 とメモリ位置 228 - 230 の相対的なサイズに応じて、ある命令は、メモリ位置 230 に示す命令によって表されるように、単一のメモリ位置に格納され得る。あるいは、メモリ位置 228 及び 229 に示す命令のセグメントによって表されるように、命令を多数の部分に分割し、それぞれ別のメモリ位置に格納してもよい。

【0050】

一般に、プロセッサ 205 には、その中で実行される一連の命令が与えられる。プロセッサ 205 は、次の入力待ち、その入力に対して、プロセッサ 205 は、別の一連の命令を実行する。各入力は、多数のソースのうち 1 以上のソースから提供され、図 2A に示された、1 以上の入力デバイス 202、203 によって生成されたデータ、ネットワーク 220、202 の一方を介して外部ソースから受信したデータ、記憶装置 206、209 の一方から取り出したデータ、または対応するリーダー 212 に挿入された記憶媒体 225 から取り出したデータを含む。一連の命令を実行すると、場合によってはデータが出力されることがある。実行により、データまたは変数をメモリ 234 に格納することもある。

【0051】

ビデオエンコーダ 114、ビデオデコーダ 134、及び説明された方法は、入力変数 254 を使用することができ、これらは、メモリ 234 の対応するメモリ位置 255、256、257 に格納される。ビデオエンコーダ 114、ビデオデコーダ 134、及び説明された方法は、出力変数 261 を生成し、これらは、メモリ 234 の対応するメモリ位置 262、263、264 に格納される。中間変数 258 は、メモリ位置 259、260、266、及び 267 に格納され得る。

【0052】

図 2B のプロセッサ 205 に戻り、レジスタ 244、245、246、算術論理演算装置 (ALU) 240、及び制御ユニット 239 は、プログラム 233 を構成する命令セット内のすべての命令について、「取り出し、復号、実行」サイクルを行うために必要な一連のマイクロオペレーションを共同で行う。各取り出し、復号、実行のサイクルは、以下を含む。

メモリ位置 228、229、230 から命令 231 を取り出しまたは読み出す取り出し処理；

制御ユニット 239 がどの命令が取り出されたかを決定する復号処理；

制御ユニット 239 及び / または ALU 240 が命令を実行する実行処理。

【0053】

その後、次の命令のためのさらなる取り出し、復号、実行のサイクルが実行される。同様に、記憶サイクルを実行してもよく、これにより、制御ユニット 239 は値をメモリ位置 232 に記憶または書き込む。

【0054】

図 11 から図 19 の方法における各ステップまたはサブプロセスは、以下に説明するように、プログラム 233 の 1 以上のセグメントに関連付けられており、通常、プロセッサ 205 内のレジスタセクション 244、245、247、ALU 240、及び制御ユニット 239 が協働して、プログラム 233 の注記されたセグメントについて命令セット内のすべての命令について取り出し、復号、実行のサイクルを実行する。

【0055】

図 3 は、ビデオエンコーダ 114 の機能モジュールを示す概略ブロック図である。図 4 は、ビデオデコーダ 134 の機能モジュールを示す概略ブロック図である。一般に、データは、ブロックを分割した固定サイズのサブブロックや行列などの、サンプルまたは係数のグループ単位で、ビデオエンコーダ 114 及びビデオデコーダ 134 内の機能モジュール間で受け渡しされる。ビデオエンコーダ 114 及びビデオデコーダ 134 は、図 2A 及

び図 2 B に示すように、汎用のコンピュータシステム 2 0 0 を使用して実施することができ、様々な機能モジュールが、システム 2 0 0 内の専用ハードウェアや、ハードディスクドライブ 2 0 5 上に常駐し、プロセッサ 2 0 5 による実行により制御される、ソフトウェアアプリケーションプログラム 2 3 3 の 1 以上のソフトウェアコードモジュール等の、コンピュータシステム 2 0 0 内で実行可能なソフトウェアにより実現され得る。あるいは、ビデオエンコーダ 1 1 4 及びビデオデコーダ 1 3 4 は、専用のハードウェアとコンピュータシステム 2 0 0 で実行可能なソフトウェアとの組み合わせによって実現され得る。あるいは、ビデオエンコーダ 1 1 4、ビデオデコーダ 1 3 4、及び説明された方法は、説明された方法の機能またはサブ機能を実行する 1 以上の集積回路等の専用ハードウェアで実現され得る。このような専用ハードウェアには、グラフィックプロセッシングユニット (GPU)、デジタル信号プロセッサ (DSP)、特定用途向け標準製品 (ASSP)、特定用途向け集積回路 (ASIC)、フィールドプログラマブルゲートアレイ (FPGA)、または 1 以上のマイクロプロセッサと関連メモリが含まれる。特に、ビデオエンコーダ 1 1 4 はモジュール 3 1 0 ~ 3 9 2 を含み、ビデオデコーダ 1 3 4 はモジュール 4 2 0 ~ 4 9 6 を含み、これらはそれぞれ、ソフトウェアアプリケーションプログラム 2 3 3 の 1 以上のソフトウェアコードモジュールとして実現され得る。

【 0 0 5 6 】

図 3 のビデオエンコーダは、Versatile Video Coding (VVC) のビデオコーディングパイプラインの例であるが、他のビデオコーデックを、本明細書に記載の処理段階を実行するために使用しても良い。ビデオエンコーダ 1 1 4 は、各フレームが 1 以上のカラーチャネルを含む一連のフレーム等の、キャプチャされたフレームデータ 1 1 3 を受信する。フレームデータ 1 1 3 は、任意のクロマフォーマット、例えば、4 : 0 : 0、4 : 2 : 0、4 : 2 : 2、または 4 : 4 : 4 クロマフォーマットであってよい。ブロック分割部 3 1 0 は、まず、フレームデータ 1 1 3 を、一般に正方形の形状であって、CTUのための特定のサイズが使用されるように構成されたCTUに分割する。CTUのサイズは、例えば、64 × 64、128 × 128、または 256 × 256 のルマサンプルである。ブロック分割部 3 1 0 はさらに、ルマ符号木及びクロマ符号木に応じて、各CTUを 1 以上のCBに分割する。CBにはさまざまなサイズがあり、正方形と非正方形アスペクト比の両方を含んでいてもよい。ブロック分割部 3 1 0 の動作は、図 1 1 乃至図 1 9 を参照してさらに説明される。ただし、VVC 標準では、CB、CU、PU、及びTUの辺の長さは、常に 2 の累乗である。従って、3 1 2 として表される現CBがブロック分割部 3 1 0 から出力され、CTUのルマ符号木及びクロマ符号木に従って、CTUの 1 以上のブロックにわたって反復することにより進行する。CTUをCBに分割するオプションについては、図 5 及び図 6 を参照して以下にさらに説明する。動作は全体的にCTU単位で記述されるが、ビデオエンコーダ 1 1 4 とビデオデコーダ 1 3 4 は、メモリ消費を削減するために、より小さなサイズの領域について動作することができる。例えば、各CTUは、サイズ 64 × 64 の「仮想パイプラインデータ単位」(VPDU)と呼ばれる、より小さな領域に分割することができる。VPDUは、ハードウェアアーキテクチャでのパイプライン処理により適した小ささのデータを形成し、これにより、CTU全体に処理を行う場合と比較して、メモリフットプリントの削減によりシリコン領域、ひいてはコストが削減される。

【 0 0 5 7 】

フレームデータ 1 1 3 を 1 回分割して生じるCTUは、ラスタースキャン順にスキャンされ、1 以上の「スライス」にグループ化され得る。スライスは「イントラ」(または「I」)スライスかもしれない。イントラスライス(イスライス)は、スライス内のすべてのCUがイントラ予測されることを示す。あるいは、スライスは、単方向予測または双方向予測(それぞれ「P」または「B」スライス)されてもよく、それぞれスライス内の単方向予測及び双方向予測の可能性を更に示す。

【 0 0 5 8 】

各CTUについて、ビデオエンコーダ 1 1 4 は 2 段階で動作する。第 1 段階(「検索」段階と呼ぶ。)において、ブロック分割部 3 1 0 は、符号木の様々な潜在的構成をテスト

10

20

30

40

50

する。符号木の各潜在的構成には、「候補」CBが関連付けられている。第1段階では、様々な候補CBをテストして、比較的低い歪みで比較的高い圧縮効率となるCBを選択する。テストは一般的にラグランジュ最適化を含み、候補CBは、レート（符号化コスト）と歪み（入力フレームデータ113に対するエラー）の重み付けされた組み合わせに基づいて評価される。「最良の」候補CB（評価されたレート/歪みが最も低いCB）が、ビットストリーム115への次の符号化のために選択される。候補CBの評価には、所定のエリアにCBを使用するか、または、様々な分割オプションに従って領域をさらに分割し、結果として得られる小さいエリアをそれぞれCBでさらに符号化する、または、領域をさらに分割する、といったオプションが含まれる。その結果、CBと符号木の両方が共に検索段階で選択される。

10

【0059】

ビデオエンコーダ114は、各CB、例えば、CB312について、矢印320によって示される予測ブロック(PB)を生成する。PB320は、関連するCB312の内容の予測である。減算モジュール322は、324（または「残差」、空間ドメインにある差異を指す）として示されるPB320とCB312との差異を生成する。差異324は、PB320とCB312における対応するサンプル間のブロックサイズの差異である。差異324は、変換され、量子化され、矢印336によって示される変換ブロック(TB)として表される。PB320及び関連するTB336は、通常、例えば評価されたコストまたは歪みに基づいて、多くの可能な候補CBのうちの1つから選択される。

【0060】

20

候補符号化ブロック(CB)は、関連するPB及び結果として生じる残差についてビデオエンコーダ114が利用可能な予測モードの1つから生じるCBである。TB336は、差異324を量子化し変換された表現である。ビデオデコーダ114において予測されたPBと組み合わせられると、ビットストリームにシグナリングを追加することで、TB336は、復号されたCBと元のCB312との間の差異を低減する。

【0061】

従って、各候補符号化ブロック(CB)、すなわち、予測ブロック(PB)と変換ブロック(TB)の組み合わせは、関連する符号化コスト（または「レート」）と、関連する差異（または「歪み」）を有する。レートは通常ビット単位で測定される。CBの歪みは、通常、絶対差の合計(SAD)や二乗差の合計(SSD)など、サンプル値の差として推定される。各候補PBから生じる推定値は、予測モード387を決定するために、差異324を使用してモードセクタ386によって決定され得る。予測モード387は、現CBについて、フレーム内予測、フレーム間予測、または行列イントラ予測(MIP)を使用する決定を示す。予測モード387は、関連する動きベクトルを伴う、イントラ予測（行列イントラ予測、DC、平面、及び角度イントラ予測を含む）またはインター予測を含む可能なモードの中から選択されたモードとして決定される。予測モード387は、通常、関連するレートをラムダ値でスケールした結果と合計された各候補モードから生じる歪みのラグランジュ最適化の結果として得られる歪みメトリックを最小化することによって選択される。行列イントラ予測が使用されている場合、現CBに対して、使用可能ないくつかの行列イントラ予測モードのどれが使用されているかを示すために、行列イントラ予測モード（矢印388で表される）も決定される。ブロック、特に比較的小さいブロックのためのMIPモードの使用及び選択を決定するための検索は、図11乃至図14、そして、別の例では図17及び図18を参照して説明されるように、制約のない検索と比較して、行列係数を取り出すための最悪の場合のメモリ帯域幅を緩和するように制約され得る。各候補予測モード及び対応する残差の符号化に関連する符号化コストの推定は、残差のエントロピー符号化よりも大幅に少ないコストで実行することができる。従って、いくつかの候補モードを評価することで、レート対歪みの点で最適なモードを決定することができる。

30

40

【0062】

レート対歪みの観点からの最適モードの決定は、通常、様々なラグランジュ最適化を使

50

用して行われる。行列イントラ予測モード 388 の選択に際して、特定の行列イントラ予測モードを適用することにより生じる残差データの符号化コストの決定が通常行われる。符号化コストは、「絶対変換差の合計」(SATD)を使用して概算することができ、これにより、アダマール変換などの比較的単純な変換を使用して、推定変換残差コストを得ることができる。比較的単純な変換を使用するいくつかの場合において、単純化された推定方法から生じるコストは、完全な評価から決定される実際のコストに単調に関連している。単調に関連する推定コストを伴う場合では、簡略化された推定方法を使用して、ビデオエンコーダ 114 における複雑さを軽減しつつ、同じ決定(つまり、イントラ予測モード)を行うことができる。推定コストと実際のコストとの関係で起こりうる非単調性を受け入れるために、簡略化された推定方法を使用して、最良の候補のリストを生成することができる。非単調性は、例えば、残差データの符号化に利用可能なさらなるモードの決定から生じる可能性がある。リスト中の最良の候補の数は、任意の数であってよい。最良の候補を使用してより完全な検索を実行することで、各候補の残差データを符号化するための最適なモードの選択肢を確立し、これにより、他のモード決定と共にイントラ予測モードの最終選択が可能となる。

10

【0063】

他のモード決定には、「変換スキップ」として知られる前方変換をスキップする機能が含まれる。変換のスキップは、変換基底関数等の式を用いて符号化コストを削減するための十分な相関関係が無い残差データに適している。コンピューターで生成された比較的単純なグラフィックス等の特定の種類のコンテンツは、同様の性質を示す場合がある。「スキップされた変換」の場合、変換自体が実行されなくても、残差係数は符号化される。

20

【0064】

ラグランジアンまたは同様の最適化処理を使用して、(ブロック分割部 310 による)CTUのCBへの最適な分割の選択、ならびに複数の可能性から最良の予測モードの選択の両方を行うことができる。モード選択モジュール 386 における候補モードのラグランジュ最適化プロセスの適用を通じて、最も少ないコスト量のイントラ予測モードが「最良の」モードとして選択される。最少コストモードは、選択されたイントラ予測モード 388 であり、エン트로ピーエンコーダ 338 によってビットストリーム 115 に符号化される。モードセレクトモジュール 386 の動作によるイントラ予測モード 388 の選択は、ブロック分割部 310 の動作にまで及ぶ。例えば、イントラ予測モード 388 の選択の候補は、所定ブロックに適用可能なモードと、所定ブロックと同じ位置にある複数のより小さなブロックに適用可能な追加モードを含み得る。所定ブロック及び同じ位置にあるより小さなブロックに適用可能なモードを含む場合、候補を選択する処理は、非明示的には、CTUからCBへの最良の階層的分解を決定する処理でもある。

30

【0065】

ビデオエンコーダ 114 の動作の第 2 段階(「符号化」段階と呼ばれる)では、選択されたルマ符号木及び選択されたクロマ符号木、すなわち、選択された各CBの反復処理が、ビデオエンコーダ 114 において行われる。反復処理において、CBは、ここでさらに説明されるように、ビットストリーム 115 に符号化される。

【0066】

40

エン트로ピーエンコーダ 338 は、構文要素の可変長符号化と構文要素の算術符号化の両方をサポートする。算術符号化は、コンテキスト適応型の二値化算術符号化処理を使用してサポートされる。算術的に符号化された構文要素は、1 以上の「ピン」のシーケンスで構成される。ピンは、ビットと同様に、「0」または「1」の値を持つ。ただし、ピンは離散ビットとしてビットストリーム 115 にエンコードされることはない。ピンは、関連する予測(または「可能性が高い」または「最も可能性が高い」)値と、「コンテキスト」として知られる関連する確率とを有する。符号化される実際のピンが予測値と一致する場合、「最も可能性の高いシンボル」(MPS)が符号化される。最も可能性の高いシンボルの符号化は、ビットストリーム 115 で消費されるビットの点においてコストが比較的少なく、1 離散ビット未満のコストも含まれる。符号化される実際のピンが想定され

50

る値と一致しない場合、「最も可能性の低いシンボル」(LPS)が符号化される。最も可能性の低いシンボルの符号化は、消費されるビットの点においてコストが比較的高くなる。ピンの符号化手法により、「0」対「1」の確率が偏っているピンの効率的な符号化が可能になる。2つの可能な値(つまり、「フラグ」)を持つ構文要素の場合、単一のピンで十分である。可能な値が多数ある構文要素の場合、一連のピンが必要となる。

【0067】

シーケンスにおける後方のピンの存在は、シーケンスにおける前方のピンの値に基づいて決定することができる。さらに、各ピンは複数のコンテキストに関連付けられている場合がある。特定のコンテキストの選択は、構文要素内の前方のピン、隣接する構文要素のピン値(つまり、隣接するブロックからのピン値)などに依存する可能性がある。コンテキスト符号化されたピンが符号化される度に、そのピンに対して選択されたコンテキスト(存在する場合)は、新しいピン値を反映するように更新される。このように、バイナリ算術符号化スキームは適応性があると言える。

【0068】

コンテキストが無いピン(「バイパスピン」)もまた、ビデオエンコーダ114によってサポートされる。バイパスピンは、「0」と「1」が等確率分布であることを想定して符号化される。従って、各ピンはビットストリーム115の1ビットを占める。コンテキストがないため、メモリが節約されると共に複雑さが軽減されるため、特定のピンの値の分布が偏っていない場合は、バイパスピンが使用される。コンテキストと適応を採用するエントロピーコーダーの一例は、当技術分野ではCABAC(コンテキスト適応バイナリ算術コーダー)として知られており、このコーダーの多くの変形例がビデオ符号化に採用されてきた。

【0069】

エントロピーエンコーダ338は、予測モード387を符号化し、現CBに使用されている場合、コンテキスト符号化及びバイパス符号化されたピンの組み合わせを使用して、行列イントラ予測モード388を符号化する。サイズ4×4のブロックは、35の可能な行列イントラ予測モードを有し、サイズが8×8を超えないブロック(つまり、4×8、8×4、及び8×8)は、19の可能な行列イントラ予測モードを有する。他のブロックサイズの場合、11の可能な行列イントラ予測モードがある。通常、「最も可能性の高い行列モード」(MPM)のリストは、ビデオエンコーダ114で生成される。最も可能性の高いモードのリストは、通常、3つのモードなど、固定長である。最も可能性の高いモードのリストは、現ブロックに隣接する先のブロックにおけるモードを含み得る。例えば、現ブロックの上または左のブロックをMIPモードで使用する場合、対応するモードが現ブロックのMPMとして存在する。現ブロックの上または左のブロックを角度イントラ予測で使用する場合、現ブロックのMPMリストは、角度イントラ予測モードをMIPモードにマッピングするルックアップテーブルを介して導出されたMIPモードで埋められる。さらに、通常のイントラ予測(DC、平面、または角度)を使用して予測された後続のCUのMPMリストには、MIPモードから候補の通常のイントラ予測モードへのマッピングテーブルとともに、MIPモードを使用して符号化されたCUから派生したイントラ予測モードの候補を含めることができる。コンテキスト符号化されたピンは、イントラ予測モードが最も可能性の高いモードの1つであるかどうかを示すフラグを符号化する。イントラ予測モード388が最も可能性の高いモードの1つである場合、バイパス符号化されたピンを使用して、更にシグナリングが符号化される。符号化されたさらなるシグナリングは、例えば、短縮された1つのピンのストリングを使用して、最も可能性の高いモードのどれが行列イントラ予測モード388に対応するかを示す。それ以外の場合、イントラ予測モード388は「残りのモード」として符号化される。残りのモードとしての符号化では、バイパス符号化されたピンを使用して符号化された、固定長符号等の代替構文を使用して、最も可能性の高いモードリストに存在するもの以外のイントラ予測モードを表す。

【0070】

いくつかの構成では、(MIPモードを使用して符号化された)現CUと、通常のイントラ予測モードを使用して符号化され、任意のMIPモードで符号化された近隣のブロックから派生したイントラ予測モードの候補を含み得る、後続のCUの両方のMPMリストの作成が複雑になることを回避できる。MPMリストの作成を省略すると、MIPモードの2値化は、MIPモードを表すために、短縮された二値符号を使用して実行される。短縮された二値符号では、各MIPモードの符号化コストが比較的均等になるが、MPMリストでは、MPMリストに追加された各MIPの符号化コストが低くなる。MIPモードの選択から得られた統計では、MPMリストに含まれていないMIPモードと比較して、MPMリストに含まれているMIPモードの選択に強い偏りがあるとは示されておらず、これは、MPMリストの生成を省略しても圧縮効率が低下しないことを示している。4×4ブロックの場合、35の可能なMIPモードを、5ビットまたは6ビットの符号を用いて符号化でき、MIPモード0～28では5ビット、MIPモード29～34では6ビットを用いる。4×8、8×4、及び8×8ブロックの場合、19の可能なMIPモードを、4ビットまたは5ビットの符号を用いて符号化でき、MIPモード0～12では4ビット、MIPモード13～18では5ビットを用いる。他のサイズのブロックの場合、11個の可能なMIPモードを、3ビットまたは4ビットの符号を用いて符号化でき、MIPモード0～4では3ビット、MIPモード5～10では4ビットを用いる。

【0071】

マルチプレクサモジュール384は、決定された最良の予測モード387に従って各候補CBのテストされた予測モードから選択して、PB320を出力する。候補予測モードは、ビデオエンコーダ114によってサポートされる考えられる予測モードをすべて含む必要は無い。

【0072】

フレーム間予測では、ブロックの予測は、ビットストリームにおけるフレームを符号化する順序で、現フレームから先行する1～2フレームからのサンプルを使用して生成される。さらに、フレーム間予測では、通常、単一の符号木がルマチャネルとクロマチャネルの両方に用いられる。ビットストリームにおけるフレームを符号化する順序は、フレームがキャプチャまたは表示されるときの順序とは異なる場合がある。1つのフレームが予測に使用される場合、ブロックは「単方向予測」と呼ばれ、1つの関連する動きベクトルを有する。2つのフレームが予測に使用される場合、ブロックは「双方向予測」と呼ばれ、2つの関連するモーションベクトルを有する。Pスライスの場合、各CUはイントラ予測または単方向予測され得る。Bスライスの場合、各CUは、イントラ予測、単方向予測、または双方向予測され得る。フレームは通常、「画像のグループ」構造を用いて符号化され、これにより、フレームの時間的階層が可能になる。フレームの時間的階層により、フレームが、フレームを表示する順序で前後の画像を参照できるようになる。画像は、各フレームを復号するための依存関係を確実に満たすために必要な順序で符号化される。

【0073】

サンプルは、動きベクトル及び参照画像インデックスに従って選択される。動きベクトルと参照画像インデックスはすべてのカラーチャネルに適用されるため、インター予測は、PBではなく、主にPUに対する処理の観点から説明される。各カテゴリ(つまり、フレーム内及びフレーム間予測)において、PUを生成するために異なる技術を適用することができる。例えば、イントラ予測は、先に再構築されたサンプルに隣接する行及び列からの値を、方向と組み合わせて使用することで、所定のフィルタ処理及び生成処理に従ってPUを生成する。あるいは、PUは少数のパラメータを用いて記述してもよい。インター予測方法は、モーションパラメータの数とその精度が異なる場合がある。モーションパラメータは通常、参照フレームのリストからどの参照フレームを使用するか、及び各参照フレームの空間変換を示す、参照フレームインデックスを含むが、より多くのフレーム、特別なフレーム、またはスケーリングや回転のような複雑なアフィンパラメータを含めることもできる。さらに、所定のモーションリファインメント処理を適用して、参照されたサンプルブロックに基づく高密度のモーション推定値を生成してもよい。

【 0 0 7 4 】

P B 3 2 0 に決定して選択し、減算器 3 2 2 で元のサンプルブロックから P B 3 2 0 を減算すると、3 2 4 として表される最も少ない符号化コストを有する残差が得られ、非可逆圧縮にかけられる。非可逆圧縮処理は、変換ステップ、量子化ステップ、及びエントロピー符号化工程で構成される。順方向一次変換モジュール 3 2 6 は、差異 3 2 4 に順方向変換を適用して、差異 3 2 4 を空間ドメインから周波数ドメインに変換し、矢印 3 2 8 により表された一次変換係数を生成する。一次変換係数 3 2 8 は、順方向二次変換モジュール 3 3 0 に渡され、分離不可能な二次変換 (N S S T) 処理を実行することによって、矢印 3 3 2 で表された変換係数を生成する。順方向一次変換は、通常、分離可能であり、各ブロックの行のセットを変換してから、列のセットを変換する。順方向一次変換モジュール 3 2 6 は、通常、タイプ I I の離散コサイン変換 (D C T - 2) を使用するが、例えば、1 6 サンプルを超えないブロック幅では水平方向に、1 6 サンプルを超えないブロック高さでは垂直方向に、タイプ V I I の離散サイン変換 (D S T - 7) 及びタイプ V I I I の離散コサイン変換 (D C T - 8) を使用してもよい。行と列の各セットの変換は、まず、一次元変換をブロックの各行に対して適用して中間結果を生成し、次に中間結果の各列に対して適用して最終結果を生成することによって実行される。モジュール 3 3 0 の順方向二次変換は、一般に、分離不可能な変換であり、イントラ予測された C U の残差にのみ適用されるが、バイパスしても良い。順方向二次変換は、1 6 サンプル (一次変換係数 3 2 8 の左上の 4×4 のサブブロックとして配置) または 6 4 サンプル (左上の 8×8 係数として配置、一次変換係数 3 2 8 の 4 つの 4×4 サブブロックとして配置) のいずれかに

10

20

【 0 0 7 5 】

変換係数 3 3 2 は、量子化モジュール 3 3 4 に渡される。モジュール 3 3 4 において、「量子化パラメータ」に従って量子化が実行され、矢印 3 3 6 によって表される残差係数が生成される。量子化パラメータは、所定の T B に対して一定であるため、T B の残差係数を生成するための均一なスケーリングが得られる。「量子化マトリックス」を適用することによって不均一なスケーリングとしてもよく、その場合、量子化パラメータと、通常、T B と同じサイズを有するスケーリングマトリックスにおける対応するエントリとの組み合わせから、各残差係数に適用されるスケーリングファクターが導出される。スケーリングマトリックスは、T B のサイズよりも小さいサイズにすることができ、T B に適用される場合、最近傍アプローチを使用して、T B のサイズよりも小さいスケーリングマトリックスから各残差係数に対してスケーリング値を提供することができる。残差係数 3 3 6 は、ビットストリーム 1 1 5 での符号化のためにエントロピーエンコーダ 3 3 8 に供給される。通常、T U の少なくとも 1 つの有意な残差係数を持つ各 T B の残差係数がスキャンされ、スキャンパターンに従って並べられた値のリストが生成される。スキャンパターンは通常、T B を 4×4 の「サブブロック」のシーケンスとしてスキャンし、これにより、T B のサイズに応じたサブブロックの配置となる、残差係数が 4×4 セットとなる細かさで通常のスキャン操作を提供する。さらに、予測モード 3 8 7、行列イントラ予測モード (使用されている場合) 3 8 8、及び対応するブロック分割もまた、ビットストリーム 1 1 5 で符号化される。

30

40

【 0 0 7 6 】

上記のように、ビデオエンコーダ 1 1 4 は、ビデオデコーダ 1 3 4 で見られるフレーム表現に対応するフレーム表現へのアクセスが必要がある。従って、残差係数 3 3 6 はまた、逆量子化モジュール 3 4 0 によって逆量子化されて、矢印 3 4 2 によって表される逆変換係数を生成する。逆変換係数 3 4 2 は、逆方向二次変換モジュール 3 4 4 を通過し、矢印 3 4 6 によって表される中間逆変換係数が生成される。中間逆変換係数 3 4 6 は、逆方向一次変換モジュール 3 4 8 に渡されて、矢印 3 5 0 によって表される T U の残差サンプルが生成される。逆方向二次変換モジュール 3 4 4 によって実行される逆変換のタイプは、順方向二次変換モジュール 3 3 0 によって実行される順方向変換のタイプに対応する。逆方向一次変換モジュール 3 4 8 によって実行される逆変換のタイプは、一次変換モジュ

50

ール 3 2 6 で実行される一次変換のタイプに対応する。加算モジュール 3 5 2 は、残差サンプル 3 5 0 及び P U 3 2 0 を加算して、C U の再構成されたサンプル（矢印 3 5 4 で示される）を生成する。

【 0 0 7 7 】

再構築されたサンプル 3 5 4 は、参照サンプルキャッシュ 3 5 6 及びループ内フィルタモジュール 3 6 8 に渡される。一般的に、A S I C 上にスタティック R A M を使用して実装される参照サンプルキャッシュ 3 5 6 は（従って、コストのかかるオフチップメモリアクセスを回避することができる）、フレーム内の後続の C U のフレーム内 P B を生成するための依存関係を満たすために必要な最小限のサンプルストレージを提供する。最小限の依存関係には、通常、C T U の行の下部に沿ったサンプルの「ラインバッファ」が含まれ、C T U の次の行と、C T U の高さによって設定された範囲の列をバッファする列で使用される。参照サンプルキャッシュ 3 5 6 は、参照サンプル（矢印 3 5 8 で表される）を参照サンプルフィルタ 3 6 0 に供給する。サンプルフィルタ 3 6 0 は、平滑化操作を適用して、フィルタ処理された参照サンプル（矢印 3 6 2 で示される）を生成する。フィルタ処理された参照サンプル 3 6 2 は、フレーム内予測モジュール 3 6 4 によって使用され、矢印 3 6 6 によって表されるサンプルのイントラ予測ブロックを生成する。各候補イントラ予測モードについて、フレーム内予測モジュール 3 6 4 は、サンプルのブロック 3 6 6 を生成する。サンプルのブロック 3 6 6 は、D C、平面または角度イントラ予測などの技術を使用してモジュール 3 6 4 によって生成されるが、行列イントラ予測ではない。

【 0 0 7 8 】

モードセクタ 3 8 6 が現 C B に対して行列イントラ予測を選択した場合、行列イントラ予測モード 3 8 8 を使用して、係数メモリ 3 9 2 から行列係数 3 6 3 を選択（読み取り）する。行列係数 3 6 3 は、行列イントラ予測モジュール 3 9 0 に渡される。行列イントラ予測モジュール 3 9 0 は、行列係数 3 6 3 及び参照サンプル 3 5 8 を使用して行列乗算を実行して、行列イントラ予測ブロック 3 9 3 を生成する。マルチプレクサ 3 8 4 は、P B 3 2 0 として、行列イントラ予測ブロック 3 9 3 を出力する。係数メモリ 3 9 2 は、行列係数 3 9 3 を提供するために必要な限られた帯域幅を有する。特に、異なる行列イントラ予測モードを、連続するブロックごとに使用して、最悪の場合のメモリ帯域幅の要件を確立する。モードセクタ 3 8 6 は、図 1 1 乃至図 1 4 の動作に関連して説明するように、係数メモリ 3 9 2 の最悪の場合のメモリ帯域幅を減少させる制約下で、ブロックに対する M I P モードを選択するように動作可能である。係数メモリ 3 9 2 の最悪の場合のメモリ帯域幅を減らすことで、ビデオエンコーダ 1 1 4 の符号化効率を比例して低下させることなく、例えば、メモリのハードウェア領域などの複雑さが低減される。制約のない検索における M I P モード選択の統計値は、通常、モードセクタ 3 8 6 での M I P モード選択に課せられる制約を引き起こさないの、符号化効率の比例的低下無く、複雑さを低減することができる。従って、最悪の場合のメモリ帯域幅は、符号化の実行において相応の損失が生じることなく、低減される。

【 0 0 7 9 】

ループ内フィルタモジュール 3 6 8 は、再構成されたサンプル 3 5 4 に数段階のフィルタ処理を行う。フィルタ処理の段階は、不連続性に起因するアーチファクトを低減するために C U 境界に沿って平滑化を適用する「デブロッキングフィルタ」（D B F）を含む。ループ内フィルタモジュール 3 6 8 に存在する別のフィルタ処理の段階は、「適応ループフィルタ」（A L F）であり、ウィーナベースの適応フィルタを適用して、歪みをさらに低減する。ループ内フィルタモジュール 3 6 8 でさらに利用可能なフィルタ処理の段階は、「サンプル適応オフセット」（S A O）フィルタである。S A O フィルタは、まず、再構成されたサンプルを 1 または複数のカテゴリに分類し、割り当てられたカテゴリに従って、サンプルレベルでオフセットを適用することによって動作する。

【 0 0 8 0 】

矢印 3 7 0 によって表されるフィルタ処理されたサンプルは、ループ内フィルタモジュール 3 6 8 から出力される。フィルタ処理されたサンプル 3 7 0 は、フレームバッファ 3

10

20

30

40

50

72に格納される。フレームバッファ372は、通常、いくつか（例えば、上限16）の画像を格納する容量を有し、メモリ206内に構成される。フレームバッファ372は、必要とされるメモリ消費量が大きいために、通常、オンチップメモリを用いて構成されない。従って、フレームバッファ372へのアクセスは、メモリ帯域幅の点でコストがかかる。フレームバッファ372は、動き推定モジュール376及び動き補償モジュール380に参照フレーム（矢印374で表される）を供給する。

【0081】

動き推定モジュール376は、それぞれが現CBの位置からのデカルト空間オフセットであって、フレームバッファ372内の参照フレームの1つにおけるブロックを参照する、いくつかの「動きベクトル」（378として示す）を推定する。参照サンプルのフィルタ処理されたブロック（382として表される）は、動きベクトルごとに生成される。フィルタ処理された参照サンプル382は、モードセクタ386による潜在的な選択で利用可能な更なる候補モードを形成する。さらに、所定のCUについて、PU320は、1つの参照ブロック（「単方向予測」）を使用して形成されても、または2つの参照ブロック（「双方向予測」）を使用して形成されてもよい。選択された動きベクトルについて、動き補償モジュール380は、動きベクトルのサブピクセル精度に準じたフィルタ処理プロセスに従ってPB320を生成する。従って、動き推定モジュール376（多くの候補動きベクトルで動作する）は、動き補償モジュール380（選択された候補のみで動作する）のフィルタ処理よりも単純化されたフィルタ処理を実行して、計算の複雑さを軽減することができる。ビデオエンコーダ114がCUのインター予測を選択すると、動きベクトル378はビットストリーム115に符号化される。

【0082】

図3のビデオエンコーダ114は、Versatile Video Coding（VVC）を参照して説明されているが、他のビデオコーディング符号化標準または実施形態も、モジュール310～386の処理段階を採用することができる。フレームデータ113（及びビットストリーム115）はまた、メモリ206、ハードディスクドライブ210、CD-ROM、Blu-ray disc（登録商標）または他のコンピュータ可読記憶媒体から読み出す（または書き込む）ことができる。さらに、フレームデータ113（及びビットストリーム115）は、通信ネットワーク220または無線周波数の受信機に接続されたサーバ等の外部ソースから受信（または送信）することができる。

【0083】

ビデオデコーダ134を図4に示す。図4のビデオデコーダ134は、Versatile Video Coding（VVC）ビデオデ復号パイプラインの例であるが、他のビデオコーデックを使用して、本明細書に記載されている処理段階を実行してもよい。図4に示すように、ビットストリーム133がビデオデコーダ134に入力される。ビットストリーム133は、メモリ206、ハードディスクドライブ210、CD-ROM、Blu-ray disc（登録商標）、または他の非一時的なコンピュータ可読記憶媒体から読み取ることができる。あるいは、ビットストリーム133は、通信ネットワーク220または無線周波数の受信機に接続されたサーバなどの外部ソースから受信することができる。ビットストリーム133は、復号されるべきキャプチャされたフレームデータを表す符号化された構文要素を含む。

【0084】

ビットストリーム133は、エントロピーデコーダモジュール420に入力される。エントロピーデコーダモジュール420は、一連の「ピン」を復号することによってビットストリーム133から構文要素を抽出し、構文要素の値をビデオデコーダ134内の他のモジュールに渡す。エントロピーデコーダモジュール420は、算術復号エンジンを使用して、各構文要素を一連の1以上のピンとして復号する。各ピンは、1以上の「コンテキスト」を使用することができ、コンテキストは、ピンの「1」と「0」の値を符号化するために使用される確率レベルを表す。あるピンに対して複数のコンテキストが使用可能である場合、「コンテキストモデリング」または「コンテキスト選択」処理が実行され、ピ

ンを復号するために使用可能なコンテキストの1つが選択される。ピンを復号する処理は、連続するフィードバックループを形成する。フィードバックループにおける処理数は、エントロピーデコーダ420がピン/秒で高いスループットを達成できるようにするために、最小化することが好ましい。コンテキストモデリングは、コンテキストを選択するときにビデオデコーダ134に認識されているビットストリームの他の属性、つまり現在のピンに先行する属性に依存する。例えば、コンテキストは、符号木内の現CUの四分木の深さに基づいて選択され得る。依存関係は、ピンを復号する前によく知られているか、長い連続処理を必要とせずに決定される属性に基づいていることが好ましい。

【0085】

符号木の四分木の深さは、容易に知られているコンテキストモデリングの依存関係の例である。イントラ予測モード、特に行列イントラ予測モードは、決定が比較的困難または計算量が多いコンテキストモデリング及び2値化の依存関係の例である。行列イントラ予測モードは、「最も可能性の高いモード」(MPM)のリストへのインデックス、または「残りのモード」のリストへのインデックスとして符号化され、復号されたコンテキスト符号化フラグに従ってMPMと残りのモードのいずれかを選択する。他のイントラ予測モードは、「最も可能性の高いモード」(MPM)のリストへのインデックス、または「残りのモード」のリストへのインデックスとしてコード化され、復号されたイントラルマ_MPMフラグに従って、MPMと残りのモードのいずれかを選択する。行列イントラ予測モードの符号化にMPMが使用されている場合、範囲が0~2の短縮された単一ピンストリングは、MPMリストからMPMの1つを選択する。残りのモードが使用されている場合、残りの(非MPM)モードのいずれを使用するかを選択するために、固定長の符号語が復号される。使用可能なMPMモードの数は、ブロックサイズに応じて、35、19または11のいずれかになる。従って、長さが3のMPMリストでは、残りのモードの数はそれぞれ32、16または8となる。残りのモードは、それぞれ長さ5、4または3の固定長の符号語により効率的に表すことができる。最も可能性の高いモードと残りのモードの両方を決定するには、かなりの数の処理が必要であり、隣接するブロックのイントラ予測モードへの依存を含む。例えば、隣接するブロックは、現ブロックの上及び左にあるブロックとしても良い。隣接するブロックが角度イントラ予測を使用する場合、テーブルルックアップを実行することで、角度イントラ予測モードを、MPMリストの作成に使用するための行列イントラ予測モードにマッピングすることができる。あるいは、構成は、各場合、すなわち、モードの数が35、19、または11であるかどうかに応じて、短縮された値符号語を使用してMIPモードを符号化することができる。エントロピーデコーダモジュール420は、算術符号化アルゴリズム、例えば、「コンテキスト適応2値算術符号化」(CABAC)を適用して、ビットストリーム133から構文要素を復号する。復号された構文要素は、ビデオデコーダ134内でパラメータを再構築するために使用される。パラメータには、残差係数(矢印424で表される)及びイントラ予測モード(矢印458で表される)等のモード選択情報が含まれる。モード選択情報には、動きベクトルや、各CTUの1以上のCBへの分割等の情報も含まれる。パラメータは、通常、先に復号されたCBからのサンプルデータと共に、PBを生成するために使用される。

【0086】

残差係数424は、逆量子化モジュール428に入力される。逆量子化モジュール428は、残差係数424に対して逆量子化(または「スケーリング」)を実行し、量子化にパラメータに応じて、矢印432に示される再構成された中間変換係数を生成する。再構成された中間変換係数432は、逆方向二次変換モジュール436に渡されて、二次変換が適用されるか、処理なし(バイパス)となる。逆方向二次変換モジュール436は、再構成された変換係数440を生成する。不均一な逆量子化行列の使用がビットストリーム133に示される場合、ビデオデコーダ134は、ビットストリーム133から量子化行列を一連のスケーリング係数として読み取り、行列として整える。逆スケーリングは、量子化行列を量子化パラメータと組み合わせて使用して、再構築された中間変換係数432を生成する。

10

20

30

40

50

【 0 0 8 7 】

再構築された変換係数 4 4 0 は、逆方向一次変換モジュール 4 4 4 に渡される。モジュール 4 4 4 は、係数を周波数ドメインから空間ドメインに戻す。モジュール 4 4 4 による動作の結果、矢印 4 4 8 が示す残留サンプルのブロックが得られる。残留サンプル 4 4 8 のブロックは、対応する C U とサイズが等しい。残差サンプル 4 4 8 は、加算モジュール 4 5 0 に供給される。加算モジュール 4 5 0 で、残差サンプル 4 4 8 は、復号された P B (4 5 2 として表される) と加算されて、矢印 4 5 6 が示す再構成されたサンプルのブロックが生成される。再構成されたサンプル 4 5 6 は、再構成サンプルキャッシュ 4 6 0 及びループ内フィルタリングモジュール 4 8 8 に供給される。ループ内フィルタリングモジュール 4 8 8 は、 4 9 2 として表される再構成されたフレームサンプルのブロックを生成する。フレームサンプル 4 9 2 は、フレームバッファ 4 9 6 に書き込まれる。

10

【 0 0 8 8 】

再構成サンプルキャッシュ 4 6 0 は、ビデオエンコーダ 1 1 4 の再構築サンプルキャッシュ 3 5 6 と同様に動作する。再構成サンプルキャッシュ 4 6 0 は、メモリ 2 0 6 無しで後続の C B をイントラ予測するために必要な再構成されたサンプルのためのストレージを提供する (例えば、代わりにデータ 2 3 2 を使用することによって行う、通常オンチップメモリ)。矢印 4 6 4 で表される参照サンプルは、再構築サンプルキャッシュ 4 6 0 から取得され、参照サンプルフィルタ 4 6 8 に供給されて、矢印 4 7 2 で示されるフィルタ処理された参照サンプルを生成する。フィルタ処理された参照サンプル 4 7 2 は、フレーム内予測モジュール 4 7 6 に供給される。モジュール 4 7 6 は、ビットストリーム 1 3 3 で信号を送られ、エントロピーデコーダ 4 2 0 によって復号されたイントラ予測モードパラメータ 4 5 8 に従って、矢印 4 8 0 によって示されたイントラ予測されたサンプルのブロックを生成する。サンプルのブロック 4 8 0 は、行列イントラ予測ではなく、D C、平面または角度イントラ予測などのモードを使用して生成される。

20

【 0 0 8 9 】

C B の予測モードが、ビットストリーム 1 3 3 において (行列イントラ予測以外の) イントラ予測を使用するように示される場合、イントラ予測されたサンプル 4 8 0 は、マルチプレクサモジュール 4 8 4 を介して復号された P B 4 5 2 を形成する。イントラ予測は、サンプルの予測ブロック (P B)、つまり、同じ色成分の「隣接サンプル」を使用して導出された、1つの色成分のブロックを生成する。隣接するサンプルは、現ブロックに隣接するサンプルであり、ブロックの復号順で先行しているため、すでに再構築されている。ルマブロックとクロマブロックが同じ位置にある場合、ルマブロックとクロマブロックは異なるイントラ予測モードを使用する場合がある。ただし、2つのクロマチャネルはそれぞれ同じイントラ予測モードを共有している。イントラ予測は3つのタイプに分類される。「D C イントラ予測」は、隣接するサンプルの平均を表す単一の値で P B を埋める処理を含む。「平面イントラ予測」は、平面に応じたサンプルで P B を埋める処理を含み、隣接するサンプルから、D C オフセットと、垂直及び水平方向の勾配が導出される。「角度イントラ予測」は、フィルタ処理され、P B 全体に特定の方向 (または「角度」) に伝播された隣接サンプルで P B を埋める処理を含む。V V C では、65の角度がサポートされており、長方形のブロックでは、正方形のブロックでは使用できない追加の角度を利用できるため、合計 87 の角度を生成する。4番目のタイプのイントラ予測はクロマ P B で利用することができ、これにより、P B は、「クロスコンポーネント線形モデル」(C C L M) モードに従って、同じ位置にあるルマ再構成サンプルから生成される。3つの異なる C C L M モードが利用可能であり、それぞれが隣接するルマサンプル及びクロマサンプルから派生した異なるモデルを使用する。そして、導出されたモデルを使用して、同じ位置にあるルマサンプルからクロマ P B のサンプルのブロックを生成する。

30

40

【 0 0 9 0 】

ビットストリーム 1 3 3 において、C B の予測モードが行列イントラ予測であることが示されると、行列イントラ予測モード 4 5 8 が復号され、係数メモリ 4 8 6 及び行列イントラ予測モジュール 4 8 2 に供給される。行列係数 4 8 1 は、選択された行列イントラ予

50

測モードの係数メモリ 4 8 6 から読み出されて、行列イントラ予測モジュール 4 8 2 に送られる。行列係数の選択は、係数メモリ 4 8 6 からのメモリ読み出し処理を含み、図 8 及び図 9 を参照して説明したように、あるブロックサイズに対して M I P モードを選択する頻度により、メモリアクセスの最悪の場合のメモリ帯域幅制限が確立される。

【 0 0 9 1 】

ビットストリーム 1 3 3 において、C B の予測モードがインター予測であることが示される場合、動き補償モジュール 4 3 4 は、動きベクトルと、フレームバッファ 4 9 6 からサンプルのブロック 4 9 8 を選択してフィルタ処理するために参照フレームインデックスとを使用して、4 3 8 で表されたインター予測サンプルのブロックを生成する。サンプルのブロック 4 9 8 は、フレームバッファ 4 9 6 に格納された過去に復号されたフレームから取得される。双方向予測の場合、2 つのサンプルのブロックが生成され、混合されて、復号された P B 4 5 2 のためのサンプルが生成される。フレームバッファ 4 9 6 は、ループ内フィルタリングモジュール 4 8 8 からのフィルタ処理されたブロックデータ 4 9 2 で埋められる。ビデオエンコーダ 1 1 4 のループ内フィルタリングモジュール 3 6 8 のように、ループ内フィルタリングモジュール 4 8 8 は、D B F、A L F 及び S A O フィルタ処理のいずれかを行う。ルマチャネル及びクロマチャネルのサブサンプル補間のフィルタ処理は異なるが、一般に、動きベクトルはルマチャネルとクロマチャネルの両方に適用される。

10

【 0 0 9 2 】

図 5 は、Versatile Video Coding の木構造における 1 以上のサブ領域に領域を分割または分岐して得ることのできる分割例 5 0 0 を示す概略ブロック図である。分割例 5 0 0 に示される分割は、エンコーダ 1 1 4 のブロック分割部 3 1 0 によって行うことができ、図 3 を参照して説明したように、ラグランジュ最適化によって決定されるように、符号木に従って、各 C T U を 1 以上の C U または C B に分割する。

20

【 0 0 9 3 】

分割例 5 0 0 は、正方形領域が、他の、おそらくは非正方形のサブ領域に分割された場合を示しているが、図 5 0 0 は分割の可能性を示すものであって、包含する領域が正方形である必要はないことを理解されたい。包含する領域が非正方形の場合、分割の結果として生じるブロックの寸法は、包含するブロックのアスペクト比に従ってスケールされる。領域がそれ以上分割されなくなると、つまり符号木の葉ノードでは、C U はその領域を占有する。ブロック分割部 3 1 0 による C T U の 1 以上の C U への特定の細分化は、C T U の「符号木」と呼ばれる。

30

【 0 0 9 4 】

領域をサブ領域に細分化する処理は、結果として生じるサブ領域が最小 C U サイズに達したときに終了しなければならない。所定の最小サイズ、例えば 1 6 個のサンプルよりも小さいブロック領域を禁止するように C U を制限することに加えて、C U は最小の幅または高さとして 4 を持つように制限される。幅及び高さの観点から、または幅または高さの観点から、他の最小値にすることも可能である。細分化の処理は、最も深いレベルへの分解前に終了することもあり、その結果、C U が最小 C U サイズよりも大きくなることもある。分割が発生せず、単一の C U が C T U 全体を占める可能性もある。C T U 全体を占める単一の C U は、利用可能な最大の符号化ユニットのサイズである。4 : 2 : 0 などのサブサンプリングされたクロマフォーマットを使用しているため、ビデオエンコーダ 1 1 4 とビデオデコーダ 1 3 4 の構成により、ルマチャネルよりも早くクロマチャネルの領域の分割が終了する場合がある。

40

【 0 0 9 5 】

符号木の葉ノードには、更に細分化することができない C U が存在する。例えば、葉ノード 5 1 0 は、1 つの C U を含む。符号木の非葉ノードでは、更に 2 つ以上のノードに分割し、各ノードは、1 つの C U を形成する葉ノード、またはより小さな領域へのさらなる分割を含む非葉ノードである可能性がある。符号木の各葉ノードには、カラーチャネル毎に 1 つの符号化ブロックが存在する。ルマとクロマの両方で同じ深さで分割が終了すると

50

、同じ位置にある3つのCBが生成される。ルマがクロマよりも深い深さで分割が終了すると、複数のルマCBがクロマチャンネルのCBと同じ位置になる。

【0096】

四分木分割512は、図5に示すように、包含する領域を4つの等しいサイズの領域に分割する。HEVCと比較して、Versatile Video Coding (VVC) は、水平2値分割514と垂直2値分割516を含む更なる柔軟性を達成する。分割514及び516はそれぞれ、包含する領域を2つの等しいサイズの領域に分割する。分割は、包含するブロック内において水平境界(514)または垂直境界(516)のいずれかに沿っている。

【0097】

水平三分割518及び垂直三分割520を追加することにより、Versatile Video Codingにおいてさらなる柔軟性が達成される。三分割518及び520は、包含する領域の幅または高さの1/4及び3/4に沿って、ブロックを水平方向(518)または垂直方向(520)のいずれかに3つの領域に分割する。四分木、二分木、及び三分木の組み合わせは、「QTBT」呼ばれる。木の根は、0個以上の四分木分割(木の「QT」セクション)を含む。QTセクションが終了すると、0個以上の二分割または三分割が発生し(「多分木」または木の「MT」セクション)、最終的に木構造の葉ノードにおけるCBまたはCUで終了する。木がすべてのカラーチャンネルを表す場合、木の葉ノードはCUとなる。木がルマチャンネルまたはクロマチャンネルを表す場合、木の葉ノードはCBとなる。

【0098】

四分木のみをサポートし、結果、平方ブロックのみをサポートするHEVCと比較して、QTBTは、特に二分木及び/または三分木分割の可能な再帰的適用を考慮することで、より多くのCUサイズが可能となる。分割の選択肢を制限することによって、ブロックサイズが異常(正方形でない)となる可能性を低減することができ、ブロック幅または高さが4未満のサンプル、または、4の倍数とならないサンプルへの分割を排除することができる。一般的に、ルマサンプルにを考慮することで制限が適用される。しかしながら、記載された構成では、制限はクロマチャンネルのブロックに別に適用することができる。クロマチャンネルに対して分割の選択肢の制限を適用すること、例えば、フレームデータが4:2:0のクロマフォーマットまたは4:2:2のクロマフォーマットの場合、ルマ対クロマで最小ブロックサイズが異なるものとなる可能性がある。各分割は、包含する領域に対して、辺の寸法が、変化無し、半分または1/4のサブ領域を生成する。そして、CTUサイズが2のべき乗倍であるため、全てのCUの辺の寸法も2のべき乗倍となる。

【0099】

図6は、Versatile Video Codingで使用されるQTBT(または「符号木」)構造のデータフロー600を示す概略フロー図である。QTBT構造は各CTUに使用され、CTUを分割して1以上のCUとすることを定義する。各CTUのQTBT構造は、ビデオエンコーダ114のブロック分割部310によって決定され、ビットストリーム115に符号化されるか、またはビデオデコーダ134のエントロピーデコーダ420によってビットストリーム133から復号される。データフロー600は、図5に示される分割状態に従って、CTUを1以上のCUに分割するためにブロック分割部310が利用可能な、許容することのできる組み合わせをさらに特徴付ける。

【0100】

階層の最上位レベル、すなわちCTUから始めて、0以上の四分木分割が最初に実行される。具体的には、四分木(QT)分割判定610は、ブロック分割部310によって行われる。「1」シンボルを返す610での決定は、四分木分割512に従って、現ノードを4つのサブノードに分割する決定を示す。その結果、620におけるように、4つの新しいノードが生成され、そして、新しいノードそれぞれについて、QT分割決定610への再帰が行われる。新しいノードはそれぞれ、ラスト(またはZスキャン)順で考慮される。あるいは、QT分割判定610が、それ以上の分割が実行されない(「0」シンボルを返す)ことを示す場合、四分木分割は停止し、続いて多分木(MT)分割が考慮される。

【 0 1 0 1 】

まず、MT分割判定 6 1 2 が、ブロック分割部 3 1 0 によって行われる。6 1 2 では、MT分割を実行する判定が示されている。判定 6 1 2 で「0」シンボルを返すことは、ノードをサブノードにこれ以上分割しないことを示す。ノードの更なる分割が実行されない場合、そのノードは符号木の葉ノードであり、CUに対応する。葉ノードは 6 2 2 で出力される。あるいは、MT分割 6 1 2 で、MT分割を実行する決定を示す（「1」シンボルを返す）場合、ブロック分割部 3 1 0 は方向判定 6 1 4 に進む。

【 0 1 0 2 】

方向判定 6 1 4 では、MT分割の方向を水平（「H」または「0」）または垂直（「V」または「1」）として示す。判定 6 1 4 で水平方向を示す「0」が返された場合、ブロック分割部 3 1 0 は判定 6 1 6 に進む。判定 6 1 4 で垂直方向を示す「1」が返された場合、ブロック分割部 3 1 0 は判定 6 1 8 に進む。

【 0 1 0 3 】

判定 6 1 6 及び 6 1 8 のそれぞれにおいて、MT分割のための分割数は、BT/TT分割における 2（二分割または「BT」ノード）または 3（三分割または「TT」）のいずれかとして示される。すなわち、6 1 4 から示された方向が水平の場合、BT/TT分割判定 6 1 6 はブロック分割部 3 1 0 によって行われ、6 1 4 から示された方向が垂直の場合、BT/TT分割判定 6 1 8 がブロック分割部 3 1 0 によって行われる。

【 0 1 0 4 】

BT/TT分割判定 6 1 6 は、水平分割が、「0」を返すことによって示される二分割 5 1 4 であるか、または「1」を返すことによって示される三分割 5 1 8 であるかを示す。BT/TT分割判定 6 1 6 が二分割を示す場合、HBT CTUノード生成ステップ 6 2 5 において、水平二分割 5 1 4 に従って、ブロック分割部 3 1 0 により 2 つのノードが生成される。BT/TT分割 6 1 6 が三分割を示す場合、HTT CTUノード生成ステップ 6 2 6 において、水平三分割 5 1 8 に従って、ブロック分割部 3 1 0 により 3 つのノードが生成される。

【 0 1 0 5 】

BT/TT分割判定 6 1 8 は、垂直分割が「0」を返すことによって示される二分割 5 1 6 であるか、または「1」を返すことによって示される三分割 5 2 0 であるかを示す。BT/TT分割 6 1 8 が二分割を示す場合、VBT CTUノード生成ステップ 6 2 7 において、垂直二分割 5 1 6 に従って、ブロック分割部 3 1 0 により 2 つのノードが生成される。BT/TT分割 6 1 8 が三分割を示す場合、VTT CTUノード生成ステップ 6 2 8 において、垂直三分割 5 2 0 に従って、ブロック分割部 3 1 0 により 3 つのノードが生成される。ステップ 6 2 5 ~ 6 2 8 で生じる各ノードについて、方向 6 1 4 に従って、左から右、上から下への順序で、MT分割決定 6 1 2 に戻るデータフロー 6 0 0 が繰り返し行われる。結果として、二分木分割及び三分木分割を適用して、様々なサイズのCUを生成することができる。

【 0 1 0 6 】

図 7 A 及び図 7 B は、CTU 7 1 0 から多数のCUまたはCBへの分割例 7 0 0 を示す。CU 7 1 2 の例を図 7 A に示す。図 7 A は、CTU 7 1 0 内のCUの空間配置を示す。また、分割例 7 0 0 を、図 7 B において符号木 7 2 0 として示す。

【 0 1 0 7 】

図 7 A の CTU 7 1 0 の各非葉ノード、例えば、ノード 7 1 4、7 1 6、7 1 8 では、含まれているノード（さらに分割されるかまたはCU）を、符号木 7 2 0 の列に示すようなノードのリストを生成するために、「Z順」にスキャンまたは精査する。四分木分割では、Z順スキャンの結果、左上から右の後、左下から右となる。水平方向の分割と垂直方向の分割の場合、Z順スキャン（精査）は、それぞれ、上から下へのスキャン及び左から右へのスキャンに単純化される。図 7 B の符号木 7 2 0 は、適用されたスキャン順序に従って全てのノード及びCUをリスト化する。各分割により、葉ノード（CU）に達するまで、次の木のレベルで、2 つ、3 つ、または 4 つの新しいノードのリストが生成される。

【 0 1 0 8 】

ブロック分割部 3 1 0 によって、画像を分解し、最終的に C U まで分割し、図 3 を参照して説明したように、C U を用いて各残差ブロック (3 2 4) を生成すると、残差ブロックはビデオエンコーダ 1 1 4 による順方向変換及び量子化の対象となる。結果として生じる T B 3 3 6 は、順番にスキャンされて、エントロピー符号化モジュール 3 3 8 の動作の一部として、残差係数の順次リストが形成される。ビットストリーム 1 3 3 から T B を得るために、同様の処理が、ビデオエンコーダ 1 1 4 によって行われる。

【 0 1 0 9 】

C P U を C U に分割した結果、小さいサイズの C U (例えば、 4×4 、 4×8 、または 8×4) はフレーム内で空間的に隣接しているように見える。さらに、符号木の階層的 Z 順スキャンの利点として、小さいサイズの C U は時間的に続けて処理される。特に、 8×8 領域の四分木分割により 4 つの一連の 4×4 C U が得られ、 8×4 または 4×8 領域の二分割により、一対の 4×4 C U が得られ、 8×8 領域の二分割により、一対の 4×8 領域または一対の 8×4 領域が得られる。得られた各領域 (4×8 または 8×4 のサイズ) は、C U を形成するか、または 4×4 C U に更に分割することができる。64 のサンプルの領域に三分割を適用すると、サイズ 16、32、及び 16、すなわち、 4×4 、 4×8 または 8×4 、及び 4×4 の 3 つの領域が生成される。128 のサンプルの領域に三分割を適用すると、サイズ 32、64、32、例えば、 4×8 、 8×8 、 4×8 の 3 つの領域が生成される。小さいブロック (例えば 4×4 、 4×8 、 8×4) は、64 または 128 のようなサイズの領域を分割した結果であるので、空間的にも、C T U の Z 順スキャンにおいても、一緒に見られる。

【 0 1 1 0 】

図 8 は、ビデオエンコーダ 1 1 4 及びビデオデコーダ 1 3 4 それぞれの行列イントラ予測モジュール 3 9 0 及び 4 8 2 の動作を、 8×8 ブロックを使用して、詳細に示すデータフロー 8 0 0 である。行列イントラ予測は、図 8 の例ではルマチャネルでのみ実行され、4 : 2 : 0 や 4 : 2 : 2 などのクロマフォーマットを使用することにより生じる異なるブロックサイズを考慮する必要はない。モジュール 3 9 0 及び 4 8 2 は、それぞれ 3 9 3 または 4 8 3 のような行列予測サンプルのブロックを出力する。モードセレクタ 3 8 6 が C U に対して行列イントラ予測を選択したときに、ブロック 3 9 3 は、ビデオエンコーダ 1 1 4 において P B 3 2 0 として用いられる。ビットストリーム 1 3 3 からの C U の予測モードを復号することによって、行列イントラ予測の使用が示された場合、ブロック 4 8 3 は、P B 4 8 3 として用いられる。

【 0 1 1 1 】

モジュール 3 9 0 及び 4 8 2 の動作は、3 つのステップを含む：1 . 平均化、2 . 行列乗算及びオフセット (バイアス) 加算、3 . 双線形補間である。双線形補間は、C U のサイズが 4×4 より大きい場合にのみ行われる。平均化ステップは以下のように動作する。参照サンプル 8 0 2 (例えば 3 5 8 または 4 6 4) がデータフロー 8 0 0 で受信され、上サンプル 8 2 2 及び左サンプル 8 2 0 として割り当てられる。ルマ C B の幅及び高さが 4 より大きい場合、上サンプル 8 2 2 は 4 つのセットに分割される (図 8 の 8×8 ブロックの例における 4 対)。4 つのセットそれぞれのコンテンツの値は平均されて、4 つのフィルタ処理された上サンプル 8 2 6 が生成される。同様に、左サンプル 8 2 0 は 4 つのセット (図 8 の 8×8 ブロックの例における 4 対) に分割される。4 つのセットそれぞれのコンテンツの値は平均されて、4 つのフィルタ処理された左サンプル 8 2 4 が生成される。従って、合計 8 つのフィルタ処理されたサンプルが行列乗算モジュール 8 2 8 への入力となる。

【 0 1 1 2 】

ブロックの幅及び高さが 4 に等しい場合 (図 8 に示されるように 8 ではなく)、4 つの上サンプル 8 2 2 は 2 対に分割され、それぞれが平均化されてフィルタ処理された 2 つの上サンプル 8 2 6 が生成される。同様に、4 つの左サンプル 8 2 0 は 2 対に分割され、それぞれが平均化されてフィルタ処理された 2 つの左サンプル 8 2 4 が生成され、合計 4 つ

のフィルタ処理されたサンプルが行列乗算モジュール 8 2 8 への入力となる。

【 0 1 1 3 】

第 2 のステップ（行列乗算）において、受信された行列係数（3 6 3 または 4 8 1）が行列イントラ予測モード（3 8 8 または 4 5 8）に従って選択され、行列乗算モジュール 8 2 8 にも入力される。また、行列乗算モジュールには、オフセット値またはバイアス値のセットも入力される。オフセットまたはバイアス値は、行列乗算の結果に加算されることで、任意の DC シフトが含まれるようになる。行列係数とバイアス値は予め決定されている。言い換えると、行列係数とバイアス値は「オフライン」トレーニングプロセスの結果であり、ビデオエンコーダ 1 1 4 とビデオデコーダ 1 3 4 にとって定数値と見なされる。

【 0 1 1 4 】

4 × 4 C B の場合、1 8 セットの行列係数とバイアス値（セット A）を備えた 3 5 の M I P モードが利用可能である。4 × 8、8 × 4、8 × 8 C B の場合、1 0 セットの行列係数とバイアス値（セット B）を備えた 1 9 の M I P モードが利用可能である。他の C B サイズの場合、6 セットの行列係数とバイアス値（セット C）を備えた 1 1 の M I P モードが利用可能である。与えられた行列係数とバイアス値のセットは、2 つの M I P モードに使用され得る。一方の M I P モードは提供された値を使用し、他方の M I P モードは提供された値の転置行列を使用する。さらに、ある場合には、行列係数とバイアス値のセットが 1 つの M I P モード専用となる。3 つのケースはそれぞれ、セット A ~ C のそれぞれに適用される。セット A ~ C のサイズは以下に示す通りであり、また、セット A ~ C のそれぞれに与えられた 1 つの P B の生成（つまり、1 つの C U に対する M I P モードの適用）に使用するための、3 6 2 または 4 8 1 によって選択されたワード数も示す。

セット A：サイズ 1 6 × 4 の 1 8 個の行列、サイズ 1 6 の 1 8 個のオフセットベクトル。C B のために選択された値のサイズは、 $1 6 \times 4 + 1 6 = 8 0$ ワード。

セット B：サイズ 1 6 × 8 の 1 0 個の行列、サイズ 1 6 の 1 0 個のオフセットベクトル。C B のために選択された値のサイズは、 $1 6 \times 8 + 1 6 = 1 4 4$ ワード。

セット C：サイズ 6 4 × 8 の 6 個の行列、サイズ 6 4 の 1 0 個のオフセットベクトル。C B のために選択された値のサイズは、 $6 4 \times 8 + 6 4 = 5 7 6$ ワード。

【 0 1 1 5 】

ビデオエンコーダ 1 1 4 及びビデオデコーダ 1 3 4 は、フレームサイズ及びフレームレートによって決まるピクセルレートでビデオデータを処理する。さらに、ルマ C B のサイズは、幅と高さが 4 の倍数である。従って、セット A ~ C ためのメモリ帯域幅要件は、4 × 4 のルマサンプルへのアクセスにより表現することができる。各 C U に対する M I P モードの使用に制約がない場合の、各セットに対する、最悪の場合のセット A ~ C それぞれ、つまり最小ブロックサイズの各セットへのアクセス密度は次の通りである。

セット A：4 × 4 C B は、4 × 4 サンプル領域あたり 8 0 ワード必要。

セット B：4 × 8 及び 8 × 4 C B は、4 × 4 サンプル領域あたり $1 4 4 \div 2 = 7 2$ ワード必要、8 × 8 C B は、4 × 4 サンプル領域あたり $1 4 4 \div 4 = 3 6$ ワード必要。

セット C：8 × 1 6 及び 1 6 × 8 C B は、4 × 4 サンプル領域あたり $5 7 6 \div 8 = 7 2$ ワード必要、8 × 3 2、1 6 × 1 6、3 2 × 8 C B は、4 × 4 サンプル領域あたり $5 7 6 \div 1 6 = 3 6$ ワード必要、更に大きいサイズの C B は、4 × 4 サンプル領域あたりより少ないワードが必要。

【 0 1 1 6 】

上記のアクセス密度によって示されるように、M I P モードの使用に制約がない場合、最悪の場合、すべての C B が M I P モードを使用する可能性があり、符号木は、各 C T U を、最悪の場合の係数メモリ（4 8 6、3 9 2）帯域幅を示すために上記で使用された小さな C B サイズに分解する可能性がある。行列係数とバイアス値の名目上のワードサイズは 1 6 ビットであるが、より少ないビット、例えば 1 0 ビットで十分な場合がある。グループ単位でのワードの取り出しは、実施可能な選択肢の 1 つである。しかしながら、メモリ帯域幅の負担はやや高いまま残る。

【 0 1 1 7 】

10

20

30

40

50

M I Pモード選択の統計値によれば、典型的には、(JVET common test conitions文書JVET-N1010で定義されるように)広いテストセットの場合の20%において、上と左のブロックを考慮した場合、隣接して配置されたC Bに対してM I Pモードが選択される。従って、M I Pモードの適用頻度を制限することにより、M I Pモードの利用可能性の観点から、圧縮効率ゲインの相応の低下を引き起こすことなく、係数メモリ486及び392に必要な最悪の場合のメモリ帯域幅を軽減することが可能である。M I Pモードの選択に関する制約については、図11~図16を参照して説明する。

【0118】

行列乗算モジュール828は、行列係数のセット(363または481)及びフィルタ処理された参照サンプル(すなわち、824及び826)を使用して行列乗算を実行する。バイアス値が行列乗算の出力に加算されて、疎ブロック830が形成される。疎ブロック830は、部分的に393または483を埋める、網掛けされたサンプルとして表されている。393または483の残りのサンプルは、上参照サンプル822とフィルタ処理された左参照サンプル824、または左参照サンプル820とフィルタ処理された上サンプル822のいずれかからの寄与により(図8に示されるように)、双線形補間を用いて導出される。

【0119】

図9Aは、C T U 9 0 0の一例を示す。C T U 9 0 0は、M I Pモードのメモリアクセス帯域幅が適用される領域を有する。行列イントラ予測は、V P D Uサイズの領域、つまり128×128のC P U 9 0 0の64×64の象限内で適用される。例えば、割当量が適用される512のルマサンプル領域は、(i)32×16の領域912、(i i)16×32の領域914、及び(i i i)64×8の領域916を含む。領域912は、更に、様々なC U、すなわち、図9Bの912a~912hに分解される。図9Bの例では、C U 9 1 2 aのサイズは16×4であるため、セットCに属し、C U 9 1 2 dのサイズは8×8であるため、セットBに属し、同様に912b、912c及び912e(すべてサイズ8×4)もセットBに属する。C U 9 1 2 f及びC U 9 1 2 gは、それぞれ4×4であり、セットAに属する。図11から図16を参照して説明するように、512のルマサンプルの領域の面積は、M I Pモードに制約を適用するかどうかを決定するための閾値として記載された構成で使うことができる。上記のセットAからCは通常、領域の分割から生じるため、512のサンプルの領域の面積を使用することは制約として適している。しかしながら、閾値は、以下に説明するように、異なる面積に関連し得る。

【0120】

図10は、図9AのC T U 9 0 0の例に対応する符号木10を示している。領域912は、領域914及び916と同様に、含まれるC Uに対する割当量がM I P行列係数の読み取りのために確立されている符号木のノードに対応する。領域における1以上のC Uへの分解は、簡略化のために不図示とする。M I Pモードを使用するように構成された各C Uについて、セットA~Cの1つにマッピングするC Uサイズに応じた割当量は、図11乃至図16を参照して以下に説明するように、領域(すなわち、912、914、916)の割当量から差し引かれる。

【0121】

図11は、画像フレームの符号化ユニットをビデオビットストリーム115に符号化するための方法1100を示している。方法1100は、構成されたF P G A、A S I C、またはA S S P等の装置によって具現化され得る。さらに、方法1100は、プロセッサ205の実行下でビデオエンコーダ114によって実行され得る。従って、方法1100は、コンピュータ可読記憶媒体及び/またはメモリ206に格納され得る。一連の符号化ユニットである画像フレームをビットストリーム115に符号化する方法1100の動作は、行列イントラ予測の使用に制限をかける。従って、方法1100の動作は、制限がない場合の最悪の場合の潜在的なメモリ帯域幅と比較して、行列イントラ予測に必要なメモリ帯域幅に制限をつけることになる。方法1100は、フレームをC T Uに分割するステップ1110から始まる。

10

20

30

40

50

【 0 1 2 2 】

フレームをC T Uに分割するステップ1 1 1 0において、ブロック分割部3 1 0は、プロセッサ2 0 5の実行下で、フレームデータ1 1 3の現フレームをC T Uの行列に分割する。分割の結果として生じるC T Uの符号化処理を開始する。プロセッサにおける制御は、ステップ1 1 1 0から符号木の決定ステップ1 1 2 0に進む。

【 0 1 2 3 】

符号木の決定ステップ1 1 2 0で、プロセッサ2 0 5の実行下で、ビデオエンコーダ1 1 4は、図5から図7を参照して説明したように、様々な分割オプションをテストすると共に、C T Uの符号木に到達するための符号化ユニットの決定ステップ1 1 3 0の動作が行われる。ステップ1 1 2 0の動作は、図1 2を参照して説明する。プロセッサ2 0 5における制御は、ステップ1 1 2 0から、符号化ユニットの予測モードの決定ステップ1 1 3 0に進む。

10

【 0 1 2 4 】

符号化ユニットの決定ステップ1 1 3 0において、プロセッサ2 0 5の実行下で、ビデオエンコーダ1 1 4は、選択された符号化ユニットをビットストリーム1 1 5に符号化する際に使用する予測モードを決定する。符号化ユニットは、スキャンパターンに従って選択することができる。ステップ1 1 3 0の動作は、図1 3を参照してさらに説明する。符号化ユニットの予測モードが一旦選択されると、プロセッサ2 0 5における制御は、ステップ1 1 3 0から符号化ユニットの符号化ステップ1 1 4 0に進む。符号化ユニットのための予測モードの選択に際して、符号化ユニットは、符号化ユニット自体が符号木の分割の階層から生じるものであるため、所定の符号化ユニットに到達するための特定の分割の組み合わせも選択され、これにより、符号木が決定される。

20

【 0 1 2 5 】

符号化ユニットの符号化ステップ1 1 4 0において、エン트로ピーエンコーダ3 3 8は、プロセッサ2 0 5の実行下で、ステップ1 1 3 0で決定された符号化ユニットをビットストリーム1 1 5に符号化する。決定された符号木は、プロセッサ2 0 5の実行下で、図5及び図6に示すように、選択された分割を示すために「分割フラグ」及び他の構文要素を使用して、エン트로ピーエンコーダ3 3 8によってステップ1 1 4 0でビットストリーム1 1 5に効率的に符号化される。

【 0 1 2 6 】

ステップ1 1 4 0での動作は、図1 4を参照してさらに説明する。プロセッサ2 0 5における制御は、ステップ1 1 4 0から最終符号化ユニットテストステップ1 1 5 0に進む。

30

【 0 1 2 7 】

最終符号化ユニットテストステップ1 1 5 0で、プロセッサ2 0 5は、現在の符号化ユニットがステップ1 1 2 0の符号木における最後の符号化ユニットであるかどうかをテストする。現符号化ユニットがステップ1 1 2 0の符号木における最後のものである場合（ステップ1 1 5 0で「Y E S」）、プロセッサ2 0 5における制御は、最終C T Uテストステップ1 1 6 0に進む。現符号化ユニットがステップ1 1 2 0の符号木における最後のものではない場合（ステップ1 1 5 0で「N O」）、ステップ1 1 2 0の符号木における次の符号化ユニットが、決定及び符号化のためのスキャンパターンを使用して選択され、プロセッサ2 0 5における制御は、ステップ1 1 3 0に進む。従って、ステップ1 1 3 0は、ステップ1 1 2 0で決定された符号木から生じる各C Uに対して実行される。

40

【 0 1 2 8 】

最終C T Uテストステップ1 1 6 0で、プロセッサ2 0 5は、現C T Uがスライスまたはフレーム内の最後のC T Uであるかどうかをテストする。そうでない場合（ステップ1 1 6 0で「N O」）、ビデオエンコーダ1 1 4はフレーム内の次のC T Uに進み、プロセッサ2 0 5における制御はステップ1 1 6 0からステップ1 1 2 0に戻り、フレーム内の残りのC T Uの処理を継続する。C T Uがフレームまたはスライスの最後のものである場合、ステップ1 1 6 0は「Y E S」を返し、方法1 1 0 0は終了する。方法1 1 0 0の動作の結果、画像フレーム全体が一連のC T Uとしてビットストリームに符号化される。

50

【 0 1 2 9 】

方法 1 1 0 0 は、ビデオシーケンス内の各画像フレームに対して実行される。方法 1 1 0 0 は、C T U 毎に C U を決定することができる。言い換えると、C T U の C U が、まず 1 つのパスまたはパイプラインステージで決定され、次に、ビットストリームに符号化する第 2 段階が続く。方法 1 1 0 0 はまた、C T U と比較して V P D U の面積は小さいためにメモリ消費が削減されるため、例えば、V P D U 毎に、より細かく C U を決定してもよい。

【 0 1 3 0 】

図 1 2 は、ステップ 1 1 2 0 で実施される C T U の符号木を決定する方法 1 2 0 0 を示す。方法 1 2 0 0 は、C T U を受信し、評価のための候補分割及び候補符号化ユニットを生成し、ビットストリーム 1 1 5 に符号化される分割及び符号化ユニットを最終的に選択する。特に、方法 1 2 0 0 は、ノード下、すなわち空間領域内で、符号化ユニットにおける M I P モードの適用を制限する、符号木内の特定のノードまたは領域における M I P モードメモリ帯域幅の割当量を確立する。方法 1 2 0 0 は、構成された F P G A、A S I C、または A S S P 等の装置によって具現化され得る。さらに、方法 1 2 0 0 は、プロセッサ 2 0 5 の実行下でビデオエンコーダ 1 1 4 によって実行され得る。従って、方法 1 2 0 0 は、コンピュータ可読記憶媒体及び/またはメモリ 2 0 6 に格納され得る。方法 1 2 0 0 は、候補符号木内の各ノード（領域）について実施される。方法 1 2 0 0 は、領域面積テストステップ 1 2 1 0 で開始する。

【 0 1 3 1 】

領域テストステップ 1 2 1 0 において、ビデオエンコーダ 1 1 4 は、プロセッサ 2 0 5 の実行下で、利用可能な分割オプションに従って（図 5 に関連して説明した分割オプションを使用して）、符号木の候補ノードによって占有される面積をテストする。説明した構成では、領域は、分割が停止する前にさらに分割し、C U が形成され得る符号木のノードに関連する。領域は、現符号化ユニットの直接の親に対応する必要はない。つまり、M I P モードのメモリアクセス帯域幅の割当量が設定された親領域は、符号木の現ノードより上の複数のノードである可能性がある。例えば、5 1 2 のルマサンプルの領域は、例えば 4 × 4、4 × 8、8 × 4 等のサイズの、多数の小さな符号化ユニットに分割できる。エンコーダによる検索が進むにつれて、候補の分割から生じるさまざまな候補ノードがテストされる。各候補ノード内で、結果として得られる符号化ユニットのさまざまな予測モードがテストされる。領域が占めるルマサンプルの面積が閾値を満たさない場合、領域面積テストは、真と評価し、プロセッサ 2 0 5 における制御は、ステップ 1 2 1 0 から割当量設定ステップ 1 2 2 0 に進む。領域により占められるルマサンプルの面積が閾値を満たしている場合、領域面積テストは偽と評価され、プロセッサ 2 0 5 における制御は、ステップ 1 2 1 0 から分割生成ステップ 1 2 3 0 に進む。一実施形態では、図 1 1 から図 1 6 の例に示されるように、閾値は 5 1 2 サンプルであり、領域が 5 1 2 サンプルより大きい面積を有するときに満たされる。以下に説明するように、他の閾値を使用することもできる。閾値は、通常、エンコーダ 1 1 4 及びデコーダ 1 3 4 の必要な性能に基づいて事前に決められている。

【 0 1 3 2 】

割当量設定ステップ 1 2 2 0 において、ビデオエンコーダ 1 1 4 は、プロセッサ 2 0 5 の実行下で、符号木の現ノードの面積に対応する領域に対する割当量を設定する。この割当量は、現領域をより小さな領域に分割した結果として生じる、現領域内のすべてのサブノード（サブ領域）で使用できる。5 1 2 以下のルマサンプルのすべての領域について、符号化ユニット予測モードが評価されているときに、M I P モードを適用している場合の割当量を考慮することができる。5 1 2 のルマサンプルの面積は、例えば、三分割の対象となる可能性があり、その結果、面積 1 2 8、2 5 6、及び 1 2 8 のルマサンプルの符号化ユニットが生成される。面積 1 2 8 のルマサンプルの符号化ユニットの寸法は、8 × 1 6 または 1 6 × 8 である可能性があり、その場合、最悪の場合のメモリ帯域幅制限に達する可能性があり、その結果、領域内における他の符号化ユニットに M I P モードを使用で

10

20

30

40

50

きなくなり、領域が更に細分化される可能性を含む。512のルマサンプルの面積が、例えば 16×16 の、ルマサンプル面積が256の2つの符号化ユニットに二分割される場合、最悪の場合のメモリ帯域幅制限に到達せず、結果として得られるCUに対するMIPモードの使用に制限は生じない。ただし、サンプル面積が256の領域それぞれを、より小さなCU、つまり、サイズ 16×8 、 8×16 、 8×8 、 8×4 、 4×8 、及び 4×4 のCUにさらに分割すると、先に結果として生じたCUが割当量を消費しているために、後で結果として生じるCUのMIPモードの使用が制限される可能性がある。

【0133】

4×4が最小のルマ符号化ユニットのサイズであり、他のすべての符号化ユニットはこのサイズの整数倍であり、従ってすべての領域もこのサイズの複数倍であるため、割当量は4×4ブロック面積あたりの最大ワード読み出しとして表現することができる。割当量は、4×4ブロックの場合は40ワード読み出しであり、ルマサンプル面積が512の場合、512ルマサンプル領域において許可される読み出し動作の最大数として、 $(512 \div (4 \times 4)) \times 40 = 1280$ ワード読み出しに対応する。CUにMIPモードを使用する度に、必要な読み取り数が割当量から差し引かれ、MIPモードをさらに使用するには割当量が不十分となると、図13及び図14に関連して説明するように、それ以上のCUはMIPモードを使用できなくなる。このようにして、より大きなサイズのCUは、MIPモードを使用することができ、例えば、面積512のサンプルの単一のCUまたは面積256のサンプルのCUのペアは、制約から制限を受けることなくMIPモードを使用することができる。符号木が面積512のサンプルを多数のより小さなCUに分解する場合、割当量が消費されることにより、領域内のそれ以上のCUによるMIPモードの使用を禁止する可能性がある。4×4あたり40ワードの割当量を有する512サンプル面積の場合、領域の合計の割当量は $512 \div (4 \times 4) \times 40 = 1280$ ワードとなる。512サンプル面積を 8×16 または 16×8 のサイズの4つのCUに分割した場合、2つのCUでMIPモードを使用すると、CUあたり576ワード、つまり1152ワードが消費される。残りの割当量である128ワード読み出しでは、領域内の他の2つのCUがMIPモードを使用するには不十分である。プロセッサ205における制御は、ステップ1220から分割生成ステップ1230に進む。

【0134】

分割生成ステップ1230において、ブロック分割部310は、プロセッサ205の実行下で、符号木における現ノードの候補分割セットを生成する。分割は図5に示す通りであり、関連する構文要素は図6に示され、図7に例示されている。「分割なし」、つまり510を生成する場合、符号化ユニットは後で生成される。他の分割タイプ、つまり512～520を生成する場合、符号木の追加のノードが各分割に従って生成される。方法1200は、方法1200が後で行われるときに「分割なし」で生成が行われた場合に、追加のノードの符号化ユニットを生成するために繰り返される。方法1200は、CTUの各ノードに対して順番に繰り返される。従って、CTUのすべての分割が決定される。また、分割内で分割の生成を繰り返し行うため、二分割、三分割、及び四分木分割を使用した再帰数を制限する可能性のある最小CUサイズ及び深さの再帰制約の制約内で、与えられたCTUにおけるすべての可能な符号化ユニットが検索される。プロセッサ205における制御は、ステップ1230からCU生成ステップ1240に進む。

【0135】

CU生成ステップ1240において、ブロック分割部310は、プロセッサ205の実行下で、分割生成ステップ1230で「分割なし」が生成された各場合に候補CUを生成する。ステップ1240では、候補CUの予測モードはまだ決定されておらず、最終モードはまだ選択されていないが、Z順スキャンで先行するCUの予測モードは分かっている。従って、隣接する(CU内の位置に関して)参照サンプルは、現CUを含む分割と同じ分割操作からの隣接CUから、または異なる親領域から生じる隣接領域から、または全く異なるCTUからのイントラ予測に使用することができる。ステップ1240は、符号木の領域を、図3に関連して説明したような予測ブロックを含む符号化ブロックに効率的に

10

20

30

40

50

分割する。方法 1 2 0 0 は、ステップ 1 2 4 0 で終了し、プロセッサ 2 0 5 における制御は方法 1 1 0 0 に戻り、生成された C U の予測モードが決定される。

【 0 1 3 6 】

図 1 3 は、ステップ 1 1 3 0 で実施される符号化ユニットを決定するための方法 1 3 0 0 を示す。方法 1 3 0 0 は、方法 1 2 0 0 を実行することによって生成される符号化ユニットの予測モードの決定を含む。予測モードは、イントラ予測、インター予測、及び現符号化ユニットを含む領域内で M I P モードを使用するための割当量に応じた M I P モードを含む。方法 1 2 0 0 で確立した割当量は、現 C U に対して M I P モードをテストするかどうかを制御するために使用される。割当量の一部は、C U 内の符号語の数に応じて、領域内の C U が最悪の場合のブロックのサイズになる度に（上記のケース A ~ C に関連して説明したように）消費される。割当量が一旦消費されると、領域内における M I P モードを潜在的に使用する可能性のある符号化ユニットはそれ以上検索されない。つまり、方法 1 3 0 0 は、方法 1 3 0 0 の以前の実施により、符号木の共通の親ノードに適用される設定されたメモリアクセス帯域幅の割当量を使い果たしていない場合にのみ、M I P モードのテストを実行する。

【 0 1 3 7 】

領域は、現符号化ユニットの直接の親に対応する必要はない、すなわち、M I P モードメモリアクセス帯域幅割当量が設定された親領域は、符号木の現ノードにおける現ノードより上の複数のノードであり得る。例えば、5 1 2 ルマサンプルの領域は、例えば 4 × 4、4 × 8、8 × 4 等のサイズの多数の小さな符号化ユニットに分割され得る。各 C U について、M I P モードの使用は、5 1 2 ルマサンプルの領域のための割当量の残りによって制限される。方法 1 3 0 0 は、モードセクタ 3 8 6 によって、または部分的にモジュール 3 9 6 0 によって実施され得る。方法 1 3 0 0 は、構成された F P G A、A S I C、または A S S P 等の装置によって具現化され得る。さらに、方法 1 3 0 0 は、プロセッサ 2 0 5 の実行下でビデオエンコーダ 1 1 4 によって実行され得る。従って、方法 1 3 0 0 は、コンピュータ可読記憶媒体及び/またはメモリ 2 0 6 に格納され得る。方法 1 3 0 0 は、イントラ予測モードのテストステップ 1 3 1 0 で開始する。

【 0 1 3 8 】

イントラ予測モードのテストステップ 1 3 1 0 において、プロセッサ 2 0 5 の実行下で、ビデオエンコーダ 1 1 4 は、現符号化ユニットを符号化する際に使用する可能性のある、「通常の」イントラ予測モード、すなわち D C、平面、及び角度イントラ予測モードをテストする。C U で使用可能なイントラ予測モードの中から最適なイントラ予測モードを選択するために、一般的に、ラグランジュ最適化が実行される。二次変換、つまり 3 3 0 と 3 4 4 の適用もテストされ、変換スキップの場合を含む異なるタイプの一次変換（D C T - 2、D C T - 8、D S T - 7）もテストされる。プロセッサ 2 0 5 における制御は、ステップ 1 3 1 0 からインター予測モードのテストステップ 1 3 2 0 に進む。

【 0 1 3 9 】

インター予測モードのテストステップ 1 3 2 0 において、プロセッサ 2 0 5 の実行下で、ビデオエンコーダ 1 1 4 は、インター予測された P U を生成するための様々な動きベクトルをテストする。インター予測の使用を評価する場合、動きベクトルは、候補の動きベクトルセットの中から選択される。候補の動きベクトルは、検索パターンに従って生成される。候補の動きベクトルのために取得された参照ブロックの歪みをテストする場合、符号木で禁止されているクロマ分割の適用が考慮される。分割がクロマで禁止され、ルマで許可されている場合、結果として得られるルマ C B はインター予測を使用しても良い。動き補償は、ルマチャネルにのみ適用されるため、歪み計算では、ルマの歪みは考慮されるが、クロマの歪みは考慮されない。クロマ分割が禁止されている場合、クロマチャネルで動き補償が実行されないため、クロマ歪みは考慮されない。クロマの場合、考慮されるイントラ予測モードと符号化されたクロマ T B（存在する場合）に起因する歪みが考慮される。ルマとクロマの両方を考慮する場合、インター予測検索では、最初にルマの歪みに基づいて動きベクトルを選択し、次にクロマの歪みも考慮して動きベクトルを「微調整」す

10

20

30

40

50

ることができる。微調整では、通常、サブピクセル変位など、動きベクトル値の小さな変動が考慮される。特定の動きベクトルは、「マージモード」によって生成されるものであり得、それにより、現CUの動きベクトルは、隣接するCUからの動きベクトルから導出される。マージモードは、選択された「動きベクトル予測子」に関連して適用される「動きベクトルデルタ」のシグナリングを必要とする可能性がある他の動きベクトルと比較して、ビットストリーム構文により、より完結に表現される。動きベクトル予測子は、一般に、空間的または時間的に隣接するCUから導出される。イントラ符号化されたスライス、例えば、一連のフレームの最初のフレームの場合、インター予測は利用できないため、ステップ1320は実行されない。プロセッサ205における制御は、ステップ1320から割当量内テストステップ1330に進む。

10

【0140】

割当量内テストステップ1330で、プロセッサ205の実行下で、ビデオエンコーダ114は、MIPモードメモリアクセス帯域幅割当量が現CUに適用可能であるかどうかをテストする。1330で実行されるテストは、現CUがステップ1220においてMIPモード割当量が確立された、512ルマサンプル領域内に含まれるかどうかを判定する。現CUが512ルマサンプル領域よりも大きい場合、適用される割当量の制約は無い。従って、CUは、MIPモードの使用に対する更なる制約を受けることはなく、プロセッサ205における制御は、MIPモードのテストステップ1340に進む(1330で「真」)。現CUのサイズが512ルマサンプルのサイズ以下の場合、図8を参照して説明したように、現CUでMIPモードを使用するために必要な割当量を、その領域の割当量の残りと比較する。MIPモードを現CUに適用するのに十分な割当量が無い場合、プロセッサ205における制御は、ステップ1330からモード選択ステップ1350に進む(ステップ1330で「偽」)。MIPモードを現CUに適用するのに十分な割当量がある場合(ステップ1340で「真」)、プロセッサ205における制御は、ステップ1330からステップ1340に進む。

20

【0141】

MIPモードのテストステップ1340において、モードセクタ386は、さまざまなMIPモードをテストして、CUのサイズに対して利用可能なMIPモードの中から、現CUを予測するために使用する最良のMIPモードを決定する。ステップ1310におけるイントラ予測モードのテストと同様に、テストされたMIPモードの符号化コストとそれらに関連する残差に対する歪みを打ち消すために、ラグランジュ最適化を実行しても良い。プロセッサ205における制御は、ステップ1340からモード選択ステップ1350に進む。

30

【0142】

モード選択ステップ1350において、プロセッサ205の実行下で、モードセクタ386は、ステップ1310、1320、及び1350で得られた候補から、CUの最終モードを選択する。方法1300は、プロセッサ205における制御が方法1100に戻ることで終了する。

【0143】

図14は、ステップ1140で実施されるように、CTUの符号木の符号化ユニットをビデオビットストリーム115に符号化するための方法1400を示す。方法1400は、構成されたFPGA、ASIC、ASSP等の装置によって具現化され得る。さらに、方法1400は、プロセッサ205の実行下で、例えばエントロピーエンコーダ338等のビデオエンコーダ114によって実行することができる。従って、方法1400は、コンピュータ可読記憶媒体及び/またはメモリ206に格納され得る。方法1400は、予測モードの符号化ステップ1410で開始する。

40

【0144】

予測モードの符号化ステップ1410において、ステップ1350で決定され、予測モード387によって示されるように、エントロピーエンコーダ338は、プロセッサ205の実行下で、イントラ予測(通常のイントラ予測モード使用またはMIPモード使用の

50

両方を含む)またはインター予測のいずれかの使用を示すコンテキスト符号化ピンを使用してフラグを符号化する。ステップ1410でのフラグの符号化は、通常のイントラ予測とMIPモードとを区別しない。通常のイントラ予測とMIPイントラ予測(適用可能な場合)との区別は、MIPモードフラグ符号化ステップ1430で符号化される。プロセッサ205における制御は、ステップ1410から割当量テストステップ1420に進む。
【0145】

割当量テストステップ1420において、プロセッサ205の実行下で、ビデオエンコーダ114は、MIPモードのメモリアクセス帯域幅割当量が現CUに適用可能であるかどうかをテストする。このテストにより、現CUが、ステップ1220でMIPモード割当量が確立された、512ルマサンプル領域に含まれるかどうかを決定する。現CUが512ルマサンプル領域よりも大きい領域に含まれる場合、適用すべき割当量の制約は無く、CUはMIPモードの使用に関してそれ以上の制約を受けない。プロセッサ205における制御は、MIPモードフラグの符号化ステップ1430に進む(1420で「TRUE」)。

10

【0146】

現CUが512ルマサンプルのサイズ以下の領域に含まれる場合、図8を参照して説明したように、現CUにMIPモードを使用するために必要な割当量を、その領域のための割当量の残りと比較する。MIPモードを現CUに適用するのに十分な割当量が無い場合、プロセッサ205における制御は、ステップ1420からTBの符号化ステップ1440に進む(ステップ1420で「FALSE」)。しかしながら、MIPモードを現CUに適用するのに十分な割当量がある場合(1420で「TRUE」)、現CUが512サンプル以下の領域にある場合でも、プロセッサ205における制御はステップ1420からステップ1430に進む。ステップ1420の動作は、ステップ1330の動作に対応し、従って、MIPフラグのみが、方法1300でMIPモードが検索されたCUについて、方法1400で符号化される。

20

【0147】

MIPモードフラグの符号化ステップ1430において、エントロピーエンコーダ338は、プロセッサ205の実行下で、ステップ1350で決定されたように、MIPモードか否かの選択を示すコンテキスト符号化されたピンをビットストリーム115に符号化する。ピンを符号化するために使用するコンテキストは、ステップ1420を参照して説明する。MIPモードが選択された場合、エントロピーエンコーダ338は、どの特定のMIPモードがビットストリーム338への符号化に使用されたかの選択も符号化する。MIPモードは、「最も可能性の高いモード」と残りのモードの間の選択を使用する代わりに、短縮された二値符号語を使用して符号化してもよい。短縮された二値符号語を使用することで、最も可能性の高いモードが隣接する角度イントラ予測されたCUから導出された潜在的なテーブルルックアップを含む、最も可能性の高いモードのリストを導出する必要がなくなる。プロセッサ205における制御は、ステップ1430からTBの符号化ステップ1440に進む。

30

【0148】

TBの符号化ステップ1440において、エントロピーエンコーダ338は、プロセッサ205の実行下で、現CUに関連するTBの残差係数をビットストリームに符号化する。一般に、各TBのフラグは、少なくとも1つの有意係数の存在を示し、係数は、最後の有意係数位置からDC(左上端)係数位置に戻るスキャンパターンに従って1つずつ符号化される。これにより方法1400は終了し、プロセッサ205における制御は方法1100に戻る。

40

【0149】

図15は、ビデオビットストリーム133から画像フレームの符号化ユニット及び変換ブロックを復号する方法1500を示す。方法1500は、構成されたFPGA、ASIC、またはASSP等の装置によって具現化され得る。さらに、方法1500は、プロセッサ205の実行下で、ビデオデコーダ134によって実行され得る。従って、方法15

50

00は、コンピュータ可読記憶媒体及び/またはメモリ206に格納され得る。方法1500は、フレームをCTUに分割するステップ1510で開始する。

【0150】

フレームをCTUに分割するステップ1510で、ビデオデコーダ134は、プロセッサ205の実行下で、フレームデータ133(復号予定)の現フレームをCTUの行列に分割する。分割の結果として生じるCTUの復号が始まる。プロセッサ205における制御は、ステップ1510から符号化ユニットの復号ステップ1520に進む。

【0151】

符号化ユニットの復号ステップ1520で、エントロピーデコーダ420は、プロセッサ205の実行下で、図5から図7を参照して説明したように、符号木に従ってビットストリーム133から分割フラグを復号する。分割フラグを復号することにより、ステップ1520は、CTU内、すなわち、CTUの符号木に従って、CUのサイズ及び位置を決定する動作を行うことができる。方法1300の進行は、ステップ1520への反復を含み、その結果、CTUの符号木全体を精査することになり、各CUが復号される。プロセッサ205における制御は、ステップ1520から符号化ユニットの復号ステップ1530に進む。

10

【0152】

符号化ユニットの復号ステップ1530において、エントロピーデコーダ420は、プロセッサ205の実行下で、ビットストリーム133から符号化ユニットを復号する。ステップ1530は、以下に図16に関連して説明する方法1600を呼び出して、CUを復号する。プロセッサ205における制御は、ステップ1530から最終符号化ユニットのテストステップ1540に進む。

20

【0153】

最終符号化ユニットのテストステップ1540で、プロセッサ205は、ステップ1520で分割フラグを復号することから決定されるように、現符号化ユニットがCTUの最後の符号化ユニットであるかどうかをテストする。現符号化ユニットがCTUの最後の符号化ユニットである場合(ステップ1540で「YES」)、プロセッサにおける制御は、最終CTUテストステップ1550に進む。現在の符号化ユニットがステップ1520の符号木における最後の符号化ユニットでない場合(ステップ1540で「NO」)、ステップ1520の符号木における次の符号化ユニットを復号するために選択し、プロセッサ205における制御は、ステップ1520に進む。

30

【0154】

最終CTUテストステップ1550で、プロセッサ205は、現CTUがスライスまたはフレーム内の最後のCTUであるかどうかをテストする。現CUが最後のCUではない場合(ステップ1550で「NO」)、ビデオデコーダ134は、フレームまたはスライス内の次のCTUに進み、フレーム内の残りのCTUの処理を続行するために、プロセッサ205における制御はステップ1550からステップ1520に戻る。CTUがフレームまたはスライスの最後のCTUである場合、ステップ1550は「YES」を返し、方法1500は終了する。方法1500の結果として、ビットストリームから画像フレーム全体が一連のCTUとして復号される。

40

【0155】

図16は、ステップ1530で実施される、ビデオビットストリーム133から符号化ユニットを復号する方法1600を示す。方法1600は、構成されたFPGA、ASIC、またはASSP等の装置によって具現化され得る。さらに、方法1600は、プロセッサ205の実行下で、ビデオデコーダ134によって実行され得る。従って、方法1600は、コンピュータ可読記憶媒体上及び/またはメモリ206内に格納され得る。方法1600は、pred_modeフラグの復号ステップ1602で開始する。

【0156】

pred_modeフラグの復号ステップ1602において、エントロピーデコーダ420は、コンテキスト符号化されたピンを復号して、現符号化ユニットがインター予測ま

50

たはイントラ予測（MIPモードを含む）を使用するかを決定する。プロセッサ205における制御は、ステップ1602からインター予測テストステップ1604に続く。ステップ1604では、pred_modeフラグからインター予測が使用されたかどうかを決定する。現在の符号化ユニットがインター予測を使用する場合（インター予測テストステップ1604で「TRUE」）、プロセッサ205における制御は、ステップ1602からインター予測の実行ステップ1606に続く。ステップ1606でインター予測が実行され、その結果、参照ブロック（434）が取り込まれると共に、フィルタ処理してPUを生成し、続いて各カラーチャネルのTBを復号し、TB（1690）をPUに加算してCUを復号する。プロセッサ205における制御は、ステップ1606からTBの加算ステップ1690に続く。

10

【0157】

現符号化ユニットがイントラ予測またはMIPモードを使用する場合（ステップ1604で「FALSE」）、プロセッサ205における制御は、MIPモード割当量テストステップ1610に続く。MIPモード割当量テストステップ1610では、割当量に応じて、現CUでMIPモードが利用できるかどうかを決定する。割当量は、ステップ1330または1420を参照して説明した割当量に対応する。説明した例では、現CUを含む領域が面積の閾値（512ルマサンプル）よりも大きい場合、適用可能な割当量の制約はなく、MIPモードを使用できる（ステップ1610でTRUE）。プロセッサ205における制御は、ステップ1610からMIPモードフラグの復号ステップ1620に進む。

【0158】

20

ステップ1610で、現CUを含む領域の面積が閾値以下（512ルマサンプル）である場合、図8を参照して説明したように、現CUにMIPモードを使用するために必要な割当量を、その領域の割当量の残りと比較する。現CUにMIPモードを適用するのに十分な割当量がある場合、MIPモードが利用可能であり（ステップ1610で「TRUE」）、方法1600はステップ1620に進む。割当量が不十分な場合、現CUではMIPモードを利用できない（MIPモード割当量テストステップ1610で「FALSE」）。そして、プロセッサ205における制御は、イントラ予測モードの復号ステップ1612に進む。

【0159】

ステップ1612では、イントラ予測モードの復号から始めて、通常のイントラ予測（DC、平面または角度）が使用される。プロセッサ205における制御は、ステップ1612からイントラ予測の実行ステップ1614に進む。そして、ステップ1614では、イントラ予測を実行する。プロセッサ205における制御は、ステップ1614からTBの加算ステップ1690に進む。

30

【0160】

ステップ1620の実行に際して、MIPフラグは、単一のコンテキストコード化されたピンを使用して、エントロピーデコーダ420によって復号される。ピンは1つのコンテキストを使用するため、例えば、隣接するCUのMIPモードの使用や、1つのピンを復号するための複数の可能なコンテキストから1つのコンテキストを選択するために使用され得るブロックサイズやその他のパラメータに応じて、コンテキストを選択する必要は無い。プロセッサにおける制御は、ステップ1620から、MIPモードが選択されたかをテストするステップ1630に進む。

40

【0161】

ステップ1630では、MIPモードが選択されているかどうかを決定する。復号されたMIPモードフラグがMIPモードが使用されていないことを示す場合（MIPモードが選択されたかをテストするステップ1630で「FALSE」）、制御はステップ1612に進んで、通常のイントラ予測モードの1つを使用してブロックを復号する。復号されたMIPモードフラグが、CUにMIPモードが使用されていることを示す場合（ステップ1630で「TRUE」）、プロセッサ205における制御は、MIPモードの復号ステップ1640に進む。

50

【 0 1 6 2 】

M I Pモードは、ステップ1 6 4 0を実行する際にビットストリーム1 3 3から復号される。プロセッサ2 0 5における制御は、ステップ1 6 4 0から行列係数の読み出しステップ1 6 5 0に進む。復号されたM I Pモードは、ステップ1 6 5 0において、行列係数4 8 1のセットを行列係数メモリ4 8 6から読み出すために使用される。割当量に使用に応じてステップ1 6 1 0でのM I Pモードの使用を制限することにより、行列係数メモリ4 8 6がマトリックス係数4 8 1を供給するために必要な最大メモリ帯域幅消費が低減される。

【 0 1 6 3 】

方法1 6 0 0は、ステップ1 6 5 0から隣接するサンプルのフィルタ処理ステップ1 6 6 0に遷移する。隣接する参照サンプル4 6 4は、ステップ1 6 6 0でフィルタ処理される。方法1 6 0 0は、ステップ1 6 6 0から行列乗算ステップ1 6 7 0に遷移する。ステップ1 6 0 0は、モジュール4 8 2が、例えば、行列係数4 8 1及びサンプル4 6 4を使用することで実施することができる。フィルタ処理された参照サンプル及び行列係数4 8 1は、図8の例8 2 8と同様に、ステップ1 6 7 0で乗算される。

10

【 0 1 6 4 】

方法1 6 0 0は、ステップ1 6 7 0からP Bの補間ステップ1 6 8 0に遷移する。ステップ1 6 7 0の実行により決定された疎ブロック（すなわち、8 3 0）は、ステップ1 6 8 0での補間処理によりP B（すなわち4 8 3）を埋めるために使用される。方法1 6 0 0は、ステップ1 6 8 0のP Bの補間ステップからT Bの加算ステップ1 6 9 0に遷移する。復号された残差は、T Bを生成するために使用され、生成されたT Bは、C Uを復号するためにステップ1 6 9 0においてP B 4 8 3に加算される。符号化ユニットは、ステップ1 6 9 0で生成された予測ブロックを用いて復号される。方法1 6 0 0は、ステップ1 6 9 0の完了時に終了する。

20

【 0 1 6 5 】

方法1 6 0 0を実行する際に、ビデオデコーダ1 3 4は、行列係数を取り出すための最悪の場合のメモリ帯域幅に制約を伴う行列イントラ予測のサポートを達成する。ここで説明した制約は、使用制限が有効になっていない場合と比較して、エラーに影響を与えないという点においてM I Pモードの選択を過度に制限するものではない。選択を過度に制限しない動作は、一般にM I Pモードで符号化されたC Uのうち、ほんの2 0 %が、同じくM I Pモードを使用する左または上のC Uに隣接している、という決定された統計的な見込みに基づいている。メモリアクセス割当量を確立するための細かさとして使用される領域よりも大きい領域間の境界、つまり、5 1 2ルマサンプルと、2 0 %の見つかったものの測定に含まれるC T U間の境界。

30

【 0 1 6 6 】

図1 1から図1 6の実施例では、制約に基づいてM I Pモードが使用されるかどうかを示す。制約は、例えば図1 2において、現C Uの領域の面積が閾値面積を満たすかどうかに応じて決まる、現C Uの領域の面積及び行列イントラ予測フラグが符号化または復号されるか、に基づいて、実施される。説明した実施では、ステップ1 3 3 0、1 4 2 0、1 6 1 0に関連して説明したように、行列イントラ予測フラグが符号化されるかどうかは、領域が閾値を満たしている場合には領域の面積に応じて、（ i i ）領域の面積が閾値を満たしていない場合には領域の割当量に応じて決まる。従って、行列イントラ予測フラグが符号化（または復号）されるかどうかは、少なくとも領域の面積が閾値面積を満たすかどうかによって決まる。

40

【 0 1 6 7 】

図1 1から図1 4に記載された実施において、M I Pモードフラグは、ステップ1 4 2 0及び1 4 3 0に関連して説明したように、割当量テストが満たされた場合にのみ、ビットストリーム1 1 5に符号化される。これに対応して、図1 6でビットストリームを復号するとき、M I Pモードの割当量のテストステップでT R U Eを返した場合にのみ、M I Pモードフラグは復号される。言い換えると、行列イントラ予測が使用されている場合

50

にのみ、C Uの行列イントラ予測フラグが復号される。

【0168】

図12及び図13を参照して説明したように、ビデオエンコーダ114及びビデオデコーダ134の別の構成では、MIPモードの検索が制限されている。ただし、この別の構成では、MIPフラグのシグナリングは、各C Uの割当量の消費状態に関わらず、各C Uのビットストリームに含まれる。つまり、行列イントラ予測が使用されているかどうかに関わらず、C Uの行列イントラ予測フラグが復号される。この別の構成では、図14の破線で示される矢印1411により示されるように、ステップ1420が省略され、制御は、ビデオエンコーダ114においてステップ1430に進む（ステップ1420で「TRUE」）。この別の構成のステップ1430では、MIPモードフラグはすべての場合において含まれるが、ステップ1350でMIPモードが選択されなかった場合にはゼロに設定することができる。同様に、ビデオデコーダ134では、ステップ1610が省略され、制御はステップ1620に移行する（効率的に、ステップ1610では常に「TRUE」が返される）。MIPモードフラグは、ステップ1620で復号され、ビットストリームを符号化するステップ1350でMIPモードが選択されなかった場合はゼロである。各C Uに対してMIPフラグを符号化するかどうかを復号するという追加の負担が、エントロピーエンコーダ338とエントロピーデコーダ420にかかることを回避できる。ビットストリームが各C UについてMIPモードフラグを含むにもかかわらず、方法1200及び1300で実行されるような検索により、ビデオデコーダ134の係数メモリ486から行列係数481を読み出すためのメモリ帯域幅は変わらずに制限されるため、この（制限された）最悪の場合のMIPモードの使用を処理するために必要なリソースの供給を削減することができる。

【0169】

上記図12から図17に関連して説明した実施は、C Uを含む領域の面積に基づいてMIPモードの使用に制限を適用する。ビデオエンコーダ114とビデオデコーダ134のさらに別の構成では、図12で実行された制約により、領域の面積に基づくのではなく、最悪の場合のメモリ帯域幅に対応するブロックサイズに対して、行列イントラ予測の使用を禁止する。事実上、各符号化ブロックのサイズに基づいて、行列イントラ予測が使用される（及び行列イントラ予測フラグが符号化または復号される）。ブロックサイズが4×4、4×8、8×4、8×8、8×16、16×8の場合、4×4ブロックあたり80ワードという最大アクセス密度（例えば、4×4ルマサンプルブロックについて測定）となる。前のブロックサイズのセットのうち、特定のブロックサイズでは、最悪の場合である4×4ブロックあたり72ワードとなるが、リソースの供給の目的では、これは4×4ブロックあたり80ワードの場合と同じカテゴリで扱うことができる。最悪の場合のブロックサイズを禁止すると、最悪の場合で4×4ブロックあたり40ワードとなる（特定のブロックサイズでは、最悪の場合で4×4ブロックあたり36ワードとなる）。

【0170】

図17は、ステップ1130で実施される符号化ユニットを決定する方法1700を示している。方法1300は、制限が、最悪の場合のメモリ帯域幅に対応するブロックサイズの行列イントラ予測の使用を禁止することに関して実施される、方法1300の別の方法を提供する。方法1700は、方法1200を実行することによって生成される符号化ユニットの予測モードを決定する工程を含む。予測モードは、イントラ予測、インター予測、及び、現符号化ユニットを含む領域内でMIPモードを使用できる場合にはMIPモードを含む。方法1700は、イントラ予測モードのテストステップ1710で開始する。

【0171】

イントラ予測モードのテストステップ1310で、ビデオエンコーダ114は、プロセッサ205の実行下で、現符号化ユニットの符号化で使用される可能性があるために、「通常の」イントラ予測モード、すなわち、DC、平面、及び角度イントラ予測モードをテストし、ステップ1310について説明した方法で動作する。プロセッサ205における制御は、ステップ1710からインター予測モードのテストステップ1720に進む。

【0172】

インター予測モードのテストステップ1720において、ビデオエンコーダ114は、プロセッサ205の実行下で、インター予測されたPUを生成するための様々な動きベクトルをテストする。インター予測の使用を評価する際に、動きベクトルは、動きベクトル候補のセットから選択される。ステップ1720は、ステップ1320と同じように動作する。プロセッサ205における制御は、ステップ1720からMIP許容サイズテストステップ1730に進む。

【0173】

MIP許容サイズテストステップ1730において、ビデオエンコーダ114は、プロセッサ205の実行下で、CUがMIPモードが許容されるサイズであるかどうかをテストする。MIPモードは、一部の実装では最悪の場合のサイズである 4×4 、 4×8 、 8×4 、 8×8 、 8×16 、 16×8 を除くすべてのサイズで許容される。現CUが許容サイズである場合(1730で「TRUE」)、プロセッサ205における制御は、ステップ1730からMIPモードのテストステップ1740に進む。現CUが禁止サイズである場合(ステップ1730で「FALSE」)、プロセッサ205における制御は、ステップ1730からモード選択ステップ1750に進む。

【0174】

MIPモードのテストステップ1740において、モードセクタ386は、様々なMIPモードをテストして、CUのサイズに対して利用可能なMIPモードの中から、現CUを予測するために使用する最良のMIPモードを決定する。ステップ1740は、ステップ1340と同様に動作する。プロセッサ205における処理は、ステップ1740からモード選択ステップ1750に進む。

【0175】

モード選択ステップ1750において、モードセクタ386は、プロセッサ205の実行下で、ステップ1710、1720、及び1750において得られた候補から、CUの最終モードを選択する。方法1700は、プロセッサ205における制御が方法1100に戻ると終了する。

【0176】

図18は、制限が、最悪の場合のメモリの帯域幅に対応するブロックサイズに対する行列イントラ予測の使用を禁止することに関連して、ステップ1140で実行されるような実施における、CTUの符号木の符号化ユニットをビデオビットストリーム115に符号化する方法1800を示す。方法1800は、構成されたFPGA、ASIC、またはASSP等の装置によって具現化され得る。さらに、方法1800は、ビデオエンコーダ114により、プロセッサ205の実行下で実行され得る。従って、方法1800は、コンピュータ可読記憶媒体及び/またはメモリ206に格納され得る。方法1800は、予測モードの符号化ステップ1810で開始する。

【0177】

予測モードの符号化ステップ1810において、エントロピーエンコーダ338は、プロセッサ205の実行下で、ステップ1750で決定されるように、イントラ予測(通常のイントラ予測モードの使用またはMIPモードの使用の両方を含む)、またはイントラ予測のいずれかの使用を示すコンテキスト符号化ピンを使用してフラグを符号化する。プロセッサ205における制御は、ステップ1810からMIP許容サイズテストステップ1820に進む。

【0178】

MIP許容サイズテストステップ1820において、ビデオエンコーダ114は、プロセッサ205の実行下で、現CUが許容サイズであるかどうかをテストする。許容サイズ及び禁止サイズは、方法1700のステップ1730の場合と同じである。現CUが、許容サイズである場合、プロセッサ205における制御は、MIPモードフラグの符号化ステップ1830に進む(1820で「TRUE」)。

【0179】

10

20

30

40

50

現C Uが許容可能なサイズではない場合（ステップ1 8 2 0で「F A L S E」）、プロセッサ2 0 5における制御は、ステップ1 8 2 0からT Bの符号化ステップ1 8 4 0に進む。

【0 1 8 0】

M I Pモードフラグの符号化ステップ1 8 3 0において、エントロピーエンコーダ3 3 8は、プロセッサ2 0 5の実行下で、ステップ1 3 5 0で決定されるように、M I Pモードが選択されたか否かを示すコンテキスト符号化ピンをビットストリーム1 1 5に符号化する。ステップ1 8 3 0は、ステップ1 4 3 0と同じように動作する。プロセッサ2 0 5における制御は、ステップ1 8 3 0からT Bの符号化ステップ1 8 4 0に進む。

【0 1 8 1】

T Bの符号化ステップ1 8 4 0では、エントロピーエンコーダ3 3 8は、プロセッサ2 0 5の実行下で、現C Uに関連するT Bの残差係数をビットストリームに符号化する。ステップ1 8 4 0は、ステップ1 4 4 0と同じように動作する。そして、方法1 8 0 0は終了し、プロセッサ2 0 5における制御は、方法1 1 0 0に戻る。

【0 1 8 2】

図1 9は、制約が、最悪の場合のブロックのM I Pモードの禁止に関連して、ステップ1 5 3 0で実行されるような実施における、ビデオビットストリーム1 3 3から符号化ユニットを復号する方法1 9 0 0を示す。方法1 9 0 0は、構成されたF P G A、A S I C、またはA S S P等の装置によって具現化され得る。さらに、方法1 9 0 0は、プロセッサ2 0 5の実行下で、ビデオデコーダ1 3 4により実行され得る。従って、方法1 9 0 0は、コンピュータ可読記憶媒体及び/またはメモリ2 0 6に格納され得る。方法1 9 0 0は、p r e d _ m o d eフラグの復号ステップ1 9 0 2で開始する。

【0 1 8 3】

p r e d _ m o d eフラグの復号ステップ1 9 0 2において、エントロピーデコーダ4 2 0は、コンテキスト符号化されたピンを復号して、現符号化ユニットがインター予測またはイントラ予測（M I Pモードを含む）を使用するかを決定する。現符号化ユニットがインター予測を使用する場合（インター予測テストステップ1 9 0 4で「T R U E」）、プロセッサにおける制御は、インター予測の実行ステップ1 9 0 6に進む。ステップ1 6 0 6でインター予測が実行され、その結果、参照ブロック（4 3 4）が取り込まれると共に、フィルタ処理してP Uを生成し、続いて各カラーチャネルのT Bを復号し、T B（1 6 9 0）をP Uに加算してC Uを復号する。同様に、プロセッサ2 0 5における制御は、ステップ1 9 0 6からT Bの加算ステップ1 9 9 0に進む。

【0 1 8 4】

現符号化ユニットがイントラ予測またはM I Pモードを使用する場合（ステップ1 9 0 4で「F A L S E」）、プロセッサ2 0 5における制御は、M I P許容サイズテストステップ1 9 1 0に進む。M I P許容サイズテストステップ1 9 1 0では、M I Pモードが現C Uサイズで許容されるかどうかを判断する。M I P許容サイズは、ステップ1 7 3 0または1 8 2 0を参照して説明されたものである。現C Uが許容サイズである場合（ステップ1 9 1 0でT R U E）、プロセッサ2 0 5における制御は、ステップ1 9 1 0からM I Pモードフラグの復号ステップ1 9 2 0に進む。ステップ1 9 1 0において、現C Uが許容サイズではない場合（ステップ1 9 1 0で「F A L S E」）、プロセッサ2 0 5における制御は、イントラ予測モードの復号ステップ1 9 1 2に遷移する。

【0 1 8 5】

ステップ1 9 1 2で、通常のイントラ予測（D C、平面または角度）が復号され、方法1 9 0 0は、イントラ予測の復号ステップ1 9 1 4に進む。ステップ1 9 1 2及び1 9 1 4は、ステップ1 6 1 2及び1 6 1 4に関連して説明したように動作する。プロセッサ2 0 5における制御は、ステップ1 9 1 4からT Bの加算ステップ1 9 9 0に続く。

【0 1 8 6】

ステップ1 9 2 0の実行時に、M I Pフラグが復号される。プロセッサ2 0 5における制御は、ステップ1 9 2 0から、M I Pモードが選択されたかをテストするステップ1 9

10

20

30

40

50

30に遷移する。

【0187】

ステップ1930は、MIPモードが選択されているかどうかを決定するために動作する。復号されたMIPモードフラグがMIPモードが使用されていないことを示す場合（MIPモードが選択されたかをテストするステップ1930で「FALSE」）、制御はステップ1912に進んで、通常のイントラ予測モードの1つを使用してブロックを復号する。復号されたMIPモードフラグが、CUにMIPモードが使用されていることを示す場合（ステップ1930で「TRUE」）、プロセッサにおける制御は、MIPモードの復号ステップ1940に進む。

【0188】

MIPモードは、ステップ1940を実行する際にビットストリーム133から復号される。プロセッサ205における制御は、ステップ1940から、行列係数の読み出しステップ1950に遷移する。復号されたMIPモードは、ステップ1950において、行列係数481のセットを行列係数メモリ486から読み出すために使用される。行列係数は、ステップ1650のように、MIPモードを使用すると決定された各予測ブロックについて読み出される。

【0189】

方法1900は、ステップ1950から隣接するサンプルのフィルタ処理ステップ1660に移行する。隣接する参照サンプル464は、ステップ1960でフィルタ処理される。方法1900は、ステップ1960から行列乗算ステップ1970に進む。フィルタ処理された参照サンプル及び行列係数481は、図8の例828と同様に、ステップ1970で乗算される。

【0190】

方法1900は、ステップ1970からPBの補間ステップ1980に遷移する。ステップ1970の実行により決定された疎ブロック（すなわち、830）は、ステップ1980での補間処理によりPB（すなわち483）を補間するために用いられる。方法1900は、PBの補間ステップ1980から、TBの加算ステップ1990に遷移する。復号された残差は、TBを生成するために使用され、生成されたTBは、ステップ1690と同様、ステップ1990でPB483に加算することにより、CUを復号する。方法1900は、ステップ1990の完了をもって終了する。

【0191】

方法1300、1400、及び1600に記載された変形例は、それぞれ、方法1700、1800、及び1900にも適用することができる。

【0192】

ブロックサイズがMIPモードの利用可能性の基準として使用される場合、方法1700におけるように、メモリ割当量を確立及び更新する必要はない、すなわち、ステップ1210及び1220は省略される。特定のブロックサイズのMIPモードを削除すると、メモリアクセスの割当量を定めるよりも実施が簡単になるが、多くの一般的なブロックサイズからのMIPモードが無いことにより、圧縮性能が低下する。係数メモリ（つまり、392、486）のサイズは、4×4ブロックが無いために、「セットA」の行列係数（18セットの行列係数とバイアス値）を格納する必要が無いために、小さくなる。4×4、4×8、8×4、8×8、8×16、及び16×8（「小さなブロック」）からのMIPモードの削除には、比較的複雑な動作であるMIPモードのフィードバックループが、これらの小さなブロックをサポートする必要が無い、という利点がある。

【0193】

さらに別の構成では、（上記にリストされているように）小さなブロックのサブセットに対するMIPモードの使用が禁止される。例えば、MIPモードはステップ1730、1820、及び1910で4×4ブロックに対してのみ禁止され、その他のすべてのブロックサイズに対しては許可されるか、または、MIPモードは4×4、4×8、及び8×4ブロックに対して禁止され、その他のすべてのブロックサイズに対しては許可されてい

10

20

30

40

50

る。つまり、行列イントラ予測は使用されず、実施によっては、符号化ユニットのサイズが禁止されているサイズの1つである場合に、行列イントラ予測フラグが復号される。最悪の場合のメモリ帯域幅は、MIPモードの使用に対する制限が完全に無い場合と比較して減少しないが、これらの非常に小さいブロックサイズの除外により、イントラ再構築フィールドバックループの厳しさは軽減される。4×4を削除することにより、「セットA」に関連付けられた行列係数を係数メモリ（つまり、392、486）に保存する必要も無くなる。あるいは、禁止されているブロックサイズのセットが4×4、4×8、8×4、及び8×8としても良く、その場合、セットA及びセットBはビデオエンコーダ114またはビデオデコーダ134に存在しない。セットAとセットBを削除すると、セットAとセットBに関連付けられた行列係数が不要になるため、圧縮性能は低下するが、メモリ消費量を削減することができる。

10

【0194】

さらに別の構成では、メモリ割当量は、4×4ルマサンプル面積あたり40ワードよりもさらに低く、例えば、4×4ルマサンプル面積あたり20または10ワードであっても良い。割当量は、1024または2048のルマサンプルの領域サイズに対応するノードなどで、より大きな領域サイズで設定される。上述した構成と同様に、制限内における後のCUは、これらの領域内における先のCUのMIPモードの使用に応じて、メモリ割当量を利用できるかが制限される。メモリ帯域幅の更なる削減は、圧縮効率の低下と引き換えに達成できる。

【0195】

20

さらに別の構成では、メモリ割当量は、サイズ4×4、4×8、及び8×4のCUに適用するために、ステップ1220、1330、1420、及び1610で64ルマサンプルの領域サイズで確立される（「小さなCUメモリ割当量」）。個別のメモリ割当量は、512の領域サイズで確立されるが、8×8、特に8×16及び16×8を超えるサイズのCUにのみ適用される（「より大きなCUメモリ割当量」）。割当量は、両方共、4×4ルマサンプル面積あたり40ワードに設定されている。小さいCUメモリ割当量が64ルマサンプル面積内に含まれるCUにのみ適用されるのに対し、より大きいCUメモリ割当量は64ルマサンプルより大きいCUにのみ適用されるため、割当量は合計の行列メモリ帯域幅に対して追加の割当量を形成する。

【0196】

30

本明細書に開示される構成は、4×4メモリ領域あたりのワードによりメモリ帯域幅を説明しているが、メモリへのアクセスは、関連するメモリに過度に高いクロック周波数を必要とすることなく行列係数を読み出せるようにするために、何らかのSIMD形式でワードをグループ化する可能性が高いことが理解される。しかしながら、そのようなより広いメモリはそれ自体コストがかかり、また、行列係数は同じメモリ内において他のデータと共有される可能性があるため、MIPモードの使用制限により減少したアクセスに競合が生じてしまう。

【0197】

領域毎に最悪の場合のメモリ帯域幅を制限するためにMIPモードの使用を制限すると、各領域で使用可能な割当量が先のCUによって消費されてしまうため、その領域の先のCUはMIPモードを使用し、その領域で後となるCUはMIPモードを使用することができない、という偏りが生じ得る。MIPモードが選択されるCUの分布は一般にややまばらであるため、このような偏りは通常発生しない。

40

【0198】

記載された構成は、いつMIPモードが使用され得るかについての制限を有するため、制限無しでMIPモードの実行を許可する場合と比較して、計算の複雑さが軽減される。制限無く検索を行った場合のMIPモード選択の統計値は、通常、モードセクタ386でのMIPモード選択に課せられる制限をトリガしないため、符号化効率を比例的に低下させることなく、複雑さが軽減される。従って、最悪の場合のメモリ帯域幅は、符号化効率の相応の損失無く、狭められる。

50

産業上の利用可能性

【 0 1 9 9 】

記載された構成は、コンピュータ及びデータ処理産業、特に、ビデオ及び画像信号等の信号の符号化及び復号のためのデジタル信号処理に適用可能であり、高い圧縮効率を達成する。

【 0 2 0 0 】

本明細書で説明される構成は、残差の符号化及び復号のために、係数パリティに従って更新し、コンテキスト及び係数の量子化器を選択する、トレリスベースの状態機械を使用することを可能にする。説明された構成により、状態更新の順次的性質による過度の遅延を課すことなく、トレリスベースの状態機械を実現することができる。

10

【 0 2 0 1 】

上記は、本発明のいくつかの実施形態を説明するのみであり、本発明の範囲及び精神から逸脱することなく、変形及び／または変更することができ、実施形態は例示的であって、限定的ではない。

20

30

40

50

【図面】

【図 1】

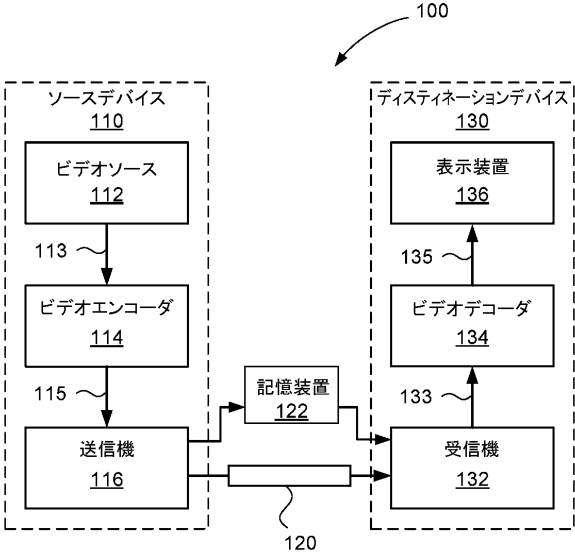


Fig. 1

【図 2 A】

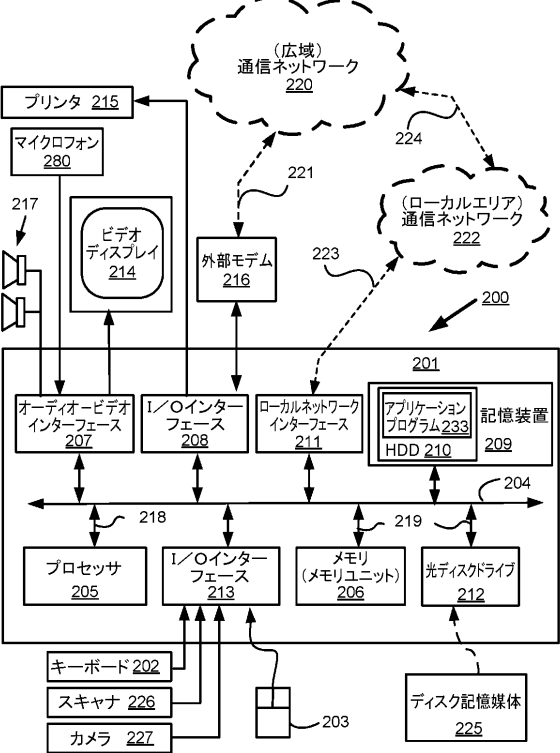


Fig. 2A

【図 2 B】

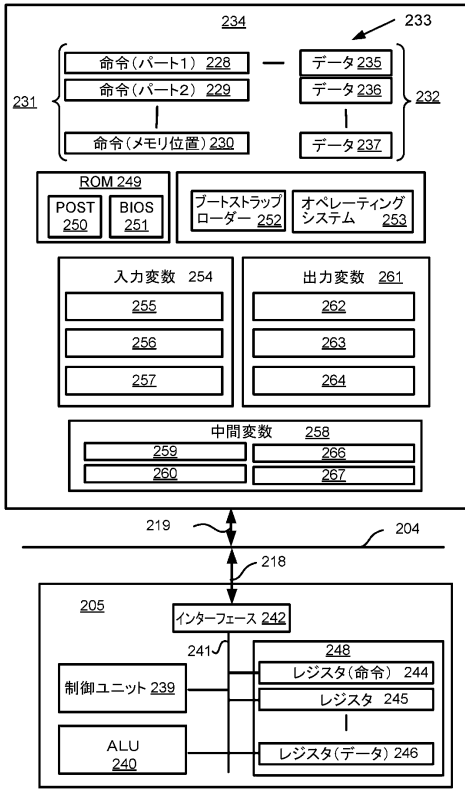


Fig. 2B

【図 3】

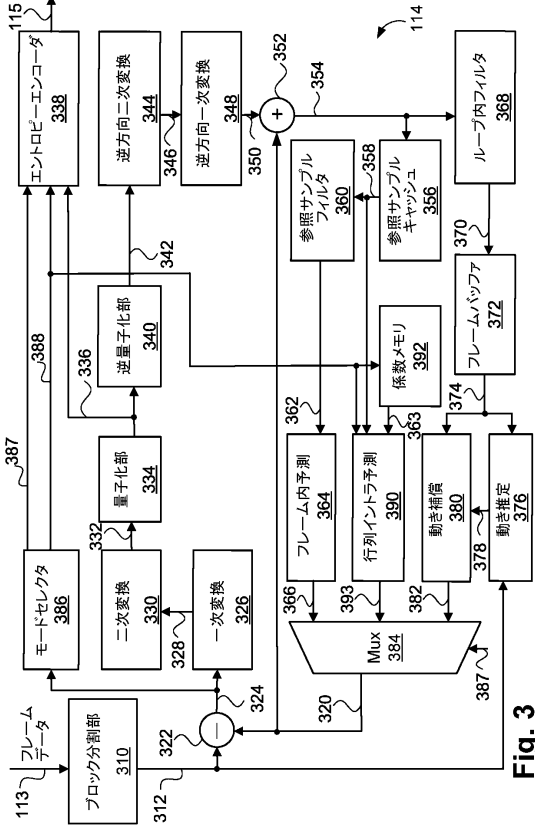


Fig. 3

【図 4】

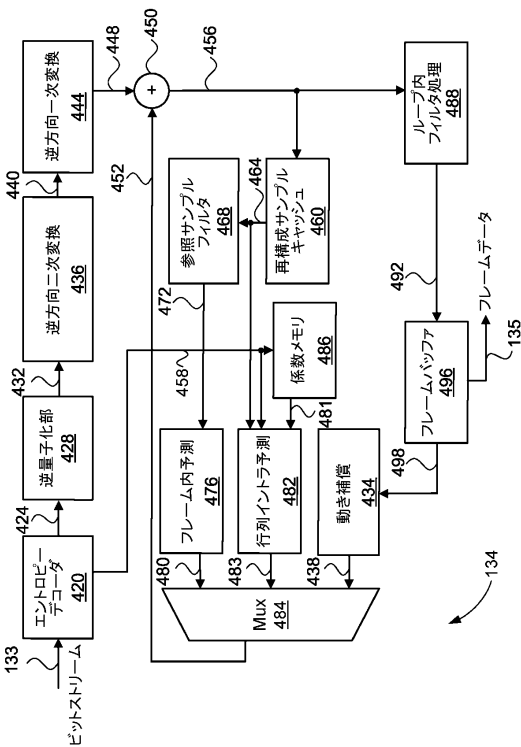


Fig. 4

【図 5】

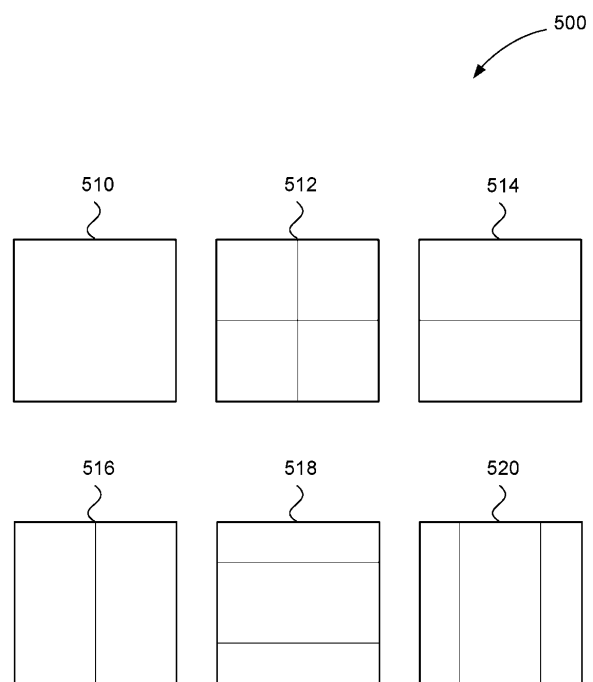


Fig. 5

10

20

30

40

50

【図 9 A】

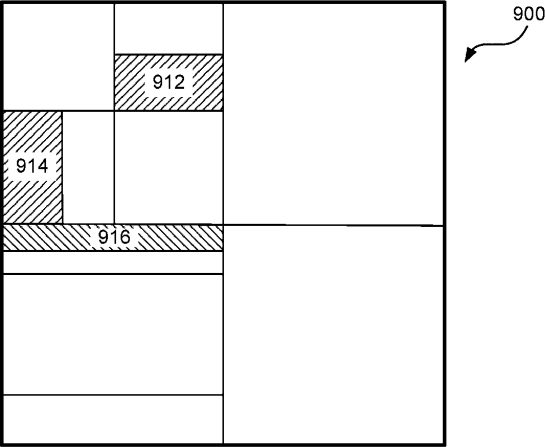


Fig. 9A

【図 9 B】

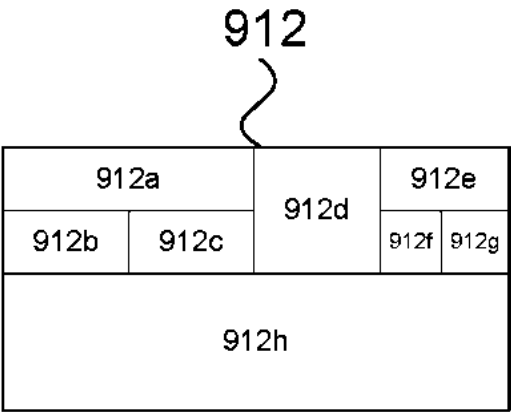


Fig. 9B

【図 1 0】

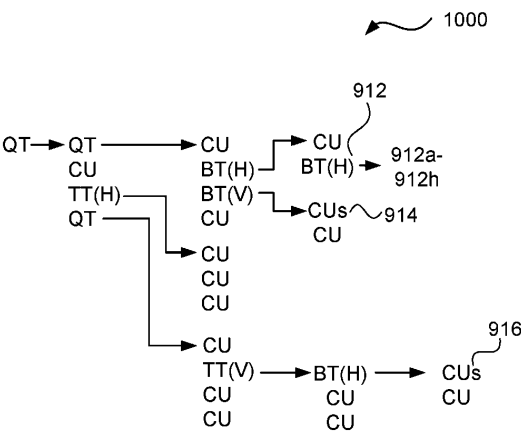


Fig. 10

【図 1 1】

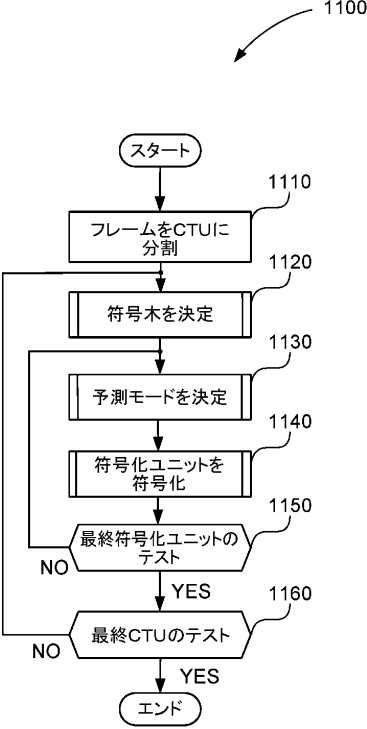


Fig. 11

10

20

30

40

50

【図 1 2】

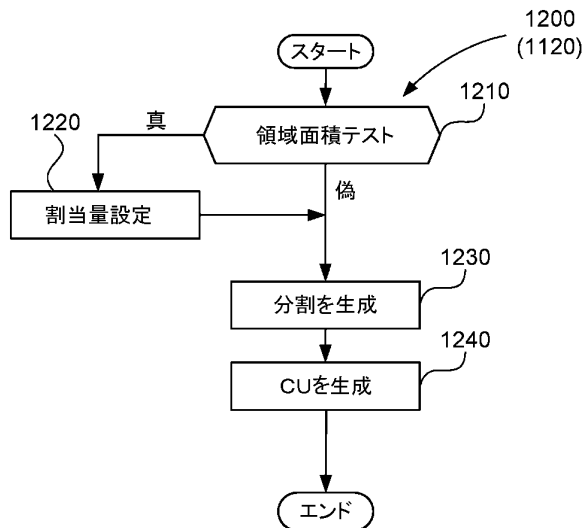


Fig. 12

【図 1 3】

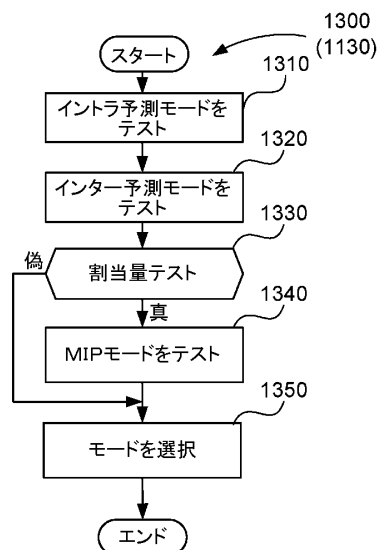


Fig. 13

【図 1 4】

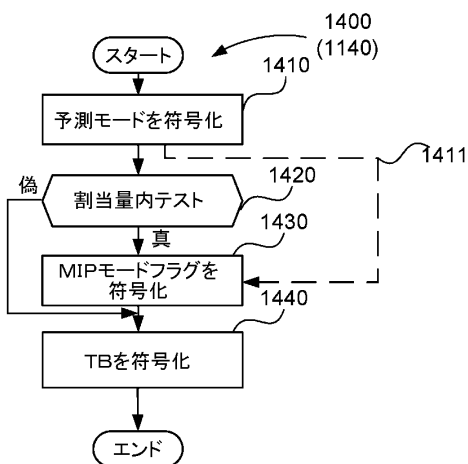


Fig. 14

【図 1 5】

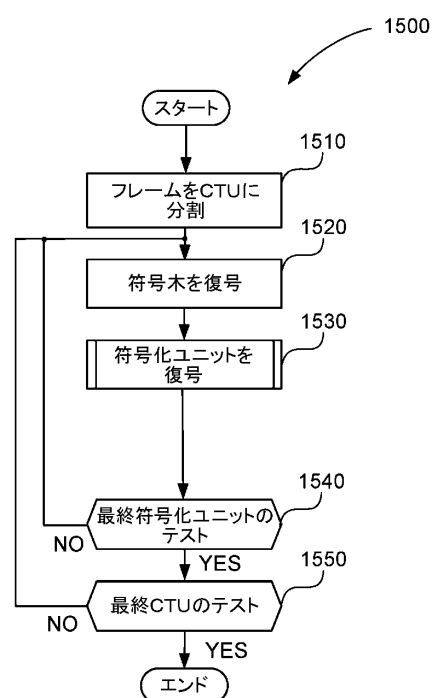


Fig. 15

10

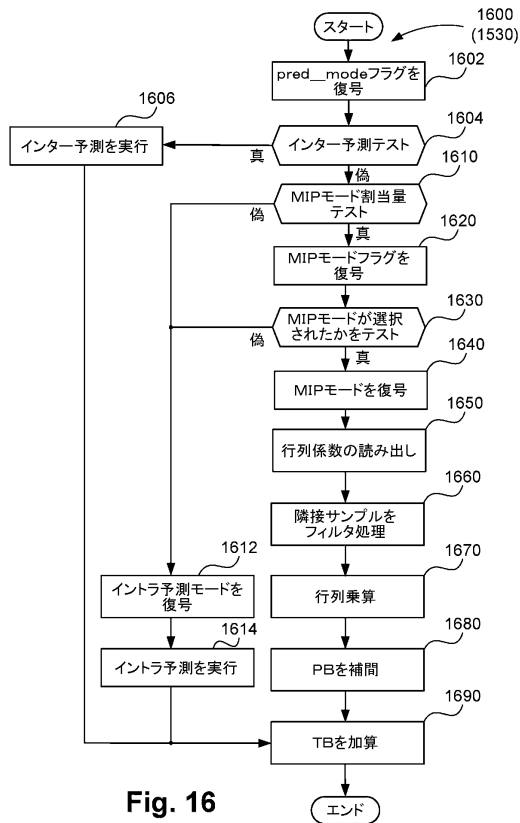
20

30

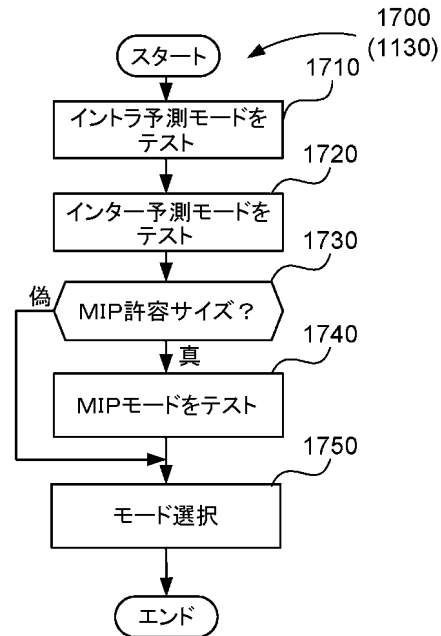
40

50

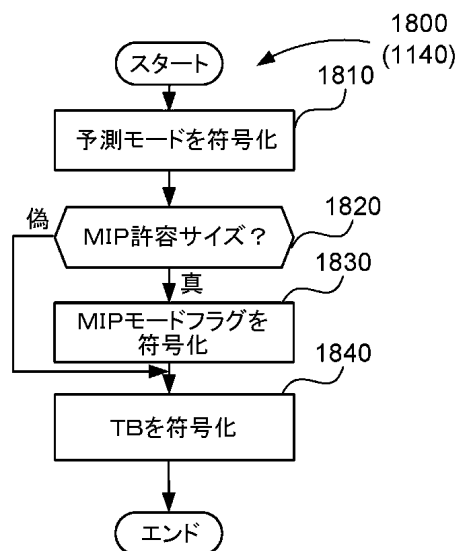
【図 16】



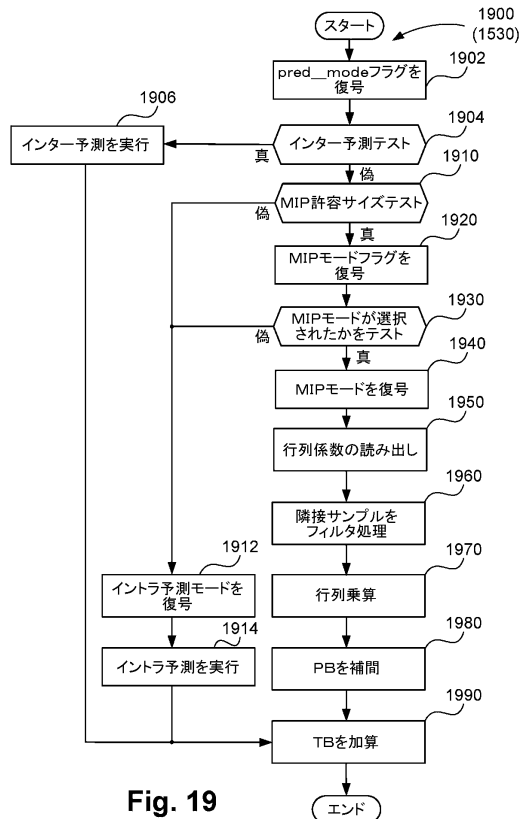
【図 17】



【図 18】



【図 19】



10

20

30

40

50

フロントページの続き

ノン オーストラリア プロプライエタリー リミテッド内

審査官 間宮 嘉誉

- (56)参考文献 国際公開第 2 0 2 0 / 2 4 6 8 0 5 (W O , A 1)
特表 2 0 2 2 - 5 3 5 3 4 9 (J P , A)
米国特許出願公開第 2 0 2 2 / 0 0 7 8 4 4 9 (U S , A 1)
ZHAO, Liang et al. , Non-CE3: Simplifications for MIP , JVET-O0171 (version 1) , ITU , 2019年06月22日 , pp.1-3 , [online] , [retrieved on 2024-06-28] , Retrieved from the Internet: URL: https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O0171-v1.zip , JVET-O0171.docx
SALEHIFAR, Mehdi et al. , CE3 Related: Low Memory and Computational Complexity Matrix Based Intra Prediction (MIP) , JVET-O0139 (version 1) , ITU , 2019年06月21日 , pp.1-6 , [online] , [retrieved on 2024-06-28] , Retrieved from the Internet: URL: https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O0139-v1.zip , JVET-O0139.docx
(58)調査した分野 (Int.Cl. , D B 名)
H 0 4 N 7 / 1 2
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8