

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7520853号
(P7520853)

(45)発行日 令和6年7月23日(2024.7.23)

(24)登録日 令和6年7月12日(2024.7.12)

(51)国際特許分類	F I
H 0 4 N 19/13 (2014.01)	H 0 4 N 19/13
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
H 0 4 N 19/70 (2014.01)	H 0 4 N 19/70

請求項の数 10 (全32頁)

(21)出願番号	特願2021-541621(P2021-541621)	(73)特許権者	518338149
(86)(22)出願日	令和2年3月6日(2020.3.6)		インターデジタル ヴイシー ホールディ ングス, インコーポレイテッド
(65)公表番号	特表2022-523903(P2022-523903 A)		アメリカ合衆国, デラウェア州 1 9 8 0 9, ウィルミントン, ベルビュー パ ークウェイ 2 0 0, スイート 3 0 0
(43)公表日	令和4年4月27日(2022.4.27)	(74)代理人	100079108
(86)国際出願番号	PCT/US2020/021281		弁理士 稲葉 良幸
(87)国際公開番号	WO2020/185530	(74)代理人	100109346
(87)国際公開日	令和2年9月17日(2020.9.17)		弁理士 大貫 敏史
審査請求日	令和5年2月28日(2023.2.28)	(74)代理人	100117189
(31)優先権主張番号	19305292.5		弁理士 江口 昭彦
(32)優先日	平成31年3月12日(2019.3.12)	(74)代理人	100134120
(33)優先権主張国・地域又は機関	欧州特許庁(EP)		弁理士 内藤 和彦
(31)優先権主張番号	19305645.4	(74)代理人	100108213
(32)優先日	令和1年5月21日(2019.5.21)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ビデオコード化のための残差コード化における通常のピンの柔軟な割り当て

(57)【特許請求の範囲】

【請求項1】

限られた数の通常のピンを使用して、変換ブロックの量子化済みの係数のサブブロックを表すコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを含むビデオ符号化方法であって、前記コード化が、C A B A C符号化であり、前記通常のピンの数が、変換ブロックのサンプルの数に基づいて変換ブロックに割り当てられたバジェットに基づいて決定される、ビデオ符号化方法。

【請求項2】

通常のピンを使用して、変換ブロックの量子化済みの係数のサブブロックを表すコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを含むビデオ復号方法であって、前記残差復号プロセスが、C A B A C復号を使用して行われ、前記通常のピンの数が、変換ブロックのサンプルの数に基づいて変換ブロックに割り当てられたバジェットに基づいて決定される、ビデオ復号方法。

【請求項3】

限られた数の通常のピンを使用して、変換ブロックの量子化済みの係数のサブブロックを表すコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを含むビデオ符号化装置であって、前記コード化が、C A B A C符号化であり、前記通常のピンの数が、変換ブロックのサンプルの数に基づいて変換ブロックに割り当てられたバジェットに基づいて決定される、ビデオ符号化装置。

【請求項4】

通常のピンを使用して、変換ブロックの量子化済みの係数のサブブロックを表すコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを含むビデオ復号装置であって、前記残差復号プロセスが、C A B A C 復号を使用して行われ、前記通常のピンの数が、変換ブロックのサンプルの数に基づいて変換ブロックに割り当てられたバジェットに基づいて決定される、ビデオ復号装置。

【請求項 5】

コード化されている変換ブロックに対し、複数のコード化グループ間で割り当てられる前記バジェットが、通常の C A B A C コード化モードを使用してバイナリ要素がコード化される度にデクリメントされる、請求項 1 に記載の方法。

【請求項 6】

復号されている変換ブロックに対し、複数のコード化グループ間で割り当てられる前記バジェットが、通常の C A B A C 復号モードを使用してバイナリ要素が復号される度にデクリメントされる、請求項 2 に記載の方法。

【請求項 7】

プロセッサによって実行されると、請求項 1、2、5 又は 6 の少なくとも一項に記載の方法のステップを実施するためのプログラムコード命令を含むコンピュータプログラム。

【請求項 8】

プロセッサによって実行されると、請求項 1、2、5 又は 6 の少なくとも一項に記載の方法のステップを実施するための命令を記憶した非一時的なコンピュータ可読記憶媒体。

【請求項 9】

コード化されている変換ブロックに対し、複数のコード化グループ間で割り当てられる前記バジェットが、通常の C A B A C コード化モードを使用してバイナリ要素がコード化される度にデクリメントされる、請求項 3 に記載のビデオ符号化装置。

【請求項 10】

復号されている変換ブロックに対し、複数のコード化グループ間で割り当てられる前記バジェットが、通常の C A B A C 復号モードを使用してバイナリ要素が復号される度にデクリメントされる、請求項 4 に記載のビデオ復号装置。

【発明の詳細な説明】

【技術分野】

【0001】

技術分野

本実施形態の少なくとも 1 つは、概して、ビデオ符号化又は復号のための残差コード化における通常のピン (bin) の割り当てに関する。

【背景技術】

【0002】

背景

高圧縮効率を達成するために、画像及びビデオコード化スキームは、通常、予測及び変換を採用してビデオコンテンツにおける空間的及び時間的冗長性を活用する。一般に、イントラ又はインターフレーム相互関係を利用するためにイントラ又はインター予測が使用され、次いで、オリジナルのブロックと予測ブロックとの差 (予測誤差又は予測残差として表される場合が多い) の変換、量子化及びエントロピーコード化が行われる。ビデオを再構築するために、圧縮データは、エントロピーコード化、量子化、変換及び予測に対する逆のプロセスによって復号される。

【発明の概要】

【0003】

概要

本実施形態の 1 つ又は複数は、高レベルの制約が尊重され、現在の手法と比べて圧縮効率が改善されるように、ブロック及びそれらの係数グループの残差コード化プロセスにおけるピンの通常の C A B A C コード化の最大使用量に対する高レベルの制約を取り扱う。

【0004】

10

20

30

40

50

少なくとも1つの実施形態の第1の態様によれば、ビデオ符号化方法は、限られた数の通常のピンを使用してコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを含み、コード化は、C A B A C符号化であり、通常のピンの数は、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される。

【0005】

少なくとも1つの実施形態の第2の態様によれば、ビデオ復号方法は、通常のピンを使用してコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを含み、復号は、C A B A C復号を使用して行われ、通常のピンの数は、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される。

【0006】

少なくとも1つの実施形態の第3の態様によれば、装置は、ピクチャ又はビデオの少なくとも1つのブロックに対するピクチャデータを符号化するためのエンコーダを含み、エンコーダは、限られた数の通常のピンを使用してコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを実行するように構成され、コード化は、C A B A C符号化であり、通常のピンの数は、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される。

【0007】

少なくとも1つの実施形態の第4の態様によれば、装置は、ピクチャ又はビデオの少なくとも1つのブロックに対するピクチャデータを復号するためのデコーダを含み、デコーダは、通常のピンを使用してコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを実行するように構成され、復号は、C A B A C復号を使用して行われ、通常のピンの数は、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される。

【0008】

少なくとも1つの実施形態の第5の態様によれば、プロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラムが提示され、コンピュータプログラムは、少なくとも第1又は第2の態様による方法のステップを実施する。

【0009】

少なくとも1つの実施形態の第6の態様によれば、非一時的なコンピュータ可読媒体上に格納され、プロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラム製品が提示され、コンピュータプログラム製品は、少なくとも第1又は第2の態様による方法のステップを実施する。

【図面の簡単な説明】

【0010】

図面の簡単な説明

【図1】高効率ビデオコード化（HEVC）エンコーダなどのビデオエンコーダ100の例のブロック図を示す。

【図2】HEVCデコーダなどのビデオデコーダ200の例のブロック図を示す。

【図3】様々な態様及び実施形態が実装されるシステムの例のブロック図を示す。

【図4A】圧縮領域におけるコード化木単位及びコード化木の例を示す。

【図4B】コード化単位、予測単位及び変換単位へのCTUの分割の例を示す。

【図5】依存性スカラ量子化における2つのスカラ量子化器の使用を示す。

【図6A】スカラ量子化器間で切り替えるためのメカニズムの例を示す。

【図6B】スカラ量子化器間で切り替えるためのメカニズムの例を示す。

【図6C】8×8変換ブロックのVVCにおいて使用されるようなCG間及び係数間の走査順序の例を示す。

【図7A】変換ブロックに対するコード化／解析のための構文要素の例を示す。

【図7B】CGレベルの残差コード化構文の例を示す。

【図8A】CUレベルの構文の例を示す。

【図8B】transform#tree構文構造の例を示す。

10

20

30

40

50

【図 8 C】変換単位レベルの構文配列を示す。

【図 9 A】C A B A C 復号プロセスを示す。

【図 9 B】C A B A C 符号化プロセスを示す。

【図 1 0】本原理の実施形態による、C A B A C オプティマイザを含む C A B A C 符号化プロセスを示す。

【図 1 1】本原理の実施形態による符号化プロセスにおいて使用される C A B A C オプティマイザのフローチャートの例を示す。

【図 1 2】C A B A C オプティマイザを使用してコード化グループを符号化 / 復号するための修正されたプロセスの例を示す。

【図 1 3 A】通常のピンのバジェットが変換ブロックレベルで決定される実施形態の例を示す。

10

【図 1 3 B】通常のピンのバジェットが最終有意係数の位置に従って変換ブロックレベルで決定される実施形態の例を示す。

【図 1 4 A】通常のピンのバジェットが変換単位レベルで決定される実施形態の例を示す。

【図 1 4 B】通常のピンのバジェットが、変換単位レベルで決定され、異なる変換ブロック間の相対表面の関数として及び考慮される変換単位の既にコード化 / 解析されている変換ブロックにおいて使用される通常のピンの関数として変換ブロック間で割り当てられる実施形態の例を示す。

【図 1 4 C】T B レベルの残差コード化 / 解析のためのプロセスの実施形態の例を示す。

【図 1 5 A】通常のピンのバジェットがコード化単位レベルで割り当てられる実施形態の例を示す。

20

【図 1 5 B】C U レベルで割り当てられた通常のピンのバジェットに基づく現在の C U と関連付けられた変換木のコード化 / 解析の実施形態の例を示す。

【図 1 5 C】バジェットが C U レベルで固定された際のコード化木コード化 / 解析プロセスの変形実施形態の例を示す。

【図 1 5 D】通常のピンのバジェットが C U レベルで固定された際の、本実施形態用に適応させた変換単位コード化 / 解析プロセスを示す。

【図 1 5 E】通常のピンのバジェットが C U レベルで固定された際の、本実施形態用に適応させた変換単位コード化 / 解析プロセスを示す。

【図 1 6 A】通常のピンのバジェットが C G レベルより高いレベルで割り当てられる実施形態の例を示し、第 1 のタイプの C G コード化 / 解析プロセスを使用するためのものである。

30

【図 1 6 B】通常のピンのバジェットが C G レベルより高いレベルで割り当てられる実施形態の例を示し、第 2 のタイプの C G コード化 / 解析プロセスを使用するためのものである。

【発明を実施するための形態】

【 0 0 1 1 】

詳細な説明

様々な実施形態は、量子化済みの変換係数のエントロピーコード化に関連する。ビデオコーデックのこの段階は、残差コード化ステップとも呼ばれる。少なくとも 1 つの実施形態は、単一のエリアあたりの最大数の通常のコード化ピンの制約の下で、ビデオコーデックのコード化効率を最適化することを目的とする。

40

【 0 0 1 2 】

この出願において説明される様々な方法及び他の態様は、図 1 及び 2 に示されるようなビデオエンコーダ 1 0 0 及びデコーダ 2 0 0 の少なくともエントロピーコード化及び / 又は復号モジュール (1 4 5、2 3 0) を修正するために使用することができる。その上、本態様は、V V C (多目的ビデオコード化) 又は H E V C (高効率ビデオコード化) 仕様の特定の草稿に関連する原理を説明しているが、V V C 又は H E V C に限定されず、例えば、先在するか又は今後開発されるかにかかわらず、他の規格及び勧告並びにそのような規格及び勧告の拡張版 (V V C 及び H E V C を含む) に適用することができる。別段の指

50

示がない限り又は技術的に除外されない限り、この出願において説明される態様は、個別に使用することも、組み合わせて使用することもできる。

【0013】

図1は、HEVCエンコーダなどのビデオエンコーダ100の例のブロック図を示す。また、図1は、HEVC規格に対する改善が行われたエンコーダ、又は、VVCにおける共同ビデオ探査チーム(JVET: Joint Video Exploration Team)による開発の下での共同探査モデル(JEM: Joint Exploration Model)若しくはVVCテストモデル(VTM: VVC Test Model)エンコーダなどのHEVCと同様の技術を採用するエンコーダも示す。

【0014】

符号化される前、ビデオシーケンスは、符号化前の処理(101)を行うことができる。例えば、この処理は、入力カラーピクチャに色変換を適用することによって(例えば、RGB 4:4:4からYCbCr 4:2:0への変換)、又は、圧縮に対する回復力がより早い信号分布を得るために入力ピクチャ成分のリマッピングを実行することによって(例えば、色成分のうちの1つのヒストグラム平坦化を使用する)実行される。メタデータは、前処理と関連付け、ビットストリームに載せることができる。

【0015】

HEVCでは、1つ又は複数のピクチャを有するビデオシーケンスを符号化するため、ピクチャは、1つ又は複数のスライスに区別化され(102)、各スライスは、1つ又は複数のスライスセグメントを含み得る。スライスセグメントは、コード化単位、予測単位及び変換単位に整理される。HEVC仕様は、「ブロック」と「単位」とを区別し、「ブロック」は、サンプルアレイの特定のエリア(例えば、ルマ、Y)に対処し、「単位」は、ブロック(例えば、動きベクトル)と関連付けられたすべての符号化済みの色成分(Y、Cb、Cr又はモノクロ)、構文要素及び予測データの配列ブロックを含む。

【0016】

HEVCでコード化するため、ピクチャは、構成可能なサイズを有する正方形のコード化木ブロック(CTB)に区別化され、コード化木ブロックの連続セットは、スライスにグループ分けされる。コード化木単位(CTU)は、符号化済みの色成分のCTBを内包する。CTBは、コード化ブロック(CB)に区別化された四分木のルートであり、コード化ブロックは、1つ又は複数の予測ブロック(PB)に区別化することができ、変換ブロック(TB)に区別化された四分木のルートを形成する。コード化ブロック、予測ブロック及び変換ブロックに対応して、コード化単位(CU)は、予測単位(PU)及び変換単位(TU)の木構造セットを含み、PUは、すべての色成分に対する予測情報を含み、TUは、各色成分に対する残差コード化構文構造を含む。ルマ成分のCB、PB及びTBのサイズは、対応するCU、PU及びTUに当てはまる。本出願では、「ブロック」という用語は、例えば、CTU、CU、PU、TU、CB、PB及びTBのいずれかを指すために使用することができる。それに加えて、「ブロック」は、H.264/AVC又は他のビデオコード化規格において指定されるようなマクロブロック及びパーティションを指すために使用することもでき、より一般的には、様々なサイズのデータのアレイを指すために使用することもできる。

【0017】

エンコーダ100の例では、ピクチャは、以下で説明されるようなエンコーダ要素によって符号化される。符号化予定のピクチャは、CUの単位で処理される。各CUは、イントラ又はインターモードを使用して符号化される。CUがイントラモードで符号化される際は、イントラ予測(160)が実行される。インターモードでは、動き推定(175)及び動き補償(170)が実行される。エンコーダは、CUの符号化に対してイントラモードとインターモードのどちらを使用するかを決定し(105)、予測モードフラグによって、イントラ/インター決定を示す。予測残差は、オリジナルの画像ブロックから予測ブロックを減じること(110)によって計算される。

【0018】

10

20

30

40

50

イントラモードにおけるCUは、同じスライス内の再構築された近隣のサンプルから予測される。HEVCでは、35個のイントラ予測モードのセットが利用可能であり、1つのDCモード、1つの平面モード及び33個の角度予測モードを含む。イントラ予測参照は、現在のブロックに隣接する行及び列から再構築される。参照は、以前に再構築されたブロックから利用可能なサンプルを使用して、水平及び垂直方向にブロックサイズの2倍以上に及ぶ。イントラ予測に対して角度予測モードが使用される際は、参照サンプルは、角度予測モードによって示される方向に沿ってコピーすることができる。

【0019】

現在のブロックに対して適用可能なルマイントラ予測モードは、2つの異なるオプションを使用してコード化することができる。適用可能なモードが3つの最確モード(MPM)の構築リストに含まれる場合は、モードは、MPMリストのインデックスによってシグナルされる。そうでなければ、モードは、モードインデックスの固定長二進化によってシグナルされる。3つの最確モードは、上隣及び左隣のブロックのイントラ予測モードから導出される。

10

【0020】

インターCUの場合は、対応するコード化ブロックは、1つ又は複数の予測ブロックにさらに区分化される。インター予測は、PBレベルで実行され、対応するPUは、インター予測がどのように実行されるかについての情報を内包する。動き情報(例えば、動きベクトル及び参照ピクチャインデックス)は、2つの方法、すなわち、「マージモード」及び「高度な動きベクトル予測(AMVP)」でシグナルすることができる。

20

【0021】

マージモードでは、ビデオエンコーダ又はデコーダは、既にコード化されているブロックに基づいて候補リストを組み立て、ビデオエンコーダは、候補リスト内の候補のうちの1つに対するインデックスをシグナルする。デコーダ側では、動きベクトル(MV)及び参照ピクチャインデックスは、シグナルされた候補に基づいて再構築される。

【0022】

AMVPでは、ビデオエンコーダ又はデコーダは、既にコード化されているブロックから決定された動きベクトルに基づいて候補リストを組み立てる。次いで、ビデオエンコーダは、動きベクトル予測因子(MVP)を識別するために候補リストのインデックスをシグナルし、動きベクトル差(MVD)をシグナルする。デコーダ側では、動きベクトル(MV)は、 $MVP + MVD$ として再構築される。また、適用可能な参照ピクチャインデックスは、AMVPに対するPU構文において明示的にコード化される。

30

【0023】

次いで、以下で説明されるクロマ量子化パラメータを適応させるための少なくとも1つの実施形態を含めて、予測残差の変換(125)及び量子化(130)が行われる。変換は、一般に、分離可能な変換に基づく。例えば、DCT変換は、最初に、水平方向に適用され、次いで、垂直方向に適用される。以前のコーデックでは、所定のブロックサイズに対する多様な2D変換は通常は限定されるが、JEMなどの最近のコーデックでは、両方向において使用される変換は異なり得(例えば、一方の方向ではDCT、他方ではDST)、それにより、多種多様な2D変換が可能になる。

40

【0024】

量子化済みの変換係数並びに動きベクトル及び他の構文要素は、ビットストリームを出力するために、エントロピーコード化(145)される。また、エンコーダは、変換をスキップして、 4×4 TUベースで非変換残差信号に量子化を直接適用することもできる。また、エンコーダは、変換と量子化の両方を回避することもできる(すなわち、残差は、変換プロセスも量子化プロセスも適用することなく直接コード化される)。直接PCMコード化では、予測は適用されず、コード化単位サンプルは、直接コード化され、ビットストリームに埋め込まれる。

【0025】

エンコーダは、さらなる予測に対する参照を提供するために、符号化済みのブロックを

50

復号する。量子化済みの変換係数は、予測残差を復号するために、逆量子化（140）及び逆変換（150）が行われる。復号済みの残差と予測ブロックを組み合わせること（155）で、画像ブロックが再構築される。インループフィルタ（165）は、例えば、符号化アーチファクトを低減するためにデブロッキング/SAO（サンプル適応オフセット）フィルタリングを実行するため、再構築されたピクチャに適用される。フィルタリング済みの画像は、参照ピクチャバッファ（180）に格納される。

【0026】

図2は、HEVCデコーダなどのビデオデコーダ200の例のブロック図を示す。デコーダ200の例では、ビットストリームは、以下で説明されるようなデコーダ要素によって復号される。ビデオデコーダ200は、一般に、ビデオデータの符号化の一部としてビデオ復号を実行する図1において説明されるような符号化パスとは逆の復号パスを実行する。また、図2は、HEVC規格に対する改善が行われたデコーダ、又は、JEM又はVVCデコーダなどのHEVCと同様の技術を採用するデコーダも示す。

10

【0027】

具体的には、デコーダの入力は、ビデオビットストリームを含み、ビデオビットストリームは、ビデオエンコーダ100によって生成することができる。ビットストリームは、最初に、変換係数、動きベクトル、ピクチャ区分化情報及び他のコード化情報を得るために、エントローピー復号される（230）。ピクチャ区分化情報は、CTUのサイズや、CTUがCUに分けられた方法を示し、適用可能な場合は、恐らくはPUに分けられた方法を示す。従って、デコーダは、復号済みのピクチャ区分化情報に従って、ピクチャをCTUに分割し、各CTUをCUに分割すること（235）ができる。変換係数は、予測残差を復号するために、以下で説明されるクロマ量子化パラメータを適応させるための少なくとも1つの実施形態を含めて、逆量子化（240）及び逆変換（250）が行われる。

20

【0028】

復号済みの予測残差と予測ブロックを組み合わせること（255）で、画像ブロックが再構築される。予測ブロックは、イントラ予測（260）又は動き補償予測（すなわち、インター予測）（275）から得ることができる（270）。上記で説明されるように、動き補償に対する動きベクトルを導出するためにAMVP及びマージモード技法を使用することができ、動き補償では、参照ブロックのサブ整数サンプルに対する補間値を計算するために補間フィルタを使用することができる。インループフィルタ（265）は、再構築されたピクチャに適用される。フィルタリング済みの画像は、参照ピクチャバッファ（280）で格納される。

30

【0029】

復号済みのピクチャは、復号後の処理（285）をさらに行うことができ、例えば、逆色変換（例えば、YCbCr 4:2:0からRGB 4:4:4への変換）又は符号化前の処理（101）において実行されるリマッピングプロセスの逆を実行する逆リマッピングを行うことができる。復号後の処理は、メタデータを使用することができ、メタデータは、符号化前の処理において導出され、ビットストリームでシグナルされる。

【0030】

図3は、様々な態様及び実施形態が実装されるシステムの例のブロック図を示す。システム300は、以下で説明される様々なコンポーネントを含むデバイスとして具体化することができる。この出願において説明される態様のうちの1つ又は複数を実行するように構成される。そのようなデバイスの例は、これらに限定されないが、パーソナルコンピュータ、ラップトップコンピュータ、スマートフォン、タブレットコンピュータ、デジタルマルチメディアセットトップボックス、デジタルテレビレシーバ、パーソナルビデオ記録システム、接続された家庭用器具、エンコーダ、トランスコーダ及びサーバなどの様々な電子デバイスを含む。システム300の要素は、単独で又は組み合わせて、単一の集積回路、複数のIC及び/又は離散コンポーネントにおいて具体化することができる。例えば、少なくとも1つの実施形態では、システム300の処理及びエンコーダ/デコーダ要素は、複数のIC及び/又は離散コンポーネントにわたって分散される。様々な実施形態では

40

50

、システム 300 の要素は、内部バス 310 を通じて通信可能に結合される。様々な実施形態では、システム 300 は、例えば、通信バスを介して或いは専用入力及び / 又は出力ポートを通じて、他の同様のシステム又は他の電子デバイスに通信可能に結合される。様々な実施形態では、システム 300 は、上記で説明される及び以下で説明されるように修正されるビデオエンコーダ 100 及びビデオデコーダ 200 など、この文書において説明される態様のうちの 1 つ又は複数を実装するように構成される。

【0031】

システム 300 は、例えば、この文書において説明される様々な態様を実装するためにその中にロードされた命令を実行するように構成された少なくとも 1 つのプロセッサ 301 を含む。プロセッサ 301 は、当技術分野で知られているような埋め込みメモリ、入力出力インタフェース及び様々な他の回路を含み得る。システム 300 は、少なくとも 1 つのメモリ 302 (例えば、揮発性メモリデバイス及び / 又は不揮発性メモリデバイス) を含む。システム 300 は、不揮発性メモリ及び / 又は揮発性メモリを含み得る記憶装置 304 を含み、これらに限定されないが、EEPROM、ROM、PROM、RAM、DRAM、SRAM、フラッシュ、磁気ディスクドライブ及び / 又は光ディスクドライブを含む。記憶装置 304 は、非限定的な例として、内部記憶装置、取り付けられた記憶装置及び / 又はネットワークアクセス可能記憶装置を含み得る。

10

【0032】

システム 300 は、例えば、符号化済みのビデオ又は復号済みのビデオを提供するためにデータを処理するように構成されたエンコーダ / デコーダモジュール 303 を含み、エンコーダ / デコーダモジュール 303 は、それ自体のプロセッサ及びメモリを含み得る。エンコーダ / デコーダモジュール 303 は、符号化及び / 又は復号機能を実行するためにデバイスに含めることができるモジュールを表す。知られているように、デバイスは、符号化及び復号モジュールの一方又は両方を含み得る。それに加えて、エンコーダ / デコーダモジュール 303 は、当業者に知られているように、システム 300 の別個の要素として実装することも、ハードウェアとソフトウェアの組合せとしてプロセッサ 301 内に組み込むこともできる。

20

【0033】

この文書において説明される様々な態様を実行するためにプロセッサ 301 又はエンコーダ / デコーダ 303 にロードされる予定のプログラムコードは、記憶装置 304 に格納し、その後、プロセッサ 301 による実行のためにメモリ 302 にロードすることができる。様々な実施形態によれば、プロセッサ 301、メモリ 302、記憶装置 304 及びエンコーダ / デコーダモジュール 303 のうちの 1 つ又は複数は、この文書において説明されるプロセスの実行の間、様々なアイテムのうちの 1 つ又は複数に格納することができる。そのような格納されるアイテムは、これらに限定されないが、入力ビデオ、復号済みのビデオ又は復号済みのビデオの一部、ビットストリーム、行列、変数、並びに、方程式、公式、演算及び演算論理の処理からの中間又は最終結果を含み得る。

30

【0034】

いくつかの実施形態では、命令を格納するため及び符号化又は復号の間に必要なものを処理するためのワーキングメモリを提供するため、プロセッサ 301 及び / 又はエンコーダ / デコーダモジュール 303 の内部のメモリが使用される。しかし、他の実施形態では、これらの機能のうちの 1 つ又は複数に対して、処理デバイス (例えば、処理デバイスは、プロセッサ 301 又はエンコーダ / デコーダモジュール 303 であり得る) の外部のメモリが使用される。外部のメモリは、メモリ 302 及び / 又は記憶装置 304 であり得、例えば、動的揮発性メモリ及び / 又は不揮発性フラッシュメモリであり得る。いくつかの実施形態では、外部の不揮発性フラッシュメモリは、テレビのオペレーティングシステムを格納するために使用される。少なくとも 1 つの実施形態では、RAM などの外部の高速動的揮発性メモリは、MPEG-2、HVC 又は VVC に対するものなどのビデオコード化及び復号動作のためのワーキングメモリとして使用される。

40

【0035】

50

システム 300 の要素への入力は、ブロック 309 に示されるような様々な入力デバイスを通じて提供することができる。そのような入力デバイスは、これらに限定されないが、(i) 例えば放送局によって無線で送信された RF 信号を受信する RF 部分、(i i) 複合入力端末、(i i i) USB 入力端末及び / 又は (i v) HDMI 入力端末を含む。

【 0036 】

様々な実施形態では、ブロック 309 の入力デバイスは、当技術分野で知られているようなそれぞれの関連入力処理要素を有する。例えば、RF 部分は、(i) 所望の周波数を選択すること (信号を選択すること又は周波数の帯域に信号を帯域制限することとも呼ばれる)、(i i) 選択された信号をダウンコンバートすること、(i i i) 信号周波数帯域 (ある実施形態では、チャンネルとも呼ぶことができる) を選択するために (例えば)、周波数のより狭い帯域に再び帯域制限すること、(i v) ダウンコンバート済みの且つ帯域制限済みの信号を復調すること、(v) 誤り訂正を実行すること、及び、(v i) データパケットの所望のストリームを選択するために逆多重化することに対して必要な要素と関連付けることができる。様々な実施形態の RF 部分は、これらの機能を実行するための 1 つ又は複数の要素を含み、例えば、周波数セレクタ、信号セレクタ、バンドリミッタ、チャンネルセレクタ、フィルタ、ダウンコンバータ、復調器、誤り訂正器及びデマルチプレクサが挙げられる。RF 部分は、チューナを含み得、チューナは、例えば、より低い周波数 (例えば、中間周波数若しくはベースバンド近くの周波数) への又はベースバンドへの受信信号のダウンコンバートを含む、これらの様々な機能を実行する。セットトップボックスの一実施形態では、RF 部分及びその関連入力処理要素は、有線 (例えば、ケーブル) 媒体上で送信された RF 信号を受信し、次いで、所望の周波数帯域を得るために、フィルタリングを行い、ダウンコンバートを行い、再びフィルタリングを行うことによって周波数選択を実行する。様々な実施形態は、上記で説明される (及び他の) 要素の順序を再構成すること、これらの要素のいくつかを取り除くこと及び / 又は同様の若しくは異なる機能を実行する他の要素を追加することを行う。要素を追加することは、既存の要素の間に要素を挿入すること (例えば、増幅器及びアナログ / デジタル変換器を挿入することなど) を含み得る。様々な実施形態では、RF 部分はアンテナを含む。

【 0037 】

それに加えて、USB 及び / 又は HDMI 端末は、USB 及び / 又は HDMI 接続を通じてシステム 300 を他の電子デバイスに接続するためのそれぞれのインタフェースプロセッサを含み得る。入力処理の様々な態様 (例えば、リードソロン誤り訂正) は、例えば、必要に応じて、別個の入力処理 IC 内又はプロセッサ 301 内で実装できることを理解されたい。同様に、USB 又は HDMI インタフェース処理の態様は、必要に応じて、別個のインタフェース IC 内又はプロセッサ 301 内で実装することができる。復調済みの、誤り訂正済みの及び逆多重化済みのストリームは、必要に応じて出力デバイス上で提示するためのデータストリームを処理するために、様々な処理要素 (例えば、メモリ及び記憶素子と連動するプロセッサ 301 及びエンコーダ / デコーダ 303 を含む) に提供される。

【 0038 】

システム 300 の様々な要素は、統合筐体内に提供することができる。統合筐体内では、様々な要素を相互接続することができ、例えば、当技術分野で知られている内部バス (I2C バス、配線及びプリント基板を含む) などの適切な接続配列を使用して、様々な要素間でデータを送信することができる。

【 0039 】

システム 300 は、通信チャンネル 320 を介して他のデバイスとの通信を可能にする通信インタフェース 305 を含む。通信インタフェース 305 は、これらに限定されないが、通信チャンネル 320 上でデータを送信及び受信するように構成されたトランシーバを含み得る。通信インタフェース 305 は、これらに限定されないが、モデム又はネットワークカードを含み得、通信チャンネル 320 は、例えば、有線及び / 又は無線媒体内で実装することができる。

【 0040 】

10

20

30

40

50

データは、様々な実施形態では、IEEE 802.11などのWi-Fiネットワークを使用して、システム300にストリーミングされる。これらの実施形態のWi-Fi信号は、Wi-Fi通信に適応させた通信チャネル320及び通信インタフェース305上で受信される。これらの実施形態の通信チャネル320は、典型的には、ストリーミングアプリケーション及び他のオーバーザトップ通信を可能にするためのインターネットを含む外部のネットワークへのアクセスを提供するアクセスポイント又はルータに接続される。他の実施形態は、入力ブロック309のHDMI接続上でデータを伝達するセットトップボックスを使用して、ストリーミングされたデータをシステム300に提供する。さらなる他の実施形態は、入力ブロック309のRF接続を使用して、ストリーミングされたデータをシステム300に提供する。

【0041】

システム300は、ディスプレイ330、スピーカ340及び他の周辺デバイス350を含む様々な出力デバイスに出力信号を提供することができる。他の周辺デバイス350は、実施形態の様々な例では、スタンドアロンDVR、ディスクプレーヤ、ステレオシステム、照明システム、及び、システム300の出力に基づいて機能を提供する他のデバイスのうちの1つ又は複数を含む。様々な実施形態では、制御信号は、AV.Link、CEC、又は、ユーザ介入の有無にかかわらずデバイス間制御を可能にする他の通信プロトコルなどのシグナリングを使用して、システム300とディスプレイ330、スピーカ340又は他の周辺デバイス350との間で伝達される。出力デバイスは、専用接続を介して、それぞれのインタフェース306、307及び308を通じて、システム300に通信可能に結合することができる。或いは、出力デバイスは、通信チャネル320を使用して、通信インタフェース305を介して、システム300に接続することができる。ディスプレイ330及びスピーカ340は、電子デバイス(例えば、テレビなど)のシステム300の他のコンポーネントと共に単一のユニットに統合することができる。様々な実施形態では、ディスプレイインタフェース306は、例えば、タイミングコントローラ(TCon)チップなどのディスプレイドライバを含む。

【0042】

例えば、入力309のRF部分が別個のセットトップボックスの一部である場合は、ディスプレイ330及びスピーカ340は、他のコンポーネントのうちの1つ又は複数から選択的に分離することができる。ディスプレイ330及びスピーカ340が外部のコンポーネントである様々な実施形態では、出力信号は、専用出力接続(例えば、HDMIポート、USBポート又はCOMP出力を含む)を介して提供することができる。本明細書で説明される実装形態は、例えば、方法若しくはプロセス、装置、ソフトウェアプログラム、データストリーム、又は、信号で実装することができる。実装の単一の形態の文脈においてのみ論じられる(例えば、方法としてのみ論じられる)場合であっても、論じられる特徴の実装形態は、他の形態(例えば、装置又はプログラム)でも実装することができる。装置は、例えば、適切なハードウェア、ソフトウェア及びファームウェアで実装することができる。方法は、例えば、コンピュータ、マイクロプロセッサ、集積回路又はプログラマブル論理デバイスを含む、例えば、一般に処理デバイスを指す装置(例えば、プロセッサなど)で実装することができる。また、プロセッサは、例えば、コンピュータ、携帯電話、ポータブル/携帯情報端末(「PDA」)、及び、エンドユーザ間の情報の通信を容易にする他のデバイスなどの通信デバイスも含む。

【0043】

図4Aは、圧縮領域におけるコード化木単位及びコード化木の例を示す。HEVCビデオ圧縮規格では、動き補償時間予測は、ビデオの連続ピクチャ間に存在する冗長性を利用するために採用される。その達成のため、ピクチャは、いわゆるコード化木単位(CTU)に区分化され、そのサイズは、典型的には、 64×64 、 128×128 又は 256×256 画素である。各CTUは、圧縮領域(例えば、CTUの四分木分割)におけるコード化木によって表される。各リーフは、コード化単位(CU)と呼ばれる。

【0044】

10

20

30

40

50

図 4 B は、コード化単位、予測単位及び変換単位への C T U の分割の例を示す。各 C U には、次いで、いくつかのイントラ又はインター予測パラメータ予測情報) が与えられる。その達成のため、各 C U は、1 つ又は複数の予測単位 (P U) に空間的に区分化され、各 P U には、動きベクトルなどのいくつかの予測情報が割り当てられる。イントラ又はインターコード化モードは、C U レベルで割り当てられる。

【 0 0 4 5 】

本出願では、「再構築される」及び「復号される」という用語は、交換可能に使用することができ、「符号化される」及び「コード化される」という用語は、交換可能に使用することができ、「画像」、「ピクチャ」及び「フレーム」という用語は、交換可能に使用することができる。通常、一概には言えないが、「再構築される」という用語は、エンコーダ側で使用され、「復号される」という用語は、デコーダ側で使用される。「ブロック」又は「ピクチャブロック」という用語は、C T U、C U、P U、T U、C B、P B 及び T B のいずれか 1 つを指すために使用することができる。それに加えて、「ブロック」又は「ピクチャブロック」という用語は、H . 2 6 4 / A V C 又は他のビデオコード化規格において指定されるようなマクロブロック、パーティション及びサブブロックを指すために使用することができ、より一般的には、多くのサイズのサンプルのアレイを指すために使用することができる。

【 0 0 4 6 】

図 5 は、依存性スカラ量子化における 2 つのスカラ量子化器の使用を示す。依存性スカラ量子化は、J V E T (寄稿 J V E T - J 0 0 1 4) において提案されるように、量子化に対する異なる再構築レベルを有する 2 つのスカラ量子化器を使用する。従来のスカラ量子化 (例えば、H E V C 及び V T M - 1 において使用されるような) と比べて、この手法の主効果は、変換係数の許容再構築値のセットが、再構築順序において現在の変換係数レベルに先行する変換係数レベルの値に依存することである。依存性スカラ量子化の手法は、(a) 異なる再構築レベルを有する 2 つのスカラの量子化器を定義することと、(b) 2 つのスカラの量子化器間で切り替えるためのプロセスを定義することとによって実現される。使用される 2 つのスカラの量子化器は、図 5 に示される (Q 0 及び Q 1 によって表される) 。利用可能な再構築レベルの場所は、量子化ステップサイズ によって一意に指定される。変換係数の実際の再構築が整数演算を使用するという事実を無視すると、2 つのスカラの量子化器 Q 0 及び Q 1 は、以下の通り特徴付けられる。

Q 0 : 第 1 の量子化器 Q 0 の再構築レベルは、量子化ステップサイズ の偶数の整数倍によって与えられる。この量子化器が使用される際は、逆量子化済みの変換係数 t' は、 $t' = 2 \cdot k$?

に従って計算され、式中、 k は、関連する量子化済みの係数 (送信された量子化インデックス) を表す。

Q 1 : 第 2 の量子化器 Q 1 の再構築レベルは、ゼロに等しい再構築レベルに加えて、量子化ステップサイズ の奇数の整数倍によって与えられる。逆量子化済みの変換係数 t' は、

$$t' = (2 \cdot k - \text{sgn}(k)) ?$$

のように、量子化済みの係数 k の関数として演算され、式中、 $\text{sgn}(\cdot)$ は、

$$\text{sgn}(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$$

として定義される符号関数である。

【 0 0 4 7 】

使用されるスカラ量子化器 (Q 0 又は Q 1) は、ビットストリームで明示的にシグナルされるものではない。代わりに、現在の変換係数に対して使用される量子化器は、コード化 / 再構築順序において現在の変換係数に先行する量子化済みの係数のパリティ及び以下で紹介される有限状態機械の状態によって決定される。

【 0 0 4 8 】

図 6 A 及び 6 B は、V V C においてスカラ量子化器間で切り替えるためのメカニズムの例を示す。2 つのスカラ量子化器 (Q 0 と Q 1) 間の切り替えは、図 6 A に示されるよう

10

20

30

40

50

に、4つの状態(0、1、2又は3としてそれぞれラベル付けされる)を有する有限状態機械を介して実現される。所定の量子化済みの係数に対して考慮される有限状態機械の状態は、コード化/再構築順序において現在の量子化済みの係数に先行する量子化済みの係数kのパリティ及びこの先行する係数を処理する際に考慮される有限状態機械の状態によって一意に決定される。変換ブロックに対する逆量子化の開始時、状態は、0に等しく設定される。変換係数は、走査順(すなわち、エントロピー復号される順序と同じ順序)に再構築される。現在の変換係数が再構築された後、状態は更新される。kは、量子化済みの係数である。次の状態は、

$state = stateTransTable[current\#state][k\&1]$

のように現在の状態及び現在の量子化済みの係数kのパリティ(k&1)に依存し、式中、stateTransTableは、図6Bに示される状態遷移表を表し、演算子&は、2の補数演算におけるビット「積」演算子を指定する。

10

【0049】

その上、いわゆる変換ブロック(TB)に含まれる量子化済みの係数は、以下で説明されるようにエントロピーコード化及び復号することができる。

【0050】

最初に、変換ブロックは、コード化グループと呼ばれる(場合により、係数グループとも名付けられ、CGと略される)量子化済みの係数の4×4サブブロックに分割される。エントロピーコード化/復号は、いくつかの走査パスから作られており、走査パスは、図6Cによって示される対角方向走査順序に従ってTBを走査する。

20

【0051】

VVCにおける変換係数コード化は、5つの主要なステップ、すなわち、走査、最終有意係数コード化、有意性マップコード化、係数レベル残余コード化、絶対レベル及び符号データコード化を伴う。

【0052】

図6Cは、8×8変換ブロックのVVCにおいて使用されるようなCG間及び係数間の走査順序の例を示す。TBにわたる走査パスは、対角方向走査順序に従って順次に各CGを処理することを含み、各CG内の16個の係数も同様に、考慮される走査順序に従って走査される。走査パスは、TBの最終有意係数から始まり、DC係数に至るまですべての係数を処理する。

30

【0053】

変換係数のエントロピーコード化は、以下のリスト内の最大で7つの構文要素を含む。

- coded#sub#block#flag: 係数グループ(CG)の有意性
- sig#flag: 係数の有意性(ゼロ/非ゼロ)
- gt1#flag: 係数レベルの絶対値が1より大きいかどうかを示す
- par#flag: 1より大きい係数のパリティを示す
- gt3#flag: 係数レベルの絶対値が3より大きいかどうかを示す
- remainder: 係数レベルの絶対値に対する残存値(以前のパスにおいてコード化されたものより値が大きい場合)

- abs#level: 係数レベルの絶対値の値(最大数のピンバジェット問題に対する現在の係数に対してCABACピンがシグナルされていない場合)

40

- sign#data: 考慮されるCGに含まれるすべての有意係数の符号。それは、一連のピンを含み、各々は、各非ゼロ変換係数の符号(0:正、1:負)をシグナルする。

【0054】

上記の要素(符号は別として)のサブセットを復号することによって量子化済みの係数の絶対値が分かった時点で、その絶対値に関するその係数に対するさらなる構文要素のコード化は行われなくなる。同じように、符号フラグは、非ゼロ係数に対してのみシグナルされる。

【0055】

所定のCGに必要なすべての走査パスは、次のCGに進む前に、そのCGのすべての量

50

子化済みの係数を再構築できるまでコード化される。

【 0 0 5 6 】

全体的な復号 T B 解析プロセスは、以下の主要なステップから作られている。

1 . 最終有意係数座標を復号する。これは、last#sig#coeff#x#prefix、last#sig#coeff#y#prefix、last#sig#coeff#x#suffix及びlast#sig#coeff#y#suffixという構文要素を含む。これにより、T B 全体における最終非ゼロ係数の空間位置 (x 座標及び y 座標) がデコーダに提供される。

【 0 0 5 7 】

次いで、T B の最終有意係数を含む C G から T B の左上の C G までの連続する C G ごとに、以下のステップが適用される。

2 . C G 有意性フラグを復号する (V V C 仕様においてcoded#sub#block#flagと呼ばれる)

3 . 考慮される C G の各係数に対する有意係数フラグを復号する。これは、sig#flag という構文要素に相当する。これは、C G のどの係数が非ゼロであることを示す。

【 0 0 5 8 】

次の解析段階は、考慮される C G の、非ゼロとして知られている係数に対する、係数レベルに関連する。これは、以下の構文要素に関連する。

4 . gt1#flag : このフラグは、現在の係数の絶対値が 1 より大きいかなを示す。1 より大きくない場合は、絶対値は 1 に等しい。

5 . par#flag : このフラグは、現在の量子化済みの係数が偶数かなを示す。現在の量子化済みの係数のgt1#flagが真である場合は、コード化される。par#flagがゼロである場合は、量子化済みの係数は偶数であり、そうでない場合は、量子化済みの係数は奇数である。par#flagがデコーダ側で解析された後、部分的に復号された量子化済みの係数は、 $(1+gt1\#flag+par\#flag)$ に等しく設定される。

6 . gt3#flag : このフラグは、現在の係数の絶対値が 3 より大きいかなを示す。3 より大きくない場合は、 $1+gt1\#flag+par\#flag$ に等しい場合は、絶対値、 $1+gt1\#flag+par\#flag$ が 2 以上である場合は、gt3#flagはコード化される。gt3#flagが解析された時点で、量子化済みの係数値は、デコーダ側で $1+gt1\#flag+par\#flag+(2^{gt3\#flag})$ になる。

7 . remainder : これは、係数の絶対値を符号化する。これは、部分的に復号された絶対値が 4 より等しいものより大きい場合に当てはまる。V V C 草稿 3 の例では、各コード化グループに対して、通常のコード化ピンのバジレットの最大数は固定されることに留意されたい。従って、いくつかの係数に対しては、sig#flag、gt1#flag及びpar#flagという要素のみをシグナルできる一方で、他の係数に対しては、gt3#flagもシグナルすることができる。従って、コード化済み及び解析済みのremainderの値は、考慮される係数に対して既に復号されているフラグに対して演算され、従って、部分的に復号された量子化済みの係数の関数として演算される。

8 . abs#level : これは、通常のコード化ピンの最大数の問題に対する、考慮される C G のどのフラグ (sig#flag、gt1#flag、par#flag又はgt3#flagの間) もコード化されていない係数の絶対値を示す。この構文要素は、remainderという構文要素と同様に、ライス - ゴロム二進化及びバイパスコード化が行われる。

9 . sign#flag : これは、非ゼロ係数の符号を示す。これは、バイパスコード化される。

【 0 0 5 9 】

図 7 A は、V V C 草稿 4 による変換ブロックに対するコード化 / 解析のための構文要素を示す。コード化 / ペアリングは、4 パスプロセスを含む。図から分かるように、所定の変換ブロックに対し、最終有意係数の位置が最初にコード化される。次いで、最終有意係数を含む C G (この C G を除く) 及び第 1 の C G からの各 C G に対し、C G の有意性がシグナルされ (coded#sub#block#flag) 、C G が有意である事例では、図 7 B によって示されるように、その C G に対する残差コード化が行われる。

【 0 0 6 0 】

図 7 B は、V V C 草稿 4 において指定されるような C G レベル残差コード化構文を示す

10

20

30

40

50

。その構文は、以前に紹介した考慮されるCGのsig#flag、gt1#flag、par#flag、gt3#flag、remainder、abs#level及びsign#dataという構文要素をシグナルする。図は、VV C草稿4による、sig#flag、gt1#flag、par#flag、gt3#flag、remainder及びabs#levelという構文要素のシグナリング及び解析を示す。EPは、「等確率の(equi-probable)」を意味し、該当ピンは算術的にはコード化されないが、バイパスモードでコード化されることを意味する。バイパスモードは、ビットの直接的な書き込み/解析を行い、それは、一般に、符号化又は解析を希望するバイナリ構文要素(ピン)に等しい。

【0061】

その上、VV C草稿4は、CUレベルから残差サブブロック(CG)レベルに至るまでの階層的な構文配列を指定する。

【0062】

図8Aは、VV C草稿4の例において指定されるようなCUレベル構文を示す。その構文は、考慮されるCUのコード化モードをシグナルすることを含む。cu#skip#flagは、CUがマージスキップモードである場合にシグナルする。CUがマージスキップモードではない場合は、cu#pred#mode#flagは、cuPredModeの変数値を示し、従って、現在のCUの予測モードがイントラ予測(cuPredMode=MODE#INTRA)又はインター予測(cuPredMode=MODE#INTER)を通じてコード化される場合は、

【0063】

非スキップ及び非イントラモードの事例では、次のフラグは、現在のCUに対するイントラブロックコピー(IBC)モードの使用を示す。構文の続きは、現在のCUのイントラ又はインター予測データを含む。構文の最後は、現在のCUと関連付けられた変換木のシグナリング専用である。この変換木のシグナリングは、cu#cbfという構文要素から始まり、それは、いくつかの非ヌル残差データが現在のCUに対してコード化されることを示す。このフラグが真に等しい事例では、図8Bによって示される構文配列に従って、現在のCUと関連付けられたtransform#treeがシグナルされる。

【0064】

図8Bは、VV C草稿4の例のtransform#treeという構文構造を示す。その構文は、基本的に、CUが1つ又はいくつかの変換単位(TU)を内包する場合にシグナルすることを含む。最初に、CUサイズが最大許容TUサイズより大きい場合は、CUは、四分木方式で4つのサブ変換木に分割される。そうでない場合及び現在のCUに対してISP(イントラサブ区分化)又はSBT(サブブロック変換)が使用されない場合は、現在の変換木は区分化されず、まさに1つのTUを内包し、図8Cの変換単位構文表を通じてシグナルされる。そうでなければ、CUがイントラモードであり、ISP(イントラサブ区分化)モードが使用される場合は、CUは、いくつか(実際には、2つ)のTUに分けられる。2つのTUの各々は、図8Cの変換単位構文表を通じて相次いでシグナルされる。そうでなければ、CUがインターモードであり、SBT(サブブロック変換)モードが使用される場合は、CUは、CUレベルのSBT関連の復号済みの構文に従って2つのTUに分けられる。2つのTUの各々は、図8Cの変換単位構文表を通じて相次いでシグナルされる。

【0065】

図8Cは、VV C草稿4の例における変換単位レベル構文配列を示す。その構文は、以下のものから作られている。最初に、tu#cbf#luma、tu#cbf#cb及びtu#cbf#crフラグはそれぞれ、各成分に対応する非ヌル残差データが現在のTUの各変換ブロックに内包されていることを示す。各成分に対し、対応するcbfフラグが真である場合は、対応する残差変換ブロックをコード化するために、変換ブロックレベルのresidual#tb構文表が使用される。また、transform#unitレベルの構文は、いくつかのコード化済み、変換済み及び量子化済みのブロック関連構文(すなわち、デルタQP情報(存在する場合)及び考慮されるTUをコード化するために使用された変換のタイプ)も含む。

【0066】

図9A及び9Bはそれぞれ、CABAC復号及び符号化プロセスを示す。CABACは

10

20

30

40

50

、コンテキスト適応型二進算術コード化 (C A B A C) を表し、優れた圧縮効率でロスレス圧縮を提供するため、例えば、H E V C 又は V V C において使用されるエントロピー符号化の形態である。図 9 A のプロセスへの入力は、コード化済みのビットストリームを含み、典型的には、H E V C 仕様又はそのさらなる進化型に準拠する。復号プロセスのいずれの時点においても、デコーダは、どの構文要素を次に復号すべきか分かっている。これは、標準化されたビットストリーム構文及び復号プロセスにおいて十分に指定されている。その上、デコーダは、復号予定の現在の構文要素をどのように二進化するか (すなわち、各々は「1」又は「0」に等しい、ピンと呼ばれるバイナリシンボルのシーケンスとして表される) 及びピン文字列の各ピンがどのように符号化されているかについても分かっている。

10

【0067】

従って、C A B A C 復号プロセスの第 1 の段階 (図 9 A の左側) は、一連のピンを復号する。各ピンに対し、デコーダは、各ピンがバイパスモードに従って符号化されているか又は通常モードに従って符号化されているかについて分かっている。バイパスモードでは、ビットは、単純にビットストリームから読み取られ、得られた値は、現在のピンに割り当てられる。このモードは、単刀直入であり、従って、高速であり、集約的な資源利用を必要としないという利点を有する。それは、典型的には効率的であり、従って、均一な統計的分布 (すなわち、「1」に等しい確率と「0」に等しい確率が等しい) を有するピンに対して使用される。

【0068】

20

それとは逆に、現在のピンがバイパスモードでコード化されていない場合は、それは、いわゆる通常モードのコード化で (すなわち、コンテキストベースの算術コード化を通じて) コード化されていることを意味する。このモードは、はるかに資源集約的なものである。

【0069】

その事例では、考慮されるピンの復号は、次のように進む。最初に、現在のピンの復号のためのコンテキストが得られる。コンテキストは、図 9 A のコンテキストモデラによって与えられる。コンテキストの目標は、何らかのコンテキストのプライア (prior) 又は情報 X を考慮して、現在のピンが「0」の値を有する条件付きの確率を得ることである。ここでのプライア X は、現在のピンが復号されている際に、エンコーダ側とデコーダ側の両方で同期的に利用可能な、何らかの既に復号されている構文要素の値である。

30

【0070】

典型的には、ピンの復号のために使用されるプライア X は、規格において指定されており、復号予定の現在のピンと統計的に相関していることを理由に選ばれる。このコンテキスト情報の使用において興味深いことは、ピンのコード化のレートコストが低減されることである。これは、ピンと X との間に相関性があることを理由に X が与えられれば、ピンの条件付きのエントロピーが低くなるという事実に基づく。以下の関係は、情報理論において周知である。

$$H(\text{bin } X) \leq H(\text{bin})$$

【0071】

それは、ピンと X が統計的に相関している場合に X が既知であるピンの条件付きのエントロピーは、ピンのエントロピーより低いことを意味する。従って、コンテキスト情報 X は、ピンが「0」である確率又は「1」である確率を得るために使用される。これらの条件付きの確率を考慮すると、図 14 の通常モードの復号エンジンは、二進値ピンの算術復号を実行する。次いで、ピンの値は、現在のコンテキスト情報 X が既知である現在のピンと関連付けられた条件付きの確率の値を更新するために使用される。これは、図 9 A のコンテキストモデル更新ステップと呼ばれる。ピンが復号されている (又はコード化されている) 限り、各ピンに対してコンテキストモデルを更新することにより、各バイナリ要素に対するコンテキストモデリングの漸進的な精緻化が可能になる。従って、C A B A C デコーダは、通常モードの符号化ピンの各々の統計的な挙動を漸進的に学習する。

40

【0072】

50

コンテキストモデラ及びコンテキストモデル更新ステップは、エンコーダ側とデコーダ側において厳密に同一の動作である。

【0073】

現在のピンの通常の算術復号又はそのバイパス復号により、どのようにコード化されたかに応じて、一連の復号ピンが得られる。

【0074】

図9Aの右側に示されるCABAC復号の第2の段階は、この一連のバイナリシンボルをより高いレベルの構文要素に変換することを含む。構文要素は、フラグの形態を取ることができ、その事例では、現在の復号ピンの値を直接取り入れる。その一方で、考慮される標準仕様に従って現在の構文要素の二進化がいくつかのピンのセットに対応する場合は、図9Aの「構文要素に対するバイナリコードワード」と呼ばれる変換ステップが行われる。

10

【0075】

このステップは、図9Bに示されるようなエンコーダによって行われた二進化ステップの逆を進める。従って、ここで実行される逆変換は、これらの構文要素のそれぞれの復号済みの二進化バージョンに基づいて、それらの構文要素の値を得ることを含む。

【0076】

図1のエンコーダ100、図2のデコーダ200及び図3のシステム1000は、以下で説明される実施形態の少なくとも1つを実装するように適応される。

【0077】

現在のビデオコード化システム（例えば、VVC草稿4）では、係数グループコード化プロセスに対して何らかのハード制約が課され、ハードコード化される最大数の通常のCABACピンは、一方の側では、sig#flag、gt1#flag及びparity#flagという構文要素に対して採用することができ、他方の側では、gt3#flagという構文要素に対して採用することができる。より正確に言えば、4×4コード化グループ（又は係数グループ若しくはCG）レベル制約は、CABACデコーダエンジンによって単位エリアあたり最大数の通常のピンを解析しなければならないことを保証する。しかし、通常のピンのCABACコード化の使用の制限により、レート歪みが非最適なビデオ圧縮につながる可能性がある。実際に、ピクチャに対して限られた数の通常のピンしか許容されない場合は、コード化予定の考慮されるピクチャの一部がスキップモード（従って、残差なし）でコード化される有意エリアを内包する一方で、ピクチャの別の有意エリアが残差コード化を採用する事例では、この総バジレットの利用率は大幅に低くなり得る。そのような事例では、最大数の通常のピンに対する低レベル（すなわち、4×4CGレベル）の制約により、ピクチャの通常のピンの総数は、許容可能なピクチャレベル閾値をはるかに下回ることになり得る。

20

30

【0078】

少なくとも1つの実施形態は、高レベルの制約が尊重され、現在の手法と比べて圧縮効率が改善されるように、ブロック及びそれらの係数グループの残差コード化プロセスにおけるピンの通常のCABACコード化の最大使用量に対する高レベルの制約を取り扱うことに関連する。言い換えれば、通常のコード化ピンのバジレットは、CGより大きなピクチャエリアにわたって割り当てられ、従って、多数のCGがカバーされ、その数は、単位エリアあたりの通常のピンの平均許容数から決定される。例えば、1つのサンプルあたり平均で1.75の通常のコード化ピンを許容することができる。次いで、このバジレットは、より小さな単位にわたってより効率的に分配することができる。異なる実施形態では、より高いレベルの制約は、変換ブロック、変換単位、コード化単位、コード化木単位又はピクチャレベルで設定される。これらの実施形態は、図10に示されるようなCABACオプティマイザによって実装することができる。

40

【0079】

少なくとも1つの実施形態では、より高いレベルの制約は、変換単位レベルで設定される。そのような実施形態では、符号化/復号のために全変換単位に対して許容される通常のピンの数は、単位エリアあたりの通常のピンの平均許容数から決定される。TUに対し

50

て得られた通常のピンのバジェットから、TUの各変換ブロック(TB)に対して許容される通常のピンの数が導出される。変換ブロックは、同じTU及び同じ色成分に属する変換係数のセットである。次に、変換ブロックにおいて許容される通常のピンの数を考慮して、全TBに対して許容される通常のピンのこの数の制約の下で、残差コード化又は復号が適用される。従って、本明細書では、修正された残差コード化及び復号プロセスが提案され、修正された残差コード化及び復号プロセスでは、CGレベルの通常のピンのバジェットの代わりに、TBレベルの通常のピンのバジェットが考慮される。この目的のため、いくつかの実施形態が提案される。

【0080】

図10は、本原理の実施形態による、CABACオプティマイザを含むCABAC符号化プロセスを示す。少なくとも1つの実施形態では、CABACオプティマイザ190は、コード化グループのセットに対して通常のコード化を使用して符号化された(又は符号化予定の)ピンの最大数を表すバジェットを取り扱う。このバジェットは、例えば、通常のコード化ピンの1つのサンプルあたりのバジェットに考慮されるデータ単位の表面(すなわち、その表面に含まれるサンプルの数)を乗じることによって決定される。例えば、TBレベルで固定された通常のコード化ピンのバジェットの事例では、1つのサンプルあたりの通常のコード化ピンの許容数には、考慮される変換ブロックに含まれるサンプルの数が乗じられる。CABACオプティマイザ190の出力は、通常のコード化モード又はバイパスコード化モードでコード化を制御し、従って、コード化の効率に大いに影響を与える。

【0081】

図11は、本原理の実施形態による符号化プロセスにおいて使用されるCABACオプティマイザのフローチャートの例を示す。このフローチャートは、考慮されるコード化グループの新しいセットの各々に対して実行される。従って、異なる実施形態によれば、このフローチャートは、各ピクチャ、各CTU、各CU、各TU又は各TBに対して起こり得る。ステップ191では、プロセッサ301は、上記で説明されるように決定された通常のコード化に対するバジェットを割り当てる。ステップ192では、プロセッサ301は、コード化グループのセットにわたってループし、ステップ193では、最後のコード化グループに達したかをチェックする。ステップ194では、各コード化グループに対し、プロセッサは、通常のコード化ピンの入力バジェットでコード化グループを処理する。通常のコード化ピンのバジェットは、考慮されるコード化済みのグループの処理の間に減少する(すなわち、通常のコード化ピンのバジェットは、ピンが通常のモードでコード化される度にデクリメントされる)。通常のコード化ピンのバジェットは、コード化グループコード化又は復号プロセスの出力パラメータとして返される。

【0082】

図12は、CABACオプティマイザを使用してコード化グループを符号化/復号するための修正されたプロセスの例を示す。上記で紹介したように、残差コード化専用の通常のCABACピンの数の割り当ては、4x4コード化グループより高いレベルで実行される。これを行うため、コード化グループを符号化/復号するためのプロセスは、従来の機能と比べて、CGがコード化又は復号されている際に通常のピンの現在のバジェットを表す入力パラメータを追加することによって修正される。従って、このバジェットは、現在のCGを符号化/復号するために使用された通常のピンの数に従って現在のCGをコード化又は復号することによって修正される。通常のピンのバジェットを表す入力/出力パラメータは、図12では、numRegBins#in#outと呼ばれる。所定のCGの残差データ自体をコード化するためのプロセスは、通常のCABACピンの許容数が外部の手段によって与えられること以外は、従来の方式と同じであり得る。図12の事例と同様に、通常のピンがコード化/復号される度、このバジェットは、1ずつデクリメントされる。

【0083】

従って、本開示の少なくとも1つの実施形態は、係数グループコード化/解析機能の入力/出力パラメータとして通常のピンのバジェットを処理することを含む。従って、通常

のピンのバジェットは、CGコード化/解析レベルより高いレベルで決定することができる。異なる実施形態は、異なるレベル（すなわち、変換ブロック、変換単位、コード化単位、コード化木単位又はピクチャレベル）でバジェットを決定することを提案する。

【0084】

図13Aは、通常のピンのバジェットが変換ブロックレベルで決定される実施形態の例を示す。そのような実施形態では、バジェットは、2つの主要なパラメータ（現在の変換ブロックのサイズ及びピクチャエリアの所定の単体に対して固定された通常のピンの基本バジェット）の関数として決定される。本明細書では、考慮されるエリア単位はサンプルである。現在のVVC草稿2では、4×4CGに対して32個の通常のピンが許容されているため、それは、1つの成分サンプルあたり平均で2つの通常のピンが許容されていることを意味する。提案される変換ブロックレベルの通常のピンの割り当ては、単一のエリアあたりの通常のピンのこの平均レートに基づく。

10

【0085】

少なくとも1つの実施形態では、考慮される変換ブロックに割り当てられるバジェットは、変換ブロック表面と1つのサンプルあたりの通常の割り当てられた数との積として計算される。次に、バジェットは、考慮される変換ブロックにおける各有意係数グループに対するCG残差コード化/解析プロセスに渡される。

【0086】

図13Bは、通常のピンのバジェットが最終有意係数の位置に従って変換ブロックレベルで決定される実施形態の例を示す。この変形実施形態によれば、現在の変換ブロックに対して許容される通常のピンの数は、考慮される変換ブロックにおける最終有意係数の位置の関数として計算される。そのような手法の利点は、通常のピンを割り当てる際、考慮される変換ブロックに含まれるエネルギーにより良く適合することである。典型的には、高エネルギー変換ブロックには、より多くのピンを割り当てることができ、低エネルギー変換ブロックには、より少ない通常のピンを割り当てることができる。

20

【0087】

図14Aは、通常のピンのバジェットが変換単位レベルで決定される実施形態の例を示す。示されるように、unitary#budget（すなわち、1つのサンプルあたり許容される通常のピンの平均レート）は、依然として2に等しい。通常のピンのTUレベルのバジェットは、このレートに考慮される変換単位に内包されるサンプルの総数を乗じることによって決定される（すなわち、4:2:0のカラーフォーマットの例では、width*height*3/2）。次いで、この総バジェットは、考慮される変換単位に内包されるすべての変換ブロックのコード化のために使用される。そのバジェットは、各色成分に対する残差変換ブロックコード化/解析プロセスに入力/出力パラメータとして渡される。言い換えれば、考慮される変換単位の変換ブロックがコード化/解析される度、バジェットは、その変換単位において相次いでコード化/解析される変換ブロックのコード化/解析において通常のピンの数だけ減少する。

30

【0088】

図14Bは、通常のピンのバジェットが、変換単位レベルで決定され、異なる変換ブロック間の相対表面の関数として及び考慮される変換単位の既にコード化/解析されている変換ブロックにおいて使用される通常のピンの関数として変換ブロック間で割り当てられる実施形態の例を示す。この実施形態では、ルマTBに割り当てられる通常のピンは、TUレベルの総バジェットの2/3である。次に、CbTBバジェットは、ルマTBがコード化/解析された後の残りのバジェットの半分である。最後に、CrTBバジェットは、2つの最初のTBがコード化/解析された後の残りのTUレベルのバジェットに設定される。その上、さらなる実施形態では、変換単位全体のバジェットは、そのTUの1つ又は複数のTBがヌル残差でコード化されるという知識に基づいて、そのTUの変換ブロック間で共有できることに留意されたい。これは、例えば、tu#cbf#luma、tu#cbf#cb又はtu#cbf#crフラグの解析を通じて分かっている。従って、TBレベルの残差コード化/解析プロセスは、図14Cに示される修正されたプロセスによって示されるように、こ

40

50

の変形実施形態に適應される。そのような実施形態は、バジェットが階層のより低いレベルで割り当てられた際より優れたコード化効率をもたらす。実際に、いくつかのCG又はTBが非常に低いエネルギーのものである場合は、それらのCG又はTBでは、他のもの比べて低減された数のピンが採用され得る。従って、より大きな数のピンのコード化/解析が必要とされるCG又はTBにおいて、より大きな数の通常のピンを使用することができる。

【0089】

図15Aは、通常のピンのバジェットがコード化単位レベルで割り当てられる実施形態の例を示す。そのバジェットは、以前のTUレベルで固定される実施形態に対するものと同様に演算されるが、通常のピンのサンプルレート及びCUサイズに基づいて演算される。次いで、演算された通常のピンのバジェットは、図15Bの変換木コード化/解析手順に渡される。図15Bは、CUレベルで割り当てられた通常のピンのバジェットに基づく現在のCUと関連付けられた変換木の適應させたコード化/解析を示す。ここでは、CUレベルは、基本的には、考慮される変換木に内包される各変換単位のコード化に相次いで渡される。また、そのバジェットは、各TUのコード化において採用される通常のピンの数だけ減少する。図15Cは、バジェットがCUレベルで固定された際のコード化木コード化/解析プロセスの変形実施形態を示す。典型的には、このプロセスは、考慮される変換木の現在のTUをコード化する際、既にコード化/解析されているTUにおいて使用される通常のピンの関数として、CUレベルの総バジェットを更新する。例えば、インターCUのSBT(サブブロック変換)モードの事例では、VVC草稿4は、CUが2つのTUに分けられ、そのうちの1つがヌル残差を有することについて述べている。その事例では、この実施形態は、CUレベルの通常のピンの総バジェットが非ヌル残差を有するTU専用であることを提案し、それにより、SBTモードでのそのようなインターCUに対するコード化効率により優れたものになる。その上、事例がISP(イントラサブ区分化)では、イントラCUのこのVVC区分化モードは、CUをいくつかのTUに分ける。その事例では、TUのコード化/解析は、同じCU内の先行するTUに割り当てられた未使用の通常のピンのバジェットの再利用を試みる。図15D及び図15Eは、通常のピンのバジェットがCUレベルで固定された際の、本実施形態用に適應させた変換単位コード化/解析プロセスを示す。それらは、図14A及び図14Bのプロセスのそれぞれを本実施形態に適應させたものである。

【0090】

少なくとも実施形態では、残差コード化において使用される通常のCABACピンのバジェットは、CTUレベルで固定される。これにより、許容される通常のピンの総バジェットをよりうまく利用することができ、従って、以前の実施形態と比べてコード化効率を改善することができる。典型的には、最初にスキップCU専用とされたピンは、有利には、他の非スキップCUに対して使用することができる。

【0091】

少なくとも実施形態では、残差コード化において使用される通常のCABACピンのバジェットは、ピクチャレベルで固定される。これにより、許容される通常のピンの総バジェットをより一層うまく利用することができ、従って、以前の実施形態と比べてコード化効率を改善することができる。典型的には、最初にCUのスキップ専用とされたピンは、有利には、他の非スキップCUに対して使用することができる。

【0092】

少なくとも実施形態では、所定のピクチャの許容される通常のピンの平均レートは、考慮されるピクチャが属する時間層/深度に従って固定される。例えば、ランダムアクセスコード化構造では、ピクチャのグループの時間的構造は、階層Bピクチャ配列に順応する。この構成では、ピクチャは、スケラブルな時間層において整理される。上位層からのピクチャは、下位時間層からの参照ピクチャに依存する。それに反して、下位層のピクチャは、いかなる上位層のピクチャにも依存しない。上位時間層は、典型的には、下位層のピクチャより高い量子化パラメータでコード化され、従って、より低い質でコード化され

10

20

30

40

50

る。その上、下位層からのピクチャは、全シーケンスにわたる全体的なコード化効率に多大な影響を与える。従って、最適なコード化効率でこれらのピクチャを符号化することが興味深い。この実施形態によれば、上位時間層からのピクチャよりも下位層からのピクチャの方に通常のC A B A Cピンのより高いサンプルベースのレートを割り当てることが提案される。

【0093】

図16A及び図16Bは、通常のピンのバジェットがCGレベルより高いレベルで割り当てられる実施形態の例を示し、2つのタイプのCGコード化/解析プロセスを使用するためのものである。図16Aによって示される第1のタイプは、「all#bypass」と呼ばれ、CGと関連付けられたすべてのピンをバイパスモードでコード化することを含む。これは、CGの変換係数の大きさがabs#levelという構文要素のみを通じてコード化されることを含意する。図16Bによって示される第2のタイプは、「all#regular」と呼ばれ、sig#flag、gt1#flag、par#flag及びgt3#flagという構文要素に対応するすべてのピンを通常モードでコード化するものである。この実施形態では、CGレベルより高いレベルで固定された通常のピンのバジェットを考慮して、TBコード化プロセスは、現在考慮されている通常のピンの考慮されるバジェットが十分に使用されているか否かによって、「all#regular」CGコード化モードと「all#bypass」CGコード化モードとの間で切り替えることができる。

10

【0094】

様々な実装形態は復号を伴う。「復号」は、この出願で使用される場合は、例えば、表示に適した最終的な出力を生成するために、受信された符号化済みのシーケンスに対して実行されるプロセスのすべて又は一部を包含し得る。様々な実施形態では、そのようなプロセスは、例えば、エントロピー復号、逆量子化、逆変換及び差分復号など、典型的にはデコーダによって実行されるプロセスのうちの一つ又は複数を含む。様々な実施形態では、そのようなプロセスは、同様に又はその代替として、例えば、図10～図16の図で提示される実施形態など、この出願において説明される様々な実装形態のデコーダによって実行されるプロセスを含む。

20

【0095】

さらなる例として、一実施形態では、「復号」は、エントロピー復号のみを指し、別の実施形態では、「復号」は、差分復号のみを指し、別の実施形態では、「復号」は、エントロピー復号と差分復号との組合せを指す。「復号プロセス」という記載が具体的にある動作のサブセットを指すことを意図するか又は概してより広範な復号プロセスを指すことを意図するかは、特定の説明の文脈に基づいて明らかになり、当業者によって十分理解されていると考えられている。

30

【0096】

様々な実装形態は符号化を伴う。「復号」についての上記の論考に類似した方法では、「符号化」は、この出願で使用される場合は、例えば、符号化済みのビットストリームを生成するために、入力ビデオシーケンスに対して実行されるプロセスのすべて又は一部を包含し得る。様々な実施形態では、そのようなプロセスは、例えば、区分化、差分符号化、変換、量子化及びエントロピー符号化など、典型的にはエンコーダによって実行されるプロセスのうちの一つ又は複数を含む。様々な実施形態では、そのようなプロセスは、同様に又はその代替として、例えば、図10～図16の図の実施形態など、この出願において説明される様々な実装形態のエンコーダによって実行されるプロセスを含む。

40

【0097】

さらなる例として、一実施形態では、「符号化」は、エントロピー符号化のみを指し、別の実施形態では、「符号化」は、差分符号化のみを指し、別の実施形態では、「符号化」は、差分符号化とエントロピー符号化との組合せを指す。「符号化プロセス」という記載が具体的にある動作のサブセットを指すことを意図するか又は概してより広範な符号化プロセスを指すことを意図するかは、特定の説明の文脈に基づいて明らかになり、当業者によって十分理解されていると考えられている。

50

【 0 0 9 8 】

本明細書で使用される構文要素は記述的用語であることに留意されたい。従って、それらの構文要素は、他の構文要素名の使用を除外しない。

【 0 0 9 9 】

この出願は、ツール、特徴、実施形態、モデル、手法などを含む、多様な態様を説明する。これらの態様の多くは、特異性を持たせて説明しており、少なくとも個々の特性を示すために、制限を意図すると思われるような方法で説明している場合が多い。しかし、これは、説明を明確にするためであり、それらの態様の応用又は範囲を制限するものではない。現に、異なる態様はすべて、さらなる態様を提供するために、組み合わせたり交換したりすることができる。その上、態様は、先の出願において説明されている態様と組み合わせたり交換したりすることもできる。この出願において説明される及び企図される態様は、多くの異なる形態で実装することができる。上記の図 1、図 2 及び図 3 の図は、いくつかの実施形態を提供するが、他の実施形態も企図され、図の論考は、実装形態の幅広さを制限しない。

10

【 0 1 0 0 】

本出願では、「再構築される」及び「復号される」という用語は、交換可能に使用することができ、「画素」及び「サンプル」という用語は、交換可能に使用することができ、「画像」、「ピクチャ」及び「フレーム」という用語は、交換可能に使用することができる。通常、一概には言えないが、「再構築される」という用語は、エンコーダ側で使用され、「復号される」という用語は、デコーダ側で使用される。

20

【 0 1 0 1 】

本明細書では様々な方法が説明されており、方法の各々は、説明される方法を達成するための 1 つ又は複数のステップ又は行動を含む。方法の正しい動作のためにステップ又は行動の特定の順序が必要でない限り、特定のステップ及び / 又は行動の順序及び / 又は使用を修正することも、組み合わせることもできる。

【 0 1 0 2 】

本出願では、例えば、ブロックサイズに関して、様々な数値が使用されている。具体的な値は、例示を目的とするものであり、説明される態様は、これらの具体的な値に限定されない。

【 0 1 0 3 】

「一実施形態」、「実施形態」、「一実装形態」又は「実装形態」及びその他の変形形態への言及は、実施形態と関係して説明される特定の特徴、構造、特性などが少なくとも 1 つの実施形態に含まれることを意味する。従って、本明細書全体を通じて様々な場所に現れる「一実施形態では」、「実施形態では」、「一実装形態では」又は「実装形態では」という記載及び他の任意の変形形態の出現は、必ずしもすべてが同じ実施形態を指すとは限らない。

30

【 0 1 0 4 】

それに加えて、この出願又はその特許請求の範囲は、様々な情報片を「決定すること」について言及し得る。情報を決定することは、例えば、情報を推定すること、情報を計算すること、情報を予測すること、又は、メモリから情報を回収することのうちの 1 つ又は複数を含み得る。

40

【 0 1 0 5 】

さらに、この出願又はその特許請求の範囲は、様々な情報片に「アクセスすること」について言及し得る。情報にアクセスすることは、例えば、情報を受信すること、情報を回収すること（例えば、メモリから）、情報を格納すること、情報を移動すること、情報をコピーすること、情報を計算すること、情報を予測すること、又は、情報を推定することのうちの 1 つ又は複数を含み得る。

【 0 1 0 6 】

それに加えて、この出願又はその特許請求の範囲は、様々な情報片を「受信すること」について言及し得る。受信することは、「アクセスすること」と同様に、幅広い用語であ

50

ることを意図する。情報を受信することは、例えば、情報にアクセスすること又は情報を回収すること（例えば、メモリ若しくは光媒体記憶装置から）のうちの1つ又は複数を含み得る。さらに、「受信すること」は、典型的には、例えば、情報を格納すること、情報を処理すること、情報を送信すること、情報を移動すること、情報をコピーすること、情報を消去すること、情報を計算すること、情報を決定すること、情報を予測すること、又は、情報を推定することなど、動作の間にいろいろな方法で関与する。

【0107】

次の「/」、「及び/又は」、「～の少なくとも1つ」のいずれかの使用は、例えば、「A/B」、「A及び/又はB」、「A及びBの少なくとも1つ」の事例では、第1のリストオプション（A）のみの選択、第2のリストオプション（B）のみの選択、又は、両方のオプション（AとB）の選択を包含することを意図することを理解されたい。さらなる例として、「A、B及び/又はC」並びに「A、B及びCの少なくとも1つ」の事例では、そのような記載は、第1のリストオプション（A）のみの選択、第2のリストオプション（B）のみの選択、第3のリストオプション（C）のみの選択、第1及び第2のリストオプション（A及びB）のみの選択、第1及び第3のリストオプション（A及びC）のみの選択、第2及び第3のリストオプション（B及びC）のみの選択、又は、すべての3つのオプション（A、B及びC）の選択を包含することを意図する。これは、この分野及び関連分野の当業者にとって容易に明らかであるように、リストされるアイテムの数だけ拡張することができる。

【0108】

当業者には明らかであるように、実装形態は、例えば、格納又は送信することができる情報を伝えるようにフォーマットされた多様な信号を生成することができる。情報は、例えば、方法を実行するための命令、又は、説明される実装形態のうちの1つによって生成されたデータを含み得る。例えば、信号は、説明される実施形態のビットストリームを伝えるようにフォーマットすることができる。そのような信号は、例えば、電磁波として（例えば、スペクトルの高周波部分を使用して）又はベースバンド信号としてフォーマットすることができる。フォーマットすることは、例えば、データストリームを符号化すること、及び、符号化済みのデータストリームを有する搬送波を変調することを含み得る。信号が伝える情報は、例えば、アナログ又はデジタル情報であり得る。信号は、知られているように、多様な異なる有線又は無線リンク上で送信することができる。信号は、プロセス

[付記1]

限られた数の通常のピンを使用してコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを含むビデオ符号化方法であって、前記コード化が、C A B A C符号化であり、前記通常のピンの数が、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される、ビデオ符号化方法。

[付記2]

通常のピンを使用してコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを含むビデオ復号方法であって、前記復号が、C A B A C復号を使用して行われ、前記通常のピンの数が、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される、ビデオ復号方法。

[付記3]

限られた数の通常のピンを使用してコード化グループを含むピクチャエリアを表す構文要素をコード化する残差符号化プロセスを含むビデオ符号化装置であって、前記コード化が、C A B A C符号化であり、前記通常のピンの数が、多数のコード化グループ間で割り当てられたバジェットに基づいて決定される、ビデオ符号化装置。

[付記4]

通常のピンを使用してコード化グループを含むピクチャエリアを表すビットストリームを解析する残差復号プロセスを含むビデオ復号装置であって、前記復号が、C A B A C復号を使用して行われ、前記通常のピンの数が、多数のコード化グループ間で割り当てられ

10

20

30

40

50

たバジェットに基づいて決定される、ビデオ復号装置。

[付記 5]

処理 / コード化又は復号されているコード化グループに対し、多数のコード化グループ間で割り当てられる前記バジェットが、通常の C A B A C コード化又は復号モードを使用してバイナリ要素が処理 / コード化又は復号される度にデクリメントされる、付記 1 若しくは 2 に記載の方法又は付記 3 若しくは 4 に記載の装置。

[付記 6]

前記多数のコード化グループが、変換ブロックである、付記 5 に記載の方法又は装置。

[付記 7]

前記通常のピンの数が、前記変換ブロックの最終有意係数の位置に基づいて決定される、付記 6 に記載の方法又は装置。

10

[付記 8]

前記バジェットが、変換ブロックの表面に基づいて変換ブロック間で割り当てられる、付記 7 に記載の方法又は装置。

[付記 9]

前記多数のコード化グループが、変換単位である、付記 5 に記載の方法又は装置。

[付記 10]

前記バジェットが、前記変換単位の異なる変換ブロック間の相対表面の関数として、前記変換単位の変換ブロック間で割り当てられる、付記 9 に記載の方法又は装置。

[付記 11]

20

前記多数のコード化グループが、コード化単位である、付記 5 に記載の方法又は装置。

[付記 12]

前記多数のコード化グループが、コード化木単位である、付記 5 に記載の方法又は装置。

[付記 13]

前記多数のコード化グループが、ピクチャである、付記 5 に記載の方法又は装置。

[付記 14]

プロセッサによって実行されると、付記 1 ~ 13 の少なくとも一項に記載の方法のステップを実施するためのプログラムコード命令を含むコンピュータプログラム。

[付記 15]

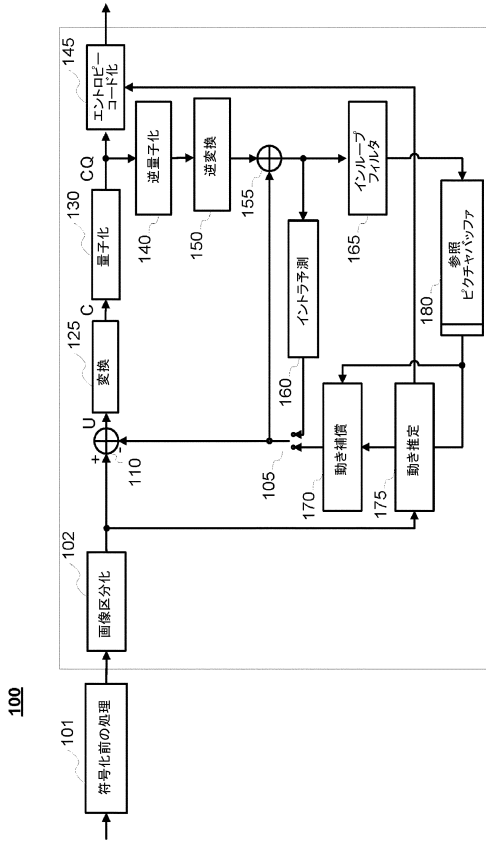
非一時的なコンピュータ可読媒体上に格納され、プロセッサによって実行されると、付記 1 ~ 13 の少なくとも一項に記載の方法のステップを実施するためのプログラムコード命令を含むコンピュータプログラム製品。

30

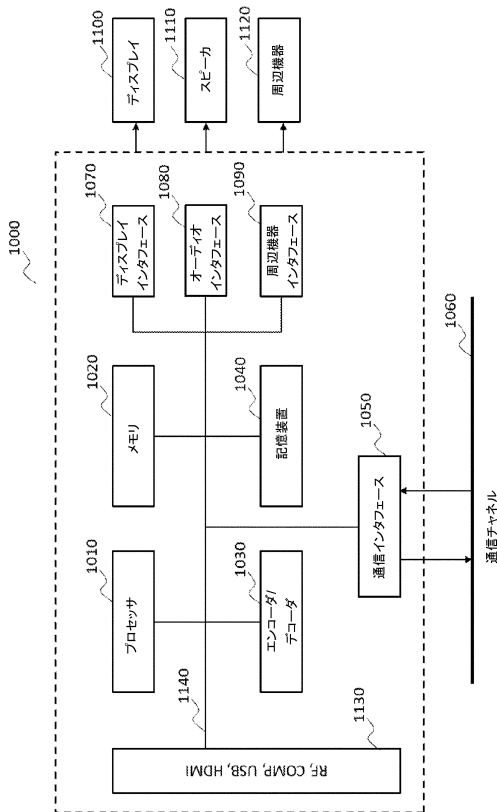
40

50

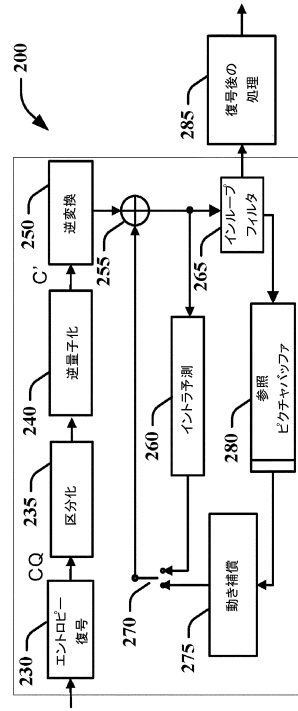
【図面】
【図 1】



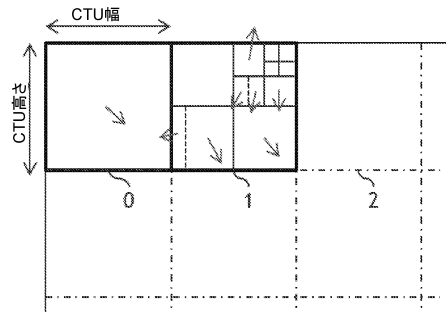
【図 3】



【図 2】



【図 4 A】



10

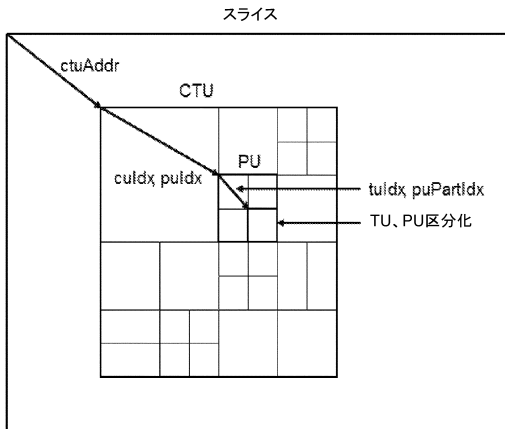
20

30

40

50

【 図 4 B 】



【 図 5 】

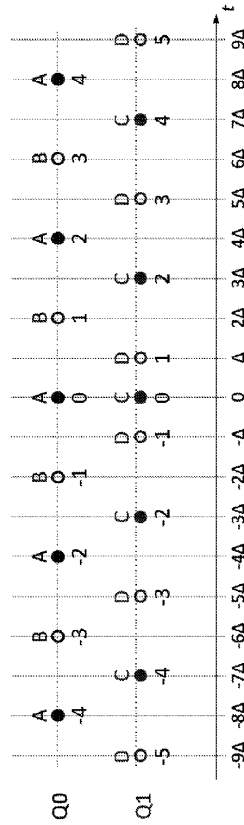
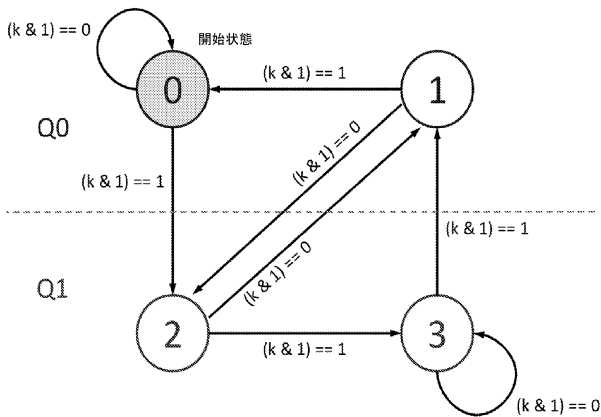


FIG. 5

10

20

【 図 6 A 】



【 図 6 B 】

現在の状態	...の次の状態	
	(k & 1) == 0	(k & 1) == 1
0	0	2
1	2	0
2	1	3
3	3	1

30

40

50

【 6 C 】

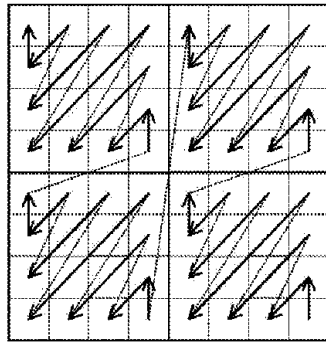


FIG. 6C

【 7 A 】

```

residual_tb( ... )
{
  last_significant_coefficient_pos_x_prefix
  last_significant_coefficient_pos_y_prefix
  last_significant_coefficient_pos_x_suffix
  last_significant_coefficient_pos_y_prefix

  for( i=lastSubBlock; i>=0; i-- ) {
    if( i < lastSubBlock ) {
      coded_sub_block_flag [ i ]
      if( codedSubBlockFlag [ i ] ) {
        residual_subblock ( ... )
      }
    }
  }
}
    
```

10

FIG. 7A

【 7 B 】

```

residual_subblock ( ... )
{
  numRegBins = is2x2CG ? 8 : 32

  for( k = firstPos ... ; remRegBins >= 4 ... ; k-- ) { // first pass
    sig_flag[ k ] // ctx depends on prev. par_flag's
    remRegBins -- // decrement remRegBins
    if( sig_flag[ k ] ) {
      gt1_flag[ k ]
      if ( gt1_flag[ k ] ) {
        par_flag[ k ] // context-based coding (regular)
        remRegBins -- // decrement remRegBins
        gt3_flag[ k ] // context-based coding (regular)
        remRegBins -- // decrement remRegBins
      }
      coeff[ k ] = 1 + parFlag + gt1Flag + (gt3Flag << 1)
    }
    firstPosMode2 = k
    for( k=firstPos; k > firstPosMode2; k-- ) { // second pass (EP bins)
      if( coeff[ k ] >= 4 )
        remainder[ k ]
    }

    for ( k = firstPosMode2 ; k > lastPos ; k-- ) { // coeff bypass (EP bins)
      abs_level[ k ]
    }
    sign_data
  }
}
    
```

FIG. 7B

【 8 A 】

```

coding_unit( ... )
{
  cu_ckpt_flag
  if ( cu_skip_flag == 0 )
    cu_pred_mode_flag

  if( cu_skip_flag == 0 && cuPredMode != MODE_INTRA ) {
    pred_mode_ibc_flag
  }

  if( cuPredMode == IMODE_NTRA ) {
    intra_prediction_data
  }
  else { // MODE_INTER or MODE_IBC
    inter_prediction_data
  }

  if ( !pcm_flag ) {
    if( cuPredMode != MODE_INTRA && merge_flag == 0 )
      cu_cbf
      if ( cu_cbf ) {
        // SBT mode related signaling
      }

    transform_tree( ... )
  }
}
    
```

20

30

FIG. 8A

40

50

【 図 8 B 】

```

Transform_tree( ... )
{
  if ( IntraSubPartSize == NO_ISP_FLAG ){
    if ( IntraSubPartSize == NO_ISP_FLAG )
      if ( width > MaxTbSizeX || height > MaxTbSizeY )
        quad-tree split into 4 sub-transform-trees
      }
    else {
      transform_unit( ... )
    }
  }
  Else if ( cu_sbt_fag ) {
    split into 2 transform-unit according to SBT syntax
    transform_unit( ... )
    transform_unit( ... )
  }
  else {
    split into 2 transform-unit according to ISP syntax
    transform_unit( ... )
    transform_unit( ... )
  }
}

```

FIG. 8B

【 図 8 C 】

```

transform_unit( ... )
{
  tu_cbf_luma
  tu_cbf_cb
  tu_cbf_cr

  if(tu_cbf_luma || tu_cbf_cb || tu_cbf_cr){
    cu_qp_delta_abs
    cu_qp_delta_sign_flag
  }

  if( width <= 32 && height <= 32 ... ){
    transform_skip_flag
  }

  if( width <= 32 && height <= 32 && !transform_skip_flag){
    tu_mts_idx
  }

  if( tu_cbf_luma ){
    Residual_tb ( ... , compId=Y,... )
  }
  if( tu_cbf_cb ){
    Residual_tb ( ... , compId=Cb,... )
  }
  if( tu_cbf_cr ){
    Residual_tb ( ... , compId=Cr,... )
  }
}

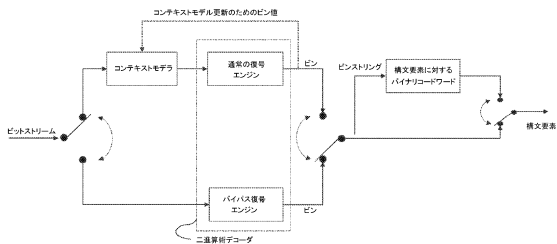
```

10

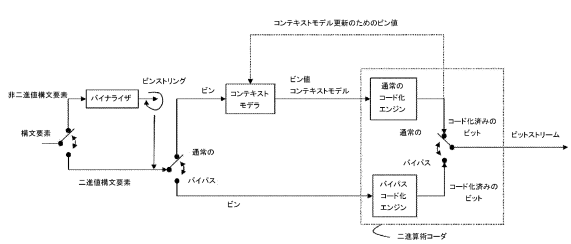
20

FIG. 8C

【 図 9 A 】



【 図 9 B 】

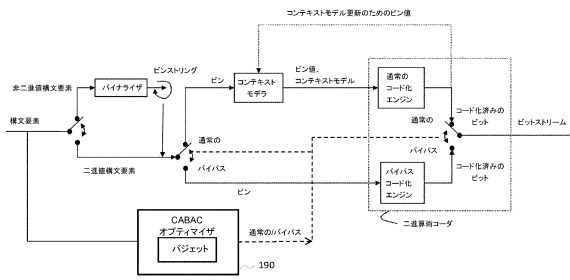


30

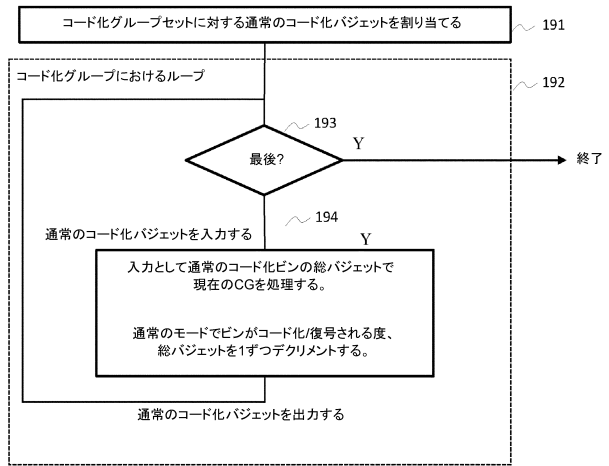
40

50

【図 10】



【図 11】



10

20

【図 12】

```

residual_subblock ( numRegBins_in_out )
{
  remRegBins = numRegBins_in_out

  for( k = firstPos ... ; remRegBins >= 4 ... ; k-- ) { // first pass
    sig_flag[ k ] // ctx depends on prev. par_flag's
    remRegBins -- // decrement remRegBins
    if( sig_flag[ k ] ) {
      gt1_flag[ k ]
      if( gt1_flag[ k ] ) {
        par_flag[ k ] // context-based coding (regular)
        remRegBins -- // decrement remRegBins
        gt3_flag[ k ] // context-based coding (regular)
        remRegBins -- // decrement remRegBins
      }
      coeff[ k ] = 1 + parFlag + gt1Flag + (gt3Flag << 1)
    }
  }
  firstPosMode2 = k
  for( k = firstPos; k > firstPosMode2; k-- ) { // second pass (EP bins)
    if( coeff[ k ] >= 4 )
      remainder[ k ]
  }

  for ( k = firstPosMode2 ; k > lastPos ; k-- ) { // coeff bypass (EP bins)
    abs_level[ k ]
  }
  sign_data
  numRegBins_in_out = remRegBins
}

```

FIG. 12

【図 13 A】

```

residual_tb( ... )
{
  unitary_budget = 2;
  TB_level_remRegBins = unitary_budget * width*height

  last_significant_coefficient_pos_x_prefix
  last_significant_coefficient_pos_y_prefix
  last_significant_coefficient_pos_x_suffix
  last_significant_coefficient_pos_y_prefix

  for( i = lastSubBlock; i >= 0; i-- ) {
    if( i < lastSubBlock ) {
      coded_sub_block_flag[ i ]
      if( codedSubBlockFlag[ i ] ) {
        residual_subblock( TB_level_remRegBins )
      }
    }
  }
}

```

FIG. 13A

30

40

50

【 1 3 B 】

```

residual_tb( ... )
{
    last_significant_coefficient_pos_x_prefix
    last_significant_coefficient_pos_y_prefix
    last_significant_coefficient_pos_x_suffix
    last_significant_coefficient_pos_y_prefix

    unitary_budget = 2;
    TB_level_remRegBins = unitary_budget * last_pos_x * last_pos_y

    for( i=lastSubBlock; i>=0; i-- ){
        if( i < lastSubBlock ){
            coded_sub_block_flag [ i ]
            if( codedSubBlockFlag [ i ] ){
                residual_subblock ( TB_level_remRegBins )
            }
        }
    }
}
    
```

FIG. 13B

【 1 4 A 】

```

transform_unit( ... )
{
    unitary_budget = 2; // average regular bins rate is 2 bins / sample
    TU_level_remRegBins = unitary_budget * width * height * 3 / 2

    tu_cbf_luma
    tu_cbf_cb
    tu_cbf_cr

    If( tu_cbf_luma || tu_cbf_cb || tu_cbf_cr ){
        cu_qp_delta_abs
        cu_qp_delta_sign_flag
    }

    If( width <= 32 && height <= 32 ... ){
        transform_skip_flag
    }

    If( width <= 32 && height <= 32 && !transform_skip_flag ){
        tu_mts_idx
    }

    if( tu_cbf_luma ){
        Residual_tb ( TU_level_remRegBins, ... , compId=Y,... )
    }
    if( tu_cbf_cb ){
        Residual_tb ( TU_level_remRegBins, ... , compId=Cb,... )
    }
    if( tu_cbf_cr ){
        Residual_tb ( TU_level_remRegBins, ... , compId=Cr,... )
    }
}
    
```

10

20

FIG. 14A

【 1 4 B 】

```

transform_unit( ... )
{
    unitary_budget = 2; // average regular bins rate is 2 bins / sample
    TU_level_RegBins = unitary_budget * width * height * 3 / 2

    tu_cbf_luma
    tu_cbf_cb
    tu_cbf_cr

    If( tu_cbf_luma || tu_cbf_cb || tu_cbf_cr ){
        cu_qp_delta_abs
        cu_qp_delta_sign_flag
    }

    If( width <= 32 && height <= 32 ... ){
        transform_skip_flag
    }

    If( width <= 32 && height <= 32 && !transform_skip_flag ){
        tu_mts_idx
    }

    if( tu_cbf_luma ){
        bins_Y = TU_level_RegBins * 2 / 3
        Residual_tb ( bins_Y, ... , compId=Y,... )
    }
    if( tu_cbf_cb ){
        bins_Cb = ( TU_level_RegBins - bins_Y ) / 2
        Residual_tb ( bins_Cb, ... , compId=Cb,... )
    }
    if( tu_cbf_cr ){
        bins_Cr = TU_level_RegBins - bins_Y - bins_Cb
        Residual_tb ( bins_Cr, ... , compId=Cr,... )
    }
}
    
```

FIG. 14B

【 1 4 C 】

```

residual_tb ( numRegBins_in_out )
{
    last_significant_coefficient_pos_x_prefix
    last_significant_coefficient_pos_y_prefix
    last_significant_coefficient_pos_x_suffix
    last_significant_coefficient_pos_y_prefix

    for( i = lastSubBlock; i>=0; i-- ){
        if( i < lastSubBlock ){
            coded_sub_block_flag [ i ]
            if( codedSubBlockFlag [ i ] ){
                residual_subblock ( numRegBins_in_out )
            }
        }
    }
}
    
```

30

40

FIG. 14C

【 1 5 A 】

```

coding_unit( ... )
{
    unitary_budget = 2; // average regular bins rate is 2 bins / sample
    CU_level_remRegBins = unitary_budget * width * height * 3 / 2

    cu_ckip_flag
    If ( cu_skip_flag == 0 )
        cu_pred_mode_flag

    if( cu_skip_flag == 0 && cuPredMode != MODE_INTRA ){
        pred_mode_ibc_flag
    }

    If( cuPredMode == IMODE_NTRA ){
        intra_prediction_data
    }
    else { // MODE_INTER or MODE_IBC
        inter_prediction_data
    }

    If ( !pcm_flag ){
        if( cuPredMode != MODE_INTRA && merge_flag == 0 )
            cu_cbf
            if ( cu_cbf ) {
                // SBT mode related signaling
            }

        transform_tree(CU_level_remRegBins , ... )
    }
}

```

FIG. 15A

【 1 5 B 】

```

Transform_tree(CU_level_remRegBins , ... )
{
    If ( IntraSubPartSize == NO_ISP_FLAG ){
        If ( IntraSubPartSize == NO_ISP_FLAG )
            if ( width > MaxTbSizeX || height > MaxTbSizeY )
                quad-tree split into 4 sub-transform-trees
        }
        else {
            transform_unit(CU_level_remRegBins , ... )
        }
    }
    Else if ( cu_sbt_fag ) {
        split into 2 transform-unit according to SBT syntax
        transform_unit(CU_level_remRegBins , ... )
        transform_unit(CU_level_remRegBins , ... )
    }
    else {
        split into 2 transform-unit according to ISP syntax
        transform_unit(CU_level_remRegBins , ... )
        transform_unit(CU_level_remRegBins , ... )
    }
}

```

10

20

FIG. 15B

【 1 5 C 】

```

Transform_tree(CU_level_remRegBins , ... )
{
    If ( IntraSubPartSize == NO_ISP_FLAG ){
        If ( IntraSubPartSize == NO_ISP_FLAG )
            if ( width > MaxTbSizeX || height > MaxTbSizeY )
                quad-tree split into 4 sub-transform-trees
        }
        else {
            transform_unit(CU_level_remRegBins , ... )
        }
    }
    Else if ( cu_sbt_fag ) {
        split into 2 transform-unit according to SBT syntax
        transform_unit(sbt_coded ? CU_level_remRegBins : 0 , ... )
        transform_unit(sbt_coded ? CU_level_remRegBins : 0 , ... )
    }
    else {
        split into 2 transform-unit according to ISP syntax
        regBins = CU_level_remRegBins / 2
        transform_unit(regBins , ... )
        transform_unit( (CU_level_remRegBins - regBins) , ... )
    }
}

```

FIG. 15C

【 1 5 D 】

```

transform_unit(TU_regBins , ... )
{
    tu_cbf_luma
    tu_cbf_cb
    tu_cbf_cr

    If( tu_cbf_luma || tu_cbf_cb || tu_cbf_cr ){
        cu_qp_delta_abs
        cu_qp_delta_sign_flag
    }

    If( width <= 32 && height <= 32 ... ){
        transform_skip_flag
    }

    If( width <= 32 && height <= 32 && !transform_skip_flag ){
        tu_mts_idx
    }

    If( tu_cbf_luma ){
        Residual_tb ( TU_regBins , ... , compId=Y,.. )
    }
    If( tu_cbf_cb ){
        Residual_tb ( TU_regBins , ... , compId=Cb,.. )
    }
    If( tu_cbf_cr ){
        Residual_tb ( TU_regBins , ... , compId=Cr,.. )
    }
}

```

30

40

FIG. 15D

50

【 1 5 E 】

```

transform_unit(TU_RegBins, ...)
{
  tu_cbf_luma
  tu_cbf_cb
  tu_cbf_cr

  If{tu_cbf_luma || tu_cbf_cb || tu_cbf_cr}{
    cu_qp_delta_abs
    cu_qp_delta_sign_flag
  }

  If{ width <= 32 && height <= 32 ... }{
    transform_skip_flag
  }

  If{ width <= 32 && height <= 32 && !transform_skip_flag}{
    tu_mts_idx
  }

  If{ tu_cbf_luma }{
    bins_Y = TU_RegBins * 2 / 3
    Residual_tb (bins_Y, ..., compId=Y,..)
  }
  If{ tu_cbf_cb }{
    bins_Cb = (TU_RegBins - bins_Y) / 2
    Residual_tb ( bins_Cb, ..., compId=Cb,..)
  }
  If{ tu_cbf_cr }{
    bins_Cr = TU_RegBins - bins_Y - bins_Cb
    Residual_tb ( bins_Cr, ..., compId=Cr,..)
  }
}

```

FIG. 15E

【 1 6 A 】

```

residual_subblock_all_bypass ()
{
  remRegBins = numRegBins_in_out

  K=firstPos
  firstPosMode2 = k

  for { k = firstPosMode2 ; k > lastPos ; k-- } {
    abs_level[ k ]
  }
  sign_data
  numRegBins_in_out = remRegBins
}

```

FIG. 16A

10

20

【 1 6 B 】

```

residual_subblock_all_regular ()
{
  for( k = firstPos ... ; remRegBins >= 4 ... ; k-- ) { // first pass
    sig_flag[ k ] // ctx depends on prev. par_flag's
    if( sig_flag[ k ] ) {
      gt1_flag[ k ]
      if ( gt1_flag[k] ) {
        par_flag[ k ] // context-based coding (regular)
        gt3_flag[ k ] // context-based coding (regular)
      }
      coeff[k] = 1 + parFlag + gt1Flag + (gt3Flag << 1)
    }
  }
  firstPosMode2 = k
  for( k=firstPos; k >lastPos; k-- ) { // second pass (EP bins)
    if( coeff[k] >= 4 )
      remainder[ k ]
  }

  sign_data
  numRegBins_in_out = remRegBins
}

```

FIG. 16B

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

欧州特許庁(EP)

(31)優先権主張番号 19305649.6

(32)優先日 令和1年5月23日(2019.5.23)

(33)優先権主張国・地域又は機関

欧州特許庁(EP)

弁理士 阿部 豊隆

(72)発明者 ルリアネック, ファブリス

フランス国, 3 5 5 7 6 セゾン - セビニエ, セーエス 1 7 6 1 6, ザック デ シャン ブラン, アベニュー デ シャン ブラン 9 7 5, エスアエス, インターデジタル アール アンド ディー フランス

(72)発明者 ロベール, アントワーヌ

フランス国, 3 5 5 7 6 セゾン - セビニエ, セーエス 1 7 6 1 6, ザック デ シャン ブラン, アベニュー デ シャン ブラン 9 7 5, エスアエス, インターデジタル アール アンド ディー フランス

(72)発明者 チェン, ヤ

フランス国, 3 5 5 7 6 セゾン - セビニエ, セーエス 1 7 6 1 6, ザック デ シャン ブラン, アベニュー デ シャン ブラン 9 7 5, エスアエス, インターデジタル アール アンド ディー フランス

審査官 岩井 健二

(56)参考文献 米国特許出願公開第 2 0 1 3 / 0 1 8 2 7 5 7 (U S , A 1)

米国特許出願公開第 2 0 1 0 / 0 2 2 0 7 8 3 (U S , A 1)

米国特許出願公開第 2 0 0 8 / 0 1 5 1 9 9 8 (U S , A 1)

Tung Nguyen, et al. , Transform Coding Techniques in HEVC , IEEE JOURNAL OF SELECTE D TOPICS IN SIGNAL PROCESSING , IEEE , 2013年12月 , VOL. 7, NO. 6 , pp.978-989

(58)調査した分野 (Int.Cl. , D B 名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8