

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7578819号  
(P7578819)

(45)発行日 令和6年11月6日(2024.11.6)

(24)登録日 令和6年10月28日(2024.10.28)

(51)国際特許分類	F I
H 04 N 19/13 (2014.01)	H 04 N 19/13
H 04 N 19/157(2014.01)	H 04 N 19/157
H 04 N 19/18 (2014.01)	H 04 N 19/18
H 04 N 19/70 (2014.01)	H 04 N 19/70

請求項の数 10 (全56頁)

(21)出願番号	特願2023-528033(P2023-528033)	(73)特許権者	520353802 テンセント・アメリカ・エルエルシー
(86)(22)出願日	令和4年4月29日(2022.4.29)		アメリカ合衆国 94306 カリフォルニア州 パロアルト パーク・ブルヴァード 2747
(65)公表番号	特表2023-549185(P2023-549185 A)	(74)代理人	100107766 弁理士 伊東 忠重
(43)公表日	令和5年11月22日(2023.11.22)	(74)代理人	100070150 弁理士 伊東 忠彦
(86)国際出願番号	PCT/US2022/072015	(74)代理人	100135079 弁理士 宮崎 修
(87)国際公開番号	WO2023/056105	(72)発明者	チエ, ビヨンドウ アメリカ合衆国 94306 カリフォルニア州 パロアルト パーク・ブルバード 2747 テンセント アメリカ エル最終頁に続く
(87)国際公開日	令和5年4月6日(2023.4.6)		
審査請求日	令和5年5月10日(2023.5.10)		
(31)優先権主張番号	63/250,166		
(32)優先日	令和3年9月29日(2021.9.29)		
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	17/710,778		
(32)優先日	令和4年3月31日(2022.3.31)		
	最終頁に続く		

(54)【発明の名称】 ビデオ復号のための方法、装置及びコンピュータプログラム

## (57)【特許請求の範囲】

## 【請求項1】

デコーダにおけるビデオ復号の方法であって、  
プロセッサによって、ビットストリーム内のコーディングされたビデオデータの第1範囲におけるコーディング制御のための第1構文要素を決定するステップであって、前記第1構文要素は、残差コーディングにおけるRiceパラメータ導出のための第1コーディングツールに代わる第2コーディングツールに関連付けられており、前記第1構文要素を決定するステップは、構文構造内の構文要素が前記構文構造内的一般制約情報の追加ビットを示すことに応答して、前記一般制約情報の前記構文構造から前記第1構文要素を復号することを含む、ステップと、

前記第1構文要素が、前記第1範囲における前記第2コーディングツールの無効化を示す第1値であることに応答して、前記プロセッサによって、前記第2コーディングツールを呼び出すことなく、コーディングされたビデオデータの1つ以上の第2範囲を含む、コーディングされたビデオデータの前記第1範囲を復号するステップと、

を含む、方法。

## 【請求項2】

前記第1構文要素は、前記デコーダにおける出力レイヤセット内のピクチャのコーディング制御のための制約情報内にある、

請求項1に記載の方法。

## 【請求項3】

前記第1構文要素の前記第1値は、前記出力レイヤセット内の各々のコーディングされたレイヤビデオシーケンス（C L V S）における第2コーディングツールの無効化を示す、請求項2に記載の方法。

**【請求項4】**

前記ビットストリーム内のコーディングされたレイヤビデオシーケンス（C L V S）のコーディング制御のための第2構文要素を、前記C L V Sを復号するために前記第2コーディングツールの呼び出しがないことを示す値を有するよう制約するステップ、を更に含む、請求項2に記載の方法。

**【請求項5】**

前記第1構文要素が第2値であることに応答して、前記ビットストリーム内のコーディングされたレイヤビデオシーケンス（C L V S）のコーディング制御のための第2構文要素の値を決定するステップを更に含み、前記第2構文要素は、前記C L V Sにおける前記第2コーディングツールの有効化／無効化を示す、

請求項2に記載の方法。

**【請求項6】**

前記第2構文要素の値を決定するステップは、

前記第2構文要素が前記C L V Sのためのシーケンスパラメータセット（S P S）内に存在しないことに応答して、前記C L V Sにおける前記第2コーディングツールの無効化を示すために前記第2構文要素の前記値を推論するステップ、

を更に含む、請求項5に記載の方法。

10

**【請求項7】**

前記第2コーディングツールは、前記デコーダによってサポートされていないレンジ拡張において定義される、

請求項1に記載の方法。

20

**【請求項8】**

前記第2コーディングツールは、残差コーディングの間の絶対値の二値化のためのR i c eパラメータ導出のためのものである、

請求項7に記載の方法。

**【請求項9】**

請求項1乃至8のいずれか一項に記載の方法を実行するよう構成される処理回路を含む、ビデオ復号のための装置。

30

**【請求項10】**

少なくとも1つのプロセッサによって実行されると、前記少なくとも1つのプロセッサに、請求項1乃至8のいずれか一項に記載の方法を実行させるコンピュータプログラム。

**【発明の詳細な説明】**

**【技術分野】**

**【0001】**

**参照による組み込み**

本出願は、2021年9月29日に出願された米国仮出願第63/250,166号、「TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION」の優先権の利益を主張する、2022年3月31日に出願された米国特許出願第17/710,778号、「TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION」の利益を主張する。先の出願の開示は、参照によりその全体が本明細書に組み込まれる。

40

**【0002】**

本開示は、一般的にビデオコーディングに関連する実施形態を説明する。

**【背景技術】**

**【0003】**

50

本明細書で提供される背景説明は、本開示の文脈を一般的に提示するためのものである。現在名前を挙げられている発明者の研究は、その研究がこの背景技術に記載された範囲において、出願時に先行技術として通常見なされ得ない記載の態様とともに、明示的にも暗黙的にも本開示に対する先行技術として認められない。

#### 【0004】

ビデオコーディング及び復号は、動き補償を伴うインパクチャ予測を使用して実行されることができる。非圧縮デジタルビデオは、一連のピクチャを含むことができ、各ピクチャは、例えば  $1920 \times 1080$  の輝度サンプル及び関連するクロミナンスサンプルの空間寸法を有する。一連のピクチャは、例えば毎秒 60 ピクチャ又は 60 Hz の固定又は可変ピクチャレート（非公式にフレームレートとしても知られる）を有することができる。非圧縮ビデオは、特定のビットレート要件を有する。例えばサンプル当たり 8 ビットの  $1080 \text{p} 60 4:2:0$  ビデオ（60 Hz のフレームレートでの  $1920 \times 1080$  の輝度サンプル解像度）は、1.5 Gbit/s に近い帯域幅を必要とする。このようないビデオの 1 時間は、600 G バイトを超える記憶領域を必要とする。

10

#### 【0005】

ビデオコーディング及び復号の 1 つの目的は、圧縮を通じた、入力ビデオ信号の冗長性の低減である可能性がある。圧縮は、前述の帯域幅及び / 又は記憶領域の要件を、場合によっては 2 衍以上低減するのを助けることができる。可逆圧縮と非可逆圧縮の両方、並びにそれらの組合せを採用することができる。可逆圧縮は、圧縮された元の信号から元の信号の正確なコピーを再構成することができる技術を指す。非可逆圧縮を使用するとき、再構成信号は、元の信号と同一でない可能性はあるが、元の信号と再構成信号との間の歪みは、再構成信号を、意図されるアプリケーションに有用であるようにするために十分に小さい。ビデオの場合、非可逆圧縮は広く使用される。許容される歪み量はアプリケーションに依存し、例えば特定の消費者ストリーミングアプリケーションのユーザは、テレビジョン分配アプリケーションのユーザよりも高い歪みを許容し得る。達成可能な圧縮比は、より高い許容可能な / 容認可能な歪みが、より高い圧縮比をもたらすことができる反応する可能性がある。

20

#### 【0006】

ビデオエンコーダ及びデコーダは、例えば動き補償、変換、量子化及びエントロピーコーディングを含む、いくつかの広範なカテゴリからの技術を利用することができる。

30

#### 【0007】

ビデオコーデック技術は、イントラコーディングとして知られる技術を含むことができる。イントラコーディングでは、サンプル値は、以前に再構成された参照ピクチャからのサンプル又は他のデータを参照することなく表される。いくつかのビデオコーデックでは、ピクチャはサンプルのブロックに空間的に細分される。イントラモードでサンプルのすべてのブロックがコーディングされるとき、そのピクチャはイントラピクチャとすることができます。独立デコーダリフレッシュピクチャ (independent decoder refresh pictures) のようなイントラピクチャ及びそれらの派生物を使用して、デコーダ状態をリセットすることができ、したがって、コーディングされたビデオビットストリーム及びビデオセッションにおける最初のピクチャとして、あるいは静止画像として使用することができる。イントラブロックのサンプルに、変換を受けさせることができ、変換係数を、エントロピーコーディングの前に量子化することができる。イントラ予測は、事前変換領域 (pre-transform domain) におけるサンプル値を最小化する技術であり得る。場合によっては、変換後の DC 値がより小さく、AC 係数がより小さいほど、エントロピーコーディング後のブロックを表すために所与の量子化ステップサイズで必要とされるビット数は少ない。

40

#### 【0008】

例えば MPEG - 2 世代のコーディング技術から知られているような伝統的なイントラコーディングは、イントラ予測を使用しない。しかしながら、いくつかのより新しいビデオ圧縮技術は、例えば空間的に隣接し、かつ復号順序において先行する、データのブロッ

50

クの符号化 / 復号中に取得される、周囲のサンプルデータ及び / 又はメタデータから試みる技術を含む。このような技術は、以下では「イントラ予測」技術と呼ばれる。少なくともいくつかの場合には、イントラ予測は、再構成中の現在のピクチャからの参照データのみを使用し、参照ピクチャからのものは使用しないことに留意されたい。

#### 【0009】

多くの異なる形式のイントラ予測が存在し得る。所与のビデオコーディング技術において、そのような技術のうちの 2 つ以上を使用することができるとき、使用中の技術を、イントラ予測モードにおいてコーディングすることができる。ある場合には、モードは、サブモード及び / 又はパラメータを有することができ、これらを個々にコーディングするか又はモードコードワード ( mode codeword ) に含めることができる。どのコードワードを、所与のモード / サブモード / パラメータの組合せに使用するかは、イントラ予測を通じたコーディング効率ゲインに影響を与える可能性があり、コードワードをビットストリームに変換するために使用されるエントロピーコーディング技術も同様であり得る。

10

#### 【0010】

イントラ予測の特定のモードは、H.264 で導入され、H.265 で精査され、JEM ( joint exploration model ) 、VVC ( versatile video coding ) 及びベンチマークセット ( BMS : benchmark set ) のようなより新しいコーディング技術で更に精査された。既に利用可能なサンプルに属している隣接するサンプル値を使用して、予測子ブロック ( predictor block ) を形成することができる。隣接するサンプルのサンプル値は、方向に従って予測子ブロックにコピーされる。使用中の方向への参照は、ビットストリームでコーディングされることができるか又はそれ自体が予測され得る。

20

#### 【0011】

図 1 A を参照すると、右下に示されているのは、H.265 の 33 の可能な予測子方向 ( predictor directions ) ( 35 のイントラモードの 33 の角度モードに対応する ) からわかる、9 つの予測子方向のサブセットである。矢印が収束する点 ( 101 ) は、予測されているサンプルを表す。矢印は、サンプルが予測されている方向を表す。例えば矢印 ( 102 ) は、サンプル ( 101 ) が、1 つのサンプル又は複数のサンプルから右上へ、水平から 45 度の角度で予測されることを示す。同様に、矢印 ( 103 ) は、サンプル ( 101 ) が、1 つのサンプル又は複数のサンプルからサンプル ( 101 ) の左下へ、水平から 22.5 度の角度で予測されることを示す。

30

#### 【0012】

引き続き図 1 A を参照すると、左上には、 $4 \times 4$  サンプルの正方形ブロック ( 104 ) が示されている ( 破線の太線で示されている )。正方形ブロック ( 104 ) は、16 個のサンプルを含み、各々「S」と、Y 次元におけるその位置 ( 例えば行インデックス ) と、X 次元におけるその位置 ( 例えば列インデックス ) とでラベリングされている。例えばサンプル S21 は、Y 次元の ( 上から ) 2 番目のサンプル及び X 次元の ( 左から ) 1 番目のサンプルである。同様に、サンプル S44 は、Y 及び X 次元の両方においてブロック ( 104 ) の 4 番目のサンプルである。ブロックのサイズが  $4 \times 4$  サンプルであるので、S44 は右下にある。さらに、同様のナンバリングスキームに従う参照サンプルが示されている。参照サンプルは、ブロック ( 104 ) に対する R、その Y 位置 ( 例えば行インデックス ) 及び X 位置 ( 列インデックス ) でラベリングされる。H.264 と H.265 の両方において、予測サンプルは再構成中のブロックに隣接し、したがって、負の値は使用される必要がない。

40

#### 【0013】

イントラピクチャ予測は、シグナリングされる予測方向によって適切であるように、隣接するサンプルから参照サンプル値をコピーすることによって機能することができる。例えばコーディングされたビデオビットストリームが、このブロックに対して、矢印 ( 102 ) と一致する予測方向を示すシグナリングを含むと仮定する、すなわち、サンプルは、1 つ又は複数の予測サンプルから右上へ、水平から 45 度の角度で予測される。この場合、サンプル S41 、 S32 、 S23 、 S14 は、同じ参照サンプル R05 から予測される

50

。次いで、サンプル S 4 4 は、参照サンプル R 0 8 から予測される。

【 0 0 1 4 】

ある場合には、特に方向が 45 度で均一に割り切れないときは、参照サンプルを計算するために、複数の参照サンプルの値が、例えば補間 (interpolation) を通して組み合わされ得る。

【 0 0 1 5 】

ビデオコーディング技術が発展するにつれて、可能な方向の数が増えている。H.264 (2003年) では、9つの異なる方向を表すことができる。これは、H.265 (2013年) では 33 に増加し、開示の時点で、JEM / VVC / BMS は、65 の方向をサポートすることができる。実験は、最も可能性がある方向を識別するために実施され、エントロピーコーディングにおける特定の技術は、それらの最も可能性がある方向を少数のビットで表すために使用され、より可能性が少ない方向に対して特定のペナルティを受け入れる。さらに、時には方向それ自体を、隣接する既に復号されたブロックにおいて使用される隣接方向から予測することができる。

10

【 0 0 1 6 】

図 1B は、JEM に従う 65 のイントラ予測方向を示す概略 (180) を示しており、経時的な予測方向の数の増加を例示す。

【 0 0 1 7 】

コーディングされたビデオビットストリーム内の、方向を表すイントラ予測方向ビットのマッピングは、ビデオコーディング技術ごとに異なる可能性があり、例えばイントラ予測モードへの予測方向の単純な直接マッピングから、コードワードへ、最確モード (most probable mode) を含む複雑な適応スキームへ及び類似の技術に及ぶ可能性がある。しかしながら、すべての場合において、特定の他の方向よりもビデオコンテンツ内において生じる可能性が統計的に低い、特定の方向が存在する可能性がある。ビデオ圧縮の目標は冗長性の低減であるので、それらのより可能性が低い方向は、良好に機能するビデオコーディング技術では、より可能性が高い方向よりも多くのビット数によって表される。

20

【 0 0 1 8 】

動き補償は、非可逆圧縮技術であり得、以前に再構成されたピクチャ又はその一部 (参照ピクチャ) からのサンプルデータのブロックが、動きベクトル (以下、MV) によって示される方向に空間的にシフトされた後に、新たに再構成されたピクチャ又はピクチャの一部の予測に使用される技術に関連する可能性がある。場合によっては、参照ピクチャは、現在再構成中のピクチャと同じものとすることができます。MV は、2 次元 X 及び Y 又は 3 次元を有することができ、3 番目の次元 (third) は、使用中の参照ピクチャの指示である (後者は、間接的に、時間次元である可能性がある)。

30

【 0 0 1 9 】

いくつかのビデオ圧縮技術では、サンプルデータのあるエリアに適用可能な MV を、他の MV から、例えば再構成中のエリアに空間的に隣接し、かつ復号順序でその MV に先行するサンプルデータの別のエリアに関連する MV から、予測することができる。そうすることは、MV をコーディングするために必要とされるデータの量を実質的に減少させることができ、それによって、冗長性を除去して、圧縮を高めることができる。MV 予測は効果的に機能することができる。なぜなら、例えばカメラから導出される入力ビデオ信号 (自然ビデオとして知られる) をコーディングする際に、単一の MV が適用可能なエリアよりも大きいエリアが、同様の方向に移動するという統計的 possibility が存在し、したがって、場合によって、隣接するエリアの MV から導出される同様の動きベクトルを用いて予測をすることができるからである。その結果、所与のエリアについて見出される MV は、周囲の MV から予測される MV と類似又は同一であり、その後、これは、エントロピーコーディングの後、MV を直接コーディングする場合に使用されるであろうものよりも、より少ない数のビットで表現することができる。場合によって、MV 予測は、元の信号 (すなわち、サンプルストリーム) から導出される信号 (すなわち、MV) の可逆圧縮の例であり得る。他の場合には、MV 予測それ自体は、例えばいくつかの周囲の MV から予測子を計

40

50

算する際の丸め誤差のために、非可逆あり得る。

【0020】

様々なMV予測メカニズムが、H.265 / HEVC (ITU-T Rec. H.265、「High Efficiency Video Coding」、2016年12月)において説明されている。H.265が提案する多くのMV予測メカニズムのうち、ここで説明されるのは「空間マージ(spatial merge)」と呼ばれる技術である。

【0021】

図2を参照すると、現在のブロック(201)は、エンコーダによって、動き探索プロセス中に、空間的にシフトされた同じサイズの前のブロックから予測可能であることが見出されたサンプルを含む。そのMVを直接コーディングする代わりに、MVを、1つ以上の参照ピクチャに関連付けられるメタデータから、例えば(復号順序で)最新の参照ピクチャから、A0、A1及びB0、B1、B2(それぞれ202から206)と示される5つの周囲のサンプルのいずれか1つに関連付けられるMVを使用して導出することができる。H.265では、MV予測は、隣接ブロックが使用しているものと同じ参照ピクチャからの予測子を使用することができる。

10

【発明の概要】

【0022】

本開示の態様は、ビデオデータ処理のための方法及び装置を提供する。いくつかの実施形態において、ビデオデータ処理のための装置は処理回路を含む。例えば処理回路は、ビットストリーム内のコーディングされたビデオデータの第1範囲(first scope)におけるコーディング制御のための第1構文要素を決定する。第1構文要素は、残差コーディングにおけるRiceパラメータ導出(Rice parameter derivation)のための第1コーディングツールに代わる第2コーディングツールに関連付けられる。第1構文要素が、第1範囲における第2コーディングツールの無効化を示す第1値であることに応答して、処理回路は、第2コーディングツールを呼び出すことなく、コーディングされたビデオデータの1つ以上の第2範囲を含む、コーディングされたビデオデータの第1範囲を復号する。

20

【0023】

いくつかの実施形態において、第1構文要素は、出力レイヤセット内のピクチャのコーディング制御のための一般制約情報(general constraint information)内にある。いくつかの例において、第1構文要素の第1値は、出力レイヤセット内の各コーディングされたレイヤビデオシーケンス(CLVS:coded layer video sequence)における第2コーディングツールの無効化を示す。

30

【0024】

いくつかの例において、処理回路は、ビットストリーム内のコーディングされたレイヤビデオシーケンス(CLVS)のコーディング制御のための第2構文要素を、CLVSを復号するために第2コーディングツールの呼び出しがないことを示す値を有するよう制約することができる。

【0025】

いくつかの実施形態において、第1構文要素が第2値であることに応答して、処理回路は、ビットストリーム内のコーディングされたレイヤビデオシーケンス(CLVS)のコーディング制御のための第2構文要素の値を決定する。第2構文要素は、CLVSにおける第2コーディングツールの有効化/無効化を示す。第2コーディングツールは、残差コーディングの間の絶対値の二値化のためのRiceパラメータ導出のためのものである。

40

【0026】

いくつかの例において、処理回路は、第2構文要素がCLVSのためのシーケンスパラメータセット(SPS:sequence parameter set)内に存在しないことに応答して、CLVSにおける第2コーディングツールの不使用を示すための第2構文要素の値を推論する。

【0027】

いくつかの例において、処理回路は、構文構造内の構文要素が構文構造内的一般制約情

50

報の追加ビットを示すことに応答して、一般制約情報の構文構造から第1構文要素を復号する。

【0028】

いくつかの例において、第2コーディングツールは、規格のレンジ拡張 (range extension) において定義され、レンジ拡張は、当該装置によってサポートされていない。

【0029】

本開示の態様はまた、ビデオ復号のためにコンピュータによって実行されると、コンピュータに、ビデオ復号のための方法を実行させる命令を記憶する、非一時的なコンピュータ読み取り可能媒体も提供する。

【図面の簡単な説明】

10

【0030】

開示される主題の更なる特徴、性質及び様々な利点は、以下の詳細な説明及び添付の図面からより明らかになる。

【0031】

【図1A】イントラ予測モードの例示的なサブセットの概略図である。

【0032】

【図1B】例示的なイントラ予測方向の図である。

【0033】

【図2】一例における、現在のブロックとその周辺の空間マージ候補の概略図である。

20

【0034】

【図3】一実施形態による通信システム (300) の簡略化されたブロック図の概略図である。

【0035】

【図4】一実施形態による通信システム (400) の簡略化されたブロック図の概略図である。

【0036】

【図5】一実施形態によるデコーダの簡略化されたブロック図の概略図である。

【0037】

【図6】一実施形態によるエンコーダの簡略化されたブロック図の概略図である。

30

【0038】

【図7】別の実施形態によるエンコーダのブロック図である。

【0039】

【図8】別の実施形態によるデコーダのブロック図である。

【0040】

【図9】本開示の実施形態による適応解像度変更 (A R C : adaptive resolution change) パラメータのシグナリングの例を示す図である。

【0041】

【図10】アップサンプル又はダウンサンプルのファクタ、コードワード及びE x t - G o l o m b コードのマッピングのための表 (1000) の例を示す図である。

40

【0042】

【図11】本開示のいくつかの実施形態によるA R C パラメータのシグナリングのいくつかの例を示す図である。

【0043】

【図12】いくつかの例におけるP T L 構文要素のセットの構文構造例を示す図である。

【0044】

【図13】いくつかの例における一般制約情報の構文構造例を示す図である。

【0045】

【図14A】本開示のいくつかの実施形態による、P T L 構文構造と一般制約情報の構文構造を含む、P T L 情報の例を示す図である。

【図14B】本開示のいくつかの実施形態による、P T L 構文構造と一般制約情報の構文

50

構造を含む、PTL情報の例を示す図である。

【0046】

【図15A】本開示の一実施形態による一般制約情報の構文構造の例を示す図である。

【図15B】本開示の一実施形態による一般制約情報の構文構造の例を示す図である。

【0047】

【図16】本開示のいくつかの実施形態による一般制約情報の構文構造を示す図である。

【0048】

【図17】本開示のいくつかの実施形態によるシーケンスパラメータセット( SPS )レンジ拡張の構文構造例を示す図である。

【0049】

【図18】本開示の実施形態によるプロセスを概説するフローチャートを示す図である。

【0050】

【図19】本開示の実施形態によるプロセスを概説するフローチャートを示す図である。

【0051】

【図20】一実施形態によるコンピュータシステムの概略図である。

【発明を実施するための形態】

【0052】

図3は、本開示の一実施形態による通信システム(300)の簡略化されたブロック図を示す。通信システム(300)は、例えばネットワーク(350)を介して互いに通信することができる複数の端末デバイスを含む。例えば通信システム(300)は、ネットワーク(350)を介して相互接続される第1の対の端末デバイス(310)及び(320)を含む。図3の例では、第1の対の端末デバイス(310)及び(320)は、データの一方向伝送を実行する。例えば端末デバイス(310)は、ネットワーク(350)を介して他の端末デバイス(320)に伝送するために、ビデオデータ(例えば端末デバイス(310)によってキャプチャされるビデオピクチャのストリーム)をコーディングし得る。符号化されたビデオデータは、1つ以上のコーディングされたビデオビットストリームの形態で伝送することができる。端末デバイス(320)は、ネットワーク(350)からコーディングされたビデオデータを受け取ることができ、コーディングされたビデオデータを復号して、ビデオピクチャを復元し、復元されたビデオデータに従ってビデオピクチャを表示す。一方向データ伝送は、メディア供給アプリケーション等において一般的であり得る。

10

20

30

40

【0053】

別の例では、通信システム(300)は、例えばビデオ会議中に生じ得るコーディングされたビデオデータの双方向伝送を実行する、第2の対の端末デバイス(330)及び(340)を含む。データの双方向伝送のために、一例では、端末デバイス(330)及び(340)の各端末デバイスは、ネットワーク(350)を介して端末デバイス(330)及び(340)のうちの他方の端末デバイスに伝送するために、ビデオデータ(例えばその端末デバイスによってキャプチャされるビデオピクチャのストリーム)をコーディングし得る。端末デバイス(330)及び(340)の各端末デバイスはまた、端末デバイス(330)及び(340)の他方の端末デバイスによって伝送された、コーディングされたビデオデータを受け取ることができ、コーディングされたビデオデータを復号してビデオピクチャを復元することができ、復元されたビデオデータに従って、アクセス可能なディスプレイデバイスでビデオピクチャを表示することができる。

【0054】

図3の例では、端末デバイス(310)、(320)、(330)及び(340)は、サーバ、パーソナルコンピュータ及びスマートフォンとして例示され得るが、本開示の原理は、そのように限定され得ない。本開示の実施形態は、ラップトップコンピュータ、タブレットコンピュータ、メディアプレーヤ及び/又は専用のビデオ会議装置への適用を見出す。ネットワーク(350)は、例えばワイヤライン(有線)及び/又は無線通信ネットワークを含め、端末デバイス(310)、(320)、(330)及び(340)の間

50

でコーディングされたビデオデータを伝達する任意の数のネットワークを表す。通信ネットワーク(350)は、回線スイッチ及び/又はパケットスイッチチャネルにおいてデータを交換し得る。代表的なネットワークは、電気通信ネットワーク、ローカルエリアネットワーク、ワイドエリアネットワーク及び/又はインターネットを含む。本議論の目的のために、ネットワーク(350)のアーキテクチャ及びトポロジは、本明細書において以下に説明されない限り、本開示の動作に対して重要ではないことがある。

#### 【0055】

図4は、開示される主題の適用の例として、ストリーミング環境におけるビデオエンコーダ及びビデオデコーダの配置を示す。開示される主題は、例えばビデオ会議やデジタルTV、CD、DVD、メモリスティック等を含むデジタルメディアにおける圧縮ビデオの記憶等を含む、他のビデオ対応アプリケーションにも同様に適用可能であり得る。

10

#### 【0056】

ストリーミングシステムは、例えば圧縮されていないビデオピクチャのストリーム(402)を作成するビデオソース(401)、例えばデジタルカメラを含むことができる、ビデオキャプチャサブシステム(413)を含み得る。一例では、ビデオピクチャのストリーム(402)は、デジタルカメラによって撮られたサンプルを含む。ビデオピクチャのストリーム(402)は、符号化ビデオデータ(404)(又はコーディングされたビデオビットストリーム)と比較したときの高データ量を強調するために太線として示されており、ビデオソース(401)に結合されるビデオエンコーダ(403)を含む電子デバイス(420)によって処理ができる。ビデオエンコーダ(403)は、以下により詳細に説明されるような、開示される主題の態様を可能にするか又は実装するために、ハードウェア、ソフトウェア又はそれらの組合せを含むことができる。符号化ビデオデータ(404)(又は符号化ビデオビットストリーム(404))は、ビデオピクチャのストリーム(402)と比較したときの低データ量を強調するために細線として示されており、将来の使用のためにストリーミングサーバ(405)に記憶されることができる。図4のクライアントサブシステム(406)及び(408)のような1つ以上のストリーミングクライアントサブシステムは、ストリーミングサーバ(405)にアクセスして、符号化ビデオデータ(404)のコピー(407)及び(409)を取り出すことができる。クライアントサブシステム(406)は、例えば電子デバイス(430)内にビデオデコーダ(410)を含むことができる。ビデオデコーダ(410)は符号化ビデオデータの入力コピー(407)を復号し、そして、ディスプレイ(412)(例えばディスプレイ画面)又は他のレンダリングデバイス(図示せず)上でレンダリングすることができる、ビデオピクチャの出力ストリーム(411)を作成する。いくつかのストリーミングシステムでは、符号化ビデオデータ(404)、(407)及び(409)(例えばビデオビットストリーム)を、特定のビデオコーディング/圧縮規格に従って符号化することができる。これらの規格の例は、ITU-T勧告H.265を含む。一例では、開発中のビデオコーディング規格は、VVC(Versatile Video Coding)として非公式に知られている。開示される主題はVVCの文脈で使用され得る。

20

30

#### 【0057】

電子デバイス(420)及び(430)は、他の構成要素(図示せず)を含むことができることに留意されたい。例えば電子デバイス(420)は、ビデオデコーダ(図示せず)を含むことができ、電子デバイス(430)は、ビデオエンコーダ(図示せず)も含むことができる。

40

#### 【0058】

図5は、本開示の一実施形態による、ビデオデコーダ(510)のブロック図を示す。ビデオデコーダ(510)は、電子デバイス(530)に含まれることができる。電子デバイス(530)は、受信機(531)(例えば受信回路)を含むことができる。図4の例のビデオデコーダ(410)の代わりに、ビデオデコーダ(510)を用いることができる。

#### 【0059】

50

受信機（531）は、ビデオデコーダ（510）によって復号されるべき1つ以上のコーディングされたビデオシーケンスを受け取ることがあり；同じ又は別の実施形態では、一度に1つのコーディングされたビデオシーケンスを受け取り、この場合、各コーディングされたビデオシーケンスの復号は、他のコーディングビデオシーケンスとは独立である。コーディングされたビデオシーケンスはチャネル（501）から受け取られてよく、該チャネルは、符号化ビデオデータを記憶するストレージデバイスへのハードウェア／ソフトウェアリンクであり得る。受信機（531）は、他のデータ、例えばコーディングされたオーディオデータ及び／又は補助データストリームとともに、符号化ビデオデータを受け取ってよく、これらのデータは、それらのそれぞれの使用エンティティ（図示せず）に転送されてよい。受信機（531）は、コーディングされたビデオシーケンスを他のデータから分離し得る。ネットワークジッタをなくすために、バッファメモリ（515）が、受信機（531）とエントロピーデコーダ／パーサ（520）（以下、「パーサ（520）」）との間に結合され得る。特定の適用では、バッファメモリ（515）は、ビデオデコーダ（510）の一部である。他では、バッファメモリ（515）は、ビデオデコーダ（510）（図示せず）の外部にある可能性がある。更に他では、例えば再生タイミングを処理するためにビデオデコーダ（510）の内部にある別のバッファメモリ（515）に加えて、例えばネットワークジッタをなくすためにビデオデコーダ（510）の外部にバッファメモリ（図示せず）が存在する可能性がある。受信機（531）が、十分な帯域幅及び制御可能性を有するストア／転送デバイスから又は等同期（isosynchronous）ネットワークからデータを受信しているとき、バッファメモリ（515）は必要でないことがあり、あるいは小さいものとすることができます。インターネットのようなベストエフォート・パケットネットワークにおける使用のために、バッファメモリ（515）が必要とされることがあります、そのサイズは比較的大きいものとすることができます、有利には適応サイズとすることができます、少なくとも部分的に、ビデオデコーダ（510）の外部のオペレーティングシステム又は類似の要素（図示せず）において実装されることがある。

#### 【0060】

ビデオデコーダ（510）は、コーディングされたビデオシーケンスからシンボル（521）を再構成するためにパーサ（520）を含み得る。これらのシンボルのカテゴリは、ビデオデコーダ（510）の動作を管理するために使用される情報と、潜在的に、電子デバイス（530）の一体部分ではないが図5に示されたように電子デバイス（530）に結合されることができるレンダリングデバイス（512）（例えばディスプレイ画面）のようなレンダリングデバイスを制御する情報とを含む。レンダリングデバイスの制御情報は、補足強化情報（S E I : Supplemental Enhancement Informationメッセージ）又はビデオユーザビリティ情報（V U I : Video Usability Information）パラメータセットフラグメント（図示せず）の形式であってよい。パーサ（520）は、受信される、コーディングされたビデオシーケンスを構文解析／エントロピー復号し得る。コーディングされたビデオシーケンスのコーディングは、ビデオコーディング技術又は規格に従うことができ、可変長コーディング、ハフマンコーディング、文脈依存（context sensitivity）を伴うか伴わない算術コーディング等を含む、様々な原理に従うことができる。パーサ（520）は、コーディングされたビデオシーケンスから、ビデオデコーダ内のピクセルのサブグループのうちの少なくとも1つのサブグループについてのサブグループパラメータのセットを、グループに対応する少なくとも1つのパラメータに基づいて抽出することができる。サブグループは、ピクチャのグループ（G O P s : Groups of Pictures）、ピクチャ、タイル、スライス、マクロブロック、コーディングユニット（C U s : Coding Units）、ブロック、変換ユニット（T U s : Transform Units）、予測ユニット（P U s : Prediction Units）等を含むことができる。パーサ（520）はまた、変換係数、量子化パラメータ値、動きベクトル等のようなコーディングされたビデオシーケンス情報から抽出してもよい。

#### 【0061】

パーサ（520）は、シンボル（521）を生成するように、バッファメモリ（515）

10

20

30

40

50

) から受け取ったビデオシーケンスに対してエントロピー復号 / 構文解析動作を実行し得る。

**【 0 0 6 2 】**

シンボル ( 5 2 1 ) の再構成は、コーディングされたビデオピクチャ又はその部分のタイプ ( 例えばインター及びイントラピクチャ、インター及びイントラブロック ) 及び他のファクタに応じて、複数の異なるユニットに関与することができる。どのユニットがどのように関与するかは、コーディングされたビデオシーケンスからパーサ ( 5 2 0 ) によって構文解析されたサブグループ制御情報によって制御されることができる。パーサ ( 5 2 0 ) と下記の複数のユニットとの間のそのようなサブグループ制御情報のフローは、明確性のために図示されていない。

10

**【 0 0 6 3 】**

既に述べた機能ブロックの他に、ビデオデコーダ ( 5 1 0 ) は、以下で説明するように、複数の機能ユニットに概念的に細分されることができる。商業的制約の下で動作する実用的な実装では、これらのユニットの多くは互いに密接に対話し、少なくとも部分的に相互へ統合されることができる。しかしながら、開示される主題を説明する目的のために、以下では、機能ユニットへの概念的な細分化が適切である。

**【 0 0 6 4 】**

第 1 ユニットは、スケーラ / 逆変換ユニット ( 5 5 1 ) である。スケーラ / 逆変換ユニット ( 5 5 1 ) は、パーサ ( 5 2 0 ) からのシンボル ( 5 2 1 ) として、どの変換を使用すべきか、ブロックサイズ、量子化係数、量子化スケーリング行列等を含む制御情報だけでなく、量子化された変換係数も受け取る。スケーラ / 逆変換ユニット ( 5 5 1 ) は、アグリゲータ ( 5 5 5 ) に入力することができるサンプル値を含むブロックを出力することができる。

20

**【 0 0 6 5 】**

場合によっては、スケーラ / 逆変換 ( 5 5 1 ) の出力サンプルは、イントラコーディングされたブロック、すなわち、以前に再構成されたピクチャからの予測情報を使用していないが、現在のピクチャの以前に再構成された部分からの予測情報を使用することができるブロックに関連する可能性がある。このような予測情報は、イントラピクチャ予測ユニット ( 5 5 2 ) によって提供されることができる。場合によっては、イントラピクチャ予測ユニット ( 5 5 2 ) は、現在のピクチャバッファ ( 5 5 8 ) からフェッチされる、周囲の既に再構成された情報を使用して、再構成中のブロックの同じサイズ及び形状のブロックを生成する。現在のピクチャバッファ ( 5 5 8 ) は、例えば部分的に再構成された現在のピクチャ及び / 又は完全に再構成された現在のピクチャをバッファする。アグリゲータ ( 5 5 5 ) は、場合によっては、サンプルごとに、イントラ予測ユニット ( 5 5 2 ) が生成した予測情報を、スケーラ / 逆変換ユニット ( 5 5 1 ) によって提供される出力サンプル情報に追加し得る。

30

**【 0 0 6 6 】**

他の場合には、スケーラ / 逆変換ユニット ( 5 5 1 ) の出力サンプルは、インターフェーディングされて潜在的に動き補償されるブロックに関連する可能性がある。このような場合、動き補償予測ユニット ( 5 5 3 ) は、予測に使用されるサンプルをフェッチするために参照ピクチャメモリ ( 5 5 7 ) にアクセスすることができる。ブロックに関連するシンボル ( 5 2 1 ) に従って、フェッチされたサンプルを動き補償した後、これらのサンプルは、アグリゲータ ( 5 5 5 ) によって、出力サンプル情報を生成するために、スケーラ / 逆変換ユニット ( 5 5 1 ) の出力 ( この場合、残差サンプル又は残差信号と呼ばれる ) に追加されることができる。動き補償予測ユニット ( 5 5 3 ) が予測サンプルをフェッチする参照ピクチャメモリ ( 5 5 7 ) 内のアドレスは、例えば X 、 Y と参照ピクチャ成分を有することができるシンボル ( 5 2 1 ) の形態で、動き補償予測ユニット ( 5 5 3 ) に利用可能な動きベクトルによって、制御されることができる。動き補償はまた、サブサンプルの正確な動きベクトルが使用されているときに参照ピクチャメモリ ( 5 5 7 ) からフェッチされるサンプル値の補間、動きベクトル予測メカニズム等も含むことができる。

40

50

## 【0067】

アグリゲータ(555)の出力サンプルは、ループフィルタユニット(556)内の様々なループフィルタリング技術の対象となることができる。ビデオ圧縮技術は、コーディングされたビデオシーケンス(コーディングされたビデオビットストリームとも呼ばれる)に含まれるパラメータによって制御され、かつパーサ(520)からシンボル(521)としてループフィルタユニット(556)に利用可能にされるが、コーディングされたピクチャ又はコーディングされたビデオシーケンスの(復号順序で)以前の部分の復号中に取得されたメタ情報に応答することもできるとともに、以前に再構成されてループフィルタリングされたサンプル値に応答することができる、ループ内フィルタ技術を含むことができる。

10

## 【0068】

ループフィルタユニット(556)の出力は、レンダデバイス(512)に出力されることができ、かつ将来のインターピクチャ予測で使用するために参照ピクチャメモリ(557)内に記憶されることができる、サンプルストリームとすることができます。

## 【0069】

ある特定のコーディングされたピクチャは、いったん完全に再構成されると、将来の予測のための参照ピクチャとして使用されることができる。例えば現在のピクチャに対応するコーディングされたピクチャが完全に再構成され、コーディングされたピクチャが(例えばパーサ(520)によって)参照ピクチャとして識別されると、現在のピクチャバッファ(558)は参照ピクチャメモリ(557)の一部となることができ、フレッシュな現在のピクチャバッファは、後続のコーディングされたピクチャの再構成を開始する前に再割り当てされることができる。

20

## 【0070】

ビデオデコーダ(510)は、ITU-T Rec. H.265のような規格の所定のビデオ圧縮技術に従って復号動作を実行し得る。コーディングされたビデオシーケンスは、該コーディングされたビデオシーケンスが、ビデオ圧縮技術又は規格の構文と、ビデオ圧縮技術又は規格で文書化されるプロファイルとの両方を守るという意味において、使用されているビデオ圧縮技術又は規格によって指定された構文に準拠し得る。具体的には、プロファイルは、そのプロファイルの下での使用に使用可能な唯一のツールとして、特定のツールを選択することができる。また、コーディングされたビデオシーケンスの複雑さが、ビデオ圧縮技術又は規格のレベルによって定義される範囲内にあることが、準拠には必要であり得る。場合によっては、レベルは、最大ピクチャサイズ、最大フレームレート、最大再構成サンプルレート(例えば毎秒メガサンプル(megasamples)で測定される)、最大参照ピクセルサイズ等を制限する。レベルによって設定される限界(limit)は、場合によっては、コーディングされたビデオシーケンスでシグナリングされる仮想リファレンスデコーダ(HRD: Hypothetical Reference Decoder)のバッファ管理のためのHRD仕様及びメタデータを通して更に制限される可能性がある。

30

## 【0071】

実施形態では、受信機(531)は、符号化ビデオとともに追加の(冗長な)データを受け取ることがある。追加のデータは、コーディングされたビデオシーケンスの一部として含まれてよい。追加のデータは、データを適切に復号し、かつ/又は元のビデオデータをより正確に再構成するために、ビデオデコーダ(510)によって使用され得る。追加のデータは、例えば時間的、空間的又は信号対ノイズ比(SNR)強化レイヤ(enhancement layers)、冗長スライス、冗長ピクチャ、順方向誤り訂正コード等の形態とすることができます。

40

## 【0072】

図6は、本開示の実施形態による、ビデオエンコーダ(603)のブロック図を示す。ビデオエンコーダ(603)は、電子デバイス(620)に含まれる。電子デバイス(620)は、送信機(640)(例えば送信回路)を含む。図4の例のビデオエンコーダ(403)の代わりに、ビデオエンコーダ(603)を使用することができる。

50

## 【0073】

ビデオエンコーダ(603)は、ビデオサンプルを、ビデオエンコーダ(603)によってコーディングされるべきビデオ画像をキャプチャし得るビデオソース(601)(図6の例では電子デバイス(620)の一部ではない)から受け取り得る。別の例では、ビデオソース(601)は、電子デバイス(620)の一部である。

## 【0074】

ビデオソース(601)は、ビデオエンコーダ(603)によってコーディングされるべきソースビデオシーケンスを、任意の適切なビット深度(例えば8ビット、10ビット、12ビット、. . .)、任意の色空間(例えばBT.601 YCrCb、RGB、. . .)及び任意の適切なサンプリング構造(例えばYCrCb 4:2:0、YCrCb 4:4:4)とすることができるデジタルビデオサンプルストリームの形態で提供し得る。メディア供給システムにおいて、ビデオソース(601)は、事前に準備されたビデオを記憶するストレージデバイスであってよい。ビデオ会議システムでは、ビデオソース(601)は、ローカル画像情報をビデオシーケンスとしてキャプチャするカメラであってもよい。ビデオデータは、シーケンスで見るときに動きを伝える複数の個々のピクチャとして提供されてもよい。ピクチャ自体は、ピクセルの空間アレイとして編成されてよく、この場合、各ピクセルは、使用中のサンプリング構造、色空間等に応じて1つ以上のサンプルを含むことができる。当業者は、ピクセルとサンプルとの関係を容易に理解することができる。以下の説明は、サンプルに焦点を当てている。

## 【0075】

一実施形態によると、ビデオエンコーダ(603)は、リアルタイムで又はアプリケーションによって要求される任意の他の時間制約の下で、ソースビデオシーケンスのピクチャをコーディング及び圧縮して、コーディングされたビデオシーケンス(643)にすることができる。適切なコーディング速度を実施することは、コントローラ(650)の1つの機能である。いくつかの実施形態において、コントローラ(650)は、以下で説明されるように、他の機能ユニットを制御し、該他の機能ユニットに機能的に結合される。この結合は、明確性のために図示されていない。コントローラ(650)によって設定されるパラメータは、レート制御関連パラメータ(ピクチャスキップ、量子化器、レート歪み最適化技術のラムダ値、. . .)、ピクチャサイズ、ピクチャのグループ(GOP)のレイアウト、最大動きベクトル探索範囲等を含むことができる。コントローラ(650)は、特定のシステム設計のために最適化された、ビデオエンコーダ(603)と関係する他の適切な機能を有するように構成されることができる。

## 【0076】

いくつかの実施形態では、ビデオエンコーダ(603)は、コーディンググループで動作するように構成される。過剰に簡略化した説明として、一例では、コーディンググループは、ソースコーダ(630)(例えばコーディングされるべき入力ピクチャ及び参照ピクチャに基づいて、シンボルストリーム等のシンボルを作成することを担当する)と、ビデオエンコーダ(603)に埋め込まれた(ローカル)デコーダ(633)とを含むことができる。デコーダ(633)は、(リモート)デコーダも作成するのと同様の方法で、シンボルを再構成してサンプルデータを作成する(シンボルとコーディングされたビデオビットストリームとの間の任意の圧縮は、開示される主題において考慮されるビデオ圧縮技術で可逆である)。再構成されたサンプルストリーム(サンプルデータ)は、参照ピクチャメモリ(634)に入力される。シンボルストリームの復号は、デコーダ位置(ローカル又はリモート)とは独立のビット正確な結果(bit-exact results)をもたらすので、参照ピクチャメモリ(634)内のコンテンツも、ローカルエンコーダとリモートエンコーダとの間でビット正確である。言い換えると、エンコーダの予測部分は、デコーダが復号中に予測を使用するときに「見る」のとまったく同じサンプル値を参照ピクチャサンプルとして「見る」。参照ピクチャのシンクロニシティ(synchronicity)のこの基本原理(及び例えばチャネルエラーのためにシンクロニシティを維持することができない場合の結果として生じるドリフト)は、いくつかの関連する技術でも同様に使用される。

10

20

30

40

50

## 【0077】

「ローカル」デコーダ（633）の動作は、ビデオデコーダ（510）のような「リモート」デコーダと同じものとすることができる、これは、既に図5に関連して上述されている。しかしながら、図5も簡単に参照すると、シンボルが利用可能であり、エントロピーコーダ（645）及びパーサ（520）によるコーディングされたビデオシーケンスへのシンボルの符号化／復号は可逆であり得るので、バッファメモリ（515）及びパーサ（520）を含むビデオデコーダ（510）のエントロピー復号部分は、ローカルデコーダ（633）では完全には実装されないことがある。

## 【0078】

この時点で可能な観察は、デコーダに存在する解析／エントロピー復号を除く任意のデコーダ技術はまた、対応するエンコーダにおいて、実質的に同一の機能形態で必ず存在する必要がある。この理由のために、開示される主題はデコーダの動作に焦点を当てる。エンコーダ技術の説明は、網羅的に説明されるデコーダ技術の反対であるので、省略される可能性がある。特定の領域のみにおいて、より詳細な説明が必要とされ、以下で提供される。

10

## 【0079】

いくつかの例において、動作中に、ソースコーダ（630）は、動き補償予測コーディングを実行してよく、動き補償予測コーディングは、「参照ピクチャ」として指定されたビデオシーケンスからの1つ以上の以前にコーディングされたピクチャに関連して予測的に入力ピクチャをコーディングする。このようにして、コーディングエンジン（632）は、入力ピクチャのピクセルブロックと、入力ピクチャに対する予測参照として選択され得る参照ピクチャのピクセルブロックとの間の差をコーディングする。

20

## 【0080】

ローカルビデオデコーダ（633）は、ソースコーダ（630）によって作成されたシンボルに基づいて、参照ピクチャとして指定され得るピクチャのコーディングされたビデオデータを復号し得る。コーディングエンジン（632）の動作は、有利には、非可逆プロセスであり得る。コーディングされたビデオデータがビデオデコーダ（図6には図示せず）で復号され得るとき、再構成ビデオシーケンスは、典型的に、いくつかの誤差を伴うソースビデオシーケンスのレプリカであり得る。ローカルビデオデコーダ（633）は、参照ピクチャに対してビデオデコーダによって実行され得る復号処理を複製し、再構成参照ピクチャを参照ピクチャキャッシュ（634）に記憶させ得る。このようにして、ビデオエンコーダ（603）は、（伝送誤差なしに）遠端ビデオデコーダによって取得される再構成参照ピクチャとして、共通のコンテンツを有する再構成参照ピクチャのコピーを、ローカルに記憶し得る。

30

## 【0081】

予測器（635）は、コーディングエンジン（632）について予測探索を実行し得る。すなわち、コーディングされるべき新しいピクチャについて、予測器（635）は、参照ピクチャメモリ（634）から、サンプルデータ（候補参照ピクセルブロックとして）又は参照ピクチャ動きベクトル、ブロック形状等のような特定のメタデータを探索してよく、これらのデータは、新しいピクチャのための適切な予測参照として機能し得る。予測器（635）は、適切な予測参照を見つけるために、サンプルブロックとピクセルブロックごと（sample block-by-pixel block basis）に動作し得る。場合によっては、予測器（635）によって取得される検索結果によって決定されるように、入力ピクチャは、参照ピクチャメモリ（634）に記憶された複数の参照ピクチャから引き出された予測参照を有してよい。

40

## 【0082】

コントローラ（650）は、例えばビデオデータを符号化するために使用されるパラメータ及びサブグループパラメータの設定を含む、ソースコーダ（630）のコーディング動作を管理してもよい。

## 【0083】

50

前述の機能ユニットのすべての出力は、エントロピーコーダ (645) におけるエントロピーコーディングの対象となり得る。エントロピーコーダ (645) は、ハフマンコーディング、可変長コーディング、算術コーディング等のような技術に従ってシンボルを可逆圧縮することによって、様々な機能ユニットによって生成されるシンボルを、コーディングされたビデオシーケンスに変換する。

#### 【0084】

送信機 (640) は、エントロピーコーダ (645) によって作成されるコーディングされたビデオシーケンスをバッファして、通信チャネル (660) を介した伝送の準備を行ってよく、該通信チャネル (660) は、符号化されたビデオデータを記憶するストレージデバイスへのハードウェア / ソフトウェアリンクであってよい。送信機 (640) は、ビデオコーダ (603) からのコーディングされたビデオデータを、伝送されるべき他のデータ、例えばコーディングされたオーディオデータ及び / 又は補助データストリーム (ソースは図示せず) とマージし得る。

10

#### 【0085】

コントローラ (650) は、ビデオエンコーダ (603) の動作を管理し得る。コーディングの間、コントローラ (650) は、各コーディングされたピクチャに、特定のコーディングピクチャタイプ (coded picture type) を割り当ててよく、該コーディングピクチャタイプは、それぞれのピクチャに適用され得るコーディングに影響を与える。例えばピクチャは、しばしば、次のピクチャタイプのうちの 1 つとして割り当てられる：

20

#### 【0086】

イントラピクチャ (I ピクチャ) は、予測のソースとしてシーケンス内のいずれの他のピクチャも使用せずにコーディング及び復号され得るものであり得る。いくつかのビデオコーデックは、例えば独立デコーダリフレッシュ (「IDR : Independent Decoder Refresh」) ピクチャを含む、異なるタイプのイントラピクチャを許容する。当業者は、I ピクチャのこれらの変形並びにそれらのそれぞれの用途及び特徴を知っている。

#### 【0087】

予測ピクチャ (P ピクチャ) は、各ブロックのサンプル値を予測するために、最大 1 つの動きベクトルと参照インデックスを用いて、イントラ予測又はインター予測を使用して、コーディング及び復号され得るものであり得る。

30

#### 【0088】

双方向予測ピクチャ (B ピクチャ) は、各ブロックのサンプル値を予測するために、最大 2 つの動きベクトルと参照インデックスを用いて、イントラ予測又はインター予測を使用して、コーディング及び復号され得るものであり得る。同様に、複数予測ピクチャ (multiple-predictive pictures) は、単一のブロックの再構成のために、2 つより多くの参照ピクチャ及び関連するメタデータを使用することができる。

#### 【0089】

ソースピクチャは、通常、空間的に複数のサンプルブロック (例えば各々  $4 \times 4$ 、 $8 \times 8$ 、 $4 \times 8$  又は  $16 \times 16$  サンプルのブロック) に細分され、ブロックごとにコーディングされ得る。ブロックは、ブロックのそれぞれのピクチャに適用されるコーディング割り当てによって決定されるように、他の (既にコーディングされた) ブロックに関連して予測的にコーディングされ得る。例えば I ピクチャのブロックは、非予測的にコーディングされてもよく、あるいはそれらは、同じピクチャの既にコーディングされたブロックに関連して予測的にコーディングされてもよい (空間予測又はイントラ予測)。P ピクチャのピクセルブロックは、以前にコーディングされた 1 つの参照ピクチャに関連して、空間予測を介して又は時間予測を介して予測的にコーディングされ得る。B ピクチャのブロックは、1 つ又は 2 つの以前にコーディングされた参照ピクチャに関連して、空間予測を介して又は時間予測を介して予測的にコーディングされ得る。

40

#### 【0090】

ビデオエンコーダ (603) は、ITU-T Rec. H.265 のような所定のビデオコーディング技術又は規格に従ってコーディング動作を実行し得る。その動作において

50

、ビデオエンコーダ(603)は、入力ビデオシーケンスにおける時間的及び空間的冗長性を利用する予測コーディング動作を含む、様々な圧縮動作を実行し得る。コーディングされたビデオデータは、したがって、使用されているビデオコーディング技術又は規格によって指定された構文に従うことができる。

#### 【0091】

一実施形態では、送信機(640)は、符号化されたビデオとともに追加データを送信してもよい。ソースコーダ(630)は、コーディングされたビデオシーケンスの一部としてそのようなデータを含んでよい。追加データは、時間／空間／S N R 強化レイヤ、冗長ピクチャ及びスライス、S E I メッセージ、V U I パラメータセットフラグメント等のような他の形式の冗長データを含んでもよい。

10

#### 【0092】

ビデオは、時間シーケンスにおいて複数のソースピクチャ(ビデオピクチャ)としてキャプチャされ得る。イントラピクチャ予測(しばしば、イントラ予測と略される)は、所与のピクチャにおける空間的相関を使用し、インターピクチャ予測は、ピクチャ間の(時間的又は他の)相関を使用する。一例では、符号化／復号中の特定のピクチャは、現在のピクチャと呼ばれ、ブロックに区分化される。現在のピクチャ内のブロックが、ビデオ内の以前にコーディングされて依然としてバッファされている参照ピクチャ内の参照ブロックに類似するとき、現在のピクチャ内のブロックは、動きベクトルと呼ばれるベクトルによってコーディングされることができる。動きベクトルは、参照ピクチャ内の参照ブロックを指し、複数の参照ピクチャが使用されているケースでは、参照ピクチャを識別する第3次元(third dimension)を有することができる。

20

#### 【0093】

いくつかの実施形態では、インターピクチャ予測において双予測技術を使用することができる。双予測技術によると、第1参照ピクチャと第2参照ピクチャのように、両方とも、復号順序でビデオ内の現在のピクチャに先行する(ただし、表示順序では、それぞれ、過去及び将来であり得る)2つの参照ピクチャが使用される。現在のピクチャ内のブロックを、第1参照ピクチャ内の第1参照ブロックを指す第1動きベクトルと、第2参照ピクチャ内の第2参照ブロックを指す第2動きベクトルとによってコーディングすることができる。ブロックは、第1参照ブロックと第2参照ブロックの組合せによって、予測されることができる。

30

#### 【0094】

さらに、コーディング効率を改善するために、インターピクチャ予測においてマージモード技術を使用することができる。

#### 【0095】

本開示のいくつかの実施形態によると、インターピクチャ予測及びイントラピクチャ予測のような予測は、ブロック単位(unit)で実行される。例えばH E V C 規格によると、ビデオピクチャのシーケンス内のピクチャは、圧縮のためにコーディングツリーユニット(C T U)に区分化され、ピクチャ内のC T U sは、 $64 \times 64$ ピクセル、 $32 \times 32$ ピクセル又は $16 \times 16$ ピクセルのように、同じサイズを有する。一般に、C T Uは、3つのコーディングツリーブロック(C T B s)を含み、該3つのC T Bは、1つのルマ(luma)C T Bと2つのクロマ(chroma)C T Bである。各C T Uは、1つ又は複数のコーディングユニット(C U s)に再帰的に四分木分裂(quadtree split)することができる。例えば $64 \times 64$ ピクセルのC T Uを、 $64 \times 64$ ピクセルの1つのC Uに又は $32 \times 32$ ピクセルの4つのC Uに又は $16 \times 16$ ピクセルの16個のC Uに分裂することができる。一例では、各C Uを分析して、インター予測タイプ又はイントラ予測タイプのような、C Uの予測タイプを決定する。C Uは、時間的及び／又は空間的予測可能性に依存して、1つ以上の予測ユニット(P U s)に分裂される。一般に、各P Uはルマ予測ブロック(P B)と2つのクロマP B sを含む。一実施形態では、コーディング(符号化／復号)における予測動作は、予測ブロックのユニットにおいて実行される。予測ブロックの例としてルマ予測ブロックを使用すると、予測ブロックは、 $8 \times 8$ ピクセル、 $16 \times 16$

40

50

ピクセル、 $8 \times 16$  ピクセル、 $16 \times 8$  ピクセル等のような、ピクセルについての値（例えばルマ値）の行列を含んでよい。

【 0 0 9 6 】

図 7 は、本開示の別の実施形態によるビデオエンコーダ（703）の図を示す。ビデオエンコーダ（703）は、ビデオピクチャのシーケンス内の現在のビデオピクチャ内のサンプル値の処理ブロック（例えば予測ブロック）を受け取り、処理ブロックを符号化して、コーディングされたビデオシーケンスの一部であるコーディングされたピクチャにするように構成される。一例では、ビデオエンコーダ（703）は、図 4 の例のビデオエンコーダ（403）の代わりに使用される。

【 0 0 9 7 】

HEVC の例では、ビデオエンコーダ（703）は、 $8 \times 8$  サンプルの予測ブロック等の処理ブロックのサンプル値の行列を受け取る。ビデオエンコーダ（703）は、処理ブロックが、例えばレート歪み最適化（rate-distortion optimization）を使用して、イントラモード、インターモード又は双予測（bi-prediction）モードを使用して最も良くコーディングされるかどうかを決定する。処理ブロックがイントラモードでコーディングされるべきであるとき、ビデオエンコーダ（703）は、イントラ予測技術を使用して、処理ブロックを符号化して、コーディングされたピクチャにしてよく、処理ブロックがインターモード又は双予測モードでコーディングされるべきであるとき、ビデオエンコーダ（703）は、それぞれインター予測技術又は双予測技術を使用して、処理ブロックを符号化して、符号化ピクチャにしてよい。特定のビデオコーディング技術では、マージモードは、予測子の外側のコーディングされた動きベクトル構成要素の利益を伴わずに、動きベクトルが 1 つ以上の動きベクトル予測子から導出される場合、インターピクチャ予測サブモードであり得る。特定の他のビデオコーディング技術では、対象ブロックに適用可能な動きベクトル構成要素が存在してもよい。一例では、ビデオエンコーダ（703）は、処理ブロックのモードを決定するために、モード決定モジュール（図示せず）のような他の構成要素を含む。

【 0 0 9 8 】

図 7 の例では、ビデオエンコーダ（703）は、図 7 に示されるように一緒に結合される、インターベンコーダ（730）、イントラエンコーダ（722）、残差計算器（723）、スイッチ（726）、残差エンコーダ（724）、一般コントローラ（721）及びエントロピーエンコーダ（725）を含む。

【 0 0 9 9 】

インターベンコーダ（730）は、現在のブロック（例えば処理ブロック）のサンプルを受け取り、該ブロックを参照ピクチャ内の 1 つ以上の参照ブロック（例えば前のピクチャ及び後のピクチャ内のブロック）と比較し、インター予測情報（例えばインター符号化技術に従った冗長情報の記述、動きベクトル、マージモード情報）を生成し、任意の適切な技術を使用して、インター予測情報に基づいてインター予測結果（例えば予測ブロック）を計算するよう構成される。いくつかの例では、参照ピクチャは、符号化されたビデオ情報に基づいて復号される復号参照ピクチャである。

【 0 1 0 0 】

イントラエンコーダ（722）は、現在のブロック（例えば処理ブロック）のサンプルを受け取り、場合によっては、該ブロックを、同じピクチャ内の既にコーディングされたブロックと比較し、変換後に量子化された係数と、場合によってはイントラ予測情報（例えば 1 つ以上のイントラ符号化技術に従ったイントラ予測方向情報）も生成するよう構成される。一例では、イントラエンコーダ（722）はまた、イントラ予測情報と、同じピクチャ内の参照ブロックに基づいて、イントラ予測結果（例えば予測ブロック）も計算する。

【 0 1 0 1 】

一般コントローラ（721）は、一般制御データを決定し、該一般制御データに基づいてビデオエンコーダ（703）の他の構成要素を制御するよう構成される。一例では、一

10

20

30

40

50

般コントローラ(721)は、ブロックのモードを決定し、該モードに基づいて制御信号をスイッチ(726)に提供する。例えばモードがイントラモードであるとき、一般コントローラ(721)は、残差計算器(723)が使用するためのイントラモードの結果を選択するようにスイッチ(726)を制御し、イントラ予測情報を選択して該イントラ予測情報をビットストリームに含めるようにエントロピーエンコーダ(725)を制御し、モードがインターモードであるとき、一般コントローラ(721)は、残差計算器(723)が使用するためのインター予測結果を選択するようにスイッチ(726)を制御し、インター予測情報を選択して該インター予測情報をビットストリームに含めるようにエントロピーエンコーダ(725)を制御する。

#### 【0102】

残差計算器(723)は、イントラエンコーダ(722)又はインターベンコーダ(730)から選択された、受け取ったブロックと予測結果との差(残差データ)を計算するよう構成される。残差エンコーダ(724)は、残差データに基づいて動作し、残差データを符号化して変換係数を生成するよう構成される。一例では、残差エンコーダ(724)は、残差データを空間領域から周波数領域に変換して、変換係数を生成するよう構成される。次いで、変換係数は、量子化処理の対象となり、量子化された変換係数を取得する。様々な実施形態では、ビデオエンコーダ(703)は、残差デコーダ(728)も含む。残差デコーダ(728)は、逆変換を実行し、復号された残差データを生成するよう構成される。復号された残差データは、イントラエンコーダ(722)及びインターベンコーダ(730)によって適切に使用することができる。例えばインターベンコーダ(730)は、復号された残差データ及びインター予測情報に基づいて、復号されたブロックを生成することができ、イントラエンコーダ(722)は、復号された残差データ及びイントラ予測情報に基づいて、復号されたブロックを生成することができる。復号されたブロックは、復号されたピクチャを生成するために適切に処理され、復号されたピクチャは、メモリ回路(図示せず)内でバッファリングされ、いくつかの例では、参照ピクチャとして使用されることができる。

#### 【0103】

エントロピーエンコーダ(725)は、符号化ブロックを含むようにビットストリームをフォーマットするように構成される。エントロピーエンコーダ(725)は、HEVC規格のような適切な規格に従って、様々な情報を含めるように構成される。一例では、エントロピーエンコーダ(725)は、一般制御データ、選択された予測情報(例えばイントラ予測情報又はインター予測情報)、残差情報及び他の適切な情報をビットストリーム内に含めるよう構成される。開示される主題によると、インターモード又は双予測モードのいずれかのマージサブモードでブロックをコーディングするとき、残差情報がないことに留意されたい。

#### 【0104】

図8は、本開示の別の実施形態による、ビデオデコーダ(810)の図を示す。ビデオデコーダ(810)は、コーディングされたビデオシーケンスの一部である、コーディングされたピクチャを受け取り、コーディングされたピクチャを復号して再構成ピクチャを生成するよう構成される。一例では、図4の例のビデオデコーダ(410)の代わりにビデオデコーダ(810)が使用される。

#### 【0105】

図8の例では、ビデオデコーダ(810)は、図8に示されるように一緒に結合される、エントロピーデコーダ(871)、インターデコーダ(880)、残差デコーダ(873)、再構成モジュール(874)及びイントラデコーダ(872)を含む。

#### 【0106】

エントロピーデコーダ(871)は、コーディングされたピクチャから、該コーディングされたピクチャが構成される構文要素を表す特定のシンボルを再構成するよう構成されることができる。そのようなシンボルは、例えばブロックがコーディングされるモード(例えばイントラモード、インターモード、双予測モード、後者2つのマージサブモード又

10

20

30

40

50

は別のサブモード等)、イントラデコーダ(872)又はインターデコーダ(880)によってそれぞれ予測のために使用される特定のサンプル又はメタデータを識別することができる予測情報(例えばイントラ予測情報又はインター予測情報等)、例えば量子化された変換係数の形態の残差情報等を含むことができる。一例では、予測モードがインターモード又は双予測モードであるとき、インター予測情報がインターデコーダ(880)に提供され、予測モードがイントラ予測モードであるとき、イントラ予測情報がイントラデコーダ(872)に提供される。残差情報は、逆量子化の対象となり得、残差デコーダ(873)に提供される。

【0107】

インターデコーダ(880)は、インター予測情報を受け取り、該インター予測情報に基づいてインター予測結果を生成するよう構成される。

10

【0108】

イントラデコーダ(872)は、イントラ予測情報を受け取り、該イントラ予測情報に基づいて予測結果を生成するよう構成される。

【0109】

残差デコーダ(873)は、逆量子化を実行して非量子化(de-quantized)変換係数を抽出し、非量子化変換係数を処理して残差を周波数領域から空間領域に変換するよう構成され得る。残差デコーダ(873)はまた、特定の制御情報(量子化器パラメータ(QP: Quantizer Parameter)を含む)も要求することがあり、この情報は、エントロピーデコーダ(871)によって提供され得る(これは低量の制御情報のみであり得るので、データ経路は図示されない)。

20

【0110】

再構成モジュール(874)は、空間領域において、残差デコーダ(873)による出力としての残差と、(場合によっては、インター又はイントラ予測モジュールによる出力としての)予測結果とを組み合わせて再構成ブロックを形成するよう構成され、該再構成ブロックは、再構成ビデオの一部であり得る再構成ピクチャの一部であり得る。視覚品質を改善するために、デブロッキング操作等の他の適切な操作を実行することができることに留意されたい。

【0111】

ビデオエンコーダ(403)、(603)及び(703)、並びにビデオデコーダ(410)、(510)及び(810)は、任意の適切な技術を使用して実装され得ることに留意されたい。一実施形態では、ビデオエンコーダ(403)、(603)及び(703)、並びにビデオデコーダ(410)、(510)及び(810)は、1つ以上の集積回路を使用して実装されることができる。別の実施形態では、ビデオエンコーダ(403)、(603)及び(703)、並びにビデオデコーダ(410)、(510)及び(810)は、ソフトウェア命令を実行する1つ以上のプロセッサを使用して実装されることができる。

30

【0112】

本開示の態様は、コーディングされたビデオストリーム内の制約フラグを用いてコーディングツール及び機能性の制御技術を提供する。

40

【0113】

本開示の態様によると、ビットストリーム内のピクチャサイズは同じままであっても、変化してもよい。いくつかの関連する例では、ビデオエンコーダ及びデコーダは、コーディングされたビデオシーケンス(CVS)、ピクチャのグループ(GOP)又は同様のマルチピクチャ時間フレームに対して定義されて、一定のままである、所与のピクチャサイズで動作することができる。MPEG-2のような一例では、システム設計は、シーンのアクティビティのようなファクタに応じて水平解像度(したがって、ピクチャサイズ)を変更することが知られているが、Iピクチャのみでは、ピクチャサイズは、GOPに対して定義されて、典型的には一定のままである。CVS内で異なる解像度を使用するための参照ピクチャの再サンプリングは、ITU-T Rec. H.263 Annex Pか

50

ら知られている。しかしながら、C V S 内のピクチャサイズは変化せず、参照ピクチャのみが再サンプリングされ、結果として潜在的に、ピクチャキャンバスの一部のみが使用されることになるか（例えばダウンサンプリングの場合）あるいはシーンの一部のみがキャプチャされたことになる（例えばアップサンプリングの場合）。H. 2 6 3 A n n e x Q のようないくつかの例では、個々のマクロブロックを各次元（例えば上方又は下方）で 2 倍にサンプリングすることが許容される。しかしながら、ピクチャサイズは同じままである。例えば H. 2 6 3 でマクロブロックのサイズを固定することができるとき、マクロブロックのサイズをシグナリングする必要はない。

#### 【 0 1 1 4 】

いくつかの関連する例では、予測されたピクチャのピクチャサイズを変更することができる。V P 9 のような例では、参照ピクチャの再サンプリングと、ピクチャ全体の解像度の変更が可能になる。いくつかの例（例えば H e n d r y 等著「O n a d a p t i v e r e s o l u t i o n c h a n g e ( A R C ) f o r V V C 」、J o i n t V i d e o T e a m d o c u m e n t J V E T - M 0 1 3 5 、J a n 9 - 1 9 、2 0 1 9 年、これらはその全体が本明細書に組み込まれる）では、異なる解像度（例えばより高解像度又はより低解像度）への参照ピクチャ全体の再サンプリングが可能になる。異なる候補解像度を、シーケンスパラメータセット（S P S ）でコーディングすることができ、ピクチャパラメータセット（P P S ）のピクチャごとの構文要素によって参照することができる。

#### 【 0 1 1 5 】

本開示の態様によると、ソースビデオを、異なる解像度のような異なる品質を有する 1 つ以上のレイヤを含むビットストリームへ、ピクチャを符号化することができる、レイヤコーディング（l a y e r e d c o d i n g ）によって圧縮することができる。ビットストリームは、デコーダ側で出力することができるレイヤ（又はレイヤのセット）を指定する構文要素を有することができる。出力すべきレイヤのセットを、出力レイヤセットとして定義することができる。例えば複数のレイヤとスケーラビリティをサポートするビデオコーデックでは、1 つ以上の出力レイヤセットをビデオパラメータセット（V P S ）でシグナリングすることができる。ビットストリーム全体又は 1 つ以上の出力レイヤセットについてプロファイル階層レベル（P T L : p r o f i l e t i e r l e v e l ）を指定する構文要素を、V P S 、いくつかの例でデコーダ能力情報（D C I : d e c o d e r c a p a b i l i t y i n f o r m a t i o n ）と呼ばれることがあるデコーダパラメータセット（D P S ）、S P S 、P P S 、S E I メッセージ等でシグナリングすることができる。P T L 情報には、コーディングツール又は機能性に対する制約を指定することができる一般制約情報が存在する可能性がある。様々なコーディングツール及び機能性の制約情報を効率的に表現し、シグナリングすることができる。

#### 【 0 1 1 6 】

いくつかの例では、「サブピクチャ」という用語は、サンプル、ブロック、マクロブロック、コーディングユニット、あるいは例えば意味的にグループ化され、変更された解像度で独立にコーディングされ得る類似のエンティティの長方形の配置を指すために使用され得る。1 つ以上のサブピクチャが、ピクチャを形成することができる。1 つ以上のコーディングされたサブピクチャが、コーディングされたピクチャを形成することができる。1 つ以上のサブピクチャを 1 つのピクチャに組み立てることができ、1 つ以上のサブピクチャを 1 つのピクチャから抽出することができる。いくつかの例では、1 つ以上のコーディングされたサブピクチャを、コーディングされたピクチャへのサンプルレベルにトランスクードすることなく、圧縮領域に組み立てることができる。いくつかの例では、1 つ以上のコーディングされたサブピクチャを、圧縮領域内のコーディングされたピクチャから抽出することができる。

#### 【 0 1 1 7 】

いくつかの例では、例えば参照ピクチャ再サンプリングによって C V S 内のピクチャ又はサブピクチャの解像度の変更を可能にするメカニズムを、適応解像度変更（A R C : a d a p t i v e r e s o l u t i o n c h a n g e ）と呼ぶことができる。適応解像度変更を行うために使用さ

10

20

30

40

50

れる制御情報を、 A R C パラメータと呼ぶことができる。 A R C パラメータは、フィルタ パラメータ、スケーリング係数、出力及び / 又は参照ピクチャの解像度、様々な制御フラグ及び / 又は同様のものを含むことができる。

#### 【 0 1 1 8 】

いくつかの例では、 A R C の符号化 / 復号はピクチャ単位であり、したがって、制御情報 ( A R C パラメータ ) のセットが、単一の意味的に独立したコーディングされたビデオ ピクチャを符号化 / 復号するために使用される。いくつかの例では、 A R C の符号化 / 復号はサブピクチャ単位であり、したがって、ピクチャ内の複数のサブピクチャを、独立した A R C パラメータで符号化 / 復号することができる。様々な技術を使用して A R C パラメータをシグナリングすることができることに留意されたい。

10

#### 【 0 1 1 9 】

図 9 は、本開示のいくつかの実施形態による、 A R C パラメータをシグナリングするための技術の実施例 ( 例えはオプション ) を示している。コーディング効率、複雑さ及びアーキテクチャは、当該実施例によって異なる可能性がある。ビデオコーディング規格又は技術は、 A R C パラメータをシグナリングするために、これらの例のうちの 1 つ以上又は他の変形を選択することができる。実施例は相互に排他的ではなく、適用のニーズ、標準技術、エンコーダの選択及び / 又は同様のものに基づいて交換されてもよい。

#### 【 0 1 2 0 】

本開示の一態様によると、 A R C パラメータは、様々な方法で A R C パラメータのクラスとして提供され得る。いくつかの例では、 A R C パラメータのクラスは、 X 次元と Y 次元で分離又は結合される、アップサンプル及び / 又はダウンサンプル係数を含む。一例では、アップサンプル及び / 又はダウンサンプル係数を含むテーブルをポイントすることができる 1 つ以上の短い構文要素を、コーディングすることができる。

20

#### 【 0 1 2 1 】

いくつかの例では、 A R C パラメータのクラスは、アップサンプル及び / 又はダウンサンプル係数を含み、時間的な次元が追加されており、所与の数のピクチャに対する一定速度のズームイン又はズームアウトを示す。一例では、時間的な次元が追加されたアップサンプル及び / 又はダウンサンプル係数を含むテーブルを指すことができる 1 つ以上の短い構文要素を、コーディングすることができる。

#### 【 0 1 2 2 】

30

いくつかの例では、 A R C パラメータのクラスは、サンプル、ブロック、マクロブロック、 C U 又は任意の他の適切な粒度の単位で、組合せ又は別個に、入力ピクチャ、出力ピクチャ、参照ピクチャ、コーディングされたピクチャの X 次元又は Y 次元の解像度を含む。いくつかの例では、ビデオコーディングで使用される 2 つ以上の解像度があり ( 例えは入力ピクチャのための 1 つの解像度と、参照ピクチャのための別の解像度 ) 、 ( 解像度のうちの 1 つに対応する ) 値のセットを、 ( 解像度のうちの別のものに対応する ) 別の値のセットから推論することができる。値の決定は、例えはフラグの使用に基づいてゲーティング ( gate ) されることができる。ゲーティングのためのフラグの使用については、更なる説明において詳細に説明される。

#### 【 0 1 2 3 】

40

いくつかの例では、 A R C パラメータのクラスは、上述のような適切な粒度で H.263 Annex P で使用されているものと同様のワーピング座標 ( warping coordinates ) を含む。 H.263 Annex P は、ワーピング座標をコーディングする効率的な方法を定義する。他の効率的な方法を考案することができる。例えは Annex P のワーピング座標の可変長可逆ハフマンスタイルのコーディングを、適切な長さのバイナリコーディングによって置き換えることができ、ここで、バイナリコードワードの長さは、最大ピクチャサイズの境界の外部でのワーピングを可能にするために、係数を掛けて値をオフセットした最大ピクチャサイズから導出されることができる。

#### 【 0 1 2 4 】

いくつかの例では、 A R C パラメータのクラスは、アップサンプル及び / 又はダウンサ

50

ンプルフィルタパラメータを含む。一例では、アップサンプリング及び／又はダウンサンプリングのための単一のフィルタのみが存在する。別の例では、複数のフィルタを使用することができる。いくつかの例では、フィルタパラメータは、フィルタ設計の更なる柔軟性を可能にするためにシグナリングされ得る。可能なフィルタ設計のリストのインデックスを使用することによって、フィルタパラメータを選択することができる。フィルタは、（例えば適切なエントロピーコーディング技術を使用して、フィルタ係数のリストを指定することによって）完全に指定され得るか、フィルタは、上述のメカニズムのいずれか及び／又は同様のものに従ってシグナリングされるアップサンプル又はダウンサンプルの比を通して暗黙的に選択され得る。

#### 【0125】

以下の説明では、アップサンプル又はダウンサンプル係数の有限セット（X次元とY次元の両方で使用される同じ係数）を使用して、コードワードを通したARCパラメータのシグナリングを例示する。いくつかの例では、コードワードは、ビデオコーディング仕様（例えばH.264及びH.265）における特定の構文要素のためにExt-Golombコードを使用して、可変長コーディングされることできる。

#### 【0126】

図10は、アップサンプル又はダウンサンプル係数、コードワード及びExt-Golombコードのマッピングの表（1000）の例を示す。

#### 【0127】

他の同様のマッピングを、ビデオ圧縮技術又は規格で利用可能なアップスケール及びダウンスケールメカニズムの適用及び能力に従って考案することができることに留意されたい。いくつかの例では、表1を追加の値に適切に拡張することができる。Ext-Golombコード以外のエントロピーコーディングメカニズムによって、例えばバイナリコーディングを使用することによって、値を表すことができることに留意されたい。一例では、Ext-Golombコード以外のエントロピーコーディングメカニズムは、再サンプリング係数が、例えばメディア認識ネットワーク要素（MANE：media-aware network elements）によって、ビデオ処理エンジン（例えばエンコーダ及びデコーダ）の外部で関心があるとき、特定の利点を有することがある。いくつかの例では、解像度変更が必要とされない（例えば表1で元の／目標解像度が1である）とき、短いExt-Golombコード（例えば表1に示されている単一ビットのみ）を選択することができ、これは、例えば最も一般的な場合にバイナリコードを使用することよりも、コーディング効率の利点を有することができる。

#### 【0128】

本開示の態様によると、表1のようなマッピングテーブルは構成可能なものとすることができます。例えば表1のエントリ数及び対応する意味は、完全に又は部分的に構成可能なものとすることができます。いくつかの例では、マッピングテーブルの基本的な概要は、SPS又はDPSのような高レベルのパラメータセットで伝達される。あるいは又は追加として、いくつかの例では、表1と同様の1つ以上のテーブルがビデオコーディング技術又は規格で定義されてよく、テーブルのうちの1つが、例えばSPS又はDPSを通して選択されてもよい。

#### 【0129】

上述のようにコーディングされるアップサンプル又はダウンサンプル係数のようなARC情報が、ビデオコーディング技術又は規格構文に含まれてもよい。1つ以上のコードワードを使用して、アップサンプル又はダウンサンプルフィルタのような他のクラスのARC情報を制御することができることに留意されたい。いくつかの例では、フィルタ又は他のデータ構造に比較的大量のデータが必要とされる。

#### 【0130】

図9を参照すると、H.263 Annex Pのような例（910）では、ARC情報（912）は4つのワーピング座標の形式とすることことができ、H.263 PLUS PTY PE（913）ヘッダ拡張のようなピクチャヘッダ（911）に含まれる。例（910）

10

20

30

40

50

は、i) ピクチャヘッダが利用可能であり、かつii) ARC情報の頻繁な変更が予想されるときに適用されることがある。しかしながら、例(910)に示されるようなH.263スタイルのシグナリングを使用するときのオーバーヘッダは高い可能性があり、ピクチャヘッダが一時的な性質である可能性があるため、スケーリング係数はピクチャ境界間で適用可能でないことがある。

#### 【0131】

図9を参照すると、JVCE-T-M135-v1のような例(920)では、ARC参照情報(925)(例えばインデックス)は、PPS(924)内に配置されることがで  
き、目標解像度(例えば解像度1~3)を含むテーブル(又は目標解像度テーブル)(926)を指すことができる。一例では、テーブル(926)はSPS(927)内に配置される。テーブル(926)内の目標解像度をSPS(927)内に配置することは、能力交換中の相互運用性ネゴシエーションポイントとしてSPSを使用することによって正当化することができる。解像度は、適切なPPS(924)内の参照(例えばARC参照情報(925))によって、テーブル(926)内の値(例えば解像度1~3)の限られたセット内で、あるピクチャから別のピクチャに変更することができる。

10

#### 【0132】

図9はまた、ARC情報をビデオビットストリームで伝達するために使用され得る、例(930)、(940)及び(950)のような追加の技術も示す。これらの技術は、同じビデオコーディング技術又は規格において、個々に使用されてもよく、適切な組合せで使用されることもできる。

20

#### 【0133】

図9を参照すると、例(930)では、再サンプリング係数(又はズーム係数)のようなARC情報を(939)が、スライスヘッダ、GOBヘッダ、タイルヘッダ、タイルグループヘッダ等のようなヘッダ内に存在し得る。例えばタイルグループヘッダ(938)が図9に図示されている。例(930)によって図示される技術は、ARC情報を(939)を単一の可変長ue(v)又は数ビットの固定長コードワードのような少ないビット数でコーディングすることができるときに使用されることがある。

#### 【0134】

本開示の態様によると、ARC情報を(939)をヘッダ(例えば図9のタイルグループヘッダ(938)、スライスヘッダ又はタイルヘッダ)に直接有することは、ARC情報を(939)が、ピクチャ全体ではなく、例えば対応するタイルグループ(又はスライス、タイル)によって表されるサブピクチャに適用可能であり得るという、追加の利点を有することができる。加えて、一例では、ビデオ圧縮技術又は規格が、(例えばタイルグループベースの適応解像度変更とは対照的に)全体ピクチャの適応解像度変更のみを想定している場合であっても、例(930)は、エラー回復性の観点から、例(910)よりも特定の利点を有することができる。

30

#### 【0135】

図9を参照すると、例(940)では、ARC情報を(942)が、PPS、ヘッダパラメータセット、タイルパラメータセット、適応パラメータセット(APS)等のようなパラメータセット(941)内に存在し得る。例えばAPS(941)が図9に図示されている。いくつかの例では、パラメータセット(941)の範囲は、ピクチャよりも大きくない可能性があり、例えばピクチャ、タイルグループ等とすることができます。ARC情報を(例えばARC情報を(942))の使用は、関連するパラメータセット(例えばAPS(941))のアクティブ化を通して暗黙的にすることができます。例えばビデオコーディング技術又は規格がピクチャベースのARCのみを考慮するとき、PPS又は同等のものが適切であり得る。

40

#### 【0136】

図9を参照すると、例(950)では、ARC参照情報(953)が、上述のような、タイルグループヘッダ(954)又は同様のデータ構造(例えばピクチャヘッダ、スライスヘッダ、タイルヘッダ又はGOPヘッダ)内に存在し得る。例えばタイルグループヘッ

50

ダ(954)が図9に図示されている。ARC参照情報(953)は、例えばSPS、DPS等のような、単一のピクチャを超える範囲を有するパラメータセット(956)において利用可能なARC情報(955)のサブセットを参照することができる。一例として、SPS(956)が図9に図示されている。

#### 【0137】

図11は、本開示のいくつかの実施形態によるARCパラメータシグナリングのいくつかの例を示す。図11は、ビデオコーディング規格で使用される構文図の例を示す。一例では、構文図の表記は大まかにCスタイルのプログラミングに従う。太字の行はビットストリームに存在する構文要素を示すことができ、太字を用いていない行は制御フロー又は変数の設定を示すことができる。

10

#### 【0138】

図11を参照すると、タイルグループヘッダ(1101)は、ピクチャの一部(例えば長方形の部分)に適用可能なヘッダの構文構造を含む。一例では、タイルグループヘッダ(1101)は、条件付きで、可変長の `Exp-Golomb` コーディングされた構文要素 `dec_pic_size_idx(1102)` (太字で示される) を含むことができる。タイルグループヘッダ(1101)内の構文要素(例えば `dec_pic_size_idx(1102)`)の存在を、例えばフラグ(例えば `adaptive_pic_resolution_change_flag(1103)`)によって表される適応解像度に基づいてゲーティングすることができる。フラグ(例えば `adaptive_pic_resolution_change_flag(1103)`)の値は太字で示されておらず、したがって、該フラグは、該フラグが構文図内で発生するポイントでビットストリーム内に存在する。ピクチャ又はピクチャの一部に適応解像度が使用されているかどうかを、ビットストリームの内部又は外部の高レベルの構文構造(例えば図11のSPS(1110))においてシグナリングすることができる。

20

#### 【0139】

図11を参照すると、SPS(1110)の抜粋が示されている。SPS(1110)は、フラグ(1111)(例えば `adaptive_pic_resolution_change_flag`)である第1構文要素(1111)を含む。フラグ(1111)が真(true)であるとき、フラグ(1111)は、特定の制御情報を必要とし得る適応解像度の使用を示すことができる。一例では、SPS(1110)内のif()ステートメント(1112)とタイルグループヘッダ(1101)によって示されるように、特定の制御情報はフラグ(1111)の値に基づいて条件付きで存在する。

30

#### 【0140】

図11の例に示されるように、適応解像度が使用されているとき、サンプル単位の出力解像度(又は出力ピクチャの解像度)(1113)をコーディングすることができる。一例では、出力解像度(1113)は、幅解像度(例えば `output_pic_width_in_luma_samples`)と高さ解像度(例えば `output_pic_height_in_luma_samples`)に基づいてコーディングされる。ビデオコーディング技術又は規格では、出力解像度(1113)の値に対する特定の制限を定義することができる。例えばレベル定義では、合計出力サンプルの数(例えば `output_pic_width_in_luma_samples` と `output_pic_height_in_luma_samples` の積)を限定し得る。いくつかの例では、ビデオコーディング技術又は規格あるいは外部技術又は規格(例えばシステム規格)は、幅解像度及び/又は高さ解像度の番号付けの範囲(例えば幅解像度及び/又は高さ解像度は2の累乗で割り切れる)、高さ解像度に対する幅解像度のアスペクト比(例えば高さ解像度に対する幅解像度の比率は4:3又は16:9である)等を限定することができる。一例では、上記の制限は、ハードウェアの実装を容易にするために導入され得る。

40

#### 【0141】

特定の適用では、エンコーダは、サイズが出力ピクチャサイズであることを暗黙的に仮定するのではなく、特定の参照ピクチャサイズを使用するようにデコーダに指示することができる。例えば構文要素(例えば `reference_pic_size_present_flag(1114)`)は、参照ピクチャ寸法(1115)の条件付きの存在をゲーティングする。参照ピクチャ

50

寸法(1115)は、例えば幅(例えばreference\_pic\_width\_in\_luma\_samples)と高さ(例えばreference\_pic\_height\_in\_luma\_samples)の両方を含むことができる。

#### 【0142】

また、図11では、ピクチャの幅と高さの適用可能な復号のテーブルが図示されている。一例では、テーブル内のエントリの数を、テーブル指示(table indication)(例えば構文要素num\_dec\_pic\_size\_in\_luma\_samples\_minus1)(1116)によって表すことができる。「minus1」は構文要素(1116)の値の解釈を指すことができる。例えばコーディングされた値が0である場合、1つのテーブルエントリが存在する。コーディングされた値が5である場合、6つのテーブルエントリが存在する。テーブル内の各エントリについて、復号されたピクチャの幅と高さが構文要素(1117)として含まれる。

10

#### 【0143】

構文要素(1117)によって表されるテーブルエントリは、タイルグループヘッダ(1101)内の構文要素dec\_pic\_size\_idx(1102)を使用してインデックス化することができ、したがって、タイルグループごとに異なる復号サイズとズーム係数を可能にする。

#### 【0144】

本開示の一態様によると、特定のビデオコーディング技術又は規格(例えばVP9)は、時間的スケーラビリティと組み合わせて特定の形式の参照ピクチャ再サンプリングを実装することによって、空間スケーラビリティを可能にすることができます。一実施形態では、参照ピクチャは、ARCスタイルの技術を使用してより高い解像度にアップサンプリングされ、空間拡張レイヤのベースを形成する。アップサンプリングされたピクチャを、例えば詳細を追加するために高解像度で通常の予測メカニズム(例えば参照ピクチャからの相互予測のための動き補償予測)を使用して精査することができる。

20

#### 【0145】

いくつかの例では、ネットワーク抽象化レイヤ(NAL)ユニットヘッダ内の値、例えば時間IDフィールドが、時間レイヤ情報と、空間レイヤ情報も示すために使用される。時間レイヤ情報と空間レイヤ情報の両方を示すためにNALユニットヘッダ内の値を使用することにより、既存の選択された転送ユニット(SFU:selected forwarding units)を、変更なしでスケーラブルな環境のために使用することが可能になる。例えばNALユニットヘッダの時間ID値に基づいて、時間レイヤ選択転送のために既存のSFUを作成して最適化することができる。次に、いくつかの例では、既存のSFUを、変更なしに空間スケーラビリティ(例えば空間レイヤの選択)のために使用することができる。いくつかの例では、コーディングされたピクチャサイズと、NALユニットヘッダ内の時間IDフィールドによって示される時間レイヤとの間のマッピングを提供することができる。

30

#### 【0146】

本開示の態様によると、コーディングされたビットストリームのいくつかの特徴は、プロファイル、階層、レベル及び一般制約情報を含む、プロファイルと階層とレベルの組合せ(PTL:profile、tier、level)情報を使用して指定することができる。いくつかの例では、プロファイルは、色再現、解像度、追加のビデオ圧縮等のようなビットストリームの機能のサブセットを定義する。ビデオコーデックは、ベースラインプロファイル(例えば低圧縮比の単純なプロファイル)、高プロファイル(高圧縮比の複雑なプロファイル)、メインプロファイル(例えばベースラインプロファイルと高プロファイルの中間の圧縮比のプロファイルは、デフォルトのプロファイル設定とすることができる。)のような様々なプロファイルを定義することができる。

40

#### 【0147】

さらに、階層とレベルを使用して、最大ビットレート、最大ルマサンプルレート、最大ルマピクチャサイズ、最小圧縮比、許容される最大スライス数、許容される最大タイル数等の観点からビットストリームを定義する特定の制約を指定することができる。下位階層は上位階層よりも制約され、下位レベルは上位レベルよりも制約される。一例では、規格は2つの階層: Main(メイン)階層とHigh(高)階層を定義し得る。Main階

50

層は *H i g h* 階層よりも下位階層である。階層は、最大ビットレートに関して異なるアプリケーションに対処するために作成される。一例では、*M a i n* 階層はほとんどのアプリケーション向けに設計されており、*H i g h* 階層は非常に要求の厳しいアプリケーション向けに設計される。規格は複数のレベルを定義することができる。レベルは、ビットストリームの制約のセットである。一例では、レベル 4 未満のレベルでは、*M a i n* 階層のみが許容される。いくつかの例では、特定の階層 / レベルに準拠するデコーダは、その階層 / レベルとすべての下位階層 / レベルに対して符号化されたすべてのビットストリームを復号することができる要求される。

#### 【 0 1 4 8 】

一般制約情報は、ビデオソースタイプ、コーディングツール及び機能性に関する制約情報を含み得る。例えば制約フラグは、インターフォーマンスツール、インストラコーディングツール、*D B F*、エントロピーコーディング、変換、パーティションング (partitioning) ( 例えばタイル、スライス )、バッファ管理、ランダムアクセス ( 例えば *I D R* )、パラメータセット ( 例えば *S P S*、*P P S* ) 及び / 又は同様のものが、コーディングされたビデオビットストリームにおいて存在するか又は使用されるかどうかを示すことができる。制約情報を、パラメータセット ( 例えば *S P S*、*V P S*、*D C I* ) でシグナリングすることができる。制約フラグを、高レベル構文構造 ( 例えば *S P S*、*V P S*、*D C I* ) でシグナリングすることができる。

#### 【 0 1 4 9 】

本開示のいくつかの態様によると、*P T L* 情報を、範囲 ( 例えばビットストリーム内のコーディングされたビデオデータの一部 ) に関連付けることができる。いくつかの例では、*P T L* 情報を、例えばビットストリーム全体、ビットストリームの *C V S*、ビットストリームの各出力レイヤセット ( *O L S* : output layer set ) 及び / 又は同様のものに対して指定することができ、*V P S*、*D P S*、*D C I*、*S P S*、*P P S*、*A P S*、*G O P*、シーケンス、ヘッダ、*S E I* メッセージ等のような高レベル構文 ( *H L S* : high-level syntax ) 構造でシグナリングすることができる。

#### 【 0 1 5 0 】

いくつかの例では、高レベル構文 ( *H L S* ) は、ブロックレベルに関して定義される。ブロックレベルのコーディングツールを使用して、ピクチャ内のピクセル又はサンプルを復号してピクチャを再構成することができる。ブロックレベルのコーディングツールは、インター予測のためのコーディングツール ( 又はインターフォーマンスツール )、インストラ予測のためのコーディングツール ( 又はインストラコーディングツール )、適応ループフィルタ ( *A L F* : adaptive loop filter )、デブロッキングフィルタ ( *D B F* : deblocking filter )、エントロピーコーディング、変換等のような、コーディングブロックの再構成において使用される任意の適切なコーディングツールを含むことができる。

#### 【 0 1 5 1 】

高レベル構文 ( *H L S* ) は、機能性、システムインターフェース、ツールのピクチャレベル制御及びバッファ制御等に関する情報を指定することができる。例えば *H L S* は、パーティション ( 例えばタイル、スライス、サブピクチャ )、バッファ管理、ランダムアクセス ( 例えば *I D R*、クリーンランダムアクセス ( *C R A* ) )、パラメータセット ( 例えば *V P S*、*S P S*、*P P S*、*A P S* )、参照ピクチャ再サンプリング ( *R P R* )、スケーラビリティ及び / 又は同様のものを指定することができる。高レベル構文は、ブロックレベルを超えるものとすることができます。

#### 【 0 1 5 2 】

制御情報は、*S P S* レベルのツール制御情報、*P P S* レベルのツール制御情報、シーケンスレベルの制御情報、ビットストリームレベルの制御情報及び / 又は同様のもののような、適切なレベルを有することができる。いくつかの例では、*P T L* 情報は制御情報の一部であり、*H L S* 構造内の制約フラグとしてシグナリングされることができ、*H L S* 構造に対応する範囲内のツールの制御又は制約を示すことができる。例えば *P T L* 情報の制約フラグを、シーケンスレベル制御情報とビットストリームレベル制御情報のうちの 1 つに

10

20

30

40

50

おいて提供することができる。一例では、特定のツールがHLS構造内の制約フラグによって無効にされる場合、それらのツールは、例えばHLSに対応する範囲内のコーディングブロックに使用されない。

#### 【0153】

図12及び図13は、本開示のいくつかの実施形態によるPTL情報の例を示す。図12は、PTL構文要素のセットの構文構造例(1200)を示し、図13は、一般制約情報の構文構造例(1300)を示す。

#### 【0154】

図12において、PTL構文要素のセットは、general\_profile\_idc、general\_tier\_flag、general\_level\_idc、num\_sub\_profiles、general\_sub\_profile\_idc、sublayer\_level\_present\_flag、ptl\_alignment\_zero\_bit及びsublayer\_level\_idcを含むことができる。

#### 【0155】

図13において、一般制約情報は、複数の制約フラグを含むことができる。一例では、1に等しい制約フラグ(例えばintra\_only\_constraint\_flag)(1305)は、パラメータsh\_slice\_typeがIでなければならないこと(すなわち、スライスがイントラスライスであること)を示すことができる。パラメータsh\_slice\_typeは、タイプI、P、Bの間でスライスのコーディングタイプを指定する、スライスヘッダ内のパラメータである。0に等しい制約フラグ(例えばintra\_only\_constraint\_flag)(1305)は、PTL情報の範囲内のすべてのコーディングされたピクチャに対して制約(例えばsh\_slice\_typeがIでなければならないこと)を課さず、この場合、他の情報(例えばprofile\_idc)は非イントラスライスを許容することができる。別の例では、1に等しい制約フラグ(例えばno\_alf\_constraint\_flag)(1306)は、PTL情報の範囲内のすべてのCVSに対してsps\_alf\_enabled\_flagが0に等しいことを示すことができ、したがって、適応ループフィルタリングが例えばprofile\_idcに基づいて許容される場合であっても、適応ループフィルタリングは使用されない。0に等しい制約フラグ(例えばno\_alf\_constraint\_flag)(1306)は、上記の制約を課さない。

#### 【0156】

別の例では、図13に示されるように、制約フラグ(例えばno\_lossless\_coding\_tool\_constraint\_flag)(1301)を一般制約情報でシグナリングすることができる。1に等しい制約フラグ(例えばno\_lossless\_coding\_tool\_constraint\_flag)(1301)は、当該制約フラグ(1301)を含むPTL情報の範囲内で、可逆コーディングに関連するコーディングツールを使用することができないことを示すことができる。0に等しい制約フラグ(例えばno\_lossless\_coding\_tool\_constraint\_flag)(1301)は、上記の制約を課さない。

#### 【0157】

別の例では、図13に示されるように、制約フラグ(例えばno\_lossy\_coding\_tool\_constraint\_flag)(1302)を一般制約情報でシグナリングすることができる。1に等しい制約フラグ(例えばno\_lossy\_coding\_tool\_constraint\_flag)(1302)は、当該制約フラグ(1302)を含むPTL情報の範囲内で、非可逆コーディングに関連するコーディングツールを使用することができないことを示すことができる。0に等しい制約フラグ(例えばno\_lossy\_coding\_tool\_constraint\_flag)(1302)は、上記の制約を課さない。

#### 【0158】

一実施形態では、制約フラグ(例えばno\_lossy\_coding\_tool\_constraint\_flag)(1302)が1に等しいとき、制約フラグ(例えばno\_lossless\_coding\_tool\_constraint\_flag)(1301)は1に等しくなくてよい。あるいは、制約フラグ(例えばno\_lossless\_coding\_tool\_constraint\_flag)(1301)が1に等しいとき、制約フラグ(例えばno\_lossy\_coding\_tool\_constraint\_flag)(1302)は1に等しくなくてよい。

#### 【0159】

10

20

30

40

50

一般制約情報内の複数の制約フラグを、特定の順序でソートすることができる。順序は、例えばそれぞれのメカニズムやツールが P T L の範囲内で使用されていない可能性に基づいて設定されることがある。順序を、優先順位と呼ぶことができる。この順序を、一般制約情報の構文構造において、高い優先度から低い優先度へと提示することができる、この場合、高い優先度は、ツール（又はメカニズム）の不使用が高い可能性であることを示し、低い優先度は、ツール（又はメカニズム）の不使用が低い可能性であることを示す。順序に影響を与える追加の要素は、特定のユースケース（例えばサブピクチャ、スケーラビリティ、インターレースサポートのためのツール）にのみ使用される可能性の高いツール、エンコーダ／デコーダ／実装の複雑さに対するツールの影響等を含むことができる。

#### 【0160】

図14A～図14Bは、本開示のいくつかの実施形態による、P T L 構文構造（P T L ブラケットとも呼ばれる）の構文構造例（1410）と、一般制約情報構文構造（一般制約情報ブラケットとも呼ばれる）の構文例（1420）を含む、P T L 情報の例を示す。いくつかの例では、制約フラグの数（例えばnum\_available\_constraint\_flags）を示す構文要素をシグナリングすることができる。一例では、制約フラグの数を示す構文要素は、一般制約情報ブラケットの構文例（1420）の外部であり得る、図14Aに示される構文例（1410）内の（1401）によって示されるような、P T L 構文構造でシグナリングされることがある。あるいは、制約フラグの数を示す構文要素を、構文例（1420）の先頭のように、一般制約情報ブラケットの先頭でシグナリングすることができる。構文要素（例えばnum\_available\_constraint\_flags）が存在し、構文要素（例えばnum\_available\_constraint\_flags）の値がNに等しいとき、最初のN個の制約フラグが、一般制約情報構文構造内に存在し得る。さらに、他の制約フラグが存在しないことがあり、特定の値と等しいと推論することができる。Nは負でない整数を指定することができる。

#### 【0161】

一実施形態では、値N（例えばnum\_available\_constraint\_flags）は、0から制約フラグの最大数（例えばパラメータMaxNumConstraintFlagsの値）までの範囲である。制約フラグの最大数は任意の正の整数とすることができる。制約フラグの最大数（例えばMaxNumConstraintFlags）の値は、16、32、64、128等であるよう予め定義されることがある。値N（例えばnum\_available\_constraint\_flags）が0に等しいとき、制約フラグは一般制約情報構文構造内に存在しない。値N（例えばnum\_available\_constraint\_flags）のコーディングは、値Nと制約フラグの対応するエントロピーコーディング表現が8で割り切れる数まで加算してバイトアライメント（byte alignment）を保証することができるように選択されることがある。

#### 【0162】

いくつかの例では、制約フラグを、1つ以上の制約情報グループに分類することができる。各制約情報グループは1つ以上の制約フラグを含むことができ、対応するゲートフラグを有することができる。対応する制約情報グループのゲートフラグは、対応する制約情報グループ内に制約フラグが存在し得るかどうかを示すことができる。一例では、ゲートフラグを、制約グループ存在フラグと呼ぶことができる。一般に、ゲートフラグは、対応する制約情報グループに関連付けられ、対応する制約情報グループ内の制約フラグに関連付けられる。一実施形態では、ゲートフラグは、対応する制約情報グループ内の制約フラグが、制約情報内に存在する（又はシグナリングされる）かどうかをゲーティングする。例えば対応する制約情報グループのゲートフラグが1に等しい場合、制約情報グループに対応する制約フラグが、例えば一般制約情報内に存在することができる。対応する制約情報グループのゲートフラグが0に等しい場合、制約情報グループに対応する制約フラグは、例えば一般制約情報内に存在し得ない。例えばすべてのゲートフラグが0に等しい場合、制約フラグは存在しない。

#### 【0163】

制約フラグは異なる範囲を有することができる。例えばD C I 内の制約フラグの範囲は

10

20

30

40

50

、コーディングされたビデオビットストリームとすることができます。VPS内の制約フラグの範囲は、複数のレイヤを有するCLVSとすることができます。SPS内の制約フラグの範囲は、単一のCLVSとすることができます。

#### 【0164】

図15A～図15Bは、本開示の実施形態による一般制約情報構文構造(1500)の例を示す。一般制約情報構文構造(1500)は、一般制約情報を表すフラグを含む。具体的には、一般制約情報構文構造(1500)は、図15Aのゲートフラグ(例えばgeneral\_frame\_structure\_constraint\_group\_flag)(1501)、ゲートフラグ(例えばhigh\_level\_functionality\_constraint\_group\_flag)(1502)、ゲートフラグ(例えばscalability\_constraint\_group\_flag)(1503)、ゲートフラグ(例えばpartitioning\_constraint\_group\_flag)(1504)、ゲートフラグ(例えばintra\_coding\_tool\_constraint\_group\_flag)(1505)、ゲートフラグ(例えばinter\_coding\_tool\_constraint\_group\_flag)(1506)、ゲートフラグ(例えばtransform\_constraint\_group\_flag)(1507)、ゲートフラグ(例えばinloop\_filtering\_constraint\_group\_flag)(1508)のような1つ以上のゲートフラグを含む。1つ以上のゲートフラグ(例えばゲートフラグ(1501)～(1508))は、図15Aに示されるように、一般制約情報構文構造(1500)の先頭に存在することができます。

#### 【0165】

ゲートフラグ(例えばgeneral\_frame\_structure\_constraint\_group\_flag)(1501)は、制約情報グループ(1510)に関連付けられ、制約情報グループ(1510)内にある制約フラグ(1511)～(1514)に関連付けられる。1に等しいゲートフラグ(例えばgeneral\_frame\_structure\_constraint\_group\_flag)(1501)は、制約情報グループ(1510)内にある制約フラグ(1511)～(1514)が存在し得ることを指定することができます。

#### 【0166】

制約情報グループ(1510)(又は制約フラグ(1511)～(1514))は、入力ソース及びフレームパッキング(例えばパック化又は投影されたフレーム)に関連することができる。図15Aを参照すると、制約フラグ(1511)～(1514)は、general\_non\_packed\_constraint\_flag(1511)、general\_frame\_only\_constraint\_flag(1512)、general\_non\_projected\_constraint\_flag(1513)及びgeneral\_one\_picture\_only\_constraint\_flag(1514)に対応する。そうでない場合、0に等しいゲートフラグ(例えばgeneral\_frame\_structure\_constraint\_group\_flag)(1501)は、制約情報グループ(1510)内にある制約フラグ(1511)～(1514)が一般制約情報構文構造(1500)内に存在し得ないことを指定することができます。

#### 【0167】

さらに、いくつかの例では、1に等しいゲートフラグ(例えばhigh\_level\_functionality\_constraint\_group\_flag)(1502)は、図15Bによって示されるように、制約情報グループ(1520)内にある高レベルの機能性(例えば参照ピクチャ再サンプリング)に関連する制約フラグが存在し得ることを指定することができます。そうでない場合、0に等しいゲートフラグ(例えばhigh\_level\_functionality\_constraint\_group\_flag)(1502)は、制約情報グループ(1520)内にある制約フラグが一般制約情報構文構造(1500)内に存在し得ないことを指定することができます。

#### 【0168】

図15Aに戻ると、1に等しいゲートフラグ(例えばscalability\_constraint\_group\_flag)(1503)は、スケーラビリティ(例えばインターレイア予測)に関連する制約フラグが存在し得ることを指定することができます。そうでない場合、スケーラビリティに関連する制約フラグは、一般制約情報構文構造(1500)内に存在し得ない。

#### 【0169】

1に等しいゲートフラグ(例えばpartitioning\_constraint\_group\_flag)(1504)

10

20

30

40

50

)は、高レベルパーティショニング(例えばサブピクチャ又はタイル)に関連する制約フラグが存在し得ることを指定することができる。そうでない場合、高レベルパーティショニングに関連する制約フラグは、一般制約情報構文構造(1500)内に存在し得ない。

#### 【0170】

1に等しいゲートフラグ(例えばintra\_coding\_tool\_constraint\_group\_flag)(1505)は、イントラコーディング(例えばイントラ予測)に関連する制約フラグが存在し得ることを指定することができる。そうでない場合、イントラコーディングに関連する制約フラグは、一般制約情報構文構造(1500)内に存在し得ない。

#### 【0171】

1に等しいゲートフラグ(例えばinter\_coding\_tool\_constraint\_group\_flag)(1506)は、インターフェースコーディングに関連する制約フラグ(例えばイントラピクチャ予測の動き補償)が存在し得ることを指定することができる。そうでない場合、インターフェースコーディングに関連する制約フラグは、一般制約情報構文構造(1500)内に存在し得ない。

10

#### 【0172】

1に等しいゲートフラグ(例えばtransform\_constraint\_group\_flag)(1507)は、変換コーディング(例えば複数の変換行列)に関連する制約フラグが存在し得ることを指定することができる。そうでない場合、変換コーディングに関連する制約フラグは、一般制約情報構文構造(1500)内に存在し得ない。

#### 【0173】

一実施形態では、すべてのゲートフラグ(例えば図15Aのゲートフラグ(1501)～(1508))が0に等しいとき、制約フラグは、一般制約情報構文構造(例えば一般制約情報構文構造(1500))内に存在しない。

20

#### 【0174】

本開示の態様によると、ゲートフラグ(例えばゲートフラグ(1501)～(1508))、関連する制約フラグ(例えば制約フラグ(1511)～(1512))、制約情報グループ(1520)内の制約フラグ)、追加の制御情報及び/又は同様のものを含む制御情報がバイトアライメントされることができるように、例えばバイトアライメントを保持するためにフラグの数が8で割り切れるように、構文を設計することができる。一例では、制約情報(例えば一般制約情報構文構造(1500))内のゲートフラグと制約フラグの数は8で割り切れる。バイトアライメントメカニズムを使用して、制御情報のバイトアライメントを実現することができる。図15Bを参照すると、バイトアライメントのために構文(例えばwhileループ)(1530)を使用することができる。

30

#### 【0175】

いくつかの実施形態では、オフセット(例えば構文要素constraint\_info\_offset[]を使用する)のようなオフセット情報と、長さ(例えば構文要素constraint\_info\_length[]を使用する)のような長さ情報が、制約情報内に(例えば一般制約情報構文構造の先頭に)存在し、制約情報内のゲートフラグに関連付けられるそれぞれの制約情報グループ内の制約フラグの提示を助ける。一実施形態では、少なくとも1つの制約情報グループのうちの1つ以上が、コーディングされたビデオビットストリーム内に存在する。制約情報グループの場合、オフセットと長さが制約情報グループの制約情報内に存在することができる。オフセットは、制約情報グループの第1制約フラグへのオフセットを示すことができ、長さは、制約情報グループの制約フラグの数を示すことができる。いくつかの例では、制約情報グループの数を、例えば構文要素num\_constraint\_info\_setによって明示的に示すことができる。num\_constraint\_info\_setの値は0以上の整数とすることができる。num\_constraint\_info\_setの値が0であるとき、constraint\_info\_offset[], constraint\_info\_length[]及び制約フラグは、一般制約情報構文構造内に存在しない。

40

#### 【0176】

一実施形態では、制約情報オフセット(例えば構文要素constraint\_info\_offset[i])と制約情報長さ(例えば構文要素constraint\_info\_length[i])は、制約情報(例えば一般制約情報構文構造)における制約情報グループi(iは正の整数)の制約フラグの提示

50

を助けることができる。例えば制約情報オフセット（例えば構文要素constraint\_info\_offset[i]）の値が5に等しく、制約情報長さ（例えば構文要素constraint\_info\_length[i]）の値が3に等しいとき、5番目、6番目及び7番目の制約フラグが制約情報グループiに関連付けられ、制約情報（例えば一般制約情報構文構造）内に存在する。

#### 【0177】

一例では、ランレンジスコーディングを使用して、予め決められた順序（又は所与の順序）で指定される制約フラグをコーディングすることができる。

#### 【0178】

一実施形態では、制約フラグが予め決められた順序（又は所与の順序）で指定される、ランコーディング（run-coding）を使用することができる。制約フラグを直接コーディングする代わりに、「スキップ」値の適切にコーディングされたリストは、0に等しい制約フラグを示すことができ、次の制約フラグが1に等しいことを意味する。上述のランコーディングは、(i)制約フラグの数が多く、(ii)制約フラグのわずかな割合が1に等しい場合に特に効率的であり得る。

10

#### 【0179】

一実施形態では、少なくとも1つの制約情報グループのうちの1つ以上が、コーディングされたビデオビットストリーム内に存在する。少なくとも1つの制約情報グループのうちの1つ以上における複数の制約フラグは、所定の順序に従ってシグナリングされる。したがって、それらの複数の制約フラグをランコーディング（例えばラン符号化又はラン復号）することができる。さらに、それらの複数の制約フラグに基づいて、コーディングブロックのサブセットの予測情報を決定することができる。

20

#### 【0180】

一実施形態では、ゲートフラグの制約情報グループ内の少なくとも1つの制約フラグは、所定の順序に従ってシグナリングされる複数の制約フラグを含む。したがって、複数の制約フラグをランコーディング（例えばラン符号化又はラン復号）することができる。

#### 【0181】

一実施形態では、制約フラグの完全なリストを、ビデオコーディング規格（例えばVV-C仕様）、外部テーブル等において指定することができる。一例では、制約フラグの利用可能な制約フラグのみが、例えば以下の1つ以上によって、すなわち、利用可能な制約フラグの数（例えばnum\_available\_constraint\_flags）、ゲートフラグ（又は制約グループ存在フラグ）、制約情報オフセット情報及び制約情報長さ情報等がコーディングされたビデオストリーム内に存在すること、によって示される。

30

#### 【0182】

一例では、制約フラグの完全なリストが指定され、エンコーダとデコーダに対して利用可能である。制約フラグの完全なリストをデコーダに記憶することができる。制約フラグの完全なリストは、100個の制約フラグを含むことができる。100個の制約フラグのうち10個は、CLVSの制約情報内に存在し、したがって、CLVS内のコーディングブロックのサブセットに利用可能である。100個の制約フラグのうち10個は、10の利用可能な制約フラグと呼ばれる。一例では、利用可能な制約フラグの数（例えば10）がシグナリングされる。一例では、10の利用可能な制約フラグは、2つの制約情報グループにあり、第1ゲートフラグと第2ゲートフラグによってゲーティングされる。したがって、第1ゲートフラグと第2ゲートフラグをシグナリングして、10の利用可能な制約フラグを示すことができる。

40

#### 【0183】

一例では、第1制約情報オフセット（例えば構文要素constraint\_info\_offset[0]）と第1制約情報長さ（例えば構文要素constraint\_info\_length[0]）がシグナリングされる。第2制約情報オフセット（例えば構文要素constraint\_info\_offset[1]）と第2制約情報長さ（例えば構文要素constraint\_info\_length[1]）がシグナリングされる。例えば構文要素constraint\_info\_offset[0]は15であり、構文要素constraint\_info\_length[0]は3であり、構文要素constraint\_info\_offset[1]は82であり、構文要素constraint\_i

50

`nfo_length[1]`は7であり、したがって、完全なリスト（例えば100個の制約フラグ）内の15番目から17番目の制約フラグと82番目から88番目の制約フラグが制約情報で利用可能又は存在することを示す。

#### 【0184】

一実施形態では、適切な制御情報を用いて、制約フラグを効率的にコーディングするための様々な技術（又は方法、実施形態、実施例）のいずれかを組み合わせることができる。組合せは、そのような技術のうちの2つ以上の適切な組合せであり得る。あるいは、様々な技術（又は方法、実施形態、実施例）のうちの1つを単独で使用することもできる。制約フラグをグループ化することができる。特定のグループにおいて、ランコーディングを使用することができ、一方、他のグループは簡単なバイナリコーディングを用いてよい。

10

#### 【0185】

制約フラグの最大数（例えば`MaxNumConstraintFlags`）の値を、16、32、64、128等あるように予め定義することができる。

#### 【0186】

制約フラグの最大数（例えば`MaxNumConstraintFlags`）の値を、`general_profile_idc`や`general_sub_profile_idc`のようなプロファイル情報又はコーデックバージョン情報によって決定することができ、その結果、制約フラグの数（例えば`num_available_constraint_flags(1401)`）の範囲をプロファイル情報又はバージョン情報によって制限することができる。例えばメインプロファイル（例えば`MaxNumConstraintFlags = 64`）における制約フラグの数（例えば`num_available_constraint_flags(1401)`）の値は、0から64の範囲内とすることができます、高度なプロファイル（例えば`MaxNumConstraintFlags = 128`）における制約フラグの数（例えば`num_available_constraint_flags(1401)`）の値は、例えば0から128の範囲内とすることができます。

20

#### 【0187】

一実施形態では、制約フラグの数（例えば`num_available_constraint_flags`）の値は、`general_profile_idc`や`general_sub_profile_idc`のようなプロファイル情報又はコーデックバージョン情報によって予め定義された値と等しいと推測することができ、その結果、明示的にシグナリングすることなく、`num_available_constraint_flags`の値を決定することができる。

30

#### 【0188】

いくつかの実施形態では、予約されたバイト情報（`reserved byte information`）が一般制約情報構文構造に存在することができる。例えば図13に示されるように、フラグ`gci_num_reserved_bytes(1303)`及び`gci_reserved_bytes[](1304)`が、一般制約情報構文構造の拡張のために、一般制約情報構文構造に存在することができる。フラグ`gci_num_reserved_bytes`は、予約された制約バイトの数を指定することができる。一例では、予約された制約バイトは、追加のフラグ（例えば追加の制約フラグ）をシグナリングするためのものである。フラグ`gci_reserved_byte[]`は、任意の適切な値を有してよい。

#### 【0189】

一実施形態では、`gci_num_reserved_bytes`の値は、`general_profile_idc`や`general_sub_profile_idc`のようなプロファイル情報又はコーデックバージョン情報によって制限又は決定され得る。ベースプロファイル（又はメインプロファイル）では、フラグ`gci_num_reserved_bytes`の値は0とすることができます。拡張プロファイル（又は高度なプロファイル）では、`gci_num_reserved_bytes`の値を0より大きいものとすることができます。

40

#### 【0190】

いくつかの実施形態では、フィールドシーケンスフラグを、コーディングされたビデオビットストリームでシグナリングすることができる。フィールドシーケンスフラグは、出力レイヤのピクチャがフィールドコーディングでコーディングされているかどうかを示すことができる。いくつかの例では、フィールドシーケンスフラグは、構文要素`sps_field_`

50

`seq_flag`を使用してS P Sでシグナリングすることができる。一実施形態では、フラグ`sps_field_seq_flag`がS P S内に存在し得る。1に等しいフラグ`sps_field_seq_flag`は、C L V Sが、フィールドを表すピクチャを伝達することを示すことができる。0に等しいフラグ`sps_field_seq_flag`は、C L V Sが、フレームを表すピクチャを伝達することを示すことができる。

#### 【0191】

図13の一般制約情報構文構造には、フラグ`general_frame_only_constraint_flag`が存在し得る。1に等しいフラグ`general_frame_only_constraint_flag`は、出力レイヤセットの範囲（例えば`0IsInScope`）が、フレームを表すピクチャを伝達することを指定することができる。0に等しいフラグ`general_frame_only_constraint_flag`は、出力レイヤセットの範囲（例えば`0IsInScope`）が、フレームを表すことも表さないこともあるピクチャを伝達することを指定する。一実施形態では、フラグ`general_frame_only_constraint_flag`は、出力レイヤセット内のピクチャが、フィールドコーディングでコーディングされているかどうかを示す。出力レイヤセットは、コーディングブロックのサブセットを含むことができる。フラグ`sps_field_seq_flag`は、ピクチャのサブセットが、フィールドコーディングでコーディングされていないことを示すフラグ`general_frame_only_constraint_flag`（例えば1である）に基づいて、偽（`false`）とすることができます。ピクチャのサブセットは、出力レイヤセットの1つのレイヤ内にすることができます。

#### 【0192】

フラグ`general_frame_only_constraint_flag`が1に等しいとき、フラグ`sps_field_seq_flag`の値は0に等しくなり得る。

#### 【0193】

一実施形態では、フラグ`pps_mixed_nalu_types_in_pic_flag`がP P S内に存在し得る。1に等しいフラグ`pps_mixed_nalu_types_in_pic_flag`は、P P Sを参照する各ピクチャが、2つ以上のV C L N A Lユニットを有し、V C L N A Lユニットが、同じ`nal_unit_type`値を有しないことを指定することができる。0に等しいフラグ`pps_mixed_nalu_types_in_pic_flag`は、P P Sを参照する各ピクチャが、1つ以上のV C L N A Lユニットを有し、P P Sを参照する各ピクチャのV C L N A Lユニットが同じ`nal_unit_type`値を有することを指定することができる。図13の一般制約情報構文構造には、フラグ`no_mixed_nalu_types_in_pic_constraint_flag`が存在し得る。1に等しいフラグ`no_mixed_nalu_types_in_pic_constraint_flag`は、`pps_mixed_nalu_types_in_pic_flag`の値が0に等しくあるべきことを指定することができる。0に等しいフラグ`no_mixed_nalu_types_in_pic_constraint_flag`は、このような制約を課さない。

#### 【0194】

一実施形態では、フラグ`general_one_picture_only_constraint_flag`が、図13に示されるような一般制約情報構文構造に存在し得る。1に等しい`general_one_picture_only_constraint_flag`は、ビットストリーム内にコーディングされたピクチャが1つだけ存在することを指定することができる。0に等しいフラグ`general_one_picture_only_constraint_flag`は、このような制約を課さない。

#### 【0195】

一実施形態では、フラグ`single_layer_constraint_flag`が、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいフラグ`single_layer_constraint_flag`は、`sps_video_parameter_set_id`が0に等しくあるべきことを指定することができる。0に等しいフラグ`single_layer_constraint_flag`は、このような制約を課さない。フラグ`general_one_picture_only_constraint_flag`が1に等しいとき、フラグ`single_layer_constraint_flag`の値は1に等しくなり得る。

#### 【0196】

一実施形態では、フラグ`all_layers_independent_constraint_flag`が、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいフラグ`all_layers_independent_constraint_flag`は、フラグ`vps_all_independent_layers_flag`が1に等しくなり

10

20

30

40

50

得ることを指定することができる。0に等しいフラグall\_layers\_independent\_constraint\_flagは、このような制約を課さない。フラグsingle\_layer\_constraint\_flagが1に等しいとき、フラグall\_layers\_independent\_constraint\_flagの値は1に等しくなり得る。

#### 【0197】

一実施形態では、フラグno\_res\_change\_in\_clvs\_constraint\_flagが、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいフラグno\_res\_change\_in\_clvs\_constraint\_flagは、フラグsps\_res\_change\_in\_clvs\_allowed\_flagが0に等しくなり得ることを指定することができる。0に等しいフラグno\_res\_change\_in\_clvs\_constraint\_flagは、このような制約を課さない。フラグno\_ref\_pic\_resampling\_constraint\_flagが1に等しいとき、フラグno\_res\_change\_in\_clvs\_constraint\_flagの値は1に等しくなり得る。

#### 【0198】

一実施形態では、フラグno\_mixed\_nalu\_types\_in\_pic\_constraint\_flagが、図13の一般制約情報構文構造に存在し得る。1に等しいフラグno\_mixed\_nalu\_types\_in\_pic\_constraint\_flagは、フラグpps\_mixed\_nalu\_types\_in\_pic\_flagの値が0に等しくなり得ることを指定する。0に等しいフラグno\_mixed\_nalu\_types\_in\_pic\_constraint\_flagは、このような制約を課さない。フラグone\_subpic\_per\_pic\_constraint\_flagが1に等しいとき、フラグno\_mixed\_nalu\_types\_in\_pic\_constraint\_flagの値は1に等しくなり得る。

#### 【0199】

一実施形態では、フラグno\_trail\_constraint\_flagが、図13の一般制約情報構文構造に存在し得る。1に等しいフラグno\_trail\_constraint\_flagは、OIsInScope内に存在するTRAIL\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる(OIsInScopeは、DPSを参照するビットストリーム全体におけるすべてのレイヤを含む出力レイヤセットである)。0に等しいフラグno\_trail\_constraint\_flagは、このような制約を課さない。フラグgeneral\_one\_picture\_only\_constraint\_flagが1に等しいとき、フラグno\_trail\_constraint\_flagは1に等しくなり得る。

#### 【0200】

一実施形態では、フラグno\_stsa\_constraint\_flagは、図13の一般制約情報構文構造に存在し得る。1に等しいフラグno\_stsa\_constraint\_flagは、OIsInScope内に存在するSTSA\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_stsa\_constraint\_flagは、このような制約を課さない。フラグgeneral\_one\_picture\_only\_constraint\_flagが1に等しいとき、フラグno\_stsa\_constraint\_flagは1に等しくなり得る。

#### 【0201】

一実施形態では、フラグno\_trail\_constraint\_flagが、図13の一般制約情報構文構造に存在し得る。1に等しいフラグno\_trail\_constraint\_flagは、OIsInScope内に存在するTRAIL\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_trail\_constraint\_flagは、このような制約を課さない。フラグgeneral\_one\_picture\_only\_constraint\_flagが1に等しいとき、フラグno\_trail\_constraint\_flagは1に等しくなり得る。

#### 【0202】

一実施形態では、フラグno\_stsa\_constraint\_flagが、図13の一般制約情報構文構造に存在し得る。1に等しいフラグno\_stsa\_constraint\_flagは、OIsInScope内に存在するSTSA\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_stsa\_constraint\_flagは、このような制約を課さない。フラグgeneral\_one\_picture\_only\_constraint\_flagが1に等しいとき、フラグno\_stsa\_constraint\_flagは1に等しくなり得る。

#### 【0203】

10

20

30

40

50

一実施形態では、フラグno\_idr\_constraint\_flagが、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいno\_idr\_constraint\_flagは、OIsInScope内に存在するIDR\_W\_RADL又はIDR\_N\_LPに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいno\_idr\_constraint\_flagは、このような制約を課さない。

#### 【0204】

一実施形態では、フラグno\_cra\_constraint\_flagが、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいフラグno\_cra\_constraint\_flagは、OIsInScope内に存在するCRA\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_cra\_constraint\_flagは、このような制約を課さない。

10

#### 【0205】

一実施形態では、フラグno\_rasl\_constraint\_flagが、図13の一般制約情報構文構造に存在し得る（フラグno\_rasl\_constraint\_flagは示されていない）。1に等しいフラグno\_rasl\_constraint\_flagは、OIsInScope内に存在するRASL\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_rasl\_constraint\_flagは、このような制約を課さない。フラグno\_cra\_constraint\_flagが1に等しいとき、フラグno\_rasl\_constraint\_flagの値は1に等しくなり得る。

#### 【0206】

一実施形態では、フラグno\_radl\_constraint\_flagが、図13に示されるような一般制約情報構文構造に存在し得る。1に等しいフラグno\_radl\_constraint\_flagは、OIsInScope内に存在するRADL\_NUTに等しいnuh\_unit\_typeを有するNALユニットが存在し得ないことを指定することができる。0に等しいフラグno\_radl\_constraint\_flagは、このような制約を課さない。フラグno\_idr\_constraint\_flagが1に等しく、フラグno\_cra\_constraint\_flagが1に等しいとき、フラグno\_rasl\_constraint\_flagの値は1に等しくなり得る。

20

#### 【0207】

本開示のいくつかの態様は、残差ライスコーディング拡張を伴うレンジ拡張のようなレンジ拡張のための制約フラグシグナリングの技術を提供する。

#### 【0208】

本開示の態様によると、残差コーディングを、Riceコーディングに基づいて実行することができる。Riceコーディングは、調整可能なパラメータを使用して入力値を2つの部分に分割し、異なるコーディング技術を使用して2つの部分をコーディングする。調整可能なパラメータは、いくつかの例ではRiceパラメータと呼ばれる。Riceパラメータは、ライスコーディングのコーディング効率に影響を与える可能性がある。技術を使用して、特定のアプリケーションのRiceパラメータを決定し、特定のアプリケーションのコーディング効率を向上させることができる。Riceパラメータを決定するいくつかの技術が、ビデオ規格のレンジ拡張に含まれ得る。

30

#### 【0209】

本開示の態様によると、いくつかの規格を当初、特定のアプリケーションのために開発することができる。規格を他のアプリケーションに適用可能にするために、他のアプリケーションをサポートするようにツールを用いてレンジ拡張を開発する。例えばHEVCは当初、サンプルあたり8～10ビットの4:2:0クロマ形式のアプリケーションを対象とする。規格を特定のクロマ形式と特定のビット深度以外の他の形式とビット深度に適用可能にするために、他のクロマ形式及び/又はより高ビット深度を使用するアプリケーションをサポートするようにレンジ拡張を開発することができる。

40

#### 【0210】

機能セットを、アプリケーションの特定のグループに必要なものに制限するために、ビデオコーディング規格はプロファイルを定義し、プロファイルは、これらの機能を使用するエンコーダとの相互運用性のためにサポートされるべき定義されたデコーダ機能セット

50

を含むことができる。例えばプロファイルは、準拠するビットストリームを生成する際に使用することができる、コーディングツール又はアルゴリズムのセットを定義することができる。プロファイルに加えて、いくつかの規格（例えばVVC、HEVC等）はまたレベルと階層を定義する。レベルは、空間解像度、ピクセルレート、ビットレート値及びデコーダ処理負荷とメモリ能力に対応し得る変動（variation）に関連してビットストリームに制限を課す。レベル制限は、最大サンプルレート、最大ピクチャサイズ、最大ビットレート、最小圧縮比、コーディングされたピクチャバッファの容量等に関して表され得る。より高いレベルの値は、より高い複雑さの限度に対応する可能性がある。階層は、各レベルについてビットレート値と変動限度（variation limits）を変更する。例えばMain階層はほとんどのアプリケーションを対象としており、一方、High階層は、ビデオ配信アプリケーションよりも大幅に高いビットレート値を有するような、より要求の厳しいビデオ寄与アプリケーション（video contribution applications）に対処するよう設計されている。プロファイル、階層及びレベルの各々が実装と復号の複雑さに影響し、3つの組合せによりビットストリームとデコーダの相互運用性ポイントを指定する。

#### 【0211】

いくつかの例では、特定の階層とレベルに準拠するデコーダは、そのレベルの同じ階層又は下位の階層又はその下の任意のレベルに準拠するすべてのビットストリームを復号することが可能である必要があり、特定のプロファイルに準拠するデコーダは、そのプロファイル内のすべての機能をサポートすることができる。いくつかの例では、エンコーダは、プロファイルでサポートされる機能の任意の特定のセットを使用する必要はないが、準拠するビットストリーム、すなわち、準拠するデコーダによって復号されることを可能にする、指定された制約に従うビットストリームを生成する必要がある。

#### 【0212】

PTL情報に加えて、PTL構文構造はまた、一般制約情報（GCI）構文構造も含んでよく、これは、制約フラグと、ビットストリームの特定の制約プロパティを示すフラグ以外の構文要素のリストを含む。

#### 【0213】

一例では、HEVCは最初、メイン（Main）プロファイル、メイン10プロファイル、メイン静止画（Main Still Picture）プロファイルと呼ばれる3つのプロファイルを含む。3つのプロファイルは、4：2：0クロマサンプリングのみをサポートすること等のような、いくつかの制限を有する。メインプロファイル及びメイン静止画プロファイルでは、サンプルあたり8ビットのビデオ精度のみがサポートされるが、一方、メイン10プロファイルは、サンプルあたり最大10ビットをサポートする。メイン静止画プロファイルでは、ビットストリーム全体で1つのコーディングされたピクチャのみを含む。

#### 【0214】

いくつかの例では、レンジ拡張を有するHEVCは、追加のプロファイルをサポートすることができる。一例では、以下のプロファイル、すなわち：モノクロ（Monochrome）プロファイル、モノクロ10プロファイル、モノクロ12プロファイル、モノクロ16プロファイル、メイン12プロファイル、メイン4：2：2 10プロファイル、メイン4：2：2 12プロファイル、メイン4：4：4プロファイル、メイン4：4：4 10プロファイル、メイン4：4：4 12プロファイル、メインイントラ（Main Intra）プロファイル、メイン10イントラプロファイル、メイン12イントラプロファイル、メイン4：2：2 10イントラプロファイル、メイン4：2：2 12イントラプロファイル、メイン4：4：4イントラプロファイル、メイン4：4：4 10イントラプロファイル、メイン4：4：4 12イントラプロファイル、メイン4：4：4 16イントラプロファイル、メイン4：4：4 16静止画プロファイル、メイン4：4：4 16静止画プロファイル、をまとめてレンジ拡張プロファイルと呼ぶ。

#### 【0215】

レンジ拡張プロファイルのいくつかは、より高ビット深度をサポートすることができ、高ビット深度の操作レンジ拡張（operation range extensions）のプロファイルと呼

ぶことができる。いくつかの例では、高ビット深度の操作レンジ拡張のプロファイルは、メイン12プロファイル、メイン12 4:4:4プロファイル、メイン16 4:4:4プロファイル、メイン12イントラプロファイル、メイン12 4:4:4イントラプロファイル、メイン16 4:4:4イントラプロファイル、メイン12 静止画プロファイル、メイン12 4:4:4 静止画プロファイル、メイン16 4:4:4 静止画プロファイル等のような、サンプルあたり10ビット超をサポートするプロファイルを含む。

#### 【0216】

具体的には、メイン12プロファイルは、イントラ予測モードとインター予測モードの両方で、4:0:0と4:2:0のクロマサンプリングをサポートし、サンプルあたり8ビットから12ビットのビット深度を可能にする。いくつかの例では、メイン12プロファイルに準拠するデコーダは、以下のプロファイル、すなわち、モノクロ、モノクロ12、メイン、メイン10、メイン12プロファイルで作成されたビットストリームを復号することが可能である。

#### 【0217】

メイン12 4:4:4プロファイルは、4:0:0、4:2:0、4:2:2及び4:4:4のクロマサンプリングと、イントラ予測モードとインター予測モードの両方をサポートし、サンプルあたり8ビットから12ビットのビット深度を可能にする。いくつかの例では、メイン12 4:4:4プロファイルに準拠するデコーダは、以下のプロファイル、すなわち、モノクロ、メイン、メイン10、メイン12、メイン10 4:2:2、メイン12 4:2:2、メイン4:4:4、メイン10 4:4:4、メイン12 4:4及びモノクロ12プロファイルで作成されたビットストリームを復号することが可能である。

#### 【0218】

メイン16 4:4:4プロファイルは、4:0:0、4:2:0、4:2:2及び4:4:4のクロマサンプリングと、イントラ予測モードとインター予測モードの両方をサポートし、サンプルあたり8ビットから16ビットのビット深度を可能にする。

#### 【0219】

メイン12 イントラプロファイルは、4:0:0及び4:2:0のクロマサンプリングと、イントラ予測モードをサポートし、サンプルあたり8ビットから12ビットのビット深度を可能にする。

#### 【0220】

メイン12 4:4:4イントラプロファイルは、4:0:0、4:2:0、4:2:2及び4:4:4のクロマサンプリングと、イントラ予測モードをサポートし、サンプルあたり8ビットから12ビットのビット深度を可能にする。

#### 【0221】

メイン16 4:4:4イントラプロファイルは、4:0:0、4:2:0、4:2:2及び4:4:4のクロマサンプリングと、イントラ予測モードをサポートし、サンプルあたり8ビットから16ビットのビット深度を可能にする。

#### 【0222】

メイン12 静止画プロファイルは、4:0:0及び4:2:0のクロマサンプリングをサポートし、サンプルあたり8ビットから12ビットのビット深度を可能にする。メイン12 静止画プロファイルでは、ビットストリーム全体で1つのコーディングされたピクチャのみを含む。

#### 【0223】

メイン12 4:4:4静止画プロファイルは、4:0:0、4:2:0、4:2:2及び4:4:4のクロマサンプリングをサポートし、サンプルごとに8ビットから12ビットのビット深度を可能にする。メイン12 4:4:4静止画プロファイルでは、ビットストリーム全体で1つのコーディングされたピクチャのみを含む。

#### 【0224】

メイン16 4:4:4静止画プロファイルは、4:0:0、4:2:0、4:2:2

10

20

30

40

50

及び 4 : 4 : 4 のクロマサンプリングをサポートし、サンプルごとに 8 ビットから 16 ビットのビット深度を可能にする。メイン 16 : 4 : 4 静止画プロファイルでは、ビットストリーム全体で 1 つのコーディングされたピクチャのみを含む。

#### 【 0 2 2 5 】

本開示のいくつかの態様によると、コーディングツール制御を、ビットストリームの範囲、コーディングされたレイヤビデオシーケンス ( C L V S ) の範囲、ピクチャ、ピクチャのスライス等のような、様々な範囲（例えばコーディングツール制御のための構文要素のインスタンスの永続性を用いてコーディングされる、コーディングされたビデオデータの一部）で実行することができる。いくつかの例では、コーディングツール制御を、デコーダに出力レイヤセットを伝えるビットストリームの制約情報を一般的に含む、一般制約情報 ( G C I ) 構文構造において提供することができる。いくつかの例では、コーディングツール制御を、 C L V S に関連付けられるシーケンスパラメータセット ( S P S ) において提供することができ、 S P S は一般的に C L V S の情報を含む。いくつかの例では、コーディングツール制御を、ピクチャのピクチャヘッダにおいて提供することができ、ピクチャヘッダは一般的にピクチャの情報を含む。いくつかの例では、コーディングツール制御を、スライスのスライスヘッダにおいて提供することができ、スライスヘッダは一般的にスライスの情報を含む。

#### 【 0 2 2 6 】

本開示の態様によると、レンジ拡張におけるコーディングツールの制御情報を、様々な範囲で提供することができる。いくつかの例では、より大きな範囲の構文要素を使用することは、コーディング効率を改善させることができる。例えば 0 より大きい G C I 構文要素の値は、ビットストリームが特定の方法で制約されることを示し、典型的には、特定のコーディングツールがビットストリームで使用されないことを示す。さらに、値 0 に等しい G C I 構文要素の値は、関連するコーディングツールが（指示されたプロファイルでその使用がサポートされている場合）ビットストリームで使用されることが許容される（ただし、必要とはされない）ように、関連する制約が適用され得ないことをシグナリングする。

#### 【 0 2 2 7 】

本開示の別の態様によると、コーディングツールが、ビットストリームにおけるビデオデータのコーディングで使用されないとき、例えば P T L 情報及び / 又は一般制約情報においてコーディングツールの不使用を示し、コーディングツールのサポートがないビデオデコーダは、該ビデオデコーダが、 P T L 情報及び / 又は一般制約情報におけるシグナリングに基づいてビットストリームを復号することが可能であると判断してよく、ビデオデコーダの機能性を拡張することができる。

#### 【 0 2 2 8 】

いくつかの実施形態では、エンコーダは、レンジ拡張を伴うビデオ規格に準拠したビットストリームを生成することができるが、レンジ拡張でサポートされる 1 つ以上の機能の使用は行わない。いくつかの例では、レンジ拡張における 1 つ以上の機能の不使用がわかっている場合、ビデオ規格に準拠しているが、レンジ拡張で 1 つ以上の機能をサポートしていないデコーダは、該デコーダがビットストリームを復号することが可能であると判断してよく、ビットストリームを拒絶する代わりに、復号のためにビットストリームを受け入れてよい。

#### 【 0 2 2 9 】

図 16 は、本開示のいくつかの実施形態による一般制約情報の構文構造 ( 1600 ) を示す。いくつかの例では、構文構造 ( 1600 ) は、デコーダへの出力レイヤセットを含むビットストリームのような、ビットストリームに適用される制約を含む。図 16 の例では、構文構造 ( 1600 ) の gci\_num\_additional\_bits によって示される構文要素は、アライメントゼロビット構文要素（存在する場合）以外の一般制約情報構文構造 ( 1600 ) 内の追加の一般制約情報 ( G C I ) ビットの数を指定するために使用される。いくつかの規格では、 gci\_num\_additional\_bits の値は、 0 又は 1 に等しいことが要求される

10

20

30

40

50

。いくつかの規格では、デコーダは、1より大きいgci\_num\_additional\_bitsの値が構文構造に表れることを許容し得る。

#### 【0230】

図16の例では、構文構造(1600)は、general\_no\_extended\_precision\_constraint\_flag、general\_no\_ts\_residual\_coding\_rice\_present\_in\_sh\_constraint\_flag、general\_no\_rrc\_rice\_extension\_constraint\_flag、general\_no\_persistent\_rice\_adaptation\_constraint\_flag及びgeneral\_no\_reverse\_last\_sig\_coeff\_constraint\_flagによって示される5つの追加GCIビット(構文要素)(1601)～(1605)を含む。5つの追加GCIビット(1601)～(1605)はそれぞれ、いくつかの例では、出力レイヤセットのビットストリームの範囲におけるコーディングツールのコーディング制御情報を提供する。

10

#### 【0231】

図17は、本開示のいくつかの実施形態によるシーケンスパラメータセット(SPS)レンジ拡張の構文構造(1700)の例を示す。構文構造(1700)は、CLVSのレンジ拡張のコーディングツールの制御を提供するためにCLVSのSPSに追加され得る。構文構造(1700)は、sps\_extended\_precision\_flag、sps\_ts\_residual\_coding\_rice\_present\_in\_sh\_flag、sps\_rrc\_rice\_extension\_flag、sps\_persistent\_rice\_adaptation\_enabled\_flag及びsps\_reverse\_last\_sig\_coeff\_enabled\_flagによって示される5つの構文要素(1701)～(1705)を含む。5つの構文要素(1701)～(1705)は、いくつかの例では、CLVSの範囲内のコーディングツールのコーディング制御情報を提供する。

20

#### 【0232】

具体的には、一実施形態では、GCIビット(1601)及び構文要素(1701)は、異なる範囲において、スケーリング及び変換プロセスにおける変換係数のため、並びにabs\_remainder[]及びdec\_abs\_level[]等のようないくつかの構文要素の二値化のための拡張ダイナミックレンジのコーディングツールの制御のような、拡張された精度を使用する制御を提供する。

#### 【0233】

1に等しい構文要素(1701)は、拡張ダイナミックレンジが、スケーリング及び変換プロセスにおける変換係数のため、並びにabs\_remainder[]及びdec\_abs\_level[]等のようないくつかの構文要素の二値化のために使用されることを指定する。構文要素abs\_remainder[scanning position n]は、スキャン位置nにおいてGolomb-Riceコードでコーディングされる変換係数レベルの残りの絶対値である。abs\_remainder[]が存在しないとき、0に等しいと推論される。構文要素dec\_abs\_level[scanning position n]は、スキャン位置nにおけるGolomb-Riceコードでコーディングされる中間値に対応することができ、スキャン位置nにおける変換係数のレベルを決定するために使用される。0に等しい構文要素(1701)は、拡張ダイナミックレンジがスケーリング及び変換プロセスにおいて使用されず、例えば構文要素abs\_remainder[]及びdec\_abs\_level[]等のようないくつかの構文要素の二値化に使用されないことを指定する。存在しないとき、構文要素(1701)の値は0に等しいと推論される。

30

#### 【0234】

一例では、Log2TransformRangeによって示される変数を使用して、スケーリング及び変換プロセスにおける変換係数と、特定の構文要素の二値化とのためのダイナミックレンジを決定する。例えば変数Log2TransformRangeは、スケーリング及び変換プロセスにおける変換係数を表すため、並びに特定の構文要素の二値化のためのビット数とすることができる。ダイナミックレンジは、ビット数を使用して表される最大数と最小数の差とすることができる。一例では、変数Log2TransformRangeは、例えば式(1)を使用して、構文要素(1701)sps\_extended\_precision\_flagに従って導出される：

Log2TransformRange = sps\_extended\_precision\_flag ? Max( 15, Min( 20, BitDepth + 6 ) ) : 15 式(1)

40

50

## 【0235】

スケーリング及び変換プロセスにおける変換係数と、特定の構文要素の二値化とのためのダイナミックレンジを、変数Log2TransformRangeに基づいて決定することができる。いくつかの例では、フラグsps\_extended\_precision\_flagが値0を有するとき、拡張ダイナミックレンジ機能（例えば拡張ダイナミックレンジのコーディングツール）は使用されず、変換係数のダイナミックレンジは、15ビットのような固定数のビットに基づく。フラグsps\_extended\_precision\_flagが値1を有するとき、拡張ダイナミックレンジ機能が有効になり、スケーリング及び変換処理における変換係数を表すビット数は、式(1)の例におけるビット深度BitDepthに基づいて、15ビット、16ビット、17ビット、18ビット、19ビット及び20ビットのうちの1つとすることができます。変換係数のダイナミックレンジを、ビット数に基づいて決定することができる。

10

## 【0236】

本開示の一態様によると、構文要素（例えばsps\_bitdepth\_minus8によって示される）を使用して、ルマ及びクロマアレイのサンプルのビット深度（例えばBitDepthによって示される）と、ルマ及びクロマ量子化パラメータレンジオフセットの値（例えばQpBdOffsetによって示される）をシグナリングすることができる。一例では、ビット深度BitDepthを、式(2)に従って計算することができ、QPレンジオフセットQpBdOffsetを、式(3)に従って計算することができる：

$$\text{BitDepth} = 8 + \text{sps_bitdepth_minus8} \quad \text{式(2)}$$

$$\text{QpBdOffset} = 6 \times \text{sps_bitdepth_minus8} \quad \text{式(3)}$$

20

## 【0237】

いくつかの例では、1に等しいGCIビット(1601)は、出力レイヤセットの範囲(OIsInScope)内のすべてのピクチャの構文要素(1701)が0に等しくなり得ることを指定する。0に等しいGCIビット(1601)は、そのような制約を課さない。したがって、1に等しいGCIビット(1601)は、ビットストリームのコーディングにおける拡張ダイナミックレンジコーディングツールの不使用を指定することができる。

## 【0238】

いくつかの実施形態では、GCIビット(1602)及び構文要素(1702)を使用して、異なる範囲において、変換スキップモードにおける残差コーディングのためのスライスベースのRiceパラメータ選択のような、変換スキップモードにおける残差コーディングのためのスライスベースのRiceコーディングのコーディングツールの制御を提供する。

30

## 【0239】

本開示の一態様によると、変換スキップ残差コーディングのためのスライスベースのRiceパラメータ選択を、ビデオ規格のレンジ拡張に含めることができる。いくつかの例では、図17に示されるように、変換スキップスライスのRiceパラメータのシグナリングが有効又は無効にされることを示すために、変換スキップモードが有効にされるとき（例えば構文要素sps\_transform\_skip\_enabled\_flagが真）、1つの制御フラグ（例えばsps\_ts\_residual\_coding\_rice\_present\_in\_sh\_flag構文要素(1702)によって示される）がシーケンスパラメータセット(SPS)においてシグナルされる。

40

## 【0240】

制御フラグが有効（例えば「1」に等しい）としてシグナルされるとき、例えばスライスヘッダにおいて、各変換スキップスライスに対して1つの構文要素（例えばsh\_ts\_residual\_dual\_coding\_rice\_idx\_minus1によって示される）が更にシグナリングされ、その変換スキップスライスのRiceパラメータの選択を示す。制御フラグが無効（例えば「0」に等しい）としてシグナルされるとき、変換スキップスライスのRiceパラメータの選択を示すために、スライスレベル（例えばスライスヘッダ）で更なる構文要素はシグナルされず、一例では、SPSを参照するコーディングされたビデオデータ内のすべての変換スキップスライスに対してデフォルトのRiceパラメータが使用され得る。

## 【0241】

50

例えば SPS 内の 1 に等しい構文要素 (1702) は、sh\_ts\_residual\_coding\_rice\_idx\_minus1 によって示されるスライスヘッダフラグが、SPS を参照するスライスのスライスヘッダ (例えば slice\_header()) 構文構造内に存在し得ることを指定する。SPS 内の 0 に等しい構文要素 (1702) は、スライスヘッダフラグ sh\_ts\_residual\_coding\_rice\_idx\_minus1 が、SPS を参照するスライスの slice\_header() 構文構造内に存在しないことを指定する。存在しないとき、sps\_ts\_residual\_coding\_rice\_present\_in\_sh\_flag の値は、いくつかの例で 0 に等しいと推論される。

#### 【0242】

いくつかの例では、構文要素を一般制約情報に含めることができ、出力レイヤセットの範囲内で、変換スキップモードにおける残差コーディングのためのスライスベースの RICE コーディングのコーディングツールの使用を制御することができる。例えば 1 に等しい構文要素 (1602) は、出力レイヤセットの範囲 (0IsInScope) 内のすべてのピクチャの構文要素 (1702) が 0 に等しくなり得ることを指定する。0 に等しい構文要素 (1602) は、このような制約を課さない。したがって、いくつかの例では、ビットストリーム内の 1 に等しい GCI ビット (1602) は、ビットストリームをコーディングするための変換スキップ残差コーディングのためのスライスベースの RICE パラメータ選択の不使用を指定することができる。

#### 【0243】

いくつかの実施形態では、GCI ビット (1603) 及び構文要素 (1703) を使用して、異なる範囲において、正規残差コーディング (RRC : regular residual coding) における abs\_remainder[] 及び dec\_abs\_level[] 等のようないくつかの構文要素の二値化のための RICE パラメータ導出のための 1 つ以上のコーディングツールの制御を提供する。いくつかの例では、正規残差コーディング (RRC) は、変換及び量子化によって取得されるブロックをコーディングするためのいくつかの技術を指す。いくつかの例では、量子化のみによって取得されるブロックのために RRC を変更することができる。いくつかの例では、変換スキップ残差コーディング (TSRC : transform skip residual coding) は、変換をバイパスすること (変換スキップとも呼ばれる) によって取得されるブロックをコーディングするための専用のいくつかの技術を指す。

#### 【0244】

いくつかの例では、ビデオコーディング規格は、abs\_remainder[] 及び dec\_abs\_level[] のような、いくつかの構文要素の二値化のための RICE パラメータ導出のための 1 つ以上のコーディングツールを含んでよく、ビデオコーディング規格のレンジ拡張は、abs\_remainder[] 及び dec\_abs\_level[] のような、いくつかの構文要素の二値化のための RICE パラメータ導出のための 1 つ以上の代替コーディングツールを含むことができる。

#### 【0245】

いくつかの例では、ビデオ規格は、RICE パラメータ導出のためにローカルテンプレートベースの技術を使用する。例えば 1 つ以上の (一例では 5 つの) 隣接する係数レベルを含むテンプレートが、RICE パラメータ導出のために使用される。例えばテンプレート内部の絶対係数値の合計を計算することができ、その合計に基づいて RICE パラメータを決定する。一例では、ルックアップテーブルを使用して、その合計に基づいて RICE パラメータを決定することができる。

#### 【0246】

RICE パラメータを他の適切なコーディングツールによって決定することができることに留意されたい。一例では、方程式を使用して、合計に基づいて RICE パラメータを決定することができる。別の例では、コンテキストモデリングを使用して、隣接する係数レベルの統計に基づいて RICE パラメータを決定することができる。いくつかの例では、ビデオ規格のレンジ拡張は、RICE パラメータ導出のための 1 つ以上の代替コーディングツールを指定することができる。

#### 【0247】

いくつかの例では、ビデオ規格のレンジ拡張は、他のシナリオでの使用のために RRC

10

20

30

40

50

に対する変更を含むことができる。一例では、レンジ拡張は、異なるコンテキストモーデリングツールと、変換スキップモードにおける残差コーディングのための残差信号回転ツールを含むことができる。

#### 【0248】

いくつかの例では、1に等しいS P S内の構文要素(1703)は、abs\_remainder[]及びdec\_abs\_level[]の二値化のための代替R i c eパラメータ導出(例えばレンジ拡張におけるR i c eパラメータ導出のための代替コーディングツール)が、S P Sを参照するC L V Sをコーディングするために使用されることを指定する。0に等しい構文要素(1703)は、abs\_remainder[]及びdec\_abs\_level[]の二値化のための代替R i c eパラメータ導出が、S P Sを参照するC L V Sをコーディングするために使用されないことを指定する。存在しないとき、構文要素(1703)の値は0に等しいと推論される。

10

#### 【0249】

いくつかの例では、1に等しい構文要素(1603)は、出力レイヤセットの範囲(OI sInScope)内のすべてのピクチャの構文要素(1703)が0に等しくなり得ることを指定する。0に等しい構文要素(1603)は、このような制約を課さない。したがって、いくつかの例では、1に等しいG C Iビット(1603)は、ビットストリームをコーディングするために、abs\_remainder[]及びdec\_abs\_level[]の二値化に、代替R i c eパラメータ導出(例えば指定されたレンジ拡張で指定されたR i c eパラメータ導出のための代替コーディングツール)の不使用を指定することができる。

20

#### 【0250】

いくつかの実施形態では、G C Iビット(1604)及び構文要素(1704)を使用して、異なる範囲において、abs\_remainder[]及びdec\_abs\_level[]の二値化のための統計ベースのR i c eパラメータ導出の制御を提供する。

#### 【0251】

本開示の一態様によると、abs\_remainder[]及びdec\_abs\_level[]の二値化のためのR i c eパラメータ導出を、以前のT Uから蓄積された統計を使用して各変換ユニット(T U)の開始時に初期化することができる。いくつかの例では、統計ベースのR i c eパラメータ導出を、ビデオ規格のレンジ拡張に含めることができる。

#### 【0252】

いくつかの例では、制御フラグ、例えばS P S内のsps\_persistent\_rice\_adaptation\_enabled\_flagによって示される構文要素(1704)を使用して、統計ベースのR i c eパラメータ導出を制御する。例えばS P S内の1に等しい構文要素(1704)は、abs\_remainder[]及びdec\_abs\_level[]の二値化のためのR i c eパラメータ導出が、以前のT Uから蓄積された統計を使用して各T Uの開始時に初期化されることを指定する。0に等しい構文要素(1704)は、以前のT U状態が現在のT UのR i c eパラメータ導出で使用されないことを指定する。存在しないとき、構文(1704)の値は0に等しいと推論される。

30

#### 【0253】

さらに、一実施形態では、1に等しい構文要素(1604)は、出力レイヤセットの範囲(OI sInScope)内のすべてのピクチャの構文要素(1704)が0に等しくなり得ることを指定する。0に等しい構文要素(1604)は、このような制約を課さない。したがって、いくつかの例では、1に等しいG C Iビット(1604)は、ビットストリームのコーディングのための統計ベースのR i c eパラメータ導出の不使用を指定することができる。

40

#### 【0254】

いくつかの実施形態では、G C Iビット(1605)及び構文要素(1705)を使用して、異なる範囲において、変換係数のエントロピーコーディング中に最後の有意係数(significant coefficient)の位置をコーディングするために使用されるコーディングツールの制御を提供する。一例では、最後の有意係数の位置を、異なるコーディングツールによってコーディングすることができる。例えばビデオ規格は、LastSignificantCoeffX変

50

数及びLastSignificantCoeffY変数によって示される位置の2つの座標をコーディングすることによって、最後の有意係数の位置を決定することができる、第1コーディングツールを指定してよく、ビデオ規格のレンジ拡張は、一例ではゼロアウト変換ブロックの右下隅を参照して、最後の有意係数の相対座標をコーディングすることによって、最後の有意係数の位置を決定することができる第2コーディングツールのような、代替のコーディングツールを指定することができる。

#### 【0255】

いくつかの例では、SPS内の1に等しい構文要素(1705)は、sh\_reverse\_last\_sig\_coeff\_flagによって示されるスライスヘッダフラグ(スライス範囲)が、SPSを参照するスライスヘッダ構文構造(例えばいくつかの例ではslice\_header())内に存在することを指定する。SPS内の0に等しい構文要素(1705)は、スライスヘッダフラグsh\_reverse\_last\_sig\_coeff\_flagが、SPSを参照するスライスヘッダ構文構造内に存在しないことを指定し、スライスヘッダフラグsh\_reverse\_last\_sig\_coeff\_flagは0であると推論され得る。存在しないとき、構文要素(1705)の値は0に等しいと推論される。

#### 【0256】

いくつかの例では、スライスのスライスヘッダフラグsh\_reverse\_last\_sig\_coeff\_flagの値を使用して、スライスのコーディングにおけるスケーリング及び変換プロセスでの変換係数の最後の有意係数の位置導出を決定する。一例では、sh\_reverse\_last\_sig\_coeff\_flagが1に等しいとき、最後の有意係数の位置は、第2コーディングツールのような、ビデオ規格のレンジ拡張における代替コーディングツールによってコーディングされ、そうでない場合、最後の有意係数の位置の現在の座標は、第1コーディングツールによってコーディングされる。

#### 【0257】

いくつかの例では、1に等しいGCIビット(1605)は、出力レイヤセットの範囲(OIsInScope)内のすべてのピクチャの構文要素(1705)が0に等しくなり得ることを指定する。0に等しいGCIビット(1605)は、このような制約を課さない。したがって、1に等しいGCIビット(1605)は、ビットストリームの範囲に対する最後の有意係数の位置導出における第2コーディングツールの不使用を指定することができる。

#### 【0258】

図18は、本開示の実施形態によるプロセス(1800)の概要を示すフローチャートを示す。プロセス(1800)は、ビデオデコーダにおいて用いられ得る。様々な実施形態において、プロセス(1800)は、端末デバイス(310)、(320)、(330)及び(340)内の処理回路、ビデオデコーダ(410)の機能を実行する処理回路、ビデオデコーダ(510)の機能を実行する処理回路等のような処理回路によって実行される。いくつかの実施形態では、プロセス(1800)はソフトウェア命令で実装され、したがって、処理回路がソフトウェア命令を実行するとき、処理回路はプロセス(1800)を実行する。プロセスは(S1801)で開始して(S1810)に進む。

#### 【0259】

(S1810)において、ビットストリーム内のコーディングされたビデオデータの第1範囲(例えば出力レイヤセット)におけるコーディング制御のための第1構文要素(例えばgeneral\_no\_rrc\_rice\_extension\_constraint\_flag)の値を決定する。第1構文要素は、残差コーディングにおけるRiceパラメータ導出のための第1コーディングツールに代わる第2コーディングツールに関連付けられる。

#### 【0260】

一例では、構文構造内の構文要素(例えばgci\_num\_additional\_bits)が構文構造内的一般制約情報の追加ビットを示すことに応答して、第1構文要素を一般制約情報の構文構造から復号する。

#### 【0261】

10

20

30

40

50

いくつかの例では、第1コーディングツールは規格で定義され、第2コーディングツールは、第1コーディングツールに対する代替コーディングツールとして規格のレンジ拡張で定義される。一例では、ビデオエンコーダは規格をサポートし得るが、該規格のレンジ拡張における第2コーディングツールはサポートし得ない。

#### 【0262】

(S1820)において、第1構文要素の値が第1値であるとき、プロセスは(1830)に進み、そうでない場合、プロセスは(S1840)に進む。第1値は、コーディングされたビデオデータの1つ以上の第2範囲(例えば出力レイヤセット内の1つ以上のCLVS)を含む、ビットストリーム内のコーディングされたビデオデータの第1範囲のコーディングにおける、第2コーディングツールの不使用を示す。

10

#### 【0263】

いくつかの例では、第1構文要素は、デコーダで出力される出力レイヤセット内のピクチャのコーディング制御のための一般制約情報内にある。一例では、第1構文要素の第1値は、出力レイヤセット内の各コーディングされたレイヤビデオシーケンス(CLVS)のコーディングにおける第2コーディングツールの不使用を示す。

#### 【0264】

(S1830)において、第1構文要素が第1値であることに応答して、ビットストリーム内のコーディングされたビデオデータの第1範囲が、第2コーディングツールを呼び出すことなく復号される。

20

#### 【0265】

いくつかの例では、ビットストリーム内のコーディングされたレイヤビデオシーケンス(CLVS)のコーディング制御のための第2構文要素(例えばsps\_extended\_precision\_flag)は、CLVSを復号するために第2コーディングツールを呼び出さないことを示す値を有するように制約される。

#### 【0266】

(S1840)において、第1構文要素が第2値であることに応答して、ビットストリーム内のコーディングされたレイヤビデオシーケンス(CLVS)のような、コーディングされたビデオデータの第2範囲のコーディング制御のための第2構文要素(例えばsps\_extended\_precision\_flag)の値が、第2範囲内のコーディングされたビデオデータの復号のために決定される。第2構文要素は、CLVS内の第2コーディングツールの使用/不使用を示す。一例では、第2構文要素は、CLVSのシーケンスパラメータセット(SPS)内に提示されず、第2構文要素の値は、CLVS内の第2コーディングツールの不使用を示すために推論される。

30

#### 【0267】

コーディングされたビデオデータの第2範囲は、次いで、(例えば第2コーディングツールを呼び出して又は呼び出さずに)第2構文要素の値に従って復号される。

#### 【0268】

プロセス(1800)を、適切に適応させることができる。プロセス(1800)のステップを、変更及び/又は省略することができる。追加のステップを追加することができる。任意の適切な実装順序を使用することができる。

40

#### 【0269】

図19は、本開示の一実施形態によるプロセス(1900)の概要を示すフローチャートを示す。プロセス(1900)を、ビデオエンコーダにおいて使用することができる。様々な実施形態において、プロセス(1900)は、端末デバイス(310)、(320)、(330)及び(340)内の処理回路、ビデオエンコーダ(403)の機能を実行する処理回路、ビデオエンコーダ(603)の機能を実行する処理回路、ビデオエンコーダ(703)の機能を実行する処理回路等のような処理回路によって実行される。いくつかの実施形態では、プロセス(1900)はソフトウェア命令で実装され、したがって、処理回路がソフトウェア命令を実行するとき、処理回路はプロセス(1900)を実行する。プロセスは(S1901)で開始して(S1910)に進む。

50

**【 0 2 7 0 】**

( S 1 9 1 0 )において、処理回路は、第 1 コーディングツールに代わる第 2 コーディングツールが、ビットストリーム内のコーディングされたビデオデータの第 1 範囲（例えば出力レイヤセット）の符号化中に使用されるかどうかを判断する。第 2 コーディングツールは、残差コーディングにおける R i c e パラメータ導出のための第 1 コーディングツールに代わるコーディングツールである。コーディングされたビデオデータの第 1 範囲は、コーディングされたビデオデータの 1 つ以上の第 2 範囲（例えば C L V S ）を含む。

**【 0 2 7 1 】**

いくつかの例では、処理回路は、ビットストリーム内のコーディングされたレイヤビデオシーケンス（ C L V S ）のコーディング制御のための第 2 構文要素（例えば sps\_excluded\_precision\_flag ）に基づいて、第 2 コーディングツールが使用されるかどうかを判断することができる。

10

**【 0 2 7 2 】**

( S 1 9 2 0 )において、第 2 コーディングツールが、コーディングされたビデオデータの第 1 範囲のコーディングにおいて使用されないとき、プロセスは（ S 1 9 3 0 ）に進み、そうでない場合、プロセスは（ S 1 9 4 0 ）に進む。

**【 0 2 7 3 】**

( S 1 9 3 0 )において、第 1 値を有する第 1 構文要素（例えば general\_no\_rrc\_rice\_extension\_constraint\_flag ）が、ビットストリームに符号化される。第 1 構文要素は、ビットストリーム内のコーディングされたビデオデータの第 1 範囲（例えば出力レイヤセット）におけるコーディング制御のためのものである。第 1 構文要素は、残差コーディングにおける R i c e パラメータ導出のための第 2 コーディングツールに関連付けられる。第 1 値は、コーディングされたビデオデータの第 1 範囲のコーディングにおける第 2 コーディングツールの不使用を示す。

20

**【 0 2 7 4 】**

一例では、第 1 構文要素は、一般制約情報のための構文構造で符号化され、該構文構造内の構文要素（例えば gci\_num\_additional\_bits ）は、該構文構造内の一般制約情報の追加ビットを示すように調整される。

**【 0 2 7 5 】**

( S 1 9 4 0 )において、第 2 値を有する第 1 構文要素が、ビットストリームに符号化される。いくつかの例では、第 1 構文要素はビットストリームに符号化されず、例えば第 2 値が第 1 構文要素のデフォルト値であり、したがって、提示されていない場合に第 1 構文要素を推論することができる場合には、次いで（ S 1 9 4 0 ）をスキップすることができる。

30

**【 0 2 7 6 】**

プロセス（ 1 9 0 0 ）を、適切に適応させることができる。プロセス（ 1 9 0 0 ）のステップを変更及び / 又は省略することができる。追加のステップを追加することができる。任意の適切な実装順序を使用することができる。

**【 0 2 7 7 】**

上述の技術（例えば制約フラグ、適応解像度パラメータ及び / 又は同様のものをシグナリングするための技術）を、コンピュータ読取可能命令を使用してコンピュータソフトウェアとして実装し、1 つ以上のコンピュータ読取可能媒体に物理的に記憶することができる。例えば図 2 0 は、開示される主題の特定の実施形態を実装するのに適したコンピュータシステム（ 2 0 0 0 ）を示す。

40

**【 0 2 7 8 】**

コンピュータソフトウェアは、アセンブリ、コンパイル、リンク又は類似のメカニズムの対象となり得る任意の適切な機械コード又はコンピュータ言語を使用してコーディングされ、1 つ以上のコンピュータ中央処理ユニット（ C P U ）、グラフィクス処理ユニット（ G P U ）等によって直接的に又は解釈やマイクロコード実行等を通して実行され得る命令を含む、コードを作成することができる。

50

## 【0279】

命令は、例えばパーソナルコンピュータ、タブレットコンピュータ、サーバ、スマートフォン、ゲームデバイス、モノのインターネットデバイス等を含む様々なタイプのコンピュータ又はその構成要素において実行されることができる。

## 【0280】

コンピュータシステム(2000)について図20に示される構成要素は、本質的に例示的なものであり、本開示の実施形態を実装するコンピュータソフトウェアの使用範囲又は機能性に関して、いかなる制限も示唆するように意図されていない。また、構成要素の構成は、コンピュータシステム(2000)の例示的実施形態に示される構成要素の任意の1つ又は組合せに関するいかなる依存性又は要件も有するものとして解釈されてはならない。

10

## 【0281】

コンピュータシステム(2000)は、特定のヒューマンインタフェース入力デバイスを含み得る。そのようなヒューマンインタフェース入力デバイスは、例えば触覚入力(キーストローク、スワイプ、データグローブの動き等)、オーディオ入力(声、拍手等)、視覚入力(ジェスチャ等)、嗅覚入力(図示せず)を通して、1人以上の人間のユーザによる入力に応答し得る。また、ヒューマンインタフェース入力デバイスは、オーディオ(音声、音楽、環境音等)、画像(スキャンされた画像、静止画像カメラから得られる写真画像等)、ビデオ(2次元ビデオ、立体映像を含む3次元ビデオ等)のような、人間による意識的入力に必ずしも直接関係しているとは限らない、特定の媒体をキャプチャするためにも使用されることができる。

20

## 【0282】

ヒューマンインタフェース入力デバイスは、キーボード(2001)、マウス(2002)、トラックパッド(2003)、タッチ画面(2010)、データグローブ(図示せず)、ジョイスティック(2005)、マイクロホン(2006)、スキャナ(2007)及びカメラ(2008)(各々の1つのみが図示される)のうちの1つ以上を含んでもよい。

## 【0283】

コンピュータシステム(2000)はまた、特定のヒューマンインタフェース出力デバイスを含み得る。そのようなヒューマンインタフェース出力デバイスは、例えば触覚出力、音響、光及び嗅覚/味覚を通して、1人以上の人間のユーザの感覚を刺激し得る。そのようなヒューマンインタフェース出力デバイスは、触覚出力デバイス(例えばタッチ画面(2010)、データグローブ(図示せず)又はジョイスティック(2005)による触覚フィードバックであるが、入力デバイスとして機能しない触覚フィードバックデバイスが存在する可能性もある)、オーディオ出力デバイス(スピーカー(2009)、ヘッドフォン(図示せず)等)、視覚出力デバイス(各々タッチ画面入力機能の有無にかかわらず、各々触覚フィードバック機能の有無にもかかわらないが、その一部は、立体画像出力や仮想現実グラス(図示せず)、ホログラフィックディスプレイ及びスマートタンク(図示せず)のような手段を通して、2次元視覚出力又は3次元超の出力を出力することができる、CRT画面、LCD画面、プラズマ画面、OLED画面を含む画面(2010)等)及びプリンタ(図示せず)を含んでよい。

30

## 【0284】

コンピュータシステム(2000)はまた、CD/DVDを有するCD/DVD ROM/RW(2020)を含む光媒体又は類似の媒体(2021)、サムドライブ(2022)、取り外し可能ハードドライブ又はソリッドステートドライブ(2023)、テープ及びフロッピーディスク(図示せず)のようなレガシーマグネット媒体、セキュリティドングル(図示せず)のような特別なROM/ASIC/PLDベースのデバイスのような、ヒューマンアクセス可能なストレージデバイス及びそれらの関連する媒体も含むことができる。

40

## 【0285】

当業者はまた、現在開示されている主題に関連して使用されるとき、「コンピュータ読

50

取可能媒体」という用語が、伝送媒体、搬送波又は他の一時的信号を包含しないことを理解すべきである。

【0286】

コンピュータシステム(2000)はまた、1つ以上の通信ネットワーク(2055)へのインターフェース(2054)も含むことができる。ネットワークは、例えば無線、有線、光とすることができます。ネットワークは更に、ローカル、ワイドエリア、メトロポリタン、車両及び産業用、リアルタイム、遅延耐性等とすることができます。ネットワークの例は、イーサネット(登録商標)、無線LAN等のローカルエリアネットワーク、GSM(登録商標)、3G、4G、5G、LTE等を含むセルラネットワーク、ケーブルTV、衛星TV及び地上放送TVを含むTV有線又は無線ワイドエリアデジタルネットワーク、CANBuses等を含む車両及び産業用ネットワークを含む。特定のネットワークは、一般に、特定の汎用データポート又は周辺バス(2049)に取り付けられ外部ネットワークインターフェースアダプタ(例えばコンピュータシステム(2000)のUSBポート等)を必要とし、他のものは、一般に、後述するシステムバスへの取り付けによって(例えばPCコンピュータシステムへのイーサネット(登録商標)インターフェース又はスマートフォンコンピュータシステムへのセルラーネットワークインターフェース)、コンピュータシステム(2000)のコアに統合される。これらのネットワークのいずれかを使用して、コンピュータシステム(2000)は、他のエンティティと通信することができる。このような通信は、例えばローカル又はワイドエリアデジタルネットワークを使用して、他のコンピュータシステムに対する、単方向の受信専用(例えば放送TV)、単方向の送信専用(例えば特定のCANbusesから特定のCANbusesデバイスへ)又は双方向であり得る。上述のように、特定のプロトコル及びプロトコルスタックを、これらのネットワーク及びネットワークインターフェースの各々において使用することができる。

10

【0287】

前述のヒューマンインターフェースデバイス、ヒューマンアクセス可能なストレージデバイス及びネットワークインターフェースを、コンピュータシステム(2000)のコア(2040)に取り付けることができる。

20

【0288】

コア(2040)は、1つ以上の中央処理ユニット(CPU)(2041)、グラフィクス処理ユニット(GPU)(2042)、フィールドプログラマブルゲートアレイ(FPGA)(2043)の形態の専用のプログラマブル処理ユニット、特定のタスク用のハードウェアアクセラレータ(2044)、グラフィクスアダプタ(2050)等を含むことができる。これらのデバイスは、読み専用メモリ(ROM)(2045)、ランダムアクセスメモリ(RAM)(2046)、内部非ユーザアクセス可能ハードドライブ、SSD等の内部大容量ストレージ(2047)とともに、システムバス(2048)を通して接続され得る。いくつかのコンピュータシステムでは、システムバス(2048)は、追加のCPU、GPU等による拡張を可能にするために、1つ以上の物理的プラグの形態でアクセス可能である。周辺デバイスを、コアのシステムバス(2048)に直接又は周辺バス(2049)を介して取り付けることができる。一例では、画面(2010)をグラフィクスアダプタ(2050)に接続することができる。周辺バスのアーキテクチャは、PCI、USB等を含む。

30

【0289】

CPU(2041)、GPU(2042)、FPGA(2043)及びアクセラレータ(2044)は、組み合わされて上述のコンピュータコードを構成することができる、特定の命令を実行することができる。そのコンピュータコードを、ROM(2045)又はRAM(2046)に記憶することができる。また、一時的なデータをRAM(2046)に記憶することができ、一方、永久的なデータを、例えば内部大容量ストレージ(2047)に記憶することができる。1つ以上のCPU(2041)、GPU(2042)、大容量ストレージ(2047)、ROM(2045)、RAM(2046)等と密接に関連付けることができるキャッシュメモリの使用を通して、メモリデバイスのいずれかに対

40

50

する高速記憶及び取り出しを可能にすることができる。

【0290】

コンピュータ読取可能媒体は、様々なコンピュータ実装される動作を実行するためのコンピュータコードをその上に有することができる。媒体及びコンピュータコードは、本開示の目的のために特別に設計及び構築されたものとすることができる、あるいはそれらは、コンピュータソフトウェア技術の当業者に周知かつ利用可能な種類のものとすることができる。

【0291】

限定ではなく例として、アーキテクチャ(2000)及び特にコア(2040)を有するコンピュータシステムは、プロセッサ(CPU、GPU、FPGA、アクセラレータ等を含む)が1つ以上の有形のコンピュータ読取可能媒体に具現化されたソフトウェアを実行する結果としての機能性を提供することができる。このようなコンピュータ読取可能媒体は、上記で紹介したようなユーザアクセス可能な大容量ストレージ、並びにコア内部大容量ストレージ(2047)又はROM(2045)のような非一時的な性質のコア(2040)の特定のストレージに関連付けられる媒体とすることができます。本開示の様々な実施形態を実装するソフトウェアを、そのようなデバイスに記憶して、コア(2040)によって実行することができる。コンピュータ読取可能媒体は、特定のニーズに従って、1つ以上のメモリデバイス又はチップを含むことができる。ソフトウェアは、コア(2040)及び特にその中のプロセッサ(CPU、GPU、FPGA等を含む)に、RAM(2046)に記憶されるデータ構造を定義することと、ソフトウェアによって定義されるプロセスに従ってそのようなデータ構造を修正することとを含む、本明細書で説明される特定のプロセス又は特定のプロセスの特定の部分を実行させることができる。追加又は代替として、コンピュータシステムは、論理ハードウェア又は他の方法で回路(例えばアクセラレータ(2044))内に具現化された結果として機能性を提供することができ、この回路は、ソフトウェアの代わりに又はソフトウェアとともに動作して、本明細書で説明される特定のプロセス又は特定のプロセスの特定の部分を実行することができる。ソフトウェアへの言及はロジックを含み、また、必要に応じて、その逆も可能である。コンピュータ読取可能媒体への参照は、実行のためのソフトウェアを記憶する回路(集積回路(I.C.)等)、実行のためのロジックを具体化する回路又は適切な場合にはその両方を包含することができる。本開示は、ハードウェアとソフトウェアの任意の適切な組合せを包含する。

付録A：頭字語

J E M : joint exploration model

V V C : versatile video coding

B M S : ベンチマークセット(benchmark set)

M V : 動きベクトル(Motion Vector)

H E V C : 高効率ビデオコーディング(High Efficiency Video Coding)

S E I : 補足強化情報(Supplementary Enhancement Information)

V U I : ビデオユーザビリティ情報(Video Usability Information)

G O P s : ピクチャのグループ(Groups of Pictures)

T U s : 変換ユニット(Transform Units)

P U s : 予測ユニット(Prediction Units)

C T U s : コーディングツリーユニット(Coding Tree Units)

C T B s : コーディングツリーブロック(Coding Tree Blocks)

P B s : 予測ブロック(Prediction Blocks)

H R D : 仮想リファレンスデコーダ(Hypothetical Reference Decoder)

S N R : 信号対ノイズ比(Signal Noise Ratio)

C P U s : 中央処理ユニット(Central Processing Units)

G P U s : グラフィックス処理ユニット(Graphics Processing Units)

C R T : ブラウン管(Cathode Ray Tube)

10

20

30

40

50

L C D : 液晶ディスプレイ (Liquid-Crystal Display)	
O L E D : 有機発光ダイオード (Organic Light-Emitting Diode)	
C D : コンパクトディスク (Compact Disc)	
D V D : デジタルビデオディスク (Digital Video Disc)	
R O M : 読取専用メモリ (Read-Only Memory)	
R A M : ランダムアクセスメモリ (Random Access Memory)	
A S I C : 特定用途向け集積回路 (Application-Specific Integrated Circuit)	
P L D : プログラマブル論理デバイス (Programmable Logic Device)	
L A N : ローカルエリアネットワーク (Local Area Network)	
G S M : 欧州デジタル移動電話方式 (Global System for Mobile communications)	10
L T E : 長期進化 (Long-Term Evolution)	
C A N B u s : コントローラエリアネットワークバス (Controller Area Network Bus)	
U S B : ユニバーサルシリアルバス (Universal Serial Bus)	
P C I : 周辺構成要素相互接続 (Peripheral Component Interconnect)	
F P G A : フィールドプログラマブルゲートアレイ (Field Programmable Gate Array)	
S S D : ソリッドステートドライブ (solid-state drive)	
I C : 集積回路 (Integrated Circuit)	
C U : コーディングユニット (Coding Unit)	

## 【0292】

本開示は、いくつかの例示的な実施形態について説明しているが、本開示の範囲内にある変更、置換及び様々な代替均等物が存在する。したがって、当業者は、本明細書に明示的に示されていないか又は説明されていないが、本開示の原理を具体化しており、よって、本開示の精神及び範囲内にある、様々なシステム及び方法を考案することができる事が理解されよう。

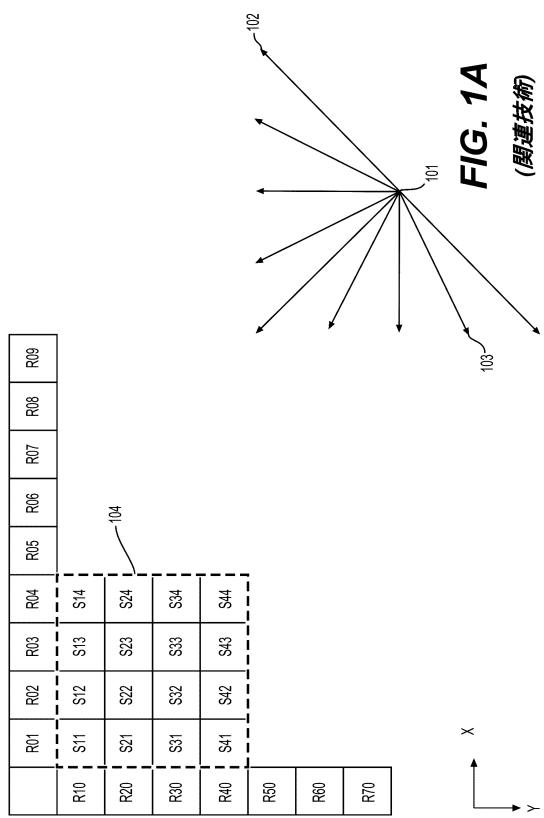
20

30

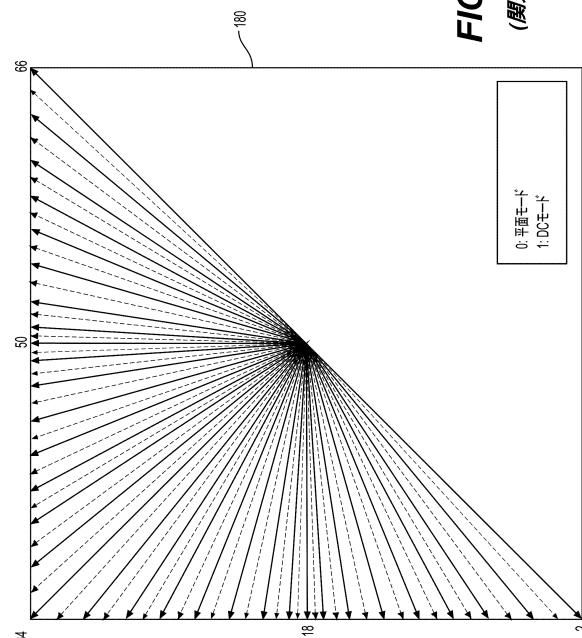
40

50

【図面】  
【図1A】



【図1B】



【 図 2 】

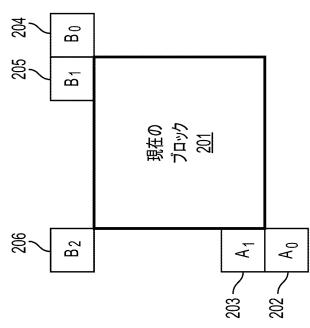
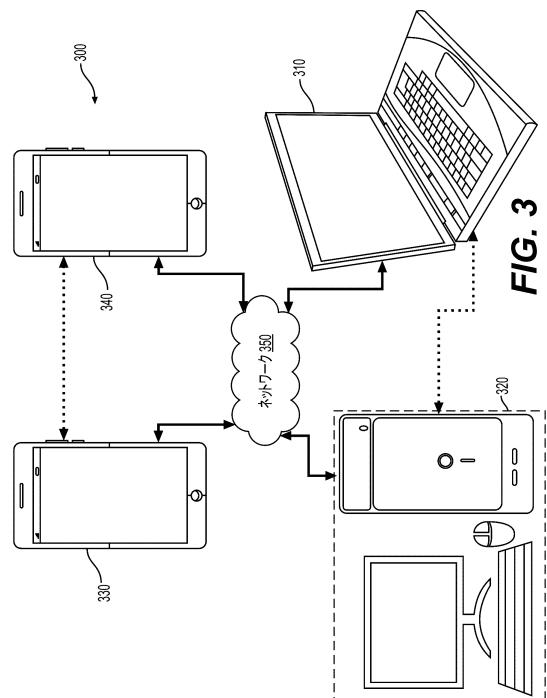


FIG. 2  
(*關連技術*)

【図3】



【図4】

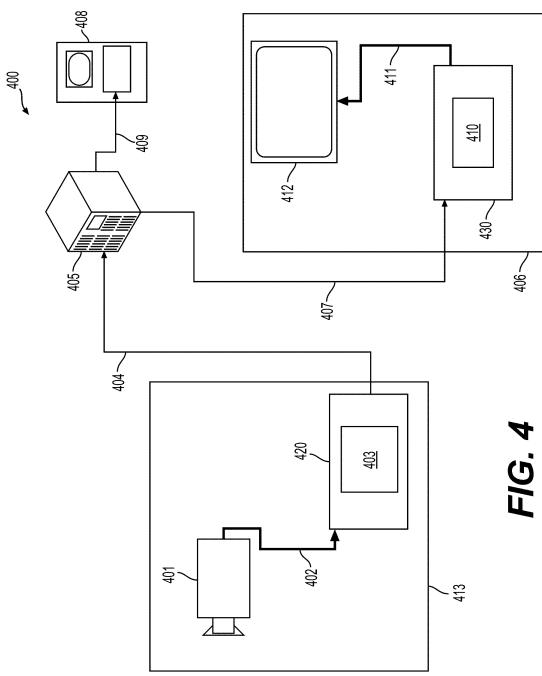


FIG. 4

【 四 5 】

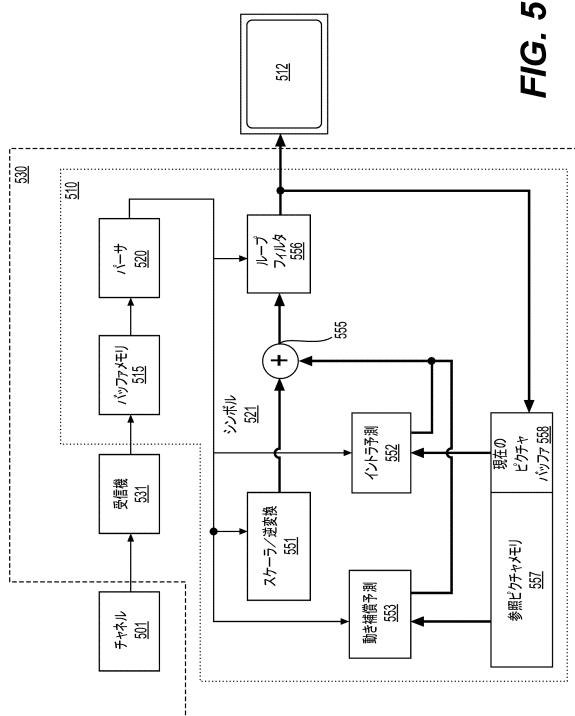


FIG. 5

【図6】

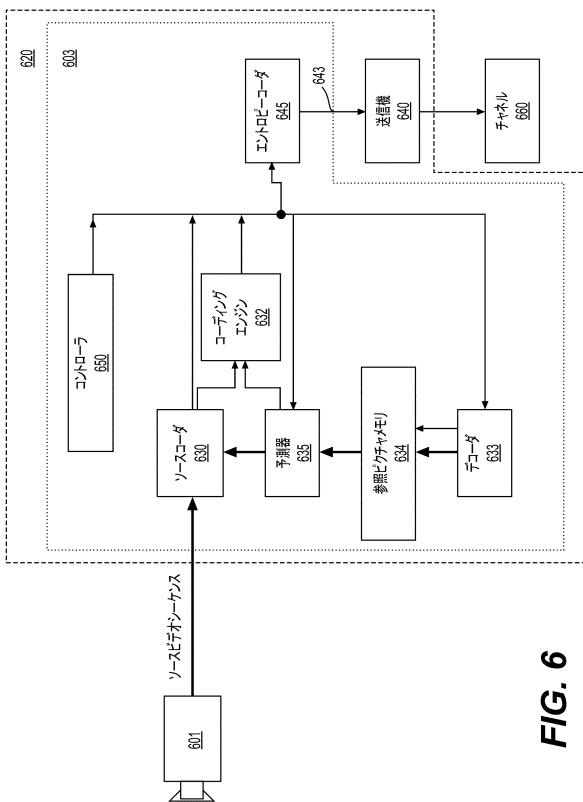


FIG. 6

【図7】

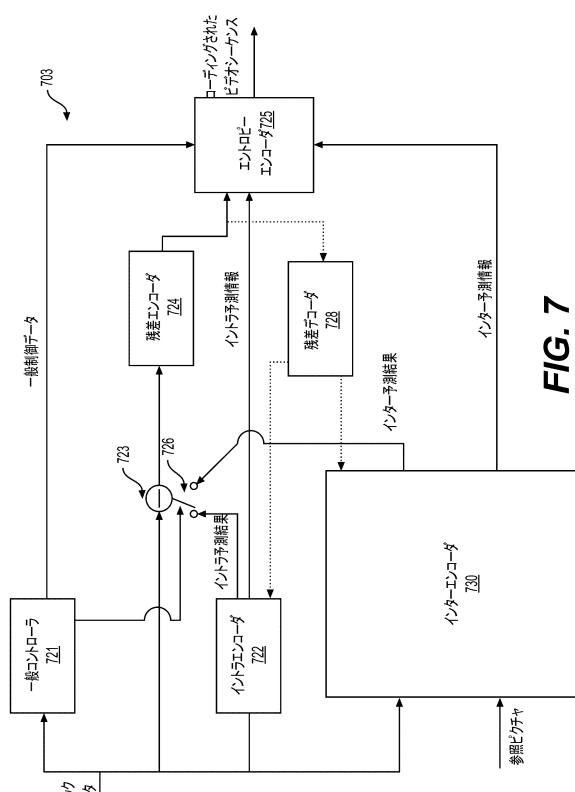


FIG. 7

【図8】

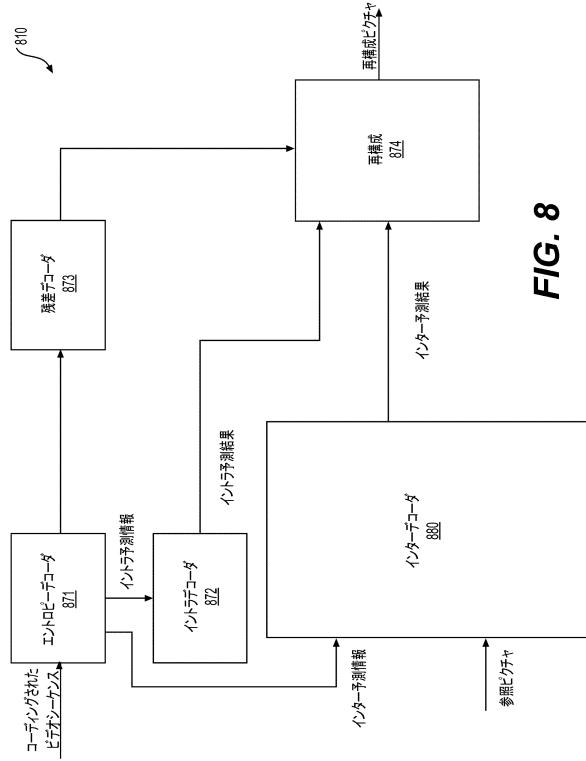


FIG. 8

【図9】

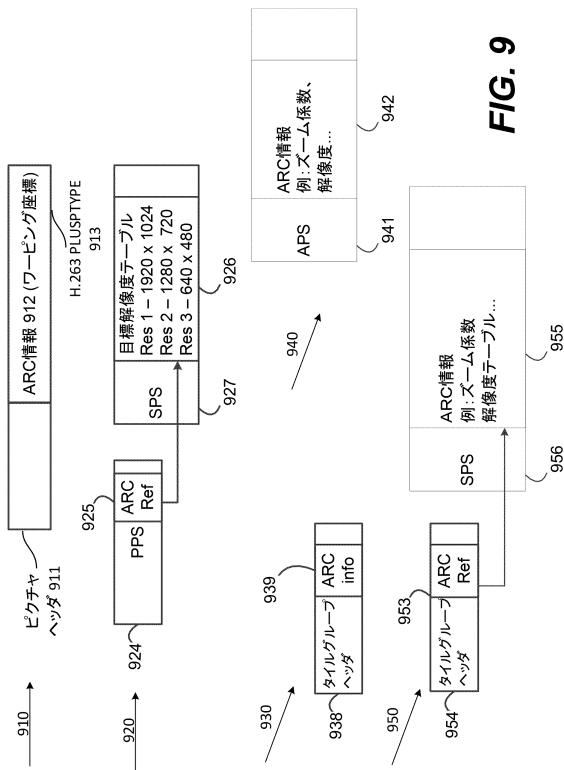


FIG. 9

【図10】

コード	Ex-Golomb コード	元のノード解像度
0	1	1/1
1	010	1/1.5
2	011	(50%アップスケーリング)
3	00100	1/2
4	00101	1/2

FIG. 10

【図11】

1101	~	tile_group_header( ) {		
	...			
1103	~	if(adaptive_pic_resolution_change_flag) {		
1102	~	dec_pic_size_idx	u(1)	
	}			
	...			
	}			
1110	~	seq_parameter_set_rbsp( ) {		記述子
	...			
1111	~	adaptive_pic_resolution_change_flag	u(1)	
1112	~	if(adaptive_pic_resolution_change_flag) {		
1113	[	output_pic_width_in_luma_samples	ue(v)	
		output_pic_height_in_luma_samples	ue(v)	
1114	~	reference_pic_size_present_flag	u(1)	
		if(reference_pic_size_present_flag)		
		{		
1115	[	reference_pic_width_in_luma_samples	ue(v)	
		reference_pic_height_in_luma_samples	ue(v)	
		}		
1116	~	num_dec_pic_size_in_luma_samples_minus1	ue(v)	
		for( i = 0; i <=		
		num_dec_pic_size_in_luma_samples_minus1;		
		i++ ) {		
1117	[	dec_pic_width_in_luma_samples[ i ]	ue(v)	
		dec_pic_height_in_luma_samples[ i ]	ue(v)	
		}		
		}		
		...		
	}			

**FIG. 11**

【図 1 2】

```

profile tier level( profileTierPresentFlag, maxNumSubLayersMinus1 )
    if( profileTierPresentFlag ) {
        general profile idc
        general tier flag
        num available constraint flags
        general constraint info( num available constraint flags
        {
            general level idc
            if( profileTierPresentFlag ) {
                num sub profiles
                for( i = 0; i < num sub profiles; i++ )
                    general sub profile idc
                {
                    for( i = 0; i < maxNumSubLayersMinus1; i++ )
                        sublayer level present flag
                    while( byte aligned() )
                        sublayer level present flag
                    while( byte aligned() )
                        sublayer level present flag
                    if( sublayer level present flag )
                        sublayer level idc
                }
            }
        }
    }
}

```

1200 ↗

FIG. 12

【図 1 3】

Description	
general constraint info( ) {	u(1)
general non packed constraint flag	u(1)
general frame only constraint flag	u(1)
general non projected constraint flag	u(1)
general one picture only constraint flag	u(1)
no lossless coding tool constraint flag	u(1)
intra cab constraint flag	u(1)
intra bidirectional constraint flag	u(1)
inter chroma format constraint flag	u(1)
single layer constraint flag	u(1)
all layers independent constraint flag	u(1)
no ec resampling constraint flag	u(1)
no ec change in chs constraint flag	u(1)
one off set plane constraint flag	u(1)
pic header in slice header constraint flag	u(1)
one slice per pic constraint flag	u(1)
one sample per pic constraint flag	u(1)
no position constraints override constraint flag	u(1)
no zero constraint flag	u(1)
no no constraint flag	u(1)
no odd constraint flag	u(1)
no even constraint flag	u(1)
no side constraint flag	u(1)
no aps constraint flag	u(1)
white byte aligned( )	u(1)
get alignment zero bit	u(1)
get num reserved bytes	u(8)
for( i = 0; i < get num reserved bytes; i++ )	u(8)
get reserved bytes i	u(8)

FIG. 13

【図 1 4 A】

```

profile tier level( profileTierPresentFlag, maxNumSubLayersMinus1 )
    if( profileTierPresentFlag ) {
        general profile idc
        general tier flag
        num available constraint flags
        general constraint info( num available constraint flags
        {
            general level idc
            if( profileTierPresentFlag ) {
                num sub profiles
                for( i = 0; i < num sub profiles; i++ )
                    general sub profile idc
                {
                    for( i = 0; i < maxNumSubLayersMinus1; i++ )
                        sublayer level present flag
                    while( byte aligned() )
                        sublayer level present flag
                    if( sublayer level present flag )
                        sublayer level idc
                }
            }
        }
    }
}

```

1440 ↗

FIG. 14A

【図 1 4 B】

Description	
general constraint info( ) {	u(1)
general non packed constraint flag	u(1)
general frame only constraint flag	u(1)
general non projected constraint flag	u(1)
general one picture only constraint flag	u(1)
no lossless coding tool constraint flag	u(1)
intra cab constraint flag	u(1)
intra bidirectional constraint flag	u(1)
inter chroma format constraint flag	u(1)
single layer constraint flag	u(1)
all layers independent constraint flag	u(1)
no ec resampling constraint flag	u(1)
no ec change in chs constraint flag	u(1)
one off set plane constraint flag	u(1)
pic header in slice header constraint flag	u(1)
one slice per pic constraint flag	u(1)
one sample per pic constraint flag	u(1)
no position constraints override constraint flag	u(1)
no zero constraint flag	u(1)
no no constraint flag	u(1)
no odd constraint flag	u(1)
no even constraint flag	u(1)
no side constraint flag	u(1)
no aps constraint flag	u(1)
white byte aligned( )	u(1)
get alignment zero bit	u(1)
get num reserved bytes	u(8)
for( i = 0; i < get num reserved bytes; i++ )	u(8)
get reserved bytes i	u(8)

FIG. 14B

【図 15 A】

general_constraint_info()		記述子
1501	general_frame_structure_constraint_group_flag	u(1)
1502	high_level_functionality_constraint_group_flag	u(1)
1503	scalability_constraint_group_flag	u(1)
1504	partitioning_constraint_group_flag	u(1)
1505	intra_coding_tool_constraint_group_flag	u(1)
1506	inter_coding_tool_constraint_group_flag	u(1)
1507	transform_constraint_group_flag	u(1)
1508	inloop_filtering_constraint_group_flag	u(1)
1509	if general_frame_structure_constraint_group_flag	
1510	general_non_packed_constraint_flag	u(1)
1511	general_frame_only_constraint_flag	u(1)
1512	general_non_projected_constraint_flag	u(1)
1513	general_one_picture_only_constraint_flag	u(1)
1514	if high_level_functionality_constraint_group_flag {	
	pic_header_in_slice_header_constraint_flag	u(1)
	no_rdl_constraint_flag	u(1)
	no_irr_constraint_flag	u(1)
	no_cra_constraint_flag	u(1)
	no_gdr_constraint_flag	u(1)
	no_ags_constraint_flag	u(1)
	}	
	if intra_coding_tool_constraint_group_flag {	
	...	
	if inter_coding_tool_constraint_group_flag {	
	...	
	}	
	while (true_aligned())	
	gei_alignment_zero_bit	u(1)
	gei_num_reserved_bytes	u(8)
	for (i = 0; i < gei_num_reserved_bytes; i++)	
	gei_reserved_bit[i]	u(8)

FIG. 15A

【図 15 B】

general_constraint_info()		記述子
1520	if high_level_functionality_constraint_group_flag {	
	pic_header_in_slice_header_constraint_flag	u(1)
	no_rdl_constraint_flag	u(1)
	no_irr_constraint_flag	u(1)
	no_cra_constraint_flag	u(1)
	no_gdr_constraint_flag	u(1)
	no_ags_constraint_flag	u(1)
	if intra_coding_tool_constraint_group_flag {	
	...	
	if inter_coding_tool_constraint_group_flag {	
	...	
	}	
	while (true_aligned())	
	gei_alignment_zero_bit	u(1)
	gei_num_reserved_bytes	u(8)
	for (i = 0; i < gei_num_reserved_bytes; i++)	
	gei_reserved_bit[i]	u(8)

FIG. 15B

【図 16】

general_constraint_info()		記述子
1601	...	
1602	gei_num_additional_bits	u(1)
1603	if (gei_num_additional_bits < 0) {	
1604	general_no_extended_precision_constraint_flag	u(1)
1605	general_no_is_residual_coding_rice_present_in_sh_constraint_flag	u(1)
	general_no_rrc_rice_extension_constraint_flag	u(1)
	general_no_persistent_rice_adaptation_constraint_flag	u(1)
	general_no_reserve_last_sig_coeff_constraint_flag	u(1)
	numAdditionalBitsUsed = 5	
	} else	
	numAdditionalBitsUsed = 0	
	for (i = 0; i < gei_num_additional_bits - numAdditionalBitsUsed; i++)	
	gei_reserved_zero_bit[i]	u(1)
	}	

FIG. 16

【図 17】

sps_range_extension()		記述子
1701	sps_extended_precision_flag	u(1)
	if (sps_transform_skip_enabled_flag)	
1702	sps_is_residual_coding_rice_present_in_sh_flag	u(1)
1703	sps_rrc_rice_extension_flag	u(1)
1704	sps_persistent_rice_adaptation_enabled_flag	u(1)
1705	sps_reverse_last_sig_coeff_enabled_flag	u(1)

FIG. 17

【図 18】

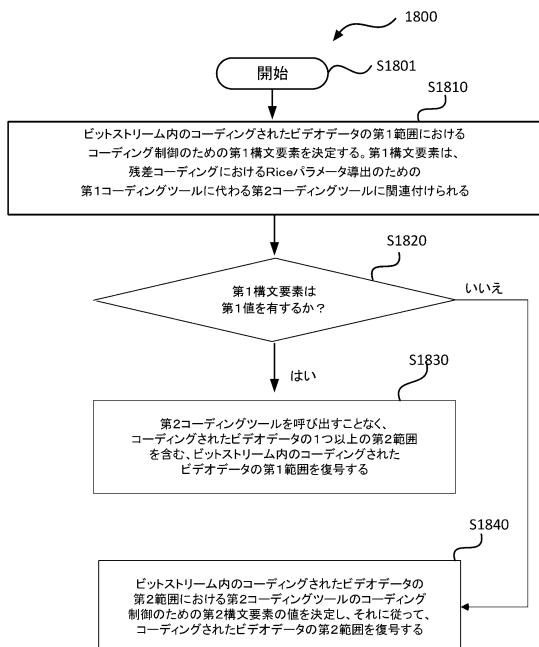


FIG. 18

【図 19】

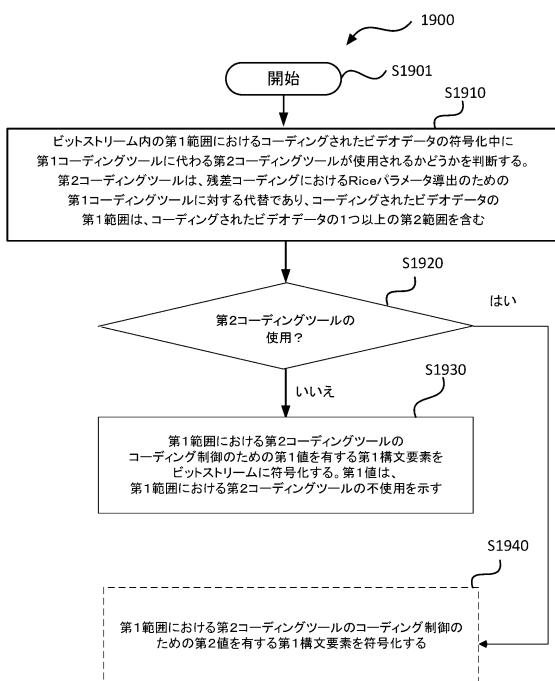


FIG. 19

【図 20】

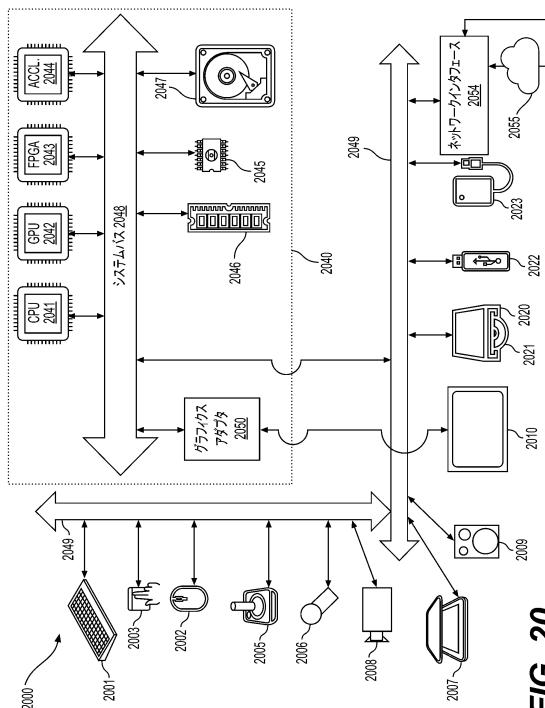


FIG. 20

10

20

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

米国(US)

エルシー内

(72) 発明者 リウ, シャン

アメリカ合衆国 94306 カリフォルニア州 パロアルト パーク・ブルバード 2747 テンセント アメリカ エルエルシー内

(72) 発明者 ウエンジャー, ステファン

アメリカ合衆国 94306 カリフォルニア州 パロアルト パーク・ブルバード 2747 テンセント アメリカ エルエルシー内

審査官 田中 純一

(56) 参考文献 特表 2024-502109 (JP, A)

国際公開第 2022/255395 (WO, A1)

国際公開第 2022/147570 (WO, A1)

特開 2023-006203 (JP, A)

Adrian Browne, et al., Algorithm description for Versatile Video Coding and Test Model 14 (VTM 14) [online], JVET-W JVET-W2002-v1, ITU-T インターネット <URL:[https://jvet-experts.org/doc\\_end\\_user/documents/23\\_Teleconference/wg11/JVET-W2002-v1.zip](https://jvet-experts.org/doc_end_user/documents/23_Teleconference/wg11/JVET-W2002-v1.zip)>, 2021年09月25日, pp.1-105

(58) 調査した分野 (Int.Cl., DB名)

H04N 7/12

H04N 19/00 - 19/98