

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6441354号  
(P6441354)

(45) 発行日 平成30年12月19日(2018.12.19)

(24) 登録日 平成30年11月30日(2018.11.30)

(51) Int.Cl.		F I
HO4N 19/70	(2014.01)	HO4N 19/70
HO4N 19/30	(2014.01)	HO4N 19/30
HO4N 19/433	(2014.01)	HO4N 19/433

請求項の数 15 (全 62 頁)

(21) 出願番号	特願2016-538600 (P2016-538600)	(73) 特許権者	595020643
(86) (22) 出願日	平成26年12月12日(2014.12.12)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2017-505019 (P2017-505019A)		QUALCOMM INCORPORATED
(43) 公表日	平成29年2月9日(2017.2.9)		ED
(86) 国際出願番号	PCT/US2014/070023		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開番号	W02015/089399		121-1714、サン・ディエゴ、モア
(87) 国際公開日	平成27年6月18日(2015.6.18)		ハウス・ドライブ 5775
審査請求日	平成29年11月15日(2017.11.15)	(74) 代理人	100108855
(31) 優先権主張番号	61/915,497		弁理士 蔵田 昌俊
(32) 優先日	平成25年12月12日(2013.12.12)	(74) 代理人	100109830
(33) 優先権主張国	米国 (US)		弁理士 福原 淑弘
(31) 優先権主張番号	14/567,710	(74) 代理人	100158805
(32) 優先日	平成26年12月11日(2014.12.11)		弁理士 井関 守三
(33) 優先権主張国	米国 (US)	(74) 代理人	100112807
			弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 マルチレイヤビデオコーディングのためのPOC値設計

(57) 【特許請求の範囲】

【請求項1】

マルチレイヤビデオデータを復号する方法であって、  
 マルチレイヤビデオデータを受信することと、前記マルチレイヤビデオデータは、複数のアクセスユニットを備え、各アクセスユニットは、共通の時間的インスタンスにおいてそれぞれの1つまたは複数のレイヤのための1つまたは複数の符号化されたピクチャを備え、前記マルチレイヤビデオデータの少なくとも1つのアクセスユニットは、第1のレイヤに少なくとも第1の符号化されたピクチャを備え、前記第1のレイヤとは異なる第2のレイヤに第2の符号化されたピクチャを備え、

マルチレイヤビデオデータの第1の受信レイヤのピクチャが、前記第1のレイヤにおける前記ピクチャのためのピクチャ順序カウント(POC)値リセットが実行されるPOCリセットピクチャであることを示すデータを復号することと、

前記第1のレイヤにおける前記ピクチャがPOCリセットピクチャであることを示す前記復号されたデータに基づいて、前記第1のレイヤにおける前記ピクチャのための前記POC値リセットを実行するより前に、前記アクセスユニットに属さず、出力のために必要とされるとマークされている、復号ピクチャバッファ(DPB)内に記憶されたすべてのレイヤのすべてのピクチャを出力することと、  
 を備える、方法。

【請求項2】

ビデオデータを符号化する方法であって、

10

20

マルチレイヤビデオデータを受信することと、前記マルチレイヤビデオデータは、複数のアクセスユニットを備え、各アクセスユニットは、共通の時間的インスタンスにおいてそれぞれの1つまたは複数のレイヤのための1つまたは複数の符号化されたピクチャを備え、前記マルチレイヤビデオデータの少なくとも1つのアクセスユニットは、第1のレイヤに少なくとも第1の符号化されたピクチャを備え、前記第1のレイヤとは異なる第2のレイヤに第2の符号化されたピクチャを備え、

ビデオエンコーダが、マルチレイヤビデオデータの第1の受信レイヤのピクチャが、前記第1のレイヤにおける前記ピクチャのためのピクチャ順序カウンタ(POC)値リセットが実行されるPOCリセットピクチャであることを示すデータを符号化することと、

前記第1のレイヤにおける前記ピクチャがPOCリセットピクチャであることを示す前記符号化されたデータに基づいて、前記第1のレイヤにおける前記ピクチャのための前記POC値リセットを実行するより前に、前記アクセスユニットに属さず、出力のために必要とされるとマークされている、復号ピクチャバッファ(DPB)内に記憶されたすべてのレイヤのすべてのピクチャを出力することと、  
を備える、方法。

【請求項3】

すべてのピクチャを出力することが、前記ピクチャのビデオデータの初期のスライスのスライスヘッダをそれぞれ復号または符号化した後、および前記スライスのペイロードを復号するより前に、すべてのピクチャを出力することを備える、請求項1または2に記載の方法。

【請求項4】

前記ピクチャがPOCリセットピクチャであることを示す前記データをそれぞれ復号または符号化することが、ビデオデータの前記ピクチャのスライスのためのPOCリセットインデックスを表すデータをそれぞれ復号または符号化することを備える、請求項1または2に記載の方法。

【請求項5】

前記ピクチャがPOCリセットピクチャであることを示す前記データをそれぞれ復号または符号化することが、前記ピクチャを包含する前記アクセスユニットから開始するPOCリセット期間を示すデータをそれぞれ復号または符号化することを備える、請求項1または2に記載の方法。

【請求項6】

前記DPBのすべてのピクチャを出力した後、前記アクセスユニットの前記ピクチャおよび任意の他のピクチャのためのPOC値リセットを実行することをさらに備える、請求項1または2に記載の方法。

【請求項7】

前記DPBのすべてのピクチャを出力した後、前記アクセスユニットの各ピクチャを復号するための参照ピクチャリストを構築することをさらに備える、請求項1または2に記載の方法。

【請求項8】

前記DPBのすべてのピクチャを出力することが、前記ピクチャのPOC値の昇順で、前記DPBのピクチャを出力することと、最小のPOC値を有する前記DPBのピクチャから開始することと、レイヤidの昇順で、同じPOC値を有するピクチャを出力することと、を備える、請求項1または2に記載の方法。

【請求項9】

ビデオデータを復号するためのデバイスであって、  
マルチレイヤビデオデータを受信するための手段と、前記マルチレイヤビデオデータは、複数のアクセスユニットを備え、各アクセスユニットは、共通の時間的インスタンスにおいてそれぞれの1つまたは複数のレイヤのための1つまたは複数の符号化されたピクチャを備え、前記マルチレイヤビデオデータの少なくとも1つのアクセスユニットは、第1のレイヤに少なくとも第1の符号化されたピクチャを備え、前記第1のレイヤとは異なる

10

20

30

40

50

第2のレイヤに第2の符号化されたピクチャを備え、

マルチレイヤビデオデータの第1の受信レイヤのピクチャが、前記第1のレイヤにおける前記ピクチャのためのピクチャ順序カウント(POC)値リセットが実行されるPOCリセットピクチャであることを示すデータを復号するための手段と、

前記第1のレイヤにおける前記ピクチャがPOCリセットピクチャであることを示す前記復号されたデータに基づいて、前記第1のレイヤにおける前記ピクチャのための前記POC値リセットを実行するより前に、前記アクセスユニットに属さず、出力のために必要とされるとマークされている、復号ピクチャバッファ(DPB)内に記憶されたすべてのレイヤのすべてのピクチャを出力するための手段と、  
を備える、デバイス。

10

【請求項10】

ビデオデータを符号化するためのデバイスであって、

マルチレイヤビデオデータを受信するための手段と、前記マルチレイヤビデオデータは、複数のアクセスユニットを備え、各アクセスユニットは、共通の時間的インスタンスにおいてそれぞれの1つまたは複数のレイヤのための1つまたは複数のピクチャを備え、前記マルチレイヤビデオデータの少なくとも1つのアクセスユニットは、第1のレイヤに少なくとも第1のピクチャを備え、前記第1のレイヤとは異なる第2のレイヤに第2のピクチャを備え、

マルチレイヤビデオデータの第1の受信レイヤのピクチャが、前記第1のレイヤにおける前記ピクチャのためのピクチャ順序カウント(POC)値リセットが実行されるPOCリセットピクチャであることを示すデータを符号化するための手段と、

20

前記第1のレイヤにおける前記ピクチャがPOCリセットピクチャであることを示す前記符号化されたデータに基づいて、前記第1のレイヤにおける前記ピクチャのための前記POC値リセットを実行するより前に、前記アクセスユニットに属さず、出力のために必要とされるとマークされている、復号ピクチャバッファ(DPB)内に記憶されたすべてのレイヤのすべてのピクチャを出力するための手段と、  
を備える、デバイス。

【請求項11】

すべてのピクチャを出力するための前記手段が、前記ピクチャのビデオデータの初期のスライスのスライスヘッダを復号した後、および前記スライスのペイロードを復号するより前に、すべてのピクチャを出力するための手段を備える、請求項9に記載のデバイス。

30

【請求項12】

前記ピクチャがPOCリセットピクチャであることを示す前記データを復号するための前記手段が、ビデオデータの前記ピクチャのスライスのためのPOCリセットインデックスを表すデータをコーディングするための手段を備える、請求項9に記載のデバイス。

【請求項13】

前記DPBのすべてのピクチャを出力することが、前記ピクチャのPOC値の昇順で、前記DPBのピクチャを出力することと、最小のPOC値を有する前記DPBのピクチャから開始することと、レイヤidの昇順で、同じPOC値を有するピクチャを出力するための手段と、を備える、請求項9に記載のデバイス。

40

【請求項14】

命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記命令が実行されると、ビデオデータを復号するためのデバイスのプロセッサに、請求項1または3～8のいずれか一項に記載の方法を実行させる、非一時的コンピュータ可読記憶媒体。

【請求項15】

命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記命令が実行されると、ビデオデータを符号化するためのデバイスのプロセッサに、請求項2～8のいずれか一項に記載の方法を実行させる、非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

50

## 【 0 0 0 1 】

[0001]本出願は、その全体が参照により本明細書に組み込まれる、2013年12月12日に出版された米国仮出願第61/915,497号の利益を主張する。

## 【 0 0 0 2 】

[0002]本開示は、ビデオコーディングに関する。

## 【 背景技術 】

## 【 0 0 0 3 】

[0003]デジタルビデオ能力は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップコンピュータもしくはデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲームデバイス、ビデオゲームコンソール、セルラー電話もしくは衛星無線電話、いわゆる「スマートフォン」、ビデオ会議デバイス、ビデオストリーミングデバイスなどを含む、広範囲のデバイスに組み込まれ得る。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4, Part 10、アドバンストビデオコーディング(AVC)、現在開発中の高効率ビデオコーディング(HEVC)規格によって定義された規格、およびそのような規格の拡張に記載されているビデオコーディング技法など、ビデオコーディング技法を実装する。ビデオデバイスは、そのようなビデオコーディング技法を実装することによって、デジタルビデオ情報をより効率的に送信、受信、符号化、復号、および/または記憶し得る。

10

20

## 【 0 0 0 4 】

[0004]ビデオコーディング技法は、ビデオシーケンスに固有の冗長性を低減または除去するために、空間的(イントラピクチャ)予測および/または時間的(インターピクチャ)予測を含む。ブロックベースのビデオコーディングでは、ビデオスライス(たとえば、ビデオフレームまたはビデオフレームの一部)は、ツリーブロック、コーディングユニット(CU)、および/またはコーディングノードと呼ばれることもあるビデオブロックに区分され得る。ピクチャのイントラコード化(I)スライス中のビデオブロックは、同じピクチャ中の隣接ブロック中の参照サンプルに対する空間的予測を使用して符号化される。ピクチャのインターコード化(PまたはB)スライス中のビデオブロックは、同じピクチャ中の隣接ブロック中の参照サンプルに対する空間的予測、または他の参照ピクチャ中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

30

## 【 0 0 0 5 】

[0005]空間的予測または時間的予測は、コーディングされるべきブロックのための予測ブロックを生じる。残差データは、コーディングされるべき元のブロックと予測ブロックとの間のピクセル差分を表す。インターコード化ブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトル、および、コード化ブロックと予測ブロックとの間の差分を示す残差データに従って符号化される。イントラコード化ブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは、ピクセル領域から変換領域に変換され、残差変換係数が生じ得、その残差変換係数は、次いで量子化され得る。最初に2次元アレイで構成される量子化された変換係数は、変換係数の1次元ベクトルを生成するために走査され得、なお一層の圧縮を達成するためにエントロピーコーディングが適用され得る。

40

## 【 発明の概要 】

## 【 0 0 0 6 】

[0006]概して、本開示は、参照ピクチャに関係するビデオデータをコーディングすることについてのエラーレジリエンスを改善するための技法について説明する。特に、本開示は、そのようなエラーレジリエンス(error resilience)を改善し得るピクチャ順序カウント(POC)値設計について説明する。これらの技法は、シングルレイヤビデオデータまたはマルチレイヤビデオデータをコーディングするときに使用され得る。概して、これ

50

らの技法は、POCリセットプロセスを実行するより前に、ピクチャ出力プロセスにおけるエラーレジリエンスを改善するために、復号ピクチャバッファ(DPB)に記憶されたピクチャを出力することを含む。たとえば、ビデオデコーダは、それにおいてPOCリセットが実行されるアクセスユニットの初期ピクチャを復号するより前に、DPBからすべてのピクチャを出力する(たとえば、出力のために必要とされるとマークされている、DPB内のすべてのピクチャを出力する)ように構成され得る。このようにして、DPBに記憶されたピクチャのPOC値は、整合された(aligned)ままであり得、それによって、アクセスユニットのピクチャが適切な順序で出力されることが可能になる。

**【0007】**

[0007]本開示はまた、マルチレイヤビットストリーム中のピクチャのPOC値を適切に決定するための技法を含む。たとえば、本開示の態様によれば、ビデオデコーダは、復号されているピクチャのピクチャタイプを決定することと、ピクチャタイプに基づいて、復号されているピクチャのためのPOC最上位ビット(MSB)値を示すデータを復号することと、を行うように構成され得る。このようにして、復号されているピクチャのためのPOC MSB値は、マルチレイヤビットストリーム中の別のピクチャのPOC MSBを参照することなく、決定され得る。このようにしてPOC MSBを決定することは、(たとえば、マルチレイヤビットストリームのアップスイッチングまたはダウンスイッチングとともに発生し得るように)比較的多数のPOC値が、コーディング順序において復号されているピクチャに先行するピクチャから、復号されているピクチャを分離する場合に、エラーレジリエンスを改善し得る。

**【0008】**

[0008]一例では、ビデオデータを復号する方法は、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウント(POC)値のためのPOCリセットを示すデータを復号することと、ここにおいて、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、復号ピクチャバッファ(DPB)内に記憶されたすべてのピクチャを出力することと、を含む。

**【0009】**

[0009]別の例では、ビデオデータを符号化する方法は、ビデオエンコーダが、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウント(POC)値のためのPOCリセットを示すデータを符号化することと、ここにおいて、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、復号ピクチャバッファ(DPB)内に記憶されたすべてのピクチャを出力することと、を含む。

**【0010】**

[0010]別の例では、ビデオデータをコーディングするためのデバイスは、マルチレイヤビデオデータを記憶するように構成された復号ピクチャバッファ(DPB)を備えるメモリを含む。このデバイスはまた、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウント(POC)値のためのPOCリセットを示すデータをコーディングすることと、ここにおいて、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、DPB内に記憶されたすべてのピクチャを出力することと、を行うように構成された、ビデオコーダを含む。

**【0011】**

[0011]別の例では、ビデオデータをコーディングするためのデバイスは、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウント(POC)値のためのPOCリセットを示すデータをコーディングするための手段と、ここにおいて、

第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、復号ピクチャバッファ(DPB)内に記憶されたすべてのピクチャを出力するための手段と、を含む。

**【0012】**

[0012]別の例では、非一時的コンピュータ可読記憶媒体は、命令を記憶しており、命令が、実行されると、ビデオデータをコーディングするためのデバイスのプロセッサに、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウンタ(POC)値のためのPOCリセットを示すデータをコーディングすることと、ここにおいて、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、復号ピクチャバッファ(DPB)内に記憶されたすべてのピクチャを出力することと、を行わせる。

10

**【0013】**

[0013]別の例では、ビデオデータを復号する方法は、ビデオデコーダが、マルチレイヤビデオデータの第1のピクチャを復号することと、ここにおいて、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、ビデオデコーダが、第1のピクチャのためのピクチャ順序カウンタ(POC)最上位ビット(MSB)値を表すデータを復号することと、を含む。

20

**【0014】**

[0014]別の例では、ビデオデータを符号化する方法は、ビデオエンコーダが、マルチレイヤビデオデータの第1のピクチャを符号化することと、ここにおいて、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、ビデオエンコーダが、第1のピクチャのためのピクチャ順序カウンタ(POC)最上位ビット(MSB)値を表すデータを符号化することと、を含む。

**【0015】**

[0015]別の例では、ビデオデータをコーディングするためのデバイスは、マルチレイヤビデオデータを記憶するように構成されたメモリを含む。このデバイスはまた、マルチレイヤビデオデータの第1のピクチャをコーディングすることと、ここにおいて、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのピクチャ順序カウンタ(POC)最上位ビット(MSB)値を表すデータをコーディングすることと、を行うように構成された、ビデオエンコーダを含む。

30

**【0016】**

[0016]別の例では、ビデオデータをコーディングするためのデバイスは、マルチレイヤビデオデータの第1のピクチャをコーディングするための手段と、ここにおいて、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのピクチャ順序カウンタ(POC)最上位ビット(MSB)値を表すデータをコーディングするための手段と、を含む。

40

**【0017】**

[0017]別の例では、非一時的コンピュータ可読記憶媒体は、命令を記憶しており、命令が、実行されると、ビデオデータをコーディングするためのデバイスのプロセッサに、マルチレイヤビデオデータの第1のピクチャをコーディングすることと、ここにおいて、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのピクチャ順序カウンタ(POC)最上位ビット(MSB)値を表すデータをコーディングすることと、を行わせる。

**【0018】**

[0018]1つまたは複数の例の詳細は、添付の図面および以下の説明において記載される

50

。他の特徴、目的、および利点は、説明および図面から、ならびに特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】

【0019】

【図1】[0019]本開示の技法による、POC値情報をコーディングするための技法を利用し得る、例示的なビデオ符号化および復号システムを示すブロック図。

【図2】[0020]本開示の技法による、POC値情報をコーディングするための技法を実装し得る、ビデオエンコーダの一例を示すブロック図。

【図3】[0021]本開示の技法による、POC値情報をコーディングするための技法を実装し得る、ビデオデコーダの一例を示すブロック図。

【図4】[0022]本開示の技法による、POC値リセットを実行するより前に、DPBのピクチャを出力することを含む、データを符号化するための例示的な方法を示すフローチャート。

【図5】[0023]本開示の技法による、POC値リセットを実行するより前に、DPBのピクチャを出力することを含む、データを復号するための例示的な方法を示すフローチャート。

【図6】[0024]本開示の技法による、POC MSB値を示すデータを符号化することを含む、データを符号化するための例示的な方法を示すフローチャート。

【図7】[0025]本開示の技法による、POC MSB値を示すデータを復号することを含む、データを復号するための例示的な方法を示すフローチャート。

【発明を実施するための形態】

【0020】

[0026]概して、本開示は、マルチレイヤビデオコーディングのために有益であり得る、ビデオコーディングのためのピクチャ順序カウント(POC)値設計について説明する。本開示は、マルチレイヤビデオコーディングにおけるPOC値のシグナリングおよび導出のための、様々な設計改良について説明する。しかしながら、本技法は、概して、マルチレイヤビデオコーディングに関して説明されるが、本開示の技法は、シングルレイヤビデオコーディングにも適用され得る。

【0021】

[0027]POC値は、ピクチャの表示順序を示し得、概して、ピクチャを識別するために使用され得る。たとえば、0のPOC値を有するピクチャ(すなわち、0に等しいPOC値を有するピクチャ)は、1のPOC値を有するピクチャより前に表示される。現在のピクチャのブロックが参照ピクチャに対してインター予測される時、参照ピクチャは、参照ピクチャのためのPOC値を使用して識別され得る。より詳細には、参照ピクチャリスト中に含まれるべき参照ピクチャのためのPOC値が、シーケンスパラメータセット(SPS)、ピクチャパラメータセット(PPS)などのパラメータセット、および/またはブロックに対応するスライスヘッダにおいてシグナリングされ得る。このようにして、ビデオエンコーダは、参照ピクチャリスト中の参照ピクチャの位置に対応する、参照ピクチャリストへのインデックスをシグナリングすることによって、参照ピクチャを識別し得、ビデオデコーダは、(POC値に基づいて)参照ピクチャリストを構築することと、参照ピクチャリスト中の参照ピクチャの位置を識別するために、参照インデックスを使用することとによって、参照ピクチャを識別し得る。

【0022】

[0028]POC値は、最下位ビット(LSB)と最上位ビット(MSB)とからなり得る。コーディング中にピクチャのPOC値を無制限に(indefinitely)増加させるのではなく、ピクチャのPOC値のLSBは、0に等しくなるように周期的にリセットされ得る。POC値は、典型的に、瞬時デコーダリフレッシュ(IDR)ピクチャについてリセットされる。加えて、ビデオコーダは、POCリセットに関連付けられた現在のピクチャの前に処理され、復号ピクチャバッファ(DPB)に記憶されている、ピクチャのPOC値をデクリメントし得る。デクリメントされた参照ピクチャのPOC値は、そのようなピクチャ

10

20

30

40

50

ヤの適切な出力順序を維持するために、負値を有し得る。

【 0 0 2 3 】

[0029]マルチレイヤビデオコーディング(たとえば、マルチビュービデオコーディング、スケーラブルビデオコーディングなど)では、DPBは「サブDPB」に区分され得、ビデオデータの各レイヤが、DPBの別々に管理されたサブDPBに割り当てられる。ビュー構成要素は、(表示順序または出力順序に関して)特定の時間における特定のレイヤまたはビューのための符号化されたピクチャを備え得る。アクセスユニットは、概して、共通の時間的インスタンスのためのすべてのビュー構成要素(たとえば、すべてのネットワークアブストラクションレイヤ(NAL)ユニット)を含むデータのユニットである。したがって、特定の時間的インスタンスのためのアクセスユニットは、それぞれのサブDPBからのそれぞれのビュー構成要素を含み得る。場合によっては、アクセスユニットは、マルチレイヤビットストリームに関連付けられたビュー構成要素のすべてよりも少ないものを含み得る。たとえば、ベースレイヤとエンハンスメントレイヤとを有するマルチレイヤビットストリームでは、場合によっては、アクセスユニットは、エンハンスメントレイヤのピクチャのみを含み得る。この例では、ビデオデコーダは、マルチレイヤビットストリームのエンハンスメントレイヤビュー構成要素のみを包含するアクセスユニットを受信し得る。

10

【 0 0 2 4 】

[0030]アクセスユニットのビュー構成要素は、典型的には、一緒に出力される(すなわち、実質的に同時に出力される)ように意図されており、その場合、ピクチャを出力することは、概して、DPBからピクチャを転送すること(たとえば、DBPから外部メモリにピクチャを記憶すること、DPBからディスプレイへピクチャを送ること、DPBからピクチャを除去することなど)を伴う。

20

ビデオコーダは、「出力のために必要とされる」および「参照のために使用されない」とマークされているピクチャをDPBから出力するために、出力およびバンピングプロセスを使用し得る。場合によっては、ピクチャ出力プロセスは、(1)ピクチャの復号前であるが、ピクチャの最初のスライスのスライスヘッダをパースした(parsing)後と、(2)ピクチャの最後の復号ユニットがコード化ピクチャバッファから除去されるときとを含めて、ピクチャごとに2回呼び出され得る。バンピングプロセスが呼び出されるとき、このプロセスは、選択されたアクセスユニットに属するピクチャを出力する。

30

【 0 0 2 5 】

[0031](たとえば、後述されるような)既存のPOC設計は、最適化されないことがある。たとえば、場合によっては、POC値をデクリメントすることは、レイヤ固有のプロセスであり得る。したがって、以下の例で説明されるように、レイヤのピクチャのPOC値またはPOC MSB値が、そのレイヤのピクチャの復号中にリセットされるとき、レイヤのためのサブDPB内のより早いピクチャ(たとえば、現在コーディングされているピクチャより前に処理され、サブDPBに記憶されているピクチャ)のPOC値は、他のサブDPB内のより早いピクチャのPOC値とクロスレイヤ整合されないことがある。クロスレイヤ整合(alignment)は、たとえば、アクセスユニットのピクチャが同時にまたはほぼ同時に出力されるように、アクセスユニットの各ピクチャが同じPOC値を有するときに発生する。既存のPOC設計では、上位レイヤ(たとえば、相対的により大きいレイヤ識別子を有するレイヤ)におけるピクチャのPOC値は、それぞれのレイヤにおけるピクチャの最初のスライスの復号が開始される後まで更新されないことがある。このようにしてPOCまたはPOC MSBを更新することは、DPB内で復号されたピクチャを包含するアクセスユニットごとのPOC値のクロスレイヤ整合を必要とする、バンピングプロセスを通じたピクチャ出力についての問題を生じ得る。

40

【 0 0 2 6 】

[0032]説明のための一例では、マルチレイヤシーケンスが3つのレイヤ(たとえば、レイヤA、B、およびC)を有すると仮定する。加えて、特定のアクセスユニットが、そのためのPOCリセットが実行されるピクチャを含むと仮定する。この例では、ビデオデコ

50



ーダは、（たとえば、P O Cリセットを実行するための指示をビットストリーム中で受信すると）アクセスユニットのレイヤAピクチャのP O C値をリセットし得る。ビデオデコーダはまた、適切な出力順序を維持するために、レイヤAのためのサブD P Bに前に記憶されたレイヤAのピクチャをデクリメントし得る。しかしながら、レイヤAの最初のピクチャを復号する間、および、アクセスユニットのレイヤBピクチャを復号するより前に、レイヤBおよびCのためのそれぞれのサブD P Bに記憶されているレイヤBおよびCのためのピクチャのP O C値は、レイヤAのサブD P Bに記憶されたピクチャのP O C値と非整合に（クロスレイヤ整合されなく）なる。すなわち、アクセスユニットのレイヤBピクチャおよびレイヤCピクチャはまだリセットされていないので、レイヤBおよびCの各々のそれぞれのサブレイヤのP O C値もまだデクリメントされていない。

10

## 【 0 0 2 7 】

[0033]復号中のそのような非整合（non-alignment）は、ピクチャが不適切な順序でD P Bから出力されることを引き起こし得る。たとえば、（たとえば、同じ時間的インスタンスを有する）アクセスユニットのピクチャは、典型的には、同じまたは実質的に同じ時間に出力されるように意図される。概して、ビデオコーダは、アクセスユニットのピクチャのP O C値に基づいて、アクセスユニットごとのベースでピクチャ（たとえば、一度に1つのアクセスユニットのピクチャ）を出力し得る。すなわち、ビデオコーダは、アクセスユニットのピクチャのP O C値と、アクセスユニットのピクチャが参照のために必要とされるか否かとに基づいて、出力のための特定のアクセスユニットを選択し得る。ビデオコーダは、D P B内で最小のP O C値を有し、もはや参照のために必要とされない、アクセスユニットのピクチャ（たとえば、出力のために必要とされるとマークされているピクチャ）を出力し得る。

20

## 【 0 0 2 8 】

[0034]ピクチャのためのP O C値が、（たとえば、上記で説明されたP O C値リセットのために）アクセスユニット内で非整合になるとき、ビデオコーダは、適切な順序でピクチャを出力しないことがある。たとえば、P O C値がアクセスユニット内でクロスレイヤ整合されなるとき、ビデオコーダは、別のアクセスユニットに移動するより前に、（単一の時間的インスタンスのための）あるアクセスユニットのすべてのピクチャを一緒に出力するのではなく、別のアクセスユニットからピクチャを出力するより前に、あるアクセスユニットのためのピクチャのすべてよりも少ないものを出力し得る。

30

## 【 0 0 2 9 】

[0035]本開示で説明されるP O C値設計は、様々な態様を含み、そのいずれかまたはすべては、単独または任意の組合せで実装され得る。本開示の技法は、上記で説明された問題を克服し得る。一例では、本開示の態様によれば、ビデオデコーダは、そのためのP O C値リセットが実行されるアクセスユニットを受信し得る。ビデオデコーダは、P O C値リセットを実行するより前に、ビデオコーダの復号ピクチャバッファ（D P B）に記憶されたすべてのピクチャを出力し得る。すなわち、ビデオデコーダは、そのためのP O C値リセットが実行されるアクセスユニットの初期ピクチャを復号するより前に、出力のために必要とされるとマークされている、D P Bに記憶されたすべてのピクチャを出力し得る。

40

## 【 0 0 3 0 】

[0036]このようにして、本開示の態様によれば、D P Bに記憶されたピクチャのP O C値が、出力のときに整合されたままであり得る。すなわち、ビデオデコーダは、D P Bに記憶されたピクチャのP O C値がそれらの前のクロスレイヤ整合されたP O C値を維持する（たとえば、マルチレイヤビットストリーム中で同じ時間的インスタンスを有するピクチャが、同じP O C値を維持する）ように、P O C値リセットより前に、D P Bに記憶されたピクチャを出力し得る。P O Cリセットより前に、このようにしてD P Bのピクチャを出力することは、（同じアクセスユニット中であり、同じP O C値を有する）同じ時間的インスタンスのピクチャが同じまたは実質的に同じ時間に出力されることを保証する助けになり得る。

50

## 【 0 0 3 1 】

[0037]本開示の他の技法は、POC値の導出に関係する。以下でより詳細に説明されるように、POC値は、LSB値とMSB値との組合せからなり得る。ビデオコーダは、典型的には、マルチレイヤビットストリームの時間的サブレイヤ0における、復号順序において前のピクチャのPOC MSB値に基づいて、POC MSB値を導出し得る。したがって、現在のピクチャのためのPOC MSB値は、典型的には、コーディング順序において現在のピクチャに先行するピクチャのPOC MSB値に依存する。

## 【 0 0 3 2 】

[0038]このタイプのPOC MSB値依存性は、マルチレイヤビデオコーディングとともにアップスイッチングおよび/またはダウンスイッチングを実行するとき問題になり得る。たとえば、アップスイッチングは、追加のレイヤがマルチレイヤビットストリームに追加されるときに発生し得る。アップスイッチングは、たとえば、利用可能な帯域幅の量が、潜在的により良い閲覧体験 (viewing experience) のために、ビデオデータの追加のレイヤが追加されることを可能にするために十分である場合に発生し得る。ダウンスイッチングは、たとえば、帯域幅または他の制約のために、ビデオデータのレイヤがマルチレイヤビットストリームから除去されるときに発生し得る。

## 【 0 0 3 3 】

[0039]1つの潜在的なPOC MSB値導出問題は、ダウンスイッチングされている比較的長い(たとえば、少なくとも、最大POC LSB期間の半分よりも長い)期間後に、ビデオコーダがアップスイッチングするとき、および、アップスイッチングを可能にするために使用されるイントラランダムアクセスポイント(IRAP)ピクチャタイプが、クリーンランダムアクセス(CRA)であるときに発生し得る。そのような場合には、ビデオコーダは、(それからPOC MSB値が導出される)コーディング順序において前のピクチャが古くなっている(obsolete)ので、アップスイッチングされている新しいレイヤの最初のピクチャのPOC MSB値を適切に決定することができないことがある。

## 【 0 0 3 4 】

[0040]説明のための一例では、マルチレイヤビットストリームは、第1のレイヤと第2のレイヤとを含み得る。ビデオコーダは、第1のレイヤと第2のレイヤとを処理することから、第1のレイヤのみを処理することへダウンスイッチングし得る。第2のレイヤの最後のピクチャは、この例では、50のPOC値を有し得る(すなわち、POC MSBは0であり、POC LSBは50である)。ビデオコーダは、第1のレイヤのみの250個のピクチャをコーディングし得る。次いで、ビデオコーダは、第1のレイヤと第2のレイヤの両方をコーディングするようにアップスイッチングし得る。概して、ビデオコーダは、(ダウンスイッチングより前の)第2のレイヤの最後のコード化されたピクチャに基づいて、(アップスイッチング後の)第2のレイヤの最初のピクチャのためのPOC値を、たとえば、 $50 + 250 = 300$ (たとえば、ダウンスイッチングより前の第2のレイヤの最後のピクチャ+第1のレイヤのみの250個のピクチャのPOC)として導出し得る。

## 【 0 0 3 5 】

[0041]ただし、8ビットLSB値では、256個の可能なPOC LSB値のみが利用可能である。上記の例では、あのそのPOC MSBは、アップスイッチング後の第2のレイヤの最初のピクチャについて、256だけ増加される(すなわち、POC MSBは256であり、POC LSBは44である)。したがって、アップスイッチング後の第2のレイヤの最初のピクチャのためのPOC値は、POC MSB値の次のサイクルにおけるものであり得る。たとえば、上記の例では、300は、256個の可能なPOC LSB値を超えるので、アップスイッチング後の第2のレイヤの最初のピクチャのためのPOC MSB値は、POC MSB値の新しいサイクルに関連付けられることになる。この例では、ビデオコーダが、POC MSB値の前のサイクル中に含まれるPOC MSB値を参照しなければならぬので、ダウンスイッチングより前の第2のレイヤの最後のピクチャからのPOC MSB値は、アップスイッチング後の第2のレイヤの最初のピク

10

20

30

40

50

チャの P O C M S B 値を決定するためには、古いものにされている (has been rendered obsolete)。すなわち、ビデオコードは、アップスイッチング後の第 2 のレイヤの最初のピクチャの P O C M S B 値を導出するために、ダウンスイッチングより前の第 2 のレイヤの最後のピクチャを使用することができないことがある。この P O C M S B 値参照問題は、ダウンスイッチングより前の第 2 のレイヤの最後のピクチャと、アップスイッチング後の第 2 のレイヤの最初のピクチャとの間の距離が、ダウンスイッチングとアップスイッチングとの間の 2 5 6 ピクチャサイクル長の半分よりも大きいときはいつでも生じ得る。

【 0 0 3 6 】

[0042]本開示のいくつかの技法は、上記で説明された問題を克服し得る。たとえば、本開示の態様によれば、ビデオデコードは、復号されているピクチャのピクチャタイプを決定することと、ピクチャタイプ(すなわち、決定されたピクチャタイプ)に基づいて、復号されているピクチャのための P O C M S B 値を示すデータを復号することと、を行うように構成され得る。たとえば、概して、そのための P O C M S B 値がコーディングされるピクチャタイプのセットと、そのための P O C M S B 値がコーディングされないピクチャタイプの別のセットとがあり得る。ビデオコードは、ピクチャのためのピクチャタイプが、そのための P O C M S B 値がコーディングされるピクチャタイプのセット中に含まれるときのみ、そのピクチャのための P O C M S B 値をコーディングし得る。このようにして、ビデオコードは、マルチレイヤビットストリーム中の別のピクチャの P O C M S B を参照することなく、P O C M S B 値を示すデータを使用して、コーディングされているピクチャのための P O C M S B 値を決定し得る。このようにして P O C M S B を決定することは、(たとえば、マルチレイヤビットストリームのレイヤによるアップスイッチングまたはダウンスイッチングとともに発生し得るように)比較的多数の P O C 値が、コーディング順序において復号されているピクチャに先行するピクチャから、復号されているピクチャを分離する場合に、エラーレジリエンスを改善し得る。

【 0 0 3 7 】

[0043]上記で説明された例に本技法を適用すると、ビデオコードは、第 2 のレイヤにアップスイッチングした後に第 2 のレイヤにおいて復号されている最初のピクチャが、C R A ピクチャタイプであると決定し得る。この例では、ビデオデコードは、ピクチャタイプが C R A であるという決定に基づいて、アップスイッチング後の第 2 のレイヤの最初のピクチャのための P O C M S B 値を示すデータを復号し得る。このようにして、ビデオデコードは、ダウンスイッチングより前に復号された第 2 のレイヤの最後のピクチャの P O C M S B を参照する必要がない。上記の例は、C R A ピクチャタイプに関して説明されているが、本技法はまた、リンク切断アクセス(B L A : broken link access)ピクチャタイプまたは他のピクチャタイプなど、他のピクチャタイプとともに実装され得る。

【 0 0 3 8 】

[0044]図 1 は、本開示の技法による、P O C 値を管理するための技法を利用し得る、例示的なビデオ符号化および復号システム 1 0 を示すブロック図である。図 1 に示されているように、システム 1 0 は、宛先デバイス 1 4 によって後で復号されるべき符号化されたビデオデータを与えるソースデバイス 1 2 を含む。具体的には、ソースデバイス 1 2 は、コンピュータ可読媒体 1 6 を介して宛先デバイス 1 4 にビデオデータを与える。ソースデバイス 1 2 および宛先デバイス 1 4 は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、ビデオストリーミングデバイスなどを含む、広範囲にわたるデバイスのいずれかを備え得る。場合によっては、ソースデバイス 1 2 および宛先デバイス 1 4 はワイヤレス通信に対応し (be equipped) 得る。

【 0 0 3 9 】

[0045]宛先デバイス 1 4 は、コンピュータ可読媒体 1 6 を介して、復号されるべき符号

10

20

30

40

50

化されたビデオデータを受信し得る。コンピュータ可読媒体 16 は、ソースデバイス 12 から宛先デバイス 14 に符号化されたビデオデータを移動することが可能な、任意のタイプの媒体またはデバイスを備え得る。一例では、コンピュータ可読媒体 16 は、ソースデバイス 12 が符号化されたビデオデータを宛先デバイス 14 にリアルタイムで直接送信することを可能にするための通信媒体を備え得る。符号化されたビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス 14 に送信され得る。通信媒体は、無線周波数 (RF) スペクトルまたは 1 つもしくは複数の物理伝送線路など、任意のワイヤレス通信媒体またはワイヤード通信媒体を備え得る。通信媒体は、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、またはインターネットなどのグローバルネットワークのような、パケットベースのネットワークの一部を形成し得る。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス 12 から宛先デバイス 14 への通信を容易にするために有用であり得る任意の他の機器を含み得る。

10

#### 【0040】

[0046]いくつかの例では、符号化されたデータは、出力インターフェース 22 から記憶デバイスへ出力され得る。同様に、符号化されたデータは、入力インターフェースによって記憶デバイスからアクセスされ得る。記憶デバイスは、ハードドライブ、Blu-ray (登録商標) ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性もしくは不揮発性のメモリ、または符号化されたビデオデータを記憶するための任意の他の好適なデジタル記憶媒体など、様々な分散されたまたはローカルにアクセスされるデータ記憶媒体のいずれをも含み得る。さらなる例では、記憶デバイスは、ソースデバイス 12 によって生成された符号化されたビデオを記憶し得るファイルサーバまたは別の中間記憶デバイスに対応し得る。宛先デバイス 14 は、ストリーミングまたはダウンロードを介して、記憶デバイスから記憶されたビデオデータにアクセスし得る。ファイルサーバは、符号化されたビデオデータを記憶し、その符号化されたビデオデータを宛先デバイス 14 に送信することが可能な任意のタイプのサーバであり得る。例示的なファイルサーバには、ウェブサーバ (たとえば、ウェブサイト用の)、FTPサーバ、ネットワーク接続記憶 (NAS) デバイス、またはローカルディスクドライブがある。宛先デバイス 14 は、インターネット接続を含む、任意の標準のデータ接続を通して符号化されたビデオデータにアクセスし得る。これは、ワイヤレスチャネル (たとえば、Wi-Fi (登録商標) 接続)、ワイヤード接続 (たとえば、DSL、ケーブルモデムなど)、または、ファイルサーバ上に記憶された符号化されたビデオデータにアクセスするために好適である、両方の組合せを含み得る。記憶デバイスからの符号化されたビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組合せであり得る。

20

30

#### 【0041】

[0047]本開示の技法は、ワイヤレス応用またはワイヤレス設定に必ずしも限定されるとは限らない。本技法は、オーバージエアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、動的適応ストリーミングオーバーHTTP (DASH: dynamic adaptive streaming over HTTP) などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されたデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の応用など、様々なマルチメディア応用のいずれかをサポートするビデオコーディングに適用され得る。いくつかの例では、システム 10 は、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、および/またはビデオ電話などの応用をサポートするために、一方向または両方向のビデオ送信をサポートするように構成され得る。

40

#### 【0042】

[0048]図 1 の例では、ソースデバイス 12 は、ビデオソース 18 と、ビデオエンコーダ 20 と、出力インターフェース 22 とを含む。宛先デバイス 14 は、入力インターフェース 28 と、ビデオデコーダ 30 と、ディスプレイデバイス 32 とを含む。本開示によれば、ソースデバイス 12 のビデオエンコーダ 20 は、本開示の技法に従って、POC 値情報をコーディングするための技法を適用するように構成され得る。他の例では、ソースデバ

50

イスおよび宛先デバイスは、他の構成要素または配置を含み得る。たとえば、ソースデバイス 1 2 は、外部カメラなどの外部のビデオソース 1 8 からビデオデータを受信し得る。同様に、宛先デバイス 1 4 は、一体型ディスプレイデバイスを含むのではなく、外部ディスプレイデバイスとインターフェースをとり得る。

【 0 0 4 3 】

[0049] 図 1 の示されるシステム 1 0 は一例にすぎない。本開示の技法による、POC 値を管理するための技法は、任意のデジタルビデオ符号化および/または復号デバイスによって実行され得る。一般に、本開示の技法は、ビデオ符号化デバイスによって実行されるが、これらの技法は、通常は「コーデック」と呼ばれるビデオエンコーダ/デコーダによっても実行され得る。その上、本開示の技法は、ビデオプロセッサによっても実行され得る。ソースデバイス 1 2 および宛先デバイス 1 4 は、ソースデバイス 1 2 が宛先デバイス 1 4 に送信するためのコード化ビデオデータを生成するようなコーディングデバイスの例にすぎない。いくつかの例では、デバイス 1 2、1 4 は、デバイス 1 2、1 4 の各々がビデオ符号化構成要素とビデオ復号構成要素とを含むように実質的に対称的に動作し得る。したがって、システム 1 0 は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、またはビデオ電話のための、ビデオデバイス 1 2 とビデオデバイス 1 4 との間の一方向または双方向のビデオ送信をサポートし得る。

【 0 0 4 4 】

[0050] ソースデバイス 1 2 のビデオソース 1 8 は、ビデオカメラ、以前にキャプチャされたビデオを含むビデオアーカイブ、および/またはビデオコンテンツプロバイダからビデオを受信するためのビデオフィールドインターフェースなどの、ビデオキャプチャデバイスを含み得る。さらなる代替として、ビデオソース 1 8 は、ソースビデオとしてコンピュータグラフィックススペースのデータ、または、ライブビデオ、アーカイブされたビデオ、およびコンピュータ生成ビデオの組合せを生成し得る。場合によっては、ビデオソース 1 8 がビデオカメラである場合、ソースデバイス 1 2 および宛先デバイス 1 4 は、いわゆるカメラ付き電話またはビデオ付き電話を形成し得る。しかしながら、上述されたように、本開示で説明される技法は、一般にビデオコーディングに適用可能であり得、ワイヤレスおよび/またはワイヤードの応用に適用され得る。各場合において、キャプチャされたビデオ、前にキャプチャされたビデオ、またはコンピュータ生成ビデオは、ビデオエンコーダ 2 0 によって符号化され得る。次いで、符号化されたビデオ情報は、出力インターフェース 2 2 によってコンピュータ可読媒体 1 6 上に出力され得る。

【 0 0 4 5 】

[0051] コンピュータ可読媒体 1 6 は、ワイヤレスブロードキャストもしくはワイヤードネットワーク送信などの一時媒体、またはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray ディスク、もしくは他のコンピュータ可読媒体などの記憶媒体（すなわち、非一時的記憶媒体）を含み得る。いくつかの例では、ネットワークサーバ（図示せず）は、たとえば、ネットワーク送信を介して、ソースデバイス 1 2 から符号化されたビデオデータを受信し、宛先デバイス 1 4 に符号化されたビデオデータを与え得る。同様に、ディスクスタンピング設備など、媒体製造設備のコンピューティングデバイスは、ソースデバイス 1 2 から符号化されたビデオデータを受信し、その符号化されたビデオデータを包含しているディスクを生成し得る。したがって、様々な例では、コンピュータ可読媒体 1 6 は、様々な形態の 1 つまたは複数のコンピュータ可読媒体を含むと理解され得る。

【 0 0 4 6 】

[0052] 宛先デバイス 1 4 の入力インターフェース 2 8 は、コンピュータ可読媒体 1 6 から情報を受信する。コンピュータ可読媒体 1 6 の情報は、ビデオエンコーダ 2 0 によって定義され、またビデオデコーダ 3 0 によって使用される、ブロックおよび他のコード化ユニット、たとえば、GOP の特性および/または処理を記述するシンタックス要素を含む、シンタックス情報を含み得る。ディスプレイデバイス 3 2 は、復号されたビデオデータをユーザに表示し、陰極線管（CRT）、液晶ディスプレイ（LCD）、プラズマディス

10

20

30

40

50

プレイ、有機発光ダイオード（O L E D）ディスプレイ、または別のタイプのディスプレイデバイスなどの様々なディスプレイデバイスのうちのいずれかを備え得る。

【 0 0 4 7 】

[0053]ビデオエンコーダ20およびビデオデコーダ30は、それぞれ、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ（D S P）、特定用途向け集積回路（A S I C）、フィールドプログラマブルゲートアレイ（F P G A）、離散論理、ソフトウェア、ハードウェア、ファームウェアまたはそれらの任意の組合せなど、様々な好適なエンコーダ回路のいずれかとして実装され得る。本技法がソフトウェアに部分的に実装される  
10  
とき、デバイスは、ソフトウェアに対する命令を好適な非一時的コンピュータ可読媒体に記憶し、本開示の技法を実行するための1つまたは複数のプロセッサを使用してハードウェアにおいてそれらの命令を実行し得る。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダに含まれ得、そのいずれも、それぞれのデバイスにおいて、複合エンコーダ/デコーダ（コーデック）の一部として統合され得る。ビデオエンコーダ20および/またはビデオデコーダ30を含むデバイスは、集積回路、マイクロプロセッサ、および/またはセルラー電話などのワイヤレス通信デバイスを備え得る。

【 0 0 4 8 】

[0054]図1には示されていないが、いくつかの態様では、ビデオエンコーダ20およびビデオデコーダ30は、それぞれ、オーディオエンコーダおよびデコーダと統合され得、  
20  
共通のデータストリームまたは別個のデータストリーム中のオーディオとビデオの両方の符号化を処理するために、適切なM U X - D E M U Xユニット、または他のハードウェアおよびソフトウェアを含み得る。適用可能であれば、M U X - D E M U Xユニットは、I T U H . 2 2 3マルチプレクサプロトコル、またはユーザデータグラムプロトコル（U D P）などの他のプロトコルに適合し得る。

【 0 0 4 9 】

[0055]本開示は全般に、ビデオエンコーダ20が、ある情報をビデオデコーダ30のような別のデバイスに「シグナリング」することに言及することがある。ただし、ビデオエンコーダ20は、いくつかのシンタックス要素をビデオデータの様々な符号化された部分に関連付けることによって情報をシグナリングし得ることを理解されたい。すなわち、ビデオエンコーダ20は、ビデオデータの様々な符号化された部分のヘッダに、いくつかの  
30  
シンタックス要素を記憶することによって、データを「シグナリング」し得る。場合によっては、そのようなシンタックス要素は、ビデオデコーダ30によって受信および復号されるより前に、符号化および記憶され得る。したがって、「シグナリング」という用語は、そのような通信がリアルタイムもしくはほぼリアルタイムで行われるか、または、符号化時にシンタックス要素を媒体に記憶し、次いで、この媒体に記憶された後の任意の時間にそのシンタックス要素が復号デバイスによって取り出され得るときなどに行われ得る、ある期間にわたって行われるかにかかわらず、概して、圧縮ビデオデータを復号するためのシンタックスまたは他のデータの通信を指し得る。

【 0 0 5 0 】

[0056]ビデオエンコーダ20およびビデオデコーダ30は、ビデオコーディング規格に従って動作し得る。例示的なビデオコーディング規格は、I T U - T H . 2 6 1と、I S O / I E C M P E G - 1 V i s u a lと、I T U - T H . 2 6 2またはI S O / I E C M P E G - 2 V i s u a lと、I T U - T H . 2 6 3と、I S O / I E C M P E G - 4 V i s u a lと、そのスケーラブルビデオコーディング（S V C）拡張およびマルチビュービデオコーディング（M V C）拡張を含む（I S O / I E C M P E G - 4 A V Cとしても知られる）I T U - T H . 2 6 4とを含む。最近、新しいビデオコーディング規格、すなわち高効率ビデオコーディング（H E V C）の設計が、I T U - T V i d e o C o d i n g E x p e r t s G r o u p（V C E G）およびI S O / I E C M o t i o n P i c t u r e E x p e r t s G r o u p（M P E G）の  
40  
J o i n t C o l l a b o r a t i o n T e a m o n V i d e o C o d i n g  
50

( J C T - V C ) によって確定された。以下で H E V C W D と呼ばれる、最新の H E V C ドラフト仕様が、[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-O1003-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-O1003-v1.zip) から入手可能である。H E V C に対するマルチビュー拡張、すなわち M V - H E V C も、J C T - 3 V によって開発されている。以下で M V - H E V C W D 6 と呼ばれる、M V - H E V C の最近のワーキングドラフト ( W D ) が、[http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/6\\_Geneva/wg11/JCT3V-F1004-v6.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1004-v6.zip) から入手可能である。S H V C と称する H E V C に対するスケラブル拡張も、J C T - V C によって開発されている。以下で S H V C W D 4 と呼ばれる、S H V C の最近のワーキングドラフト ( W D ) は、[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-O1008-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-O1008-v1.zip) から入手可能である。ただし、本開示の技法は、いかなる特定のコーディング規格にも限定されない。ビデオコーディング規格の他の例には、M P E G - 2 および I T U - T H . 2 6 3 が含まれる。

10

**【 0 0 5 1 】**

[0057] H E V C の規格化の取組みは、H E V C テストモデル ( H M ) と呼ばれるビデオコーディングデバイスのモデルに基づく。H M は、たとえば I T U - T H . 2 6 4 / A V C に従う既存のデバイスに対して、ビデオコーディングデバイスのいくつかの追加の能力を仮定する。たとえば、H . 2 6 4 は、9 つのイントラ予測符号化モードを提供するが、H M は、3 3 ものイントラ予測符号化モードを提供し得る。

20

**【 0 0 5 2 】**

[0058] 概して、H M の作業モデルは、ビデオフレームまたはピクチャが、ルーマサンプルとクロマサンプルの両方を含むツリーブロックまたは最大コーディングユニット ( L C U ) のシーケンスに分割され得ることを記載している。ビットストリーム内のシンタックスデータは、L C U にとってのサイズを定義し得、L C U は、ピクセルの個数に関して最大のコーディングユニットである。スライスは、コーディング順序でいくつかの連続するツリーブロックを含む。ビデオフレームまたはピクチャは、1 つまたは複数のスライスに区分され得る。各ツリーブロックは、4 分木に従ってコーディングユニット ( C U ) にスプリットされ得る。概して、4 分木データ構造は C U ごとに 1 つのノードを含み、ルートノードはツリーブロックに対応する。C U が 4 つのサブ C U にスプリットされる場合、C U に対応するノードは、4 つのリーフノードを含み、その各々は、サブ C U の 1 つに対応する。

30

**【 0 0 5 3 】**

[0059] 4 分木データ構造の各ノードは、対応する C U にシンタックスデータを与え得る。たとえば、4 分木内のノードは、そのノードに対応する C U がサブ C U にスプリットされるか否かを示すスプリットフラグを含み得る。C U のシンタックス要素は、繰り返して ( recursively ) 定義され得、C U がサブ C U にスプリットされるか否かに依存し得る。C U がさらにスプリットされない場合、その C U はリーフ C U と呼ばれる。本開示では、元のリーフ C U の明示的スプリットが存在しない場合でも、リーフ C U の 4 つのサブ C U はリーフ C U とも呼ばれることになる。たとえば、1 6 × 1 6 サイズの C U がさらにスプリットされない場合、この 1 6 × 1 6 C U が決してスプリットされなくても、4 つの 8 × 8 サブ C U はリーフ C U とも呼ばれることになる。

40

**【 0 0 5 4 】**

[0060] C U は、C U がサイズの特異性 ( distinction ) を有しないことを別にすれば、H . 2 6 4 規格のマクロブロックと同様の目的を有する。たとえば、ツリーブロックは、( サブ C U とも呼ばれる ) 4 つの子ノードにスプリットされ得、各子ノードは、今度は親ノードとなり、別の 4 つの子ノードにスプリットされ得る。4 分木のリーフノードと呼ばれる、最後のスプリットされていない子ノードは、リーフ C U とも呼ばれるコーディングノードを備える。コード化ビットストリームに関連するシンタックスデータは、最大 C U

50

深度と呼ばれる、ツリーブロックがスプリットされ得る最大回数を定義し得、また、コーディングノードの最小サイズを定義し得る。それに応じて、ビットストリームはまた、最小コーディングユニット（SCU）を定義し得る。本開示は、「ブロック」という用語を、HEVCのコンテキストにおいて、CU、PU、もしくはTUのうちのいずれか、または他の規格のコンテキストにおいて、同様のデータ構造（たとえば、H.264/AVCのマクロブロックおよびそのサブブロック）を指すために使用する。

【0055】

[0061] CUは、コーディングノードと、コーディングノードに関連付けられた予測ユニット（PU）および変換ユニット（TU）とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状が正方形でなければならない。CUのサイズは、8×8ピクセルから最大64×64ピクセル以上をもつツリーブロックのサイズにまでわたり得る。各CUは、1つまたは複数のPUと1つまたは複数のTUとを包含し得る。CUに関連付けられたシンタックスデータは、たとえば、1つまたは複数のPUへのCUの区分を記述し得る。区分モードは、CUがスキップモード符号化もしくは直接モード符号化されるのか、イントラ予測モード符号化されるのか、またはインター予測モード符号化されるのかの間で異なり得る。PUは、形状が非正方形に区分され得る。CUに関連付けられたシンタックスデータはまた、たとえば、4分木に従う1つまたは複数のTUへのCUの区分を記述し得る。TUは、形状が正方形または非正方形（たとえば、長方形）であってよい。

【0056】

[0062] HEVC規格は、CUごとに異なり得る、TUに従う変換を可能にする。TUは、一般に、区分されたLCUのために定義された所与のCU内のPUのサイズに基づいてサイズ決定されるが、これは常にそうであるとは限らない。TUは、一般に、PUと同じサイズであるか、またはPUよりも小さい。いくつかの例では、CUに対応する残差サンプルは、「残差4分木」（RQT）として知られる4分木構造を使用して、より小さいユニットに再分割され得る。RQTのリーフノードは、変換ユニット（TU）と呼ばれることがある。TUに関連付けられたピクセル差分値は、変換係数を生成するために変換され得、変換係数は量子化され得る。

【0057】

[0063] リーフCUは、1つまたは複数の予測ユニット（PU）を含み得る。概して、PUは、対応するCUのすべてまたは一部分に対応する空間的エリアを表し、そのPUの参照サンプルを取り出すためのデータを含み得る。その上、PUは、予測に関係するデータを含む。たとえば、PUがイントラモード符号化されるとき、PUに関するデータは、残差4分木（RQT）に含まれ得、残差4分木は、PUに対応するTUに関するイントラ予測モードを記述するデータを含み得る。別の例として、PUがインターモード符号化されるとき、PUは、PUのための1つまたは複数の動きベクトルを定義するデータを含み得る。PUのための動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルについての解像度（たとえば、1/4ピクセル精度または1/8ピクセル精度）、動きベクトルが指す参照ピクチャ、および/または動きベクトルのための参照ピクチャリスト（たとえば、リスト0、リスト1、またはリストC）を記述し得る。

【0058】

[0064] インター予測PU（より詳細には、PUが対応するCUのインター予測部分）のための動き情報は、参照ピクチャリスト識別子と、参照ピクチャリスト中の参照ピクチャの位置に対応する参照インデックスとを含み得る。ビデオエンコーダ20は、たとえば、（元の構築された参照ピクチャリストに対して行われたいかなる修正をも含む）参照ピクチャリストを構築し、（たとえば、絶対差分和（SAD）メトリクス、または同様のメトリクスに基づいて）参照ブロックとも呼ばれる、厳密に一致するブロックを識別するために、参照ピクチャリスト中の参照ピクチャの中で動き探索を実行し得る。参照ブロックのロケーションをシグナリングするために、ビデオエンコーダ20は、PUのための参照ピクチャリスト識別子と参照インデックスとを符号化し得る。さらに、ビデオエンコーダ20

10

20

30

40

50



0 は、たとえば、スライスヘッダ中、および/または、PPSもしくはSPSなどのパラメータセット中で、参照ピクチャのためのPOC値を表すデータをシグナリングすることによって、参照ピクチャリスト中に含まれるべき参照ピクチャを表すデータを符号化し得る。

#### 【0059】

[0065] 1つまたは複数のPUを有するリーフCUはまた、1つまたは複数の変換ユニット(TU)を含み得る。変換ユニットは、上記で説明されたように、(TU4分木構造とも呼ばれる)RQTを使用して指定され得る。たとえば、スプリットフラグは、リーフCUが4つの変換ユニットにスプリットされるか否かを示し得る。次いで、各変換ユニットは、さらなるサブTUにさらにスプリットされ得る。TUがこれ以上スプリットされないとき、そのTUはリーフTUと呼ばれることがある。一般に、イントラコーディングの場合、リーフCUに属するすべてのリーフTUは同じイントラ予測モードを共有する。すなわち、一般に、リーフCUのすべてのTUの予測値を計算するために同じイントラ予測モードが適用される。イントラコーディングの場合、ビデオエンコーダは、イントラ予測モードを使用して各リーフTUの残差値を、TUに対応するCUの一部分と元のブロックとの間の差分として計算し得る。TUは、必ずしも、PUのサイズに限定されとは限らない。したがって、TUは、PUよりも大きくまたは小さくなり得る。イントラコーディングの場合、PUは、同じCUの対応するリーフTUとコロケートされ得る。いくつかの例では、リーフTUの最大サイズは、対応するリーフCUのサイズに対応し得る。

#### 【0060】

[0066] その上、リーフCUのTUは、残差4分木(RQT)と呼ばれる、それぞれの4分木データ構造にも関連付けられ得る。すなわち、リーフCUは、リーフCUがどのようにTUに区分されるかを示す4分木を含み得る。TU4分木のルートノードは一般に、リーフCUに対応し、一方、CU4分木のルートノードは一般に、ツリーブロック(またはLCU)に対応する。スプリットされないRQTのTUは、リーフTUと呼ばれる。一般に、本開示は、別段の注記がない限り、CUおよびTUという用語を、それぞれ、リーフCUおよびリーフTUを指すために使用する。

#### 【0061】

[0067] ビデオシーケンスは、通常、一連のビデオフレームまたはピクチャを含む。ピクチャグループ(GOP)は、一般に、ビデオピクチャのうちの一連の1つまたは複数を用意する。GOPは、GOP中に含まれるいくつかのピクチャを記述するシンタックスデータを、GOPのヘッダ中、ピクチャの1つもしくは複数のヘッダ中、または他の場所に含み得る。ピクチャの各スライスは、それぞれのスライスの符号化モードを記述するスライスシンタックスデータを含み得る。ビデオエンコーダ20は通常、ビデオデータを符号化するために、個々のビデオスライス内のビデオブロックに対して動作する。ビデオブロックは、CU内のコーディングノードに対応し得る。ビデオブロックは、固定のサイズまたは可変のサイズを有してよく、指定されたコーディング規格に従ってサイズが異なり得る。

#### 【0062】

[0068] 一例として、HMは、様々なPUサイズでの予測をサポートする。特定のCUのサイズが $2N \times 2N$ であると仮定すると、HMは、 $2N \times 2N$ または $N \times N$ のPUサイズでのイントラ予測と、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称PUサイズでのインター予測とをサポートする。HMは、また、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ のPUサイズでのインター予測のための、非対称区分をサポートする。非対称区分では、CUの一方の方向は区分されず、一方、他方の方向は、25%と75%とに区分される。25%の区分に対応するCUの部分は、「n」とその後ろに付く「Up」、「Down」、「Left」、または「Right」という表示によって示される。したがって、たとえば、「 $2N \times nU$ 」は、上部で $2N \times 0.5N$  PU、および下部で $2N \times 1.5N$  PUに水平に区分される $2N \times 2N$  CUを指す。

#### 【0063】

[0069] 本開示では、「 $N \times N (NxN)$ 」および「 $N \times N (N \text{ by } N)$ 」は、垂直寸法およ

10

20

30

40

50

び水平寸法の観点からビデオブロックのピクセル寸法、たとえば、 $16 \times 16$  ( $16 \times 16$ )ピクセルまたは $16 \times 16$  ( $16 \text{ by } 16$ )ピクセルを指すために互換的に使用され得る。概して、 $16 \times 16$ ブロックは、垂直方向に $16$ ピクセル ( $y = 16$ )、および水平方向に $16$ ピクセル ( $x = 16$ )を有する。同様に、 $N \times N$ ブロックは、概して、垂直方向に $N$ ピクセル、および水平方向に $N$ ピクセルを有し、ここで、 $N$ は非負整数値を表す。ブロック中のピクセルは、行および列に配置され得る。その上、ブロックは、必ずしも、水平方向において垂直方向と同じ数のピクセルを有する必要があるとは限らない。たとえば、ブロックは $N \times M$ ピクセルを備え得、ここで、 $M$ は必ずしも $N$ に等しいとは限らない。

【0064】

[0070]CUのPUを使用したイントラ予測コーディングまたはインター予測コーディングに続いて、ビデオエンコーダ20は、CUのTUのための残差データを計算し得る。PUは、空間的領域(ピクセル領域とも呼ばれる)において予測ピクセルデータを生成する方法またはモードを記述するシンタックスデータを備え得、TUは、変換、たとえば、残差ビデオデータへの離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または概念的に同様の変換の適用後に、変換領域において係数を備え得る。残差データは、符号化されていないピクチャのピクセルと、PUに対応する予測値との間のピクセル差分に対応し得る。ビデオエンコーダ20は、CUのための残差データを含むTUを形成し、次いで、CUのための変換係数を生成するためにTUを変換し得る。

【0065】

[0071]変換係数を生成するための任意の変換に続いて、ビデオエンコーダ20は、変換係数の量子化を実行し得る。量子化は、一般に、係数を表すために使用されるデータの量をできるだけ低減するために、変換係数が量子化され、さらなる圧縮を実現するプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。たとえば、 $n$ ビット値は、量子化中に $m$ ビット値に切り捨てられ得、ここで、 $n$ は $m$ よりも大きい。

【0066】

[0072]量子化に続いて、ビデオエンコーダは、変換係数を走査し、量子化された変換係数を含む2次元行列から1次元ベクトルを生成し得る。走査は、アレイの前部により高いエネルギー(したがって、より低い周波数)係数を配置し、アレイの後部により低いエネルギー(したがって、より高い周波数)係数を配置するように設計され得る。いくつかの例では、ビデオエンコーダ20は、エントロピー符号化され得るシリアル化ベクトルを生成するために、量子化された変換係数を走査するためにあらかじめ定義された走査順序を利用し得る。他の例では、ビデオエンコーダ20は、適応走査を実行し得る。1次元ベクトルを形成するために、量子化された変換係数を走査した後、ビデオエンコーダ20は、たとえば、コンテキスト適応型可変長コーディング(CAVLC)、コンテキスト適応型バイナリ算術コーディング(CABAC)、シンタックスベースコンテキスト適応型バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピー符号化方法に従って、1次元ベクトルをエントロピー符号化し得る。ビデオエンコーダ20はまた、ビデオデータを復号する際にビデオデコーダ30によって使用するための、符号化ビデオデータに関連付けられたシンタックス要素をエントロピー符号化し得る。

【0067】

[0073]CABACを実行するために、ビデオエンコーダ20は、コンテキストモデル内のコンテキストを、送信されるべきシンボルに割り当て得る。コンテキストは、たとえば、シンボルの隣接値が非ゼロであるか否かに関係し得る。CAVLCを実行するために、ビデオエンコーダ20は、送信されるべきシンボルの可変長コードを選択し得る。VLC中のコードワードは、比較的短いコードが優勢シンボル(more probable symbols)に対応し、一方、より長いコードが劣勢シンボル(less probable symbols)に対応するように、構成され得る。このようにして、VLCの使用は、たとえば、送信されるべき各シンボルのための等長コードワードを使用することに勝るビット節約を達成し得る。確率決定

10

20

30

40

50

は、シンボルに割り当てられたコンテキストに基づき得る。

【 0 0 6 8 】

[0074]ビデオエンコーダ 2 0 は、さらに、ブロックベースのシンタックスデータ、フレームベースのシンタックスデータ、および GOP ベースのシンタックスデータなどのシンタックスデータを、たとえば、フレームヘッダ、ブロックヘッダ、スライスヘッダ、または GOP ヘッダ中で、ビデオデコーダ 3 0 へ送り得る。GOP シンタックスデータは、それぞれの GOP 中のいくつかのフレームを記述し得、フレームシンタックスデータは、対応するフレームを符号化するために使用された符号化 / 予測モードを示し得る。

【 0 0 6 9 】

[0075]コード化されたビデオセグメントは、ビデオ電話、記憶、ブロードキャスト、またはストリーミングなどのアプリケーションに対処する「ネットワークフレンドリ」なビデオ表現を提供する、NAL ユニットに編成され得る。NAL ユニットは、ビデオコーディングレイヤ (VCL) NAL ユニットおよび非 VCL NAL ユニットとしてカテゴリー分類され得る。VCL ユニットは、コア圧縮エンジンからの出力を包含し得、ブロック、マクロブロック、および / またはスライスレベルのデータを含み得る。他の NAL ユニットは、非 VCL NAL ユニットであり得る。いくつかの例では、通常は 1 次コード化ピクチャとして提示される、1 つの時間インスタンス中のコード化されたピクチャは、1 つまたは複数の NAL ユニットを含み得るアクセスユニット中に包含され得る。

【 0 0 7 0 】

[0076]上述されたように、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、一般に、MV - HEVC または SHVC など、ビデオコーディング規格またはビデオコーディング規格の拡張に従って動作するように構成され得る。したがって、例として、本開示の技法は、MV - HEVC に関して以下で説明されるが、これらの技法は、SHVC など、他のビデオコーディング規格または拡張に適用され得ることを理解されたい。

【 0 0 7 1 】

[0077]HEVC では、NAL ユニットタイプによって識別され得るいくつかの異なるピクチャタイプがある。1 つのピクチャタイプは、概して、IDR ピクチャ、クリーンランダムアクセス (CRA) ピクチャ、時間的レイヤアクセス (TLA) ピクチャ、リンク切断アクセス (BLA) ピクチャ、STSA (ステップワイズ時間的サブレイヤアクセス) ピクチャ、リーディングピクチャ (すなわち、後述される RASL ピクチャおよび RADL ピクチャ)、またはいくつかの他の異なるピクチャタイプを含み得る、ランダムアクセスピクチャタイプである。

【 0 0 7 2 】

[0078]CRA ピクチャシンタックスは、オープン GOP 動作として知られる効率的な時間的コーディング順序をサポートする、ランダムアクセスポイント (RAP) のロケーション、すなわち、デコーダがビットストリーム中でより早く現れたいかなるピクチャを復号する必要もなしに、ピクチャを正常に復号することを開始することができるビットストリーム中のロケーションにおける、独立してコード化されたピクチャの使用を指定する。ランダムアクセスのためのサポートは、チャンネルスイッチングと、シーク動作と、動的ストリーミングサービスとを可能にするために必要とされ得る。復号順序において CRA ピクチャに後続し、表示順序において CRA ピクチャに先行する、いくつかのピクチャは、デコーダにおいて利用可能ではないピクチャへのインターピクチャ予測参照を包含し得る。したがって、これらの復号不可能なピクチャは、CRA ポイントにおいてその復号プロセスを開始するデコーダによって破棄される。このために、そのような復号不可能なピクチャは、ランダムアクセススキップリーディング (RASL : random access skipped leading) ピクチャとして識別される。異なる元のコード化されたビットストリームからのスプライスポイント (splice points) のロケーションは、リンク切断アクセス (BLA) ピクチャによって示され得る。ビットストリームスプライシング動作は、単に、あるビットストリーム中の CRA ピクチャの NAL ユニットタイプを、BLA ピクチャを示す値に変更することと、他のビットストリーム中の RAP ピクチャの位置において新しいピッ

10

20

30

40

50

トストリームを連結することによって実行され得る。

【 0 0 7 3 】

[0079]ビデオエンコーダ20およびビデオデコーダ30は、ビデオシーケンスの各ピクチャに、ピクチャの識別のためのPOC値を割り当て得る。ビデオエンコーダ20は、たとえば、スライスヘッダ中、および/または、PPSもしくはSPSなどのパラメータセット中で、参照ピクチャのためのPOC値を表すデータをシグナリングすることによって、参照ピクチャリスト中に含まれるべき参照ピクチャを表すデータを符号化し得る（および、ビデオデコーダ30は復号し得る）。特定のコード化されたピクチャのための、シンタックス要素PicOrderCntValによって表されるPOC値は、同じコード化されたビデオシーケンス中の他のピクチャに対する、ピクチャ出力プロセスにおけるピクチャの相対的順序を示す。

10

【 0 0 7 4 】

[0080]POC値は、LSBとMSBとを含む。POCは、MSBとLSBとを連結または加算することによって取得され得る。LSBは、スライスヘッダ中でシグナリングされ得、MSBは、現在のピクチャのNALユニットタイプと、NALユニットタイプランダムアクセススキップリーディング(RASL: random access skipped leading)もしくはランダムアクセス復号可能リーディング(RADL: random access decodable leading)のものではないか、またはサブレイヤ非参照ピクチャではなく、0に等しいtemporal\_id値を有する、復号順序における前のピクチャのMSBおよびLSBとに基づいて、ビデオデコーダ30によって計算され得る。0に等しいTemporalIdと、現在のピクチャのnuh\_layer\_idに等しいnuh\_layer\_idとを有し、RASLピクチャ、RADLピクチャ、またはサブレイヤ非参照ピクチャではない、そのようなピクチャは、POCアンカーピクチャと呼ばれる。

20

【 0 0 7 5 】

[0081]現在のピクチャが、1に等しいNoRaslOutputFlagを有するイントラランダムアクセスポイント(IRAP)ピクチャであるか、またはビットストリームの最初のピクチャであるCRAピクチャであるとき、POC MSBの値は0に等しいと推論される（すなわち、コードによって決定される）。マルチレイヤビットストリーム（たとえば、2つ以上のレイヤを有するSHVCまたはMV-HEVCビットストリーム）では、1つまたは2つ以上のピクチャがIRAPピクチャであるとともに1つまたは複数のピクチャが非IRAPピクチャであるアクセスユニットが存在し得る。そのようなAUは、非整合IRAP AUと呼ばれることがある。非整合IRAP AUを包含するビットストリームを復号するとき、シグナリングされたPOC LSB値に基づいてピクチャについて導出されたPOCが、アクセスユニット中のすべてのピクチャ（たとえば、各レイヤのピクチャ）がPicOrderCntValの同じ値を有するべきであるというビットストリーム要件に違反するようになることがあり得る。

30

【 0 0 7 6 】

[0082]したがって、場合によっては、POC値は、特定のアクセスユニット中のピクチャのPOC整合を維持するために、そのアクセスユニットのすべてのピクチャについてリセットされ得る。MV-HEVC WD5では、非整合IRAP AUがビットストリーム中に存在するときでも、AU中のすべてのピクチャのPOCが同じであるように現在のピクチャおよびDPB内のピクチャのPicOrderCntValの値が調整されるように、フラグpoc\_reset\_flagシンタックス要素が、ピクチャのPOCをリセットするために使用され得る。

40

【 0 0 7 7 】

[0083]2014年4月4日に出願された、Chen他の「CROSS-LAYER POC ALIGNMENT FOR MULTI-LAYER BITSTREAMS THAT MAY INCLUDE NON-ALIGNED IRAP PICTURES」という米国特許出願第14/245,115号は、2つのフラグであるpoc\_msb\_reset\_flagとpoc\_lsb\_reset\_flagとを使用する、P

50

OCリセットを達成する別の方法について説明している。前者のフラグは、PicOrderCntValのMSBをリセットし、後者のフラグは、PicOrderCntValのLSBをリセットする。これらのフラグの両方が、スライスヘッダ中でシグナリングされる。

【0078】

[0084] 2013年10月14日に出願された米国仮第61/890,868号は、POC MSBまたはPOCリセット指示のいずれかを包含するピクチャが失われるときの、正確なPOC値の回復のための情報を包含する、SEIメッセージを提案している。

【0079】

[0085] Hannukseela他、「MV-HEVC/SHVC HLS: on POC value derivation」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコーディング共同研究部会(JCT-VC)、第15回会合:ジュネーブ、スイス、2013年10月23日~11月1日、JCTVC-00275v3([http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-00275-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00275-v3.zip)において入手可能)と、Sjoberg他、「HLS: Error robust POC alignment」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコーディング共同研究部会(JCT-VC)、第15回会合:ジュネーブ、スイス、2013年10月23日~11月1日、JCTVC-00176v3([http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-00176-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00176-v3.zip)において入手可能)とは、マルチレイヤビデオコーディングにおけるPOC値をシグナリングおよび導出するための他の方法を提案している。

【0080】

[0086] 2014年11月24日に出願された米国出願第14/552,129号は、2つのフラグ(すなわち、poc\_msb\_reset\_flagおよびpoc\_lsb\_reset\_flag)を2ビットPOCリセットインジケータに置き換えることを提案し、POCリセット期間の概念を導入しており、ここにおいて、各POCリセット期間は、2つの連続するPOCリセット期間について異なることが必要とされるPOCリセット期間識別子によって識別される。そのような概念によって、POCリセットが実行されるアクセスユニット中の1つまたは複数のピクチャの損失の場合に、デコーダがPOCのクロスレイヤ整合を保つことが可能であるように、POCリセット機構のエラーレジリエンスが改善される。

【0081】

[0087] ビデオエンコーダ20は、スライスをそれぞれのPOCリセット期間に割り当て得る。各POCリセット期間は、1つまたは複数のスライスを含み得る。したがって、POC値がPOCリセット期間の間にリセットされる(コーディング順序においてPOCリセット期間に先行する参照ピクチャのPOC値をリセットすることを含む)とき、ビデオエンコーダ20は、リセットされたPOC値に基づいて、参照ピクチャリスト中に含まれるべき参照ピクチャのPOC値をシグナリングし得る。

【0082】

[0088] ビデオエンコーダ20およびビデオデコーダ30は、POC値を使用して、DPBに記憶されたピクチャのピクチャ出力とバンピング(bumping)とを実行し得る。ピクチャ出力およびバンピングプロセスは、DPBから「出力のために必要とされる」および「参照のために使用されない」とマークされているピクチャを出力するためのプロセスである。Ramasubramonian他、「MV-HEVC/SHVC HLS: Sub-DPB based DPB operations」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコーディング共同研究部会(JCT-VC)、第15回会合:ジュネーブ、スイス、2013年10月23日~

10

20

30

40

50

11月1日、JCTVC-00217 ([http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-00217-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00217-v1.zip)において入手可能)の採用後、HEVC拡張のためのDPBのピクチャを出力およびバンピングするための概念は、次のことを組み込むように更新されている。

- ・レイヤごとの別個のDPB容量、いわゆるサブDPBを指定する
- ・レイヤにわたるサブDPB容量の共有はない
- ・各レイヤは、それ自体のパラメータ(最大ピクチャ、最大レイテンシ、最大並べ替え)を有する
- ・「出力レイヤセット」ごとの異なるパラメータ
- ・動作点の定義を、「レイヤセット」ではなく、出力レイヤセットに固有であるように変更する

10

変更後、ピクチャ出力およびバンピングプロセスは、次のように動作し、すなわち、デコーダ適合のために、ピクチャ出力プロセスは、(1)ピクチャの復号前であるが、ピクチャの最初のスライスのスライスヘッダをパースした後と、(2)ピクチャの最後の復号ユニットがコード化ピクチャバッファから除去される時を含めて、ピクチャごとに2回呼び出される。バンピングプロセスが呼び出される時、このプロセスは、選択されたアクセスユニットに属するすべてのピクチャを出力する。言い換えれば、バンピングプロセスは、それらのレイヤ識別にかかわらず、同じアクセスユニットに属するすべてのピクチャを出力するので、レイヤ固有のプロセスではない。

20

【0083】

[0089]ピクチャ出力プロセスは、ピクチャが「出力のために必要とされない」および「参照のために必要とされない」とマークされているとき、復号ピクチャバッファのための記憶バッファを空にするためのプロセスである。ピクチャ出力プロセスは、特定のレイヤのために呼び出されるピクチャの除去が、他のレイヤからピクチャを除去しないものとするので、レイヤ固有に動作する。

【0084】

[0090]ピクチャ出力、バンピングおよび除去を含む、DPBのための出力順序の例示的な動作は、JCTVC-00217における追加と、MV-HEVC WD6への、Ramassubramonian他、「MV-HEVC/SHVC HLS: On flushing of decoded pictures from DPB based on NoOutputOfPriorPicsFlag」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコーディング共同研究部会(JCT-VC)、第15回会合:ジュネーブ、スイス、2013年10月23日~11月1日、00266 ([http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/15\\_Geneva/wg11/JCTVC-00266-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00266-v1.zip)において入手可能)の追加とを含めた後、次のようになる。

30

C.5.2 出力順序DPBの動作

C.5.2.1 一般

40

復号ピクチャバッファはサブDPBからなり、各サブDPBは、1つのレイヤのみの復号されたピクチャの記憶のためのピクチャ記憶バッファを包含する。サブDPBのピクチャ記憶バッファの各々は、「参照のために使用される」とマークされているか、または将来の出力のために保持されている、復号されたピクチャを包含する。

DPBからのピクチャの出力および除去のためのプロセスと、ピクチャ復号、マーキング、追加のバンピング、および記憶のためのプロセスと、「バンピング」プロセスとが、以下で指定される。これらのプロセスは、ベースレイヤから開始して、ビットストリーム中のレイヤのnuh\_layer\_id値の昇順に、レイヤごとに独立して適用される。これらのプロセスが特定のレイヤに対して適用されるとき、ピクチャをクロップおよび出力し、ピクチャに「出力のために必要とされない」とマークし、いかなるレイヤのための

50

ピクチャ記憶バッファをも空にし得る「バンピング」プロセスを除いて、特定のレイヤのためのサブDPBのみが影響を受ける。

ピクチャ $n$ および現在のピクチャを、`nuh_layer_id`の特定の値のためのアクセスユニット $n$ のコード化されたピクチャまたは復号されたピクチャとし、ここにおいて、 $n$ は非負整数である。

これらのプロセスが、`currLayerId`に等しい`nuh_layer_id`を有するレイヤに対して適用されるとき、変数`MaxNumReorderPics`、`MaxLatencyIncreasePlus1`、`MaxLatencyPictures`、および`MaxDecPicBufferingMinus1`は、次のように導出される。

- Annex GまたはHにおいて指定されているプロファイルのうちの1つまたは複数に適合するCVSが、節2~10、Annex F、およびAnnex GまたはHにおいて指定されている復号プロセスを適用することによって復号される場合、次のことが適用される。

- `MaxNumReorderPics`は、アクティブなVPSの`max_vps_num_reorder_pics[TargetOptLayerSetIdx][HighestTid]`に等しく設定される。

- `MaxLatencyIncreasePlus1`は、アクティブなVPSのシンタックス要素`max_vps_latency_increase_plus1[TargetOptLayerSetIdx][HighestTid]`の値に等しく設定される。

- `MaxLatencyPictures`は、アクティブなVPSの`VpsMaxLatencyPictures[TargetOptLayerSetIdx][HighestTid]`に等しく設定される。

- `MaxDecPicBufferingMinus1`は、アクティブなVPSのシンタックス要素`max_vps_dec_pic_buffering_minus1[TargetOptLayerSetIdx][currLayerId][HighestTid]`の値に等しく設定される。

- そうでない(Annex Aにおいて指定されているプロファイルのうちの1つまたは複数に適合するCVSが、節2~10において指定されている復号プロセスを適用することによって復号される)場合、次のことが適用される。

- `MaxNumReorderPics`は、ベースレイヤのためのアクティブなSPSの`sps_max_num_reorder_pics[HighestTid]`に等しく設定される。

- `MaxLatencyIncreasePlus1`は、ベースレイヤのためのアクティブなSPSの`sps_max_latency_increase_plus1[HighestTid]`に等しく設定される。

- `MaxLatencyPictures`は、ベースレイヤのためのアクティブなSPSの`SpsMaxLatencyPictures[HighestTid]`に等しく設定される。

- `MaxDecPicBufferingMinus1`は、ベースレイヤのためのアクティブなSPSの`sps_max_dec_pic_buffering_minus1[HighestTid]`に等しく設定される。

#### C.5.2.2 DPBからのピクチャの出力および除去

現在のピクチャが現在のレイヤにおけるピクチャ0ではない(すなわち、現在のピクチャが0のPOC値を有していない)とき、現在のピクチャ、すなわち、ピクチャ $n$ の復号前であるが、現在のピクチャの最初のスライスのスライスヘッダをパースした後の、DPBからの現在のレイヤにおけるピクチャの出力および除去は、現在のピクチャの最初の復号ユニットがCPBから除去されるときに瞬時に起こり、次のように進む。

- 従属節8.3.2において指定されているようなRPSのための復号プロセスが呼び出される。

10

20

30

40

50

- 現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであるか、または現在のアクセスユニット中のベースレイヤピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、NoClrasOutputFlagが1に等しい場合、次の順序付けられたステップが適用される。

1. 変数NoOutputOfPriorPicsFlagは、次のようにテスト中のデコーダのために導出される。

- 現在のピクチャが、1に等しいNoRaslOutputFlagを有するCRAピクチャである場合、NoOutputOfPriorPicsFlagは、1に等しく設定される(no\_output\_of\_prior\_pics\_flagの値にかかわらず)。

- そうでない場合、現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、現在のレイヤのためのアクティブなSPSから導出されたpic\_width\_in\_luma\_samples、pic\_height\_in\_luma\_samples、またはsps\_max\_dec\_pic\_buffering\_minus1[HighestTid]の値が、現在のレイヤにおいて先行するピクチャを復号するときに現在のレイヤについてアクティブであったSPSからそれぞれ導出されたpic\_width\_in\_luma\_samples、pic\_height\_in\_luma\_samples、またはsps\_max\_dec\_pic\_buffering\_minus1[HighestTid]の値と異なる場合、NoOutputOfPriorPicsFlagは、no\_output\_of\_prior\_pics\_flagの値にかかわらず、テスト中のデコーダによって1に設定され得る(が、設定されるべきでない)。

注 - これらの条件下では、NoOutputOfPriorPicsFlagをno\_output\_of\_prior\_pics\_flagに等しく設定することが好ましいが、テスト中のデコーダは、この場合にはNoOutputOfPriorPicsFlagを1に設定することが可能にされる。

- そうでない場合、現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャである場合、NoOutputOfPriorPicsFlagは、no\_output\_of\_prior\_pics\_flagに等しく設定される。

- そうでない(現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャではなく、現在のアクセスユニット中のベースレイヤピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、NoClrasOutputFlagが1に等しい)場合、NoOutputOfPriorPicsFlagは1に等しく設定される。

2. テスト中のデコーダのために導出されたNoOutputOfPriorPicsFlagの値は、次のようにHRDに対して適用される。

- NoOutputOfPriorPicsFlagが1に等しい場合、サブDPB内のすべてのピクチャ記憶バッファは、それらが包含しているピクチャを出力せずに空にされ、サブDPBフルネス(sub-DPB fullness)は0に等しく設定される。

- そうでない(NoOutputOfPriorPicsFlagが0に等しい)場合、「出力のために必要とされない」および「参照のために使用されない」とマークされているピクチャを包含するすべてのピクチャ記憶バッファは(出力せずに)空にされ、サブDPB内のすべての空でないピクチャ記憶バッファは、従属節C.5.2.4において指定されている「バンピング」プロセスを繰り返し呼び出すことによって空にされ、サブDPBフルネスは0に等しく設定される。

- そうでない場合、現在のレイヤにおけるピクチャを包含しており、「出力のために必要とされない」および「参照のために使用されない」とマークされている、すべてのピクチャ記憶バッファは(ピクチャが出力されることなく)空にされる。空にされる各ピクチャ記憶バッファに対して、サブDPBフルネスは1だけデクリメントされる。次の条件の

10

20

30

40

50



うちの1つまたは複数が真であるとき、空にされる各追加のピクチャ記憶バッファに対してサブDPBフルネスを1だけさらにデクリメントしながら、次の条件のいずれも真でなくなるまで、従属節C.5.2.4において指定されている「バンピング」プロセスが繰り返し呼び出される。

- 「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号されたピクチャを包含するアクセスユニットの数は、`MaxNumReorderPics`よりも大きい。

- `MaxLatencyIncreasePlus1`は0に等しくなく、それについて関連付けられた変数`PicLatencyCount`が`MaxLatencyPictures`以上である、「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号されたピクチャを包含する、少なくとも1つのアクセスユニットがある。

- サブDPB内の現在のレイヤにおけるピクチャの数は、`MaxDecPicBufferingMinus1+1`以上である。

C.5.2.3 ピクチャ復号、マーキング、追加のバンピング、および記憶

この従属節において指定されているプロセスは、ピクチャnの最後の復号ユニットがCPBから除去されるときに瞬時に起こる。

`PicOutputFlag`は次のように更新される。

- 現在のアクセスユニットが、ターゲット出力レイヤにおいてピクチャを包含しておらず、`alt_output_layer_flag`が1に等しい場合、次の順序付けられたステップが適用される。

- リスト`nonOutputLayerPictures`は、1に等しい`PicOutputFlag`を有し、ならびに、`TargetDecLayerIdList`中に含まれており、ターゲット出力レイヤ上にない`nuh_layer_id`値を有する、アクセスユニットのピクチャのリストである。

- リスト`nonOutputLayerPictures`の中で最高の`nuh_layer_id`値を有するピクチャは、リスト`nonOutputLayerPictures`から除去される。

- リスト`nonOutputLayerPictures`中に含まれる各ピクチャのための`PicOutputFlag`は、0に等しく設定される。

- そうでない場合、ターゲット出力レイヤ中に含まれないピクチャのための`PicOutputFlag`は、0に等しく設定される。

現在のピクチャが、1に等しい`PicOutputFlag`を有するとき、「出力のために必要とされる」とマークされており、出力順序において現在のピクチャに後続する、サブDPB内の現在のレイヤにおける各ピクチャについて、関連付けられた変数`PicLatencyCount`は、`PicLatencyCount+1`に等しく設定される。

現在のピクチャは、ピクチャの最後の復号ユニットが復号された後に復号されると見なされる。現在の復号されたピクチャは、サブDPB内の空のピクチャ記憶バッファ内に記憶され、次のことが適用される。

- 現在の復号されたピクチャが、1に等しい`PicOutputFlag`を有する場合、それは「出力のために必要とされる」とマークされ、その関連付けられた変数`PicLatencyCount`は、0に等しく設定される。

- そうでない(現在の復号されたピクチャが、0に等しい`PicOutputFlag`を有する)場合、それは「出力のために必要とされない」とマークされる。

現在の復号されたピクチャは、「短期参照のために使用される」とマークされる。

次の条件のうちの1つまたは複数が真であるとき、従属節C.5.2.4において指定されている「バンピング」プロセスが、次の条件のいずれも真でなくなるまで繰り返し呼び出される。

- 「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号されたピクチャを包含するアクセスユニットの数は、`MaxNumReorderPics`よりも大きい。

10

20

30

40

50

- Max Latency Increase Plus 1は0に等しくなく、それについて関連付けられた変数 Pic Latency CountがMax Latency Pictures以上である、「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号されたピクチャを包含する、少なくとも1つのアクセスユニットがある。

#### C.5.2.4 「バンピング」プロセス

「バンピング」プロセスは、次の順序付きステップからなる。

1. 出力のための最初のものである1つまたは複数のピクチャが、「出力のために必要とされる」とマークされたDPB内のすべてのピクチャのうちのPicOrderCntValの最小値を有するピクチャとして選択される。

2. これらのピクチャの各々が、nuh\_layer\_idの昇順で、ピクチャのためのアクティブSPS中で指定された適合クロッピングウィンドウを使用してクロップされ、クロップされたピクチャが出力され、そのピクチャが「出力のために必要とされない」とマークされる。

3. 「参照のために使用されない」とマークされ、クロップおよび出力されたピクチャのうちの一つであったピクチャを包含する、各ピクチャ記憶バッファが空にされる。

#### 【0085】

[0091]本開示の技法によれば、ビデオエンコーダ20およびビデオデコーダ30は、POC値リセットを実行するより前に、DPBからピクチャを出力し得る。たとえば、本開示の態様によれば、特定のアクセスユニットのためのPOC値リセットを実行するより前に、ビデオエンコーダ20およびビデオデコーダ30は、DPBに記憶されたすべてのピクチャを出力するように構成され得る。ビデオエンコーダ20およびビデオデコーダ30は、そのためのPOC値リセットが実行されるアクセスユニットの初期ピクチャをコーディングするより前に、「出力のために必要とされる」とマークされている、DPBに記憶されたすべてのピクチャを出力し得る。たとえば、ビデオエンコーダ20は、スライスヘッダをコーディングした後であるが、スライスのペイロードを符号化し、POC値を導出し、参照ピクチャリストを構築するより前に、出力プロセスを実行し得る。同様に、ビデオデコーダ30は、スライスヘッダをコーディングした後であるが、スライスのペイロードを復号し、POC値の導出、および参照ピクチャを復号するより前に、出力プロセスを実行し得る。

#### 【0086】

[0092]説明のための一例では、現在コーディング（符号化または復号）されているピクチャは、POCリセットピクチャ（たとえば、POCリセット期間においてコーディング順序で現在のレイヤの最初のピクチャである、1もしくは2に等しいpoc\_reset\_idcシンタックス要素、または3に等しいpoc\_reset\_idcシンタックス要素を有するピクチャ、）であり得る。この例では、現在のピクチャがPOCリセットピクチャであることに基づいて、ビデオエンコーダ20およびビデオデコーダ30は、現在のアクセスユニット（たとえば、現在のピクチャを包含するアクセスユニット）に属するピクチャを除いて、まだ出力されていないDPB内のすべての既存のピクチャを出力し得る。

#### 【0087】

[0093]このようにして、本開示の態様によれば、ビデオエンコーダ20およびビデオデコーダ30は、POC値リセットを実行するとき、DPBに記憶されたピクチャのためのPOC値整合を維持するように構成され得る。すなわち、ビデオエンコーダ20およびビデオデコーダ30は、POC値リセットより前に、（アクセスユニットごとではなく）一緒にDPBのピクチャを出力するように構成され得る。POCリセットより前に、このようにしてDPBのすべてのピクチャを出力することは、（同じアクセスユニット中であり、同じPOC値を有する）同じ時間的インスタンスのピクチャが同じまたは実質的に同じ時間に出力されることを保証する助けになり得る。加えて、場合によっては、すべてのより早いピクチャ（たとえば、DPBに記憶され、コーディング順序において現在コーディングされているピクチャより前に発生するピクチャ）を出力することは、ビデオエンコー

10

20

30

40

50

ダ 2 0 およびビデオデコーダ 3 0 が、アクセスユニットごとのベースでピクチャを出力するために必要とされるよりも比較的小さいバッファサイズを有して構成され得ることである。

#### 【 0 0 8 8 】

[0094]本開示の技法による、ピクチャ出力と、バンピングと、除去とを含む、DPBからピクチャを出力する例示的な動作が、以下に記載される。このプロセスは、たとえば、上述された文書JCTVC-00217およびJCTVC-00266によって修正されるようなMV-HEVCのプロセスに関して説明される。以下の例示的な説明、シンタックステーブルおよびセマンティクスでは、(文書JCTVC-00217およびJCTVC-00266によって修正されるような)MV-HEVCに対する追加が、ital<<イタリック体>>italを使用して表され(なお、本翻訳文中において、本願の英文明細書におけるイタリック体表記部分を、ital<<...>>italのように表記する)、削除が、「除去」が前に付いた括弧付きのテキスト(たとえば、[除去: “除去されたテキスト”])を使用して表される。概して、「要件」に関するステートメントは、規格または規格拡張のテキストの一部を形成するものであり、本開示の技法の目的のための要件を形成しないものと理解されたい。場合によっては、そのような「要件」は、適用可能であると決定され得、次いで、たとえば、その決定に基づいてビデオコーデックによって遵守され(adhered to)得る、ビットストリーム制約を含み得る)。

#### C.5.2.2 DPBからのピクチャの出力および除去

現在のピクチャが現在のレイヤにおけるピクチャ0ではないとき、現在のピクチャ、すなわち、ピクチャnの復号前であるが、現在のピクチャの最初のスライスのスライスヘッダをパースした後の、ital<<およびピクチャ順序カウンタのための復号プロセスの呼出し前の>>ital、DPBからの現在のレイヤにおけるピクチャの出力および除去は、現在のピクチャの最初の復号ユニットがCPBから除去されるときに瞬時に起こり、次のように進む。

- ital<<現在のピクチャのpic\_reset\_idcが0よりも大きく、現在のアクセスユニットがPOCリセット期間における最初のアクセスユニットであるとき、現在のアクセスユニットに属しておらず、「出力のために必要とされる」とマークされている、DPB内のすべてのピクチャが、DPB内の現在のアクセスユニット中にあるものを除いて、すべてのピクチャのうちPicOrderCntValの最小値を有するピクチャから開始して、PicOrderCntVal値の昇順で出力され、PicOrderCntValの同じ値を有するピクチャが、nuh\_layer\_id値の昇順で出力される。ピクチャが出力されるとき、ピクチャは、ピクチャのためのアクティブSPS中で指定された適合クロッピングウィンドウを使用してクロップされ、クロップされたピクチャが出力され、そのピクチャが「出力のために必要とされない」とマークされる>>ital。

- ital<<ピクチャ順序カウンタおよび>>ital、RPSのための復号プロセスが呼び出される。ital<<節2~10において指定されている復号プロセスを使用して、Annex Aにおいて指定されているプロファイルのうちの一つまたは複数に適合するCVSを復号するとき、呼び出されるピクチャ順序カウンタおよびRPSのための復号プロセスは、それぞれ従属節8.3.1および8.3.2において指定されているものである。Annex F、およびAnnex GまたはHにおいて指定されている復号プロセスを使用して、Annex GまたはHにおいて指定されているプロファイルのうちの一つまたは複数に適合するCVSを復号するとき、呼び出されるピクチャ順序カウンタおよびRPSのための復号プロセスは、それぞれ従属節、エラー!参照元が見つかりません。およびエラー!参照元が見つかりません。において指定されているものである>>ital。

- 現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであるか、または現在のアクセスユニット中のベースレイヤピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、NoClrasOutputFlagが1に等しい場合、次の順序付けられたステップが適用される。

1. 変数NoOutputOfPriorPicsFlagは、次のようにテスト中の

10

20

30

40

50

デコーダのために導出される。

- 現在のピクチャが、1に等しいNoRaslOutputFlagを有するCRAピクチャである場合、NoOutputOfPriorPicsFlagは、1に等しく設定される(no\_output\_of\_prior\_pics\_flagの値にかかわらず)。

- そうでない場合、現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、現在のレイヤのためのアクティブなSPSから導出されたpic\_width\_in\_luma\_samples、pic\_height\_in\_luma\_samples、またはsps\_max\_dec\_pic\_buffering\_minus1[HighestTid]の値が、現在のレイヤにおいて先行するピクチャを復号するときに現在のレイヤについてアクティブであったSPSからそれぞれ導出されたpic\_width\_in\_luma\_samples、pic\_height\_in\_luma\_samples、またはsps\_max\_dec\_pic\_buffering\_minus1[HighestTid]の値と異なる場合、NoOutputOfPriorPicsFlagは、no\_output\_of\_prior\_pics\_flagの値にかかわらず、テスト中のデコーダによって1に設定され得る(が、設定されるべきでない)。

注 - これらの条件下では、NoOutputOfPriorPicsFlagをno\_output\_of\_prior\_pics\_flagに等しく設定することが好ましいが、テスト中のデコーダは、この場合にはNoOutputOfPriorPicsFlagを1に設定することが可能にされる。

- そうでない場合、現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャである場合、NoOutputOfPriorPicsFlagは、no\_output\_of\_prior\_pics\_flagに等しく設定される。

- そうでない(現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャではなく、現在のアクセスユニット中のベースレイヤピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャであり、NoClrasOutputFlagが1に等しい)場合、NoOutputOfPriorPicsFlagは1に等しく設定される。

2. テスト中のデコーダのために導出されたNoOutputOfPriorPicsFlagの値は、次のようにHRDに対して適用される。

- NoOutputOfPriorPicsFlagが1に等しい場合、サブDPB内のすべてのピクチャ記憶バッファは、それらが包含しているピクチャを出力せずに空にされ、サブDPBフルネスは0に等しく設定される。

- そうでない(NoOutputOfPriorPicsFlagが0に等しい)場合、「出力のために必要とされない」および「参照のために使用されない」とマークされているピクチャを包含するすべてのピクチャ記憶バッファは(出力せずに)空にされ、サブDPB内のすべての空でないピクチャ記憶バッファは、従属節C.5.2.4において指定されている「バンピング」プロセスを繰り返し呼び出すことによって空にされ、サブDPBフルネスは0に等しく設定される。

- そうでない場合、現在のレイヤにおけるピクチャを包含しており、「出力のために必要とされない」および「参照のために使用されない」とマークされている、すべてのピクチャ記憶バッファは(出力せずに)空にされる。空にされる各ピクチャ記憶バッファに対して、サブDPBフルネスは1だけデクリメントされる。次の条件のうちの1つまたは複数が真であるとき[削除:空にされる各追加のピクチャ記憶バッファに対してサブDPBフルネスを1だけさらにデクリメントしながら]、次の条件のいずれも真でなくなるまで、従属節C.5.2.4において指定されている「バンピング」プロセスが繰り返し呼び出される。

- 「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号

10

20

30

40

50

されたピクチャを包含するアクセスユニットの数は、`MaxNumReorderPics`よりも大きい。

- `MaxLatencyIncreasePlus1`は0に等しくなく、それについて関連付けられた変数`PicLatencyCount`が`MaxLatencyPictures`以上である、「出力のために必要とされる」とマークされた、`DPB`内の少なくとも1つの復号されたピクチャを包含する、少なくとも1つのアクセスユニットがある。

- サブ`DPB`内の現在のレイヤにおけるピクチャの数は、`MaxDecPicBufferingMinus1+1`以上である。

C.5.2.3 ピクチャ復号、マーキング、追加のバンピング、および記憶

この従属節において指定されているプロセスは、ピクチャ`n`の最後の復号ユニットが`CPB`から除去されるときに瞬時に起こる。[`Ed.(MH)`:バージョン1デコーダが、ベースレイヤピクチャの`CPB`除去時間よりも遅くなり得る、`AU`の`CPB`除去時間において、ベースレイヤピクチャにマークし、ベースレイヤピクチャを記憶することになるので、この変更はバージョン1に準拠しないことがある。]

`PicOutputFlag`は次のように更新される。

- 現在のアクセスユニットが、ターゲット出力レイヤにおいてピクチャを包含しておらず、`alt_output_layer_flag`が1に等しい場合、次の順序付けられたステップが適用される。

- リスト`nonOutputLayerPictures`は、1に等しい`PicOutputFlag`を有し、ならびに、`TargetDecLayerIdList`中に含まれており、ターゲット出力レイヤ上にない`nuh_layer_id`値を有する、アクセスユニットのピクチャのリストである。

- リスト`nonOutputLayerPictures`の中で最高の`nuh_layer_id`値を有するピクチャは、リスト`nonOutputLayerPictures`から除去される。

- リスト`nonOutputLayerPictures`中に含まれる各ピクチャのための`PicOutputFlag`は、0に等しく設定される。

- そうでない場合、ターゲット出力レイヤ中に含まれないピクチャのための`PicOutputFlag`は、0に等しく設定される。

現在のピクチャが、1に等しい`PicOutputFlag`を有するとき、「出力のために必要とされる」とマークされており、出力順序において現在のピクチャに後続する、サブ`DPB`内の現在のレイヤにおける各ピクチャについて、関連付けられた変数`PicLatencyCount`は、`PicLatencyCount+1`に等しく設定される。

現在のピクチャは、ピクチャの最後の復号ユニットが復号された後に復号されると見なされる。現在の復号されたピクチャは、サブ`DPB`内の空のピクチャ記憶バッファ内に記憶され、次のことが適用される。

- 現在の復号されたピクチャが、1に等しい`PicOutputFlag`を有する場合、それは「出力のために必要とされる」とマークされ、その関連付けられた変数`PicLatencyCount`は、0に等しく設定される。

- そうでない(現在の復号されたピクチャが、0に等しい`PicOutputFlag`を有する)場合、それは「出力のために必要とされない」とマークされる。

現在の復号されたピクチャは、「短期参照のために使用される」とマークされる。

次の条件のうちの1つまたは複数が真であるとき、従属節C.5.2.4において指定されている「バンピング」プロセスが、次の条件のいずれも真でなくなるまで繰り返し呼び出される。

- 「出力のために必要とされる」とマークされた、`DPB`内の少なくとも1つの復号されたピクチャを包含するアクセスユニットの数は、`MaxNumReorderPics`よりも大きい。

- `MaxLatencyIncreasePlus1`は0に等しくなく、それについて関連付けられた変数`PicLatencyCount`が`MaxLatencyPictu`

10

20

30

40

50

res 以上である、「出力のために必要とされる」とマークされた、DPB内の少なくとも1つの復号されたピクチャを包含する、少なくとも1つのアクセスユニットがある。

#### C. 5. 2. 4 「バンピング」プロセス

「バンピング」プロセスは以下の順序付きステップからなる。

1. 出力のための最初のものである1つまたは複数のピクチャが、「出力のために必要とされる」とマークされたDPB内のすべてのピクチャのうちのPicOrderCntValの最小値を有するピクチャとして選択される。

2. これらのピクチャの各々が、nuh\_layer\_idの昇順で、ピクチャのためのアクティブSPS中で指定された適合クロッピングウィンドウを使用してクロップされ、クロップされたピクチャが出力され、そのピクチャが「出力のために必要とされない」と

10

マークされる。「参照のために使用されない」とマークされ、クロップおよび出力されたピクチャのうち

の1つであったピクチャを包含する、各ピクチャ記憶バッファが空にされ、ital<<関連付けられたサブDPBのフルネスが1だけデクリメントされる>>ital。

#### 【0089】

[0095]したがって、上記の例では、および本開示の態様によれば、ビデオエンコーダ20およびビデオデコーダ30は、マルチレイヤビデオデータの第1の受信レイヤのピクチャのPOC値のためのPOCリセットを示すデータ(たとえば、ital<<poc\_reset\_id>>italシンタックス要素)に基づいて、POC値リセットより前に、(アクセスユニットごとではなく)一緒にDPBのピクチャを出力するように構成され得る。POCリセットより前に、このようにしてDPBのすべてのピクチャを出力することは、(同じアクセスユニット中であり、同じPOC値を有する)同じ時間的インスタンスのピクチャが同じまたは実質的に同じ時間に出力されることを保証する助けになり得る。

20

#### 【0090】

[0096]上述されたように、レイヤ固有のPOCリセット期間は、スライスセグメントヘッダ中でシグナリングされたPOCリセット期間識別子に基づいて指定され得る。すなわち、ビデオエンコーダ20およびビデオデコーダ30は、スライスセグメントヘッダ中でPOCリセット期間識別子を表すデータをそれぞれコーディングし得る。少なくとも1つのIRAPピクチャを包含するアクセスユニットに属する各非IRAPピクチャは、非IRAPピクチャを包含するレイヤにおいてPOCリセット期間の開始であり得る。すなわち、ビデオエンコーダ20は、非IRAPピクチャが新しいPOCリセット期間の開始であることを示すために、少なくとも1つのIRAPピクチャを包含するアクセスユニットの非IRAPピクチャのためのPOCリセットタイプを設定し得る。そのようなアクセスユニットにおいて、各ピクチャは、そのピクチャを包含するレイヤにおいてPOCリセット期間の開始となる。POC\_MSBのみ、またはPOC\_MSBとPOC\_LSBの両方のいずれかである、POCリセット、および、DPB内の同じレイヤのピクチャのPOC値の更新は、各POCリセット期間内の最初のピクチャのみに対して適用され得る。

30

#### 【0091】

[0097]ビデオエンコーダ20は、ビデオデコーダ30がスライスセグメントヘッダを有するスライスを含むレイヤにおけるピクチャのPOC値を導出するために使用し得る、スライスセグメントヘッダ中のPOC\_LSB値をシグナリングし得る。スライスセグメントヘッダ中でシグナリングされたPOC\_LSB値は、DPB内の同じレイヤのピクチャのPOC値を更新するために使用される、デルタPOC値の導出のため、および、現在のピクチャのPOC値のPOC\_MSBの導出のためにも使用され得る。現在のピクチャが、シグナリングされたそのようなPOC\_LSB値を有するとき、および、現在のピクチャに関連付けられたPOCアンカーピクチャがビットストリーム中に存在するとき、POCアンカーピクチャは、POCリセットまたはPOC\_MSBリセットのいずれかの指示を有する。現在のピクチャのためにシグナリングされたそのようなPOC\_LSB値は、現在のピクチャとしてのPOCリセット期間の最初のピクチャでもある、POCアンカーピクチャのPOC\_LSB値に等しい。

40

50

## 【 0 0 9 2 】

[0098] 上述されたように、ビデオエンコーダ 20 およびビデオデコーダ 30 は、コーディング順序において現在のピクチャより前に発生するピクチャの POC MSB 値に基づいて、POC MSB 値を導出し得る。たとえば、ビデオデコーダ 30 は、現在のピクチャの NAL ユニットタイプと、NAL ユニットタイプランダムアクセススキップリーディング (RASL: random access skipped leading) もしくはランダムアクセス復号可能リーディング (RADL: random access decodable leading) のものではないか、またはサブレイヤ非参照ピクチャではなく、0 に等しい temporal\_id 値を有する、復号順序における前のピクチャの MSB および LSB とに基づいて、現在のピクチャ (現在復号されているピクチャ) のための MSB POC 値を決定し得る。現在のピクチャが、1 に等しい NoRaslOutputFlag を有する IRAP ピクチャであるか、またはビットストリームの最初のピクチャである CRA ピクチャであるとき、POC MSB の値は 0 に等しいと推論され (すなわち、ビデオデコーダ 30 によって決定され) 得る。

10

## 【 0 0 9 3 】

[0099] 本開示の態様によれば、ビデオエンコーダ 20 およびビデオデコーダ 30 は、コーディングされているピクチャのピクチャタイプを決定することと、ピクチャタイプに基づいて、コーディングされているピクチャのための POC MSB 値を示すデータをコーディング (符号化または復号) することとを行うように構成され得る。より詳細には、ビデオコーダは、ピクチャタイプに基づいて、POC MSB 値を示すデータを選択的にコーディングし得る。たとえば、概して、そのための POC MSB 値がコーディングされるピクチャタイプのセットと、そのための POC MSB 値がコーディングされないピクチャタイプの別のセットとがあり得る。ビデオコーダは、ピクチャのためのピクチャタイプが、そのための POC MSB 値がコーディングされるピクチャタイプのセット中に含まれるときのみ、そのピクチャのための POC MSB 値をコーディングし得る。このようにして、ビデオエンコーダ 20 は、マルチレイヤビットストリーム中の別のピクチャの POC MSB を参照することなく、コーディングされているピクチャのための POC MSB 値を示し得、ビデオデコーダ 30 はそれを決定し得る。このようにして POC MSB を示し、決定することは、(たとえば、マルチレイヤビットストリームのアップスイッチングまたはダウンスイッチングとともに発生し得るように) 比較的多数の POC 値が、コーディング順序において復号されているピクチャに先行するピクチャから、コーディングされているピクチャを分離する場合に、エラーレジリエンスを改善し得る。現在のピクチャのピクチャタイプが、そのための POC MSB 値がコーディングされないピクチャタイプのセット中に含まれる場合、ビデオコーダは、現在のピクチャのための POC MSB 値を決定するために、前に復号されたピクチャの POC MSB 値を使用し得る。

20

30

## 【 0 0 9 4 】

[0100] 同じく、次の開示および表は、本開示の技法による例示的なシンタックスを含む。このシンタックスは、たとえば、上述された文書 JCTVC - 00217 および JCTVC - 00266 によって修正されるような MV - HEVC のシンタックスに関して修正され得る。以下の例示的な説明、シンタックステーブルおよびセマンティクスでは、(文書 JCTVC - 00217 および JCTVC - 00266 によって修正されるような) MV - HEVC に対する追加が、イタリック体を使用して表され、削除が、「除去」が前に付いた括弧付きのテキスト (たとえば、[ 除去: “ 除去されたテキスト ” ]) を使用して表される。概して、「要件」に関するステートメントは、規格または規格拡張のテキストの一部を形成するものであり、本開示の技法の目的のための要件を形成しないものと理解されたい。場合によっては、そのような「要件」は、適用可能であると決定され得、次いで、たとえば、その決定に基づいてビデオコーダによって遵守され得る、ビットストリーム制約を含み得る)。

40

## 【 0 0 9 5 】

[0101] 次の数学関数の定義が、MV - HEVC に、たとえば、MV - HEVC の従属節

50

5.8に追加され得る。

【数1】

$$GetCurrMsb(cl, pl, pm, ml) = \begin{cases} pm + ml & ; \quad pl - cl \geq ml / 2 \\ pm - ml & ; \quad cl - pl > ml / 2 \\ pm & ; \quad \text{他の場合} \end{cases}$$

【0096】

[0102] 次の追加のシンタックス関数が追加される。

10

`ital<<more_data_in_slice_segment_header_extension()` は、次のように指定される>>`ital`。

- `ital<<(slice_segment_header())` シンタックス構造における現在の位置) - (`slice_segment_header_extension_length`の直後の位置) が、(`slice_segment_header_extension_length * 8`) 未満である場合、`more_data_in_slice_segment_header_extension()` の戻り値はTRUEに等しい>>`ital`。

- `ital<<そうでない場合、more_data_in_slice_segment_header_extension()` の戻り値はFALSEに等しい>>`ital`。

20

【0097】

[0103] 次の定義が、従属節F.3に追加され得る。

F.3.1 ピクチャ順序カウント(POC)リセット期間: 0に等しい`poc_reset_idc`を有するアクセスユニットでインターリーブされてもされなくてもよい、0よりも大きい`poc_reset_idc`と、`poc_reset_period_id`の同じ値とをすべてが有する、復号順序における`ital<<アクセスユニット>>ital`のシーケンス。

【0098】

[0104] 1つの代替として、POCリセット期間におけるピクチャは、レイヤ内で復号順序において連続であることが必要とされる。しかしながら、これらのピクチャが、同じレイヤにおいて0に等しい`poc_reset_idc`を有するピクチャでインターリーブされることを可能にすることは、より柔軟であり、クラスタ化されたパケット損失に対するより良いエラーレジリエンスを達成するために、同じオーバーヘッドを使用することを可能にする。

30

【0099】

[0105] 次の表は、本開示の技法による、PPSのためのシンタックスの一例である。



【表 1】

pic_parameter_set_rbsp( ) {	記述子	
...		
<b>pps_extension_flag</b>	u(1)	
if( pps_extension_flag ) {		
<b>poc_reset_info_present_flag</b>	u(1)	10
<b>pps_extension2_flag</b>	u(1)	
if( pps_extension2_flag )		
while( more_rbsp_data( ) )		
<b>pps_extension_data_flag</b>	u(1)	
}		
rbsp_trailing_bits( )		
}		20

## 【0100】

[0106]上の例では、0に等しいpps\_extension\_flagは、pps\_extension\_data\_flagシンタックス要素がPPS RBSPシンタックス構造中に存在しないことを指定する。slice\_segment\_header\_extension\_present\_flagが0に等しいとき、pps\_extension\_flagは、本仕様の本バージョンに適合するビットストリーム中で0に等しいものとし、pps\_extension\_flagのための1の値は、ITU-T|ISO/IECによる将来の使用のために確保され、デコーダは、PPS NALユニット中のpps\_extension\_flagのための値1に後続するすべてのデータを無視するものとする。

30

## 【0101】

[0107]加えて、0に等しいpoc\_reset\_info\_present\_flagは、シンタックス要素poc\_reset\_idcが、PPSを参照するスライスのスライスセグメントヘッダ中に存在しないことを指定する。加えて、1に等しいpoc\_reset\_info\_present\_flagは、シンタックス要素poc\_reset\_idcが、PPSを参照するスライスのスライスセグメントヘッダ中に存在することを指定する。

40

## 【0102】

[0108]加えて、0に等しいpps\_extension2\_flagは、pps\_extension\_data\_flagシンタックス要素が、PPS RBSPシンタックス構造中に存在しないことを指定する。pps\_extension2\_flagは、本仕様の本バージョンに適合するビットストリーム中で0に等しいものとする。pps\_extension2\_flagのための1の値は、ITU-T|ISO/IECによる将来の使用のために確保される。デコーダは、PPS NALユニット中のpps\_extension2\_flagのための値1に後続するすべてのデータを無視するものとする。

## 【0103】

50

[0109] 次の表は、本開示の技法による、スライスヘッダのためのシンタックスの一例である。

【表 2】

	記述子	
slice_segment_header() {		
...		
if( !dependent_slice_segment_flag ) {		
i = 0		
if( num_extra_slice_header_bits > i ) {		10
i++		
discardable_flag	<b>u(1)</b>	
}		
if( num_extra_slice_header_bits > i ) {		
i++		
cross_layer_bla_flag	<b>u(1)</b>	
}		
if( num_extra_slice_header_bits > i ) {		20
i++		
poc_reset_flag	<b>u(1)</b>	
}		
...		
if( slice_segment_header_extension_present_flag ) {		
slice_segment_header_extension_length	<b>ue(v)</b>	
if( poc_reset_info_present_flag )		
poc_reset_idc	<b>u(2)</b>	
if( poc_reset_idc > 0 )		30
poc_reset_period_id	<b>u(6)</b>	
if( poc_reset_idc == 3 ) {		
full_poc_reset_flag	<b>u(1)</b>	
poc_lsb_val	<b>u(v)</b>	
}		
if( nal_unit_type == BLA_W_LP    nal_unit_type == BLA_N_LP    nal_unit_type == BLA_W_RADL    nal_unit_type == CRA_NUT )		40
poc_msb_val	<b>ue(v)</b>	
while( more_data_in_slice_segment_header_extension() )		
slice_segment_header_extension_data_bit	<b>u(1)</b>	
}		
byte_alignment()		
}		

【 0 1 0 4 】

[0110] 代替的に、poc\_reset\_period\_id シンタックス要素は、たと 50

えば、 $u(14)$ としてコーディングされた、異なるビット数を使用してシグナリングされ得る。

【0105】

[0111]存在するとき、スライスセグメントヘッダシンタックス要素 `slice__pic__parameter__set__id`、`pic__output__flag`、`no__output__of__prior__pics__flag`、`slice__pic__order__cnt__lsb`、`short__term__ref__pic__set__sps__flag`、`short__term__ref__pic__set__idx`、`num__long__term__sps`、`num__long__term__pics`、`slice__temporal__mvp__enabled__flag`、`discardable__flag`、`cross__layer__bla__flag`、`inter__layer__pred__enabled__flag`、`num__inter__layer__ref__pics__minus1`、`poc__reset__idc`、`poc__reset__pic__id`、`full__poc__reset__flag`、および `poc__lsb__val`、および `ital<<poc__msb__val>>ital`の値は、コード化されたピクチャのすべてのスライスセグメントヘッダ中で同じものとする。存在するとき、スライスセグメントヘッダシンタックス要素 `lt__idx__sps[i]`、`poc__lsb__lt[i]`、`used__by__curr__pic__lt__flag[i]`、`delta__poc__msb__present__flag[i]`、`delta__poc__msb__cycle__lt[i]`、および `inter__layer__pred__layer__idc[i]`の値は、 $i$ の可能な値ごとに、コード化されたピクチャのすべてのスライスセグメントヘッダ中で同じものとする。

【0106】

[0112]0に等しいシンタックス要素 `poc__reset__idc`は、現在のピクチャのためのピクチャ順序カウント値の最上位ビットまたは最下位ビットのいずれもリセットされないことを指定する。1に等しいシンタックス要素 `poc__reset__idc`は、現在のピクチャのためのピクチャ順序カウント値の最上位ビットのみがリセットされ得ることを指定する。2に等しいシンタックス要素 `poc__reset__idc`は、現在のピクチャのためのピクチャ順序カウント値の最上位ビットと最下位ビットの両方がリセットされ得ることを指定する。3に等しいシンタックス要素 `poc__reset__idc`は、現在のピクチャのためのピクチャ順序カウント値の最上位ビットのみ、または、最上位ビットと最下位ビットの両方のいずれかがリセットされ得、追加のピクチャ順序カウント情報がシグナリングされることを指定する。存在しないとき、シンタックス要素 `poc__reset__poc`の値は、0に等しいと推論される（すなわち、明示的なシグナリングなしに、コードによって決定される）。

【0107】

[0113]次の制約が適用されることが、ビットストリーム適合の要件である。

- `poc__reset__idc`の値は、RASLピクチャ、RADLピクチャ、サブレイヤ非参照ピクチャ、または0よりも大きいTemporalId、`ital<<または1に等しいdiscardable__flag>>ital`を有するピクチャについて、1または2に等しくないものとする。

- アクセスユニット中のすべてのピクチャの `poc__reset__idc`の値は、同じものとする。

- アクセスユニット中の1つのピクチャが、`nal__unit__type`の特定の値を有するIRAPピクチャであり、`nal__unit__type`の異なる値を有する、同じアクセスユニット中の少なくとも1つの他のピクチャがあるとき、`poc__reset__idc`の値は、アクセスユニット中のすべてのピクチャについて1または2に等しいものとする。

- アクセスユニット中の0に等しい`nuh__layer__id`を有するピクチャが、IDRピクチャであり、同じアクセスユニット中の少なくとも1つの非IDRピクチャがあるとき、`poc__reset__idc`の値は、アクセスユニット中のすべてのピクチャ

10

20

30

40

50

について2に等しいものとする。

- アクセスユニット中の0に等しい`nuh_layer_id`を有するピクチャが、IDRピクチャではないとき、`poc_reset_idc`の値は、アクセスユニット中の任意のピクチャについて2に等しくないものとする。

【0108】

[0114]アクセスユニットの`poc_reset_idc`の値は、アクセスユニット中のピクチャの`poc_reset_idc`の値を指す。

【0109】

[0115]シンタックス要素`poc_reset_period_id`は、POCリセット期間を識別する。シンタックス要素`poc_reset_period_id`の同じ値と、1または2に等しいシンタックス要素`poc_reset_idc`とを有する、同じレイヤにおける復号順序で連続する2つのピクチャがないものとする。

10

注 - あるレイヤにおける複数のピクチャが、`poc_reset_pic_id`の同じ値を有することと、1または2に等しい`poc_reset_idc`を有することとは、そのようなピクチャが復号順序で連続する2つのアクセスユニットにおいて発生しない限り、禁止されない。ピクチャ損失、ビットストリーム抽出、シーク、またはスプライシング動作のために、そのような2つのピクチャがビットストリーム中に現れる見込みを最小限に抑えるために、エンコーダは、(上記で指定された制約を受けて)`poc_reset_pic_id`の値を、POCリセット期間ごとにランダム値になるように設定すべきである。

20

【0110】

[0116]次の制約が適用されることが、ビットストリーム適合の要件である。

- 1つのPOCリセット期間は、1または2に等しい`poc_reset_idc`を有する2つ以上のアクセスユニットを含まないものとする。

- 1または2に等しい`poc_reset_idc`を有するアクセスユニットは、POCリセット期間における最初のアクセスユニットであるものとする。

【0111】

[0117]1に等しいシンタックス要素`full_poc_reset_flag`は、同じレイヤにおける復号順序において前のピクチャが同じPOCリセット期間に属していないとき、現在のピクチャのためのピクチャ順序カウンタ値の最上位ビットと最下位ビットの両方がリセットされることを指定する。0に等しいシンタックス要素`full_poc_reset_flag`は、同じレイヤにおける復号順序において前のピクチャが同じPOCリセット期間に属していないとき、現在のピクチャのためのピクチャ順序カウンタ値の最上位ビットのみがリセットされることを指定する。

30

【0112】

[0118]シンタックス要素`poc_lsb_val`は、現在のピクチャのピクチャ順序カウンタを導出するために使用され得る値を指定する。`poc_lsb_val`シンタックス要素の長さは、 $\log_2 \text{max\_pic\_order\_cnt} \text{ lsb} \text{ minus } 4 + 4$ ビットである。

【0113】

40

[0119]`poc_reset_idc`が3に等しく、また、現在のピクチャと同じレイヤ内である、1または2に等しい`poc_reset_idc`を有する、および同じPOCリセット期間に属する、復号順序における前のピクチャ`picA`がビットストリーム中に存在するとき、`picA`は、現在のピクチャと同じレイヤ内である、RASLピクチャ、RADLピクチャまたはサブレイヤ非参照ピクチャではない、ならびに0に等しい`TemporalId`、`ital<<および0に等しいdiscardable_flag>>ital`を有する、復号順序における前のピクチャと同じピクチャであるものとする、ならびに、現在のピクチャの`poc_lsb_val`の値は、`picA`の`slice_pic_order_cnt_lsb`の値に等しいものとする、ことが、ビットストリーム適合の要件である。

50

## 【0114】

[0120] シンタックス要素 `ital<<poc__msb__val` は、現在のピクチャが、0 に等しい `NoRaslOutputFlag` を有する CRA ピクチャであるとき、ピクチャ順序カウンタ値を導出するために使用される値を指定する。 `poc__msb__val` の値は、両端値を含む  $0 \sim 2^{32} - \log_2 \text{max\_pic\_order\_cnt\_lsb} - \text{minus} 4 - 4$  の範囲内にあるものとする。>>ital。

## 【0115】

[0121] ここで、本開示の態様によれば、ビデオデコーダ 30 は、現在復号されているピクチャのピクチャタイプを決定し得る。(たとえば、上記の例示的なスライスセグメントヘッダの表における「`ifnal__unit__type...`」ステートメントにおいて記載されたように) ピクチャのNALユニットタイプがCRAまたはBLAである場合、ビデオデコーダ 30 は、現在復号されているピクチャのPOC MSB値を示すデータ(たとえば、`poc__msb__val` シンタックス要素)を復号し得る。この例では、特定のピクチャタイプ(BLAおよびCRA)とシンタックス要素(`poc__msb__val`)が参照されるが、本技法はこのように限定されないことを理解されたい。すなわち、ビデオデコーダ 30 は、現在復号されているピクチャタイプが様々な他のピクチャタイプであることに基づいて、POC MSB値を示す様々な他のシンタックス要素を復号し得る。

## 【0116】

[0122] ビデオデコーダ 30 は、0 よりも大きい `nuh__layer__id` を有するコード化されたピクチャの復号を開始するための、次の復号プロセスを実行し得る。以下のプロセスにおいて参照される各「ピクチャ」は、完全コード化されたピクチャである。復号プロセスは、現在のピクチャ `CurrPic` について次のように動作する。

- 本明細書で説明されるように、NALユニットを復号する。
- 従属節 F. 8. 3 におけるプロセスは、スライスセグメントレイヤおよびそれ以上におけるシンタックス要素を使用した次の復号プロセスを指定する。
  - ピクチャ順序カウンタに関係する変数および関数は、従属節 F. 8. 3. 1 において導出される。これは、ピクチャの最初のスライスセグメントについてのみ呼び出される必要がある。 `PicOrderCntVal` がアクセスユニット内で不変のままであるものとするのが、ビットストリーム適合の要件である。

- 従属節 F. 8. 3. 2 におけるRPSのための復号プロセスが呼び出され、ここにおいて、`CurrPic` の `nuh__layer__id` に等しい `nuh__layer__id` を有する参照ピクチャのみが、「参照のために使用されない」または「長期参照のために使用される」とマークされ得、`nuh__layer__id` の異なる値を有するいかなるピクチャもマークされない。これは、ピクチャの最初のスライスセグメントについてのみ呼び出される必要がある。

- `FirstPicInLayerDecodedFlag[nuh__layer__id]` が0に等しいとき、従属節 F. 8. 1. 3 において指定されている利用不可能な参照ピクチャを生成するための復号プロセスが呼び出され、これは、ピクチャの最初のスライスセグメントについてのみ呼び出される必要がある。

- `FirstPicInLayerDecodedFlag[nuh__layer__id]` が0に等しくなく、現在のピクチャが、1に等しい `NoRaslOutputFlag` を有するIRAPピクチャであるとき、従属節 F. 8. 1. 3 において指定されている利用不可能な参照ピクチャを生成するための復号プロセスが呼び出され、これは、ピクチャの最初のスライスセグメントについてのみ呼び出される必要がある。

## 【0117】

[0123] 本開示の態様によれば、ビデオデコーダ 30 は、POC値のための次の復号プロセスを実行し得る。このプロセスの出力は、`PicOrderCntVal`、現在のピクチャのピクチャ順序カウンタである。

10

20

30

40

50

ピクチャ順序カウンタは、マージモードにおける動きパラメータと動きベクトル予測とを導出するために、およびデコード適合チェック（従属節 C.5 参照）のために、ピクチャを識別するために使用される。

各コード化されたピクチャは、PicOrderCntValとして示されるピクチャ順序カウンタ変数に関連付けられる。

FirstPicInLayerDecodedFlag[nuh\_layer\_id]が1に等しく、poc\_reset\_idcが0よりも大きく、現在のピクチャがital<<現在のレイヤのための>>ital、POCリセット期間において復号順序でital<<最初のピクチャ>>italである場合、次のことが適用される。

- 変数pocMsbDelta、pocLsbDelta、およびDeltaPocValは、次のように導出される。

【数2】

```

if( poc_reset_idc == 3 )
    pocLsbVal = poc_lsb_val
else
    pocLsbVal = slice_pic_order_cnt_lsb
if( 現在のピクチャが、0に等しいNoRaslOutputFlagを有するCRAピクチャである )
    pocMsbDelta = poc_msb_val * MaxPicOrderCntLsb
else {
    prevPicOrderCntLsb =
    PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
    prevPicOrderCntMsb =
    PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
    pocMsbDelta = getCurrMsb( pocLsbVal, prevPicOrderCntLsb,
    prevPicOrderCntMsb, MaxPicOrderCntLsb )
}
if( poc_reset_idc == 2 || ( poc_reset_idc == 3 && full_poc_reset_flag ) )
    pocLsbDelta = pocLsbVal
else
    pocLsbDelta = 0
DeltaPocVal = pocMsbDelta + pocLsbDelta

```

- ita<<DPB内にあり、現在のピクチャと同じレイヤに属する各ピクチャのPicOrderCntValは、DeltaPocValだけデクリメントされる>>ital。
- 現在のピクチャのPicOrderCntValは、次のように導出される。

## 【数3】

```

if( poc_reset_idc == 1 )
    PicOrderCntVal = slice_pic_order_cnt_lsb
else if( poc_reset_idc == 2 )
    PicOrderCntVal = 0
else { // poc_reset_idc == 3
    PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb, poc_lsb_val, 0,
    MaxPicOrderCntLsb )
    PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
}

```

10

- PrevPicOrderCnt[nuh\_layer\_id]の値は、次のように導出される。

- 現在のピクチャが、RASLピクチャ、RADLピクチャ、またはサブレイヤ非参照ピクチャではなく、現在のピクチャが、0に等しいTemporalId、ital<<および0に等しいdiscardable\_flag>>italを有する場合、PrevPicOrderCnt[nuh\_layer\_id]は、PicOrderCntValに等しく設定される。

20

- そうでない場合、poc\_reset\_idcが3に等しいとき、PrevPicOrderCnt[nuh\_layer\_id]は、full\_poc\_reset\_flag?0:poc\_lsb\_valに等しく設定される。

そうでない場合、次のことが適用される。

- 現在のピクチャのPicOrderCntValは、次のように導出される。

## 【数4】

```

if( !FirstPicInLayerDecodedFlag[ nuh_layer_id ] ) {
  if( poc_reset_idc == 1 )
    PicOrderCntVal = slice_pic_order_cnt_lsb
  else if( poc_reset_idc == 2 )
    PicOrderCntVal = 0
  else if( poc_reset_idc == 3 ) {
    PicOrderCntMsb = getCurrMsb(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb )
    PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
  } else // 現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャである
    PicOrderCntVal = slice_pic_order_cnt_lsb
} else { // HEVCバージョン1と同様のPOC導出
  if( 現在のピクチャが、1に等しいNoRaslOutputFlagを有するIRAPピクチャである )
    PicOrderCntMsb = 0
  else if( 現在のピクチャが、CRAピクチャである )
    PicOrderCntMsb = poc_msb_val * MaxPicOrderCntLsb
  else {
    prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
    prevPicOrderCntMsb =
PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
    PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb,
prevPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb )
  }
  PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
}

```

40  
- PrevPicOrderCnt[ nuh\_layer\_id ] の値は、次のように導出される。

- 現在のピクチャが、RASLピクチャ、RADLピクチャ、またはサブレイヤ非参照ピクチャではなく、現在のピクチャが、0に等しいTemporalId、ital<<および0に等しいdiscardable\_flag>>italを有する場合、PrevPicOrderCnt[ nuh\_layer\_id ] は、PicOrderCntValに等しく設定される。

そうでない場合、FirstPicInLayerDecodedFlag[ nuh\_layer\_id ] が0に等しく、poc\_reset\_idcが3に等しいとき、PrevPicOrderCnt[ nuh\_layer\_id ] は、full\_poc\_ 50



`reset_flag?0:poc_lsb_val`に等しく設定される。

`PicOrderCntVal`の値は、両端値を含む  $-2^{31} \sim 2^{31} - 1$  の範囲内にあるものとする。1つのCVSの中で、同じレイヤにおける任意の2つのコード化されたピクチャのための `PicOrderCntVal` 値は、同じではないものとする。

関数 `PicOrderCnt(picX)` は、次のように指定される。

【数5】

$$\text{PicOrderCnt}(\text{picX}) = \text{ピクチャpicXの PicOrderCntVal} \quad (\text{F-25})$$

関数 `DiffPicOrderCnt(picA, picB)` は、次のように指定される。

【数6】

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB}) \quad (\text{F-26})$$

ビットストリームは、両端値を含む  $-2^{15} \sim 2^{15} - 1$  の範囲内にはない、復号プロセスにおいて使用される `DiffPicOrderCnt(picA, picB)` の値をもたらすデータを含むものとする。

注 - Xを現在のピクチャとし、YおよびZを同じシーケンス中の2つの他のピクチャとすると、YおよびZは、`DiffPicOrderCnt(X, Y)` と `DiffPicOrderCnt(X, Z)` の両方が正であるか、または両方が負であるとき、Xからの同じ出力順序方向であると見なされる。

【0118】

[0124]このようにして、ビデオエンコーダ20およびビデオデコーダ30は、マルチレイヤビデオデータの第1のピクチャをコーディングすることと、その場合、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのための `POC_MSB` 値を表すデータをコーディングすることと、を行うように構成された、ビデオコードの例を表す。ビデオエンコーダ20およびビデオデコーダ30はまた、本開示の態様によれば、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャの `POC` 値のための `POC` リセットを示すデータをコーディングすることと、その場合、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャの `POC` 値のための `POC` リセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、`DPB` 内に記憶されたすべてのピクチャを出力することと、を行うように構成され得る。

【0119】

[0125]図2は、本開示の技法による、`POC` 値を管理するための技法を実装し得る、ビデオエンコーダ20の一例を示すブロック図である。特に、ビデオエンコーダ20は、本開示の技法に従って、マルチレイヤビットストリームのピクチャの `POC` 値を管理するように構成され得る。上述されたように、ビデオエンコーダ20は、マルチビューおよび/またはスケーラブルビデオコーディングを実行するように適応され得る。たとえば、ビデオエンコーダ20は、`SHVC`、`MV-HEVC`、または `3D-HEVC` などの1つまたは複数のビデオコーディング規格の拡張に適合するビットストリームを符号化するように構成され得る。ただし、特定のコーディング規格が参照されるが、本技法は、任意の1つのコーディング規格に固有のものではなく、将来のおよび/またはまだ開発されていない規格とともに実装され得ることを理解されたい。

【0120】

[0126]ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターコーディングとを実行し得る。イントラコーディングは、所与のビデオフレームまたはピクチャ内のビデオ中の空間的冗長性を低減または除去するために、空間

10

20

30

40

50

的予測に依拠する。インターコーディングは、ビデオシーケンスの隣接するフレームまたはピクチャ内のビデオ中の時間的冗長性を低減または除去するために、時間的予測に依拠する。イントラモード（Iモード）は、いくつかの空間ベースのコーディングモードのいずれかを指し得る。単一方向予測（Pモード）または双予測（Bモード）などのインターモードは、いくつかの時間ベースのコーディングモードのいずれかを指し得る。

#### 【0121】

[0127]図2に示されているように、ビデオエンコーダ20は、符号化されるべきビデオフレーム内で現在のビデオブロックを受信する。図2の例では、ビデオエンコーダ20は、ビデオデータメモリ38と、モード選択ユニット40と、参照ピクチャメモリ64と、加算器50と、変換処理ユニット52と、量子化ユニット54と、エントロピー符号化ユニット56とを含む。モード選択ユニット40は、動き補償ユニット44と、動き推定ユニット42と、イントラ予測ユニット46と、区分ユニット48とを含む。ビデオブロックの再構築のために、ビデオエンコーダ20は、また、逆量子化ユニット58と、逆変換ユニット60と、加算器62とを含む。デブロッキングフィルタ（図2に図示せず）も、再構築されたビデオからブロックネスアーティファクトを除去するために、ブロック境界をフィルタリングするために含まれ得る。所望される場合、デブロッキングフィルタは、通常、加算器62の出力をフィルタリングすることになる。追加のフィルタ（ループ内またはループ後）も、デブロッキングフィルタに加えて使用され得る。そのようなフィルタは、簡潔のために示されていないが、所望される場合、（ループ内フィルタとして）加算器50の出力をフィルタリングし得る。

#### 【0122】

[0128]符号化プロセス中に、ビデオエンコーダ20は、コーディングされるべきビデオフレームまたはスライスを受信する。フレームまたはスライスは、複数のビデオブロックに分割され得る。ビデオデータメモリ38は、ビデオエンコーダ20の構成要素によって符号化されるべきビデオデータを記憶し得る。ビデオデータメモリ38内に記憶されるビデオデータは、たとえば、ビデオソース18から取得され得る。参照ピクチャメモリ64は、たとえば、イントラコーディングモードまたはインターコーディングモードで、ビデオエンコーダ20によってビデオデータを符号化する際に使用するための参照ビデオデータを記憶する、DPBと呼ばれることがある。ビデオデータメモリ38および参照ピクチャメモリ64は、同期DRAM（SDRAM）、磁気抵抗RAM（MRAM）、抵抗RAM（RRAM（登録商標））、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ（DRAM）など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ38および参照ピクチャメモリ64は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ38は、ビデオエンコーダ20の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

#### 【0123】

[0129]動き推定ユニット42および動き補償ユニット44は、時間的予測を行うために、1つまたは複数の参照フレーム中の1つまたは複数のブロックに対する受信されたビデオブロックのインター予測コーディングを実行する。イントラ予測ユニット46は、代替的に、空間的予測を行うために、コーディングされるべきブロックと同じフレームまたはスライス中の1つまたは複数の隣接ブロックに対する受信されたビデオブロックのイントラ予測コーディングを実行し得る。ビデオエンコーダ20は、たとえば、ビデオデータのブロックごとに適当なコーディングモードを選択するために、複数のコーディングパスを実行し得る。

#### 【0124】

[0130]その上、区分ユニット48は、前のコーディングパスでの前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分し得る。たとえば、区分ユニット48は、最初にフレームまたはスライスをLCUに区分し、レートひずみ分析（たとえば、レートひずみ最適化）に基づいて、LCUの各々をサブCUに区分し得る。モード選

扱ユニット40は、サブCUへのLCUの区分を示す4分木データ構造をさらに生成し得る。4分木のリーフノードCUは、1つまたは複数のPUと1つまたは複数のTUとを含み得る。

【0125】

[0131]モード選択ユニット40は、たとえば、誤差結果に基づいて、コーディングモードのうちの一つ、すなわち、イントラまたはインターを選択し得、得られたイントラコード化ブロックまたはインターコード化ブロックを、残差ブロックデータを生成するために加算器50に、参照フレームとしての使用のために符号化されたブロックを再構築するために加算器62に与える。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのようなシンタックス情報などのシンタックス要素をエントロピー符号化ユニット56に与える。

10

【0126】

[0132]動き推定ユニット42および動き補償ユニット44は、高度に統合され得るが、概念上の目的から別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックに関する動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、たとえば、現在のスライス（または、他のコード化ユニット）内で符号化されている現在のブロックに対する参照ピクチャ（または、他のコード化ユニット）内の予測ブロックに対する現在のビデオフレームまたはピクチャ内のビデオブロックのPUの変位を示し得る。予測ブロックは、絶対差分和（SAD）、2乗差分和（SSD）、または他の差分メトリクスによって決定され得るピクセル差分の観点で、符号化されるべきブロックと厳密に一致することが見出されるブロックである。

20

【0127】

[0133]動き推定ユニット42は、PUの位置を参照ピクチャの予測ブロックの位置と比較することによって、インターコード化スライス中のビデオブロックのPUの動きベクトルを計算する。参照ピクチャは、第1の参照ピクチャリスト（リスト0）または第2の参照ピクチャリスト（リスト1）から選択され得、それらの参照ピクチャリストの各々は、参照ピクチャメモリ64に記憶された1つまたは複数の参照ピクチャを識別する。動き推定ユニット42は、計算された動きベクトルをエントロピー符号化ユニット56と動き補償ユニット44とに送る。

【0128】

30

[0134]動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定された動きベクトルに基づいて、予測ブロックをフェッチまたは生成することを伴い得る。同じく、動き推定ユニット42および動き補償ユニット44は、いくつかの例では、機能的に統合され得る。現在のビデオブロックのPUのための動きベクトルを受信すると、動き補償ユニット44は、動きベクトルが参照ピクチャリストのうちの一つにおいてそれを指す予測ブロックの位置を特定し得る。加算器50は、以下で説明されるように、符号化されている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42は、ルーマ成分に対して動き推定を実行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方に関して、ルーマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40は、ビデオスライスのビデオブロックを復号する際のビデオデコーダ30による使用のために、ビデオブロックとビデオスライスとに関連付けられたシンタックス要素を生成することもできる。

40

【0129】

[0135]動き推定ユニット42は、参照ピクチャメモリ64のDPB内に記憶された参照ピクチャを探索する。モード選択ユニット40は、ピクチャ、ピクチャのスライス、および/またはピクチャのシーケンスを符号化するために、どの参照ピクチャが使用されるべきかを決定し得る。ビデオエンコーダ20は、ピクチャ、スライス、またはシーケンスを符号化するために使用される参照ピクチャのためのピクチャ順序カウンタ（POC）値を符号化し得る。ビデオエンコーダ20は、スライスヘッダ、ならびに/または、P

50

P Sおよび/もしくはS P Sなどのパラメータセット中で、参照ピクチャリスト中に含まれるべき参照ピクチャを示すP O C値を符号化し得る。

【 0 1 3 0 】

[0136]このようにして、ビデオデコーダ30などのビデオデコーダは、スライスヘッダおよび/または(1つまたは複数の)パラメータセット中で示された参照ピクチャを含めることによって、参照ピクチャリストを再生し得る。さらに、動き推定ユニット42によって生成された動きベクトルを使用して、ブロックを符号化した後、ビデオエンコーダ20は、ブロックについての動き情報を符号化し得、ただし、動き情報は、動きベクトルを表すデータと、参照ピクチャリストのための識別子と、参照ピクチャリスト中の参照ピクチャを識別する参照インデックスとを含み得る。

10

【 0 1 3 1 】

[0137]いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ64内に記憶された参照ピクチャのサブ整数ピクセル位置のための値を計算し得る。たとえば、ビデオエンコーダ20は、参照ピクチャの1/4ピクセル位置、1/8ピクセル位置、または他の分数ピクセル位置の値を補間し得る。したがって、動き推定ユニット42は、フルピクセル位置および分数ピクセル位置に対して動き探索を実行し、分数ピクセル精度で動きベクトルを出力し得る。

【 0 1 3 2 】

[0138]ビデオエンコーダ20はまた、P O Cリセット期間を形成し、それぞれのP O Cリセット期間にスライスを割り当てるために、ピクチャのスライスのためのP O Cリセット期間識別子のための値を符号化し得る。ビデオエンコーダ20は、ビデオデータの各レイヤ内で個々にP O Cリセット期間にスライスを割り当て得る。その上、ビデオエンコーダ20は、スライスのためのP O Cリセットタイプを決定し、スライスのためのP O Cリセットタイプの指示のための値(たとえば、2ビット値)を符号化し得る。

20

【 0 1 3 3 】

[0139]図1に関して説明されたように、ビデオエンコーダ20は、適用可能なビデオコーディング規格によって確立された制限に従って構成され得る。たとえば、ビデオエンコーダ20は、各P O Cリセット期間における最初のピクチャが、P O C値が完全または部分的にリセットされるべきであることを示すP O Cリセットタイプを有することを保証し得る。さらに、ビデオエンコーダ20は、いくつかの例では、P O Cリセット期間のP O CアンカーピクチャのためのP O C L S B値を示し得る、P O C L S B値を示す追加の情報を符号化し得る。

30

【 0 1 3 4 】

[0140]概して、上記で説明されたように、ビデオエンコーダ20は、コーディング順序において現在のピクチャより前にコーディングされたピクチャのP O C M S B値のP O C M S B値に基づいて、現在のピクチャのためのP O C M S B値を決定し得る。本開示の態様によれば、ビデオエンコーダ20はまた、ビットストリームのいくつかのピクチャのためのP O C M S B値を示す情報を符号化し得る。たとえば、本開示の態様によれば、ビデオエンコーダ20は、特定のピクチャタイプを有するピクチャのためのP O C M S B値を示すデータを符号化し得る。いくつかの例では、ビデオエンコーダ20は、ピクチャがC R AまたはB L Aピクチャタイプを有することに基づいて、データ、C R AまたはB L Aピクチャタイプを有するピクチャのためのP O C M S B値を符号化するように構成され得る。すなわち、そのような例では、ビデオエンコーダ20は、C R AピクチャタイプとB L Aピクチャタイプとを含むピクチャタイプの最初のセットのみのためのP O C M S B値を符号化し得る。ビデオエンコーダ20は、他のピクチャタイプのためのP O C M S B値を符号化せず、代わりに、現在のピクチャより前に符号化されたピクチャに基づいて、P O C M S B値を決定し得る。

40

【 0 1 3 5 】

[0141]場合によっては、ビデオエンコーダ20は、上記で説明された、指定されたピクチャタイプを有するピクチャ以外のピクチャのためのP O C M S B値を示すデータを符

50

号化し得る。たとえば、場合によっては、ビデオエンコーダ 20 は、そのための POC 値リセットが実行される任意のピクチャのための POC MSB 値を示すデータを符号化し得る。

**【0136】**

[0142]ビデオエンコーダ 20 はまた、参照ピクチャメモリ 64 の DPB 内の参照ピクチャのバンピングおよび出力プロセスを管理し得る。本開示の態様によれば、ビデオエンコーダ 20 は、POC 値リセットを実行するより前に、参照ピクチャメモリ 64 の DPB からピクチャを出力するように構成され得る。たとえば、本開示の態様によれば、特定のアクセスユニットのための POC 値リセットを実行するより前に、ビデオエンコーダ 20 は、「出力のために必要とされる」とマークされている、参照ピクチャメモリの DPB に記憶されたすべてのピクチャを出力するように構成され得る。場合によっては、ビデオエンコーダ 20 は、アクセスユニットの初期ピクチャのためのスライスヘッダを符号化した後であるが、POC 値を導出し、参照ピクチャリストを構築するより前に、出力プロセスを実行し得る。たとえば、ビデオエンコーダ 20 は、アクセスユニットの初期ピクチャのためのスライスヘッダを符号化した後であるが、スライスのペイロード（たとえば、スライスのビデオデータ）を符号化する前に、出力プロセスを実行し得る。

10

**【0137】**

[0143]イントラ予測ユニット 46 は、上記で説明されたように、動き推定ユニット 42 と動き補償ユニット 44 とによって実行されるインター予測の代替として、現在のブロックをイントラ予測し得る。特に、イントラ予測ユニット 46 は、現在のブロックを符号化するために使用すべきイントラ予測モードを決定し得る。いくつかの例では、イントラ予測ユニット 46 は、たとえば別々の符号化パス中に、様々なイントラ予測モードを使用して現在のブロックを符号化し得、イントラ予測ユニット 46（または、いくつかの例では、モード選択ユニット 40）は、テストされたモードから使用すべき適当なイントラ予測モードを選択し得る。

20

**【0138】**

[0144]たとえば、イントラ予測ユニット 46 は、様々なテストされたイントラ予測モードのためのレートひずみ分析を使用してレートひずみ値を計算し、テストされたモードの中で最良のレートひずみ特性を有するイントラ予測モードを選択し得る。レートひずみ分析は、概して、符号化ブロックと、符号化ブロックを生成するために符号化された元の符号化されていないブロックとの間のひずみ（または誤差）の量、ならびに符号化ブロックを生成するために使用されるビットレート（すなわち、ビット数）を決定する。イントラ予測ユニット 46 は、どのイントラ予測モードがブロックについて最良のレートひずみ値を呈するかを決定するために、様々な符号化ブロックのひずみおよびレートから比を計算し得る。

30

**【0139】**

[0145]ブロックのためのイントラ予測モードを選択した後、イントラ予測ユニット 46 は、ブロックのための選択されたイントラ予測モードを示す情報をエントロピー符号化ユニット 56 に与え得る。エントロピー符号化ユニット 56 は、選択されたイントラ予測モードを示す情報を符号化し得る。ビデオエンコーダ 20 は、複数のイントラ予測モードインデックステーブルおよび複数の修正されたイントラ予測モードインデックステーブル（コードワードマッピングテーブルとも呼ばれる）と、様々なブロックの符号化コンテキストの定義と、コンテキストの各々について使用すべき、最確イントラ予測モード、イントラ予測モードインデックステーブル、および修正されたイントラ予測モードインデックステーブルの指示とを含み得る構成データを送信ビットストリーム中に含め得る。

40

**【0140】**

[0146]ビデオエンコーダ 20 は、モード選択ユニット 40 からの予測データを、コーディングされている元のビデオブロックから減算することによって、残差ビデオブロックを形成する。加算器 50 は、この減算演算を実行する、1 つまたは複数の構成要素を表す。変換処理ユニット 52 は、離散コサイン変換（DCT）または概念的に類似する変換など

50

の変換を残差ブロックに適用し、残差変換係数値を備えるビデオブロックを生成する。変換処理ユニット52は、DCTに概念的に類似する他の変換を実行することができる。ウェーブレット変換、整数変換、サブバンド変換、または他のタイプの変換も使用され得る。いずれの場合でも、変換処理ユニット52は、変換を残差ブロックに適用し、残差変換係数のブロックを生成する。変換は、ピクセル値領域からの残差情報を、周波数領域などの変換領域に変換し得る。

#### 【0141】

[0147]変換処理ユニット52は、得られた変換係数を量子化ユニット54へ送り得る。量子化ユニット54は、ビットレートをさらに低減するために変換係数を量子化する。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって、修正され得る。いくつかの例では、量子化ユニット54は次いで、量子化された変換係数を含む行列の走査を実行し得る。代替的に、エントロピー符号化ユニット56が、走査を実行し得る。

10

#### 【0142】

[0148]量子化に続いて、エントロピー符号化ユニット56は、量子化された変換係数をエントロピーコーディングする。たとえば、エントロピー符号化ユニット56は、コンテキスト適応型可変長コーディング(CAVLC)、コンテキスト適応型バイナリ算術コーディング(CABAC)、シンタクスペースコンテキスト適応型バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディングまたは別のエントロピーコーディング技法を実行し得る。コンテキストベースのエントロピーコーディングの場合、コンテキストは、隣接ブロックに基づき得る。エントロピー符号化ユニット56によるエントロピーコーディングに続いて、符号化されたビットストリームは、別のデバイス(たとえば、ビデオデコーダ30)に送信されるか、または後で送信するかもしくは取り出すためにアーカイブされ得る。

20

#### 【0143】

[0149]逆量子化ユニット58および逆変換ユニット60は、たとえば、参照ブロックとして後で使用するために、ピクセル領域中で残差ブロックを再構築するために、それぞれ逆量子化および逆変換を適用する。動き補償ユニット44は、残差ブロックを参照ピクチャメモリ64のフレームのうちの1つの予測ブロックに加算することによって参照ブロックを計算し得る。動き補償ユニット44はまた、動き推定において使用するためのサブ整数ピクセル値を計算するために、再構築された残差ブロックに1つまたは複数の補間フィルタを適用し得る。加算器62は、参照ピクチャメモリ64に記憶するための再構築されたビデオブロックを生成するために、動き補償ユニット44によって生成された動き補償された予測ブロックに再構築された残差ブロックを加算する。再構築されたビデオブロックは、動き推定ユニット42および動き補償ユニット44によって、後続のビデオフレーム中のブロックをインターコーディングするために参照ブロックとして使用され得る。

30

#### 【0144】

[0150]このようにして、図2のビデオエンコーダ20は、マルチレイヤビデオデータの第1のピクチャを符号化することと、その場合、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのPOC MSB値を表すデータを符号化することと、を行うように構成された、ビデオエンコーダの一例を表す。ビデオエンコーダ20はまた、本開示の態様によれば、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのPOC値のためのPOCリセットを示すデータを符号化することと、その場合、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを符号化するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、DPB内に記憶されたすべてのピクチャを出力することと、を行うように構成され得る。

40

#### 【0145】

[0151]図3は、本開示の技法による、POC値を管理するための技法を実装し得る、ピ

50

デオデコーダ30の一例を示すブロック図である。上述されたように、ビデオデコーダ30は、マルチビューおよび/またはスケーラブルビデオコーディングを実行するように適応され得る。たとえば、ビデオデコーダ30は、SHVC、MV-HEVC、または3D-HEVCなどの1つまたは複数のビデオコーディング規格の拡張に適合するビットストリームを復号するように構成され得る。ただし、特定のコーディング規格が参照されるが、本技法は、任意の1つのコーディング規格に固有のものではなく、将来のおよび/またはまだ開発されていない規格とともに実装され得ることを理解されたい。

【0146】

[0152]図3の例では、ビデオデコーダ30は、ビデオデータメモリ68と、エントロピー復号ユニット70と、動き補償ユニット72と、イントラ予測ユニット74と、逆量子化ユニット76と、逆変換ユニット78と、参照ピクチャメモリ82と、加算器80とを含む。ビデオデコーダ30は、いくつかの例では、ビデオエンコーダ20(図2)に関して説明された符号化パスに対して概ね逆の復号パスを実行し得る。動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルに基づいて、予測データを生成し得、一方、イントラ予測ユニット74は、エントロピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて、予測データを生成し得る。

【0147】

[0153]ビデオデータメモリ68は、ビデオデコーダ30の構成要素によって復号されるべき、符号化されたビデオビットストリームなどのビデオデータを記憶し得る。ビデオデータメモリ68内に記憶されるビデオデータは、たとえば、コンピュータ可読媒体から、たとえば、カメラなどのローカルビデオソースから、ビデオデータのワイヤードもしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって取得され得る。ビデオデータメモリ68は、符号化されたビデオビットストリームからの符号化されたビデオデータを記憶するコード化ピクチャバッファ(CPB)を形成し得る。参照ピクチャメモリ82は、たとえば、イントラコーディングモードまたはインターコーディングモードで、ビデオデコーダ30によってビデオデータを復号する際に使用するための参照ビデオデータを記憶する、DPBと呼ばれることがある。ビデオデータメモリ68および参照ピクチャメモリ82は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM)、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ68および参照ピクチャメモリ82は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ68は、ビデオデコーダ30の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

【0148】

[0154]復号プロセス中に、ビデオデコーダ30は、ビデオエンコーダ20から、符号化ビデオスライスのビデオブロックと、関連付けられるシンタックス要素とを表す、符号化されたビデオビットストリームを受信する。ビデオデコーダ30のエントロピー復号ユニット70は、量子化された係数と、動きベクトルまたはイントラ予測モードインジケータと、他のシンタックス要素とを生成するために、ビットストリームをエントロピー復号する。エントロピー復号ユニット70は、動きベクトルと他のシンタックス要素とを動き補償ユニット72へ転送する。ビデオデコーダ30は、ビデオスライスレベルおよび/またはビデオブロックレベルでシンタックス要素を受信し得る。

【0149】

[0155]ビデオスライスが、イントラコード化(I)スライスとしてコーディングされる時、イントラ予測ユニット74は、現在のフレームまたはピクチャの以前に復号されたブロックから、シグナリングされたイントラ予測モードおよびデータに基づいて、現在のビデオスライスのビデオブロックのための予測データを生成し得る。ビデオフレームが、インターコード化(すなわち、B、P、またはGPB)スライスとしてコーディングされるとき、動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベ

10

20

30

40

50

クトルと他のシンタックス要素とに基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリストのうちの1つの中の参照ピクチャのうちの1つから生成され得る。ビデオデコーダ30は、参照ピクチャメモリ82のDPB内に記憶された参照ピクチャに基づいて、デフォルトの構築技法を使用して、参照フレームリストのリスト0とリスト1とを構築し得る。

**【0150】**

[0156]より詳細には、ビデオデコーダ30は、PPSに対応するピクチャのスライスがPOCリセット情報を含むか否かを示すPPSを復号し得る。スライスがPOCリセット情報を含むと仮定すると、ビデオデコーダ30は、POCリセット情報を含むピクチャのスライスのスライスセグメントヘッダを復号し得る。POCリセット情報は、POCリセット期間識別子とPOCリセットタイプとを含み得る。

10

**【0151】**

[0157]POCリセット期間識別子は、スライスが対応するPOCリセット期間を示し得る。POCリセットタイプは、スライスを含むピクチャのPOC値が完全にリセットされるか否か、または、POC値のMSBのみがリセットされることを示し得る。さらに、POCリセットタイプは、POCリセットが実行されるべきではないこと、または追加の情報がシグナリングされることを示し得る。POCリセットタイプが、追加の情報がシグナリングされることを示す場合、ビデオデコーダ30は、POC LSB値と、POC値が完全にリセットされるか、MSBのみがリセットされるかを示し得る、追加の情報を復号し得る。

20

**【0152】**

[0158]本開示の態様によれば、ビデオデコーダ30は、POCリセットを示す情報に基づいて、復号ピクチャバッファからピクチャを出力し得る。たとえば、ビデオデコーダ30は、POC値リセットを実行するより前に、参照ピクチャメモリ84のDPBからピクチャを出力するように構成され得る。すなわち、本開示の態様によれば、特定のアクセスユニットのためのPOC値リセットを実行するより前に、ビデオデコーダ30は、「出力のために必要とされる」とマークされている、参照ピクチャメモリのDPBに記憶されたすべてのピクチャを出力するように構成され得る。場合によっては、ビデオデコーダ30は、アクセスユニットの初期ピクチャのためのスライスヘッダを復号した後であるが、POC値を導出し、参照ピクチャリストを構築するより前に、出力プロセスを実行し得る。たとえば、ビデオデコーダ30は、アクセスユニットの初期ピクチャのためのスライスヘッダを復号した後であるが、スライスのペイロード（たとえば、スライスのビデオデータ）を復号する前に、出力プロセスを実行し得る。

30

**【0153】**

[0159]概して、ビデオデコーダ30は、現在のピクチャのスライスのスライスヘッダ中で受信されたデータに基づいて、現在復号されているピクチャのためのPOC LSB値を決定し得る。ビデオデコーダ30は、コーディング順序において現在のピクチャより前にコーディングされたピクチャのPOC MSB値のPOC MSB値に基づいて、現在のピクチャのためのPOC MSB値を決定し得る。

**【0154】**

40

[0160]本開示の態様によれば、場合によっては、ビデオデコーダ30は、ビットストリームのいくつかのピクチャのための符号化されたビットストリームから、POC MSB値を示すデータを復号し得る。たとえば、本開示の態様によれば、ビデオデコーダ30は、現在復号されているピクチャのためのピクチャタイプを決定し得る。ピクチャタイプに基づいて、ビデオデコーダ30は、復号順序において現在のピクチャより前にコーディングされたピクチャのPOC MSB値に基づいて、ピクチャのためのPOC MSB値を決定するのではなく、ピクチャのためのPOC MSB値を示すデータを復号し得る。

**【0155】**

[0161]場合によっては、ビデオデコーダ30は、ピクチャタイプがCRAまたはBLAピクチャタイプであることに基づいて、現在のピクチャのためのPOC MSB値を示す

50



データを復号し得る。たとえば、ビデオデコーダ30は、CRAピクチャタイプとBLAピクチャタイプとを含むピクチャタイプの最初のセットのみのためのPOC MSB値を復号し得る。ビデオデコーダ30は、他のピクチャタイプのためのPOC MSB値を復号せず、代わりに、現在のピクチャより前に符号化されたピクチャに基づいて、POC MSB値を決定し得る。他の例では、ビデオデコーダ30は、追加または代替として、他のピクチャおよびピクチャタイプのためのPOC MSB値を示すデータを復号し得る。たとえば、本開示の態様によれば、ビデオデコーダ30は、そのためのPOC値リセットが実行される任意のピクチャのためのPOC MSB値を示すデータを復号し得る。

【0156】

[0162]ビデオデコーダ30はまた、参照ピクチャリスト中に含まれるべき参照ピクチャのPOC値を復号し得る。ビデオデコーダ30は、スライスセグメントヘッダ中で、および/または、PPSもしくはSPSなどのパラメータセットから、これらのPOC値を復号し得る。次いで、ビデオデコーダ30は、復号されたPOC値によって識別された参照ピクチャを含む参照ピクチャリストを構築し得る。

10

【0157】

[0163]スライスのための参照ピクチャリストを構築した後、ビデオデコーダ30は、スライスのブロックを復号し得る。動き補償ユニット72は、動きベクトルと他のシンタックス要素とをパースすることによって現在のビデオスライスのビデオブロックに関する予測情報を決定し、復号されている現在のビデオブロックの予測ブロックを生成するために、その予測情報を使用する。たとえば、動き補償ユニット72は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード（たとえば、イントラ予測またはインター予測）と、インター予測スライスタイプ（たとえば、Bスライス、Pスライス、またはGPBスライス）と、スライスの各インター符号化ビデオブロックのための動きベクトルと、スライスの各インターコード化ビデオブロックのためのインター予測ステータスと、現在のビデオスライス中のビデオブロックを復号するための他の情報とを決定するために、受信されたシンタックス要素のいくつかを使用する。インター予測ブロックのための動き情報は、ブロックを予測するために使用すべき参照ピクチャリスト中の参照ピクチャを識別するために、参照ピクチャリスト識別子と参照インデックスとを含み得る。

20

【0158】

[0164]動き補償ユニット72は、補間フィルタに基づいて補間を実行することもできる。動き補償ユニット72は、参照ブロックのサブ整数ピクセルのための補間された値を計算するために、ビデオブロックの符号化中にビデオエンコーダ20によって使用された補間フィルタを使用し得る。この場合、動き補償ユニット72は、受信されたシンタックス要素からビデオエンコーダ20によって使用された補間フィルタを決定し、予測ブロックを生成するために、その補間フィルタを使用し得る。

30

【0159】

[0165]逆量子化ユニット76は、ビットストリーム中で与えられ、エントローピー復号ユニット70によって復号された、量子化された変換係数を逆量子化（inverse quantize）、すなわち、逆量子化（de-quantize）する。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するための、ビデオスライス中の各ビデオブロックに関してビデオデコーダ30によって計算される量子化パラメータ $Q_{P\gamma}$ の使用を含み得る。

40

【0160】

[0166]逆変換ユニット78は、ピクセル領域において残差ブロックを生成するために、逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換プロセスを変換係数に適用する。

【0161】

[0167]動き補償ユニット72が、動きベクトルおよび他のシンタックス要素に基づいて現在のビデオブロックのための予測ブロックを生成した後、ビデオデコーダ30は、逆変

50

換ユニット78からの残差ブロックを動き補償ユニット72によって生成された対応する予測ブロックと加算することによって、復号ビデオブロックを形成する。加算器80は、この加算演算を実行する1つまたは複数の構成要素を表す。所望される場合、ブロックネスアर्टィファクトを除去するために、復号されたブロックをフィルタリングするためのデブロッキングフィルタも適用され得る。(コーディンググループ中の、またはコーディンググループ後のいずれかにおける)他のループフィルタも、ピクセルの遷移を平滑化し、または場合によっては、ビデオ品質を改善するために使用され得る。所与のフレームまたはピクチャ中の復号ビデオブロックは、次いで、参照ピクチャメモリ82に記憶され、この参照ピクチャメモリ82は、後続の動き補償のために使用される参照ピクチャを記憶する。参照ピクチャメモリ82はまた、復号されたビデオを、図1のディスプレイデバイス32などのディスプレイデバイス上に後で提示するために記憶する。

10

【0162】

[0168]このようにして、図3のビデオデコーダ30は、マルチレイヤビデオデータの第1のピクチャを復号することと、その場合、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのPOC\_MSB値を表すデータをコーディングすることと、を行うように構成された、ビデオデコーダの一例を表す。ビデオデコーダ30はまた、本開示の態様によれば、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのPOC値のためのPOCリセットを示すデータを復号することと、その場合、第1のピクチャがアクセスユニット中に含まれ、第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット中に含まれない、DPB内に記憶されたすべてのピクチャを出力することと、を行うように構成され得る。

20

【0163】

[0169]図4は、本開示の技法による、POC値リセットを実行するより前に、DPBのピクチャを出力することを含む、データを符号化するための例示的な方法を示すフローチャートである。加えて、図4の方法は、現在のブロックを符号化することを含む。現在のブロックは、現在のCU、または現在のCUの一部を備え得る。ビデオエンコーダ20(図1および図2)に関して説明されるが、他のデバイスが図4の方法と同様の方法を実行するように構成され得ることを理解されたい。

30

【0164】

[0170]この例では、ビデオエンコーダ20は、現在のスライスをPOCリセット期間に割り当て得る(148)。たとえば、現在のスライスがアンカーピクチャの一部を形成する場合、ビデオエンコーダ20は、現在のスライスが前のPOCリセット期間識別子とは異なるPOCリセット期間識別子を有する新しいPOCリセット期間の開始を形成することをシグナリングし得る。代替的に、現在のスライスがアンカーピクチャの一部を形成しない場合、ビデオエンコーダ20は、現在のスライスが既存のPOCリセット期間の一部を形成することをシグナリングし得る。

【0165】

[0171]次いで、ビデオエンコーダ20は、DPBに記憶されているすべてのピクチャを出力する(150)。たとえば、ビデオエンコーダ20は、現在のアクセスユニット中に含まれる、DPBに記憶されたいかなるピクチャをも除いて、「出力のために必要とされる」とマークされたすべてのピクチャを出力し得る。ビデオエンコーダ20は、現在のスライスをPOCリセット期間に割り当てた(たとえば、スライスヘッダを符号化した)後であるが、最初のスライスのビデオデータを符号化するより前に、DPBからピクチャを出力し得る。

40

【0166】

[0172]次いで、ビデオエンコーダ20は、POCリセットを実行し、参照ピクチャリストを形成し得る(152)。ビデオエンコーダ20は、参照ピクチャリスト中に含まれた参照ピクチャのPOC値をさらにシグナリングする(154)。たとえば、ビデオエンコ

50

ーダ20は、SPSもしくはPPSなどのパラメータセット中、および/またはスライスのためのスライスヘッダ中で、参照ピクチャのためのPOC値(または、POC値の部分、たとえば、POC LSB値)を符号化し得る。いくつかの参照ピクチャ(たとえば、長期参照ピクチャ)は、パラメータセット中でシグナリングされ得るのに対して、他の参照ピクチャ(たとえば、短期参照ピクチャ)は、スライスヘッダ中でシグナリングされ得る。

#### 【0167】

[0173]たとえば、現在のスライスのブロックのための、最良のレートひずみ特性を生じる参照ピクチャのセットを決定するために、参照ピクチャリストを形成するステップと、どのピクチャが参照ピクチャリスト中に含まれるかをシグナリングするステップとが、いくつかの異なる符号化パスを介して複数回実行され得ることを理解されたい。すなわち、ビデオエンコーダ20は、単一のブロックの個々の特性のみに基づくのではなく、現在のスライス中のすべてのブロックの特性に基づいて、参照ピクチャリスト中に含まれた参照ピクチャのセットを選択し得る。

10

#### 【0168】

[0174]次いで、ビデオエンコーダ20は、現在のスライスの現在のブロックを予測し得る(156)。たとえば、ビデオエンコーダ20は、現在のブロックの1つまたは複数の予測ユニット(PU)を計算し得る。より詳細には、動き推定ユニット42は、たとえば、SAD、SSD、MAD、MSD、または他の誤差計算メトリクスに基づいて、参照ブロックとして使用される一致するブロックを識別するために、参照ピクチャリストの参照ピクチャの中で現在のブロックの動き探索を実行し得る。動き推定ユニット42は、動き探索に基づいて、現在のブロックのための動きベクトルを生成し得る。次いで、ビデオエンコーダ20は、ブロックのための動き情報を符号化し得(158)、ブロックのための動き情報は、参照ブロックと一緒に識別する、動きベクトルのための動きベクトル差分値と、参照ピクチャリスト識別子と、参照インデックスとを含み得る。

20

#### 【0169】

[0175]次いで、ビデオエンコーダ20は、たとえば、変換ユニット(TU)を生成するために、現在のブロックのための残差ブロックを計算し得る(160)。残差ブロックを計算するために、ビデオエンコーダ20は、元のコーディングされていないブロックと現在のブロックのための予測ブロックとの間の差分を計算し得る。次いで、ビデオエンコーダ20は、残差ブロックの係数を変換および量子化し得る(162)。次に、ビデオエンコーダ20は、残差ブロックの量子化された変換係数を走査し得る(164)。走査中に、または走査に続いて、ビデオエンコーダ20は、係数をエントロピー符号化し得る(166)。たとえば、ビデオエンコーダ20は、CAVLCまたはCABACを使用して係数を符号化し得る。次いで、ビデオエンコーダ20は、ブロックのエントロピーコード化データを出力し得る(168)。

30

#### 【0170】

[0176]このようにして、図4の方法は、ビデオエンコーダが、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのPOC値のためのPOCリセットを示す符号化データを符号化することと、その場合、第1のピクチャがアクセスユニット中に含まれ、ならびに第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを符号化するより前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット内に含まれない、DPB内に記憶されたすべてのピクチャを出力することを含む、方法の一例を表す。

40

#### 【0171】

[0177]図5は、本開示の技法による、POC値リセットを実行するより前に、DPBのピクチャを出力することを含む、データを復号するための例示的な方法を示すフローチャートである。加えて、図5の方法は、現在のブロックを復号することを含む。現在のブロックは、現在のCU、または現在のCUの一部を備え得る。ビデオデコーダ30(図1および図3)に関して説明されるが、他のデバイスが図5の方法と同様の方法を実行する

50

ように構成され得ることを理解されたい。

【0172】

[0178]最初に、ビデオデコーダ30は、POCリセットを示すデータを復号する(200)。いくつかの例では、本明細書で説明されるように、POCリセットを示すデータは、現在のスライスのスライスセグメントヘッダ中に含まれるPOCリセット識別子であり得る。このような例では、ビデオデコーダ30は、POCリセットタイプ識別子をさらに復号し得る。図5の方法は、POCリセット期間識別子が、現在のスライスが新しいPOCリセット期間の一部であることを示すという仮定に基づいている。ビデオデコーダ30は、現在のピクチャが特定のアクセスユニット中の第1のピクチャ(たとえば、特定のアクセスユニットのすべてのピクチャの中の最下位レイヤ識別子を有するピクチャ)であることをさらに決定し得る。

10

【0173】

[0179]ビデオデコーダ30は、DPBに記憶されたからすべてのピクチャを出力し得る(202)。たとえば、本開示の態様によれば、POC値リセットを示すデータに基づいて、ビデオデコーダ30は、現在のアクセスユニット中に含まれる、DPBに記憶されたいかなるピクチャをも除いて、「出力のために必要とされる」とマークされているすべてのピクチャをDPBから出力し得る。ビデオデコーダ30は、現在のスライスのためのスライスヘッダを復号した後であるが、POC値を導出し、参照ピクチャを復号するより前に、DPBからピクチャを出力し得る。

【0174】

[0180]次いで、ビデオデコーダ30はPOC値をリセットし、たとえば、スライスセグメントヘッダ、および/または、PPSもしくはSPSなどのパラメータセットから、参照ピクチャリスト中に含まれるべき参照ピクチャのPOC値を復号する(204)。次いで、ビデオデコーダ30は、参照ピクチャリストを形成する(206)。

20

【0175】

[0181]次に、ビデオデコーダ30は、現在のブロックのための動き情報を復号する(208)。動き情報は、たとえば、参照ピクチャリスト識別子と、参照ピクチャリストへの参照インデックスとを含み得る。次いで、ビデオデコーダ30は、たとえば、現在のブロックのための予測ブロックを計算するために、インター予測モードを使用して、現在のブロックを予測する(200)。より詳細には、ビデオデコーダ30は、どの参照ピクチャリストを使用すべきかを識別するために参照ピクチャリスト識別子と、参照ピクチャリスト中の参照インデックスを識別するために参照インデックスとを使用する。次いで、ビデオデコーダ30は、現在のブロックのための動きベクトルを復号し、識別された参照ピクチャ中の参照ブロックを識別する。

30

【0176】

[0182]ビデオデコーダ30はまた、現在のブロックに対応する残差ブロックの係数のためのエン트로ピーコード化データなど、現在のブロックのためのエン트로ピーコード化データを受信し得る(212)。ビデオデコーダ30は、残差ブロックの係数を再生するために、エン트로ピーコード化データをエン트로ピー復号し得る(214)。次いで、ビデオデコーダ30は、量子化された変換係数のブロックを作成するために、再生された係数を逆走査し得る(216)。すなわち、逆走査を使用して、ビデオデコーダ30は、1次元ベクトルを2次元行列に変換する。次いで、ビデオデコーダ30は、残差ブロックを生成するために、係数を逆量子化および逆変換し得る(218)。最終的に、ビデオデコーダ30は、予測ブロックと残差ブロックとを組み合わせることによって現在のブロックを復号し得る(220)。

40

【0177】

[0183]このようにして、図5の方法は、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのPOC値のためのPOCリセットを示すデータを復号すること、その場合、第1のピクチャがアクセスユニット中に含まれ、ならびに第1のピクチャのPOC値のためのPOCリセットを示すデータに基づいて、および第1のピクチャを復号するよ

50

り前に、コーディング順序において第1のピクチャに先行する、およびアクセスユニット内に含まれない、DPB内に記憶されたすべてのピクチャを出力することを含む、方法の一例を表す。

【0178】

[0184]図6は、本開示の技法による、POC MSB値を示すデータを符号化することを含む、データを符号化するための例示的な方法を示すフローチャートである。ビデオエンコーダ20(図1および図2)に関して説明されるが、他のデバイスが図6の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0179】

[0185]この例では、ビデオエンコーダ20は、現在符号化されているピクチャのためのピクチャタイプを決定し得る(240)。場合によっては、現在のピクチャのピクチャタイプは、上記で説明されたCRA、BLA、または他のランダムアクセスピクチャタイプなど、ランダムアクセスピクチャタイプであり得る。

10

【0180】

[0186]本開示の態様によれば、ピクチャタイプに基づいて、ビデオエンコーダ20は、現在のピクチャのためのPOC MSB値を示すデータを符号化すべきか否かを決定し得る(242)。たとえば、ビデオエンコーダ20は、POC MSB値いくつかのピクチャタイプを示すデータを符号化し得る。すなわち、ビデオエンコーダ20は、ピクチャがピクチャタイプの最初のセットに含まれるときのみ、POC MSB値を符号化し得る。ビデオエンコーダ20は、ピクチャがピクチャタイプの最初のセットに含まれないとき、POC MSB値を符号化しなくてよい。説明のための一例では、ビデオエンコーダ20は、CRAまたはBLAピクチャタイプを有するピクチャのためのPOC MSB値を示すデータを符号化し得る。他の例では、ビデオエンコーダ20は、追加または代替として、他のピクチャおよびピクチャタイプのためのPOC MSB値を示すデータを符号化し得る。たとえば、本開示の態様によれば、ビデオエンコーダ20は、そのためのPOC値リセットが実行される任意のピクチャのためのPOC MSB値を示すデータを符号化し得る。

20

【0181】

[0187]POC MSB値がシグナリングされる場合(ステップ242のyes分岐)、ビデオエンコーダ20は、POC MSB値を示すデータを符号化し得る。たとえば、ビデオエンコーダ20は、現在のピクチャのためのそのPOC MSB値を示す1つまたは複数のシンタックス要素を符号化し得る。

30

【0182】

[0188]次いで、ビデオエンコーダ20は、決定されたPOC値を現在のピクチャの現在のスライスに割り当て得る246(246)。ビデオエンコーダ20はまた、現在のスライスを符号化し得る(248)。いくつかの例では、ビデオエンコーダ20は、図4の例のステップ156~168に関して説明された技法と同様の技法を使用して、現在のスライスを符号化し得る。

【0183】

[0189]このようにして、図6の方法は、ビデオエンコーダが、マルチレイヤビデオデータの第1のピクチャを符号化することと、その場合、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのPOC MSB値を表すデータを符号化することと、を含む方法の一例を表す。

40

【0184】

[0190]図7は、本開示の技法による、POC MSB値を示すデータを復号することを含む、データを復号するための例示的な方法を示すフローチャートである。ビデオデコーダ30(図1および図3)に関して説明されるが、他のデバイスが図7の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0185】

50

[0191]この例では、ビデオデコーダ30は、現在復号されているピクチャのためのピクチャタイプを決定し得る(260)。場合によっては、現在のピクチャのピクチャタイプは、上記で説明されたCRA、BLA、または他のランダムアクセスピクチャタイプなど、ランダムアクセスピクチャタイプであり得る。

【0186】

[0192]本開示の態様によれば、ピクチャタイプに基づいて、ビデオデコーダ30は、現在のピクチャのためのPOC MSB値を示すデータを復号すべきか否かを決定し得る(262)。たとえば、ビデオデコーダ30は、復号順序において現在のピクチャより前にコーディングされたピクチャのPOC MSB値に基づいて、ピクチャのためのPOC MSB値を決定するのではなく、ピクチャのためのPOC MSB値を示すデータを復号し得る。すなわち、ビデオデコーダ30は、現在のピクチャがピクチャタイプの最初のセット中に含まれるときのみ、POC MSB値を復号し得、ビデオデコーダ30は、ピクチャがピクチャタイプの最初のセット中に含まれないとき、POC MSB値を復号しなくてよい。場合によっては、ビデオデコーダ30は、ピクチャタイプがCRAまたはBLAピクチャタイプであることに基づいて、現在のピクチャのためのPOC MSB値を示すデータを復号し得る。他の例では、ビデオデコーダ30は、追加または代替として、他のピクチャおよびピクチャタイプのためのPOC MSB値を示すデータを復号し得る。たとえば、本開示の態様によれば、ビデオデコーダ30は、そのためのPOC値リセットが実行される任意のピクチャのためのPOC MSB値を示すデータを復号し得る。

【0187】

[0193]ビデオデコーダ30が、POC MSB値を示すデータを復号する場合(ステップ262のyes分岐)、ビデオデコーダ30は、POC MSB値を示すデータを復号し得る(264)。たとえば、ビデオデコーダ30は、現在のピクチャのためのPOC MSB値を示す1つまたは複数のシンタックス要素を復号し得る。ビデオデコーダ30が、POC MSB値を示すデータを復号しない場合(ステップ262のno分岐)、ビデオデコーダ30は、復号順序において現在のピクチャより前に復号されたピクチャのPOC MSB値に基づいて、現在のピクチャのためのPOC値を決定し得る。

【0188】

[0194]次いで、ビデオデコーダ30は、決定されたPOC値を現在のピクチャの現在のスライスに割り当て得る246(266)。ビデオデコーダ30はまた、現在のスライスを復号し得る(270)。いくつかの例では、ビデオデコーダ30は、図5の例のステップ206~220に関して説明された技法と同様の技法を使用して、現在のスライスを復号し得る。

【0189】

[0195]このようにして、図7の方法は、ビデオデコーダが、マルチレイヤビデオデータの第1のピクチャを復号することと、その場合、第1のピクチャが第1のピクチャタイプを有し、第1のピクチャが第1のピクチャタイプを有することに基づいて、第1のピクチャのためのPOC MSB値を表すデータを復号することと、を含む方法の一例を表す。

【0190】

[0196]例によっては、本明細書で説明された技法のうちのいずれかの、いくつかの動作またはイベントは、異なる順序で実行され得、追加、統合、または完全に除外され得る(たとえば、すべての説明された動作またはイベントが、本技法の実施のために必要であるとは限らない)ことを認識されたい。その上、いくつかの例では、動作またはイベントは、たとえば、マルチスレッドの処理、割込み処理、または多数のプロセッサを用いて、連続的ではなく同時に実行され得る。

【0191】

[0197]1つまたは複数の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、コンピュータ可読媒体上の1つもしくは複数の命令もしくはコード上に記憶され、またはこれを介して送信され、ハードウェアベースの処理ユニットによって実行

され得る。コンピュータ可読媒体は、データ記憶媒体などの有形の媒体に対応するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従う、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含み得る。このようにして、コンピュータ可読媒体は、一般に、(1)非一時的である有形のコンピュータ可読記憶媒体、または(2)信号もしくは搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明された技法の実装のために命令、コード、および/またはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスされ得る任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【0192】

[0198]限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM(登録商標)、CD-ROMもしくは他の光ディスク記憶装置、磁気ディスク記憶装置もしくは他の磁気記憶デバイス、フラッシュメモリ、または命令もしくはデータ構造の形態で所望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る任意の他の媒体を備え得る。また、任意の接続が、コンピュータ可読媒体と適切に呼ばれる。たとえば、命令が、ウェブサイト、サーバ、または他の遠隔ソースから、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的媒体を含むのではなく、代わりに、非一時的な有形の記憶媒体を対象とすることを理解されたい。本明細書で使用されるディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)、およびBlu-rayディスク(disc)を含み、ここで、ディスク(disk)は、通常、データを磁氣的に再生し、一方、ディスク(disc)は、データをレーザーで光学的に再生する。上記の組合せも、コンピュータ可読媒体の範囲の中に含まれるべきである。

【0193】

[0199]命令は、1つまたは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、または他の同等の集積された論理回路もしくは離散論理回路などの、1つまたは複数のプロセッサによって実行され得る。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造のいずれか、または本明細書で説明された技法を実装するために好適な任意の他の構造を指し得る。加えて、いくつかの態様では、本明細書で説明された機能は、符号化および復号のために構成されるか、または複合コーデックに組み込まれる、専用のハードウェアモジュールおよび/またはソフトウェアモジュール内で提供され得る。また、本技法は、1つまたは複数の回路または論理素子において完全に実装され得る。

【0194】

[0200]本開示の技法は、ワイヤレスハンドセット、集積回路(IC)またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置で実装され得る。本開示では、開示される技法を実行するように構成されたデバイスの機能的態様を強調するために様々な構成要素、モジュール、またはユニットが説明されたが、それらの構成要素、モジュール、またはユニットを、必ずしも異なるハードウェアユニットによって実現する必要があるとは限らない。むしろ、上記で説明されたように、様々なユニットが、好適なソフトウェアおよび/またはファームウェアとともに、上記で説明されたような1つまたは複数のプロセッサを含めて、コーデックハードウェアユニットにおいて組み合わされるか、または相互動作ハードウェアユニットの集合によって提供され得る。

10

20

30

40

50

## 【 0 1 9 5 】

[0201]様々な例が、説明された。これらおよび他の例は、次の特許請求の範囲内に入る。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

## [ C 1 ]

ビデオデータを復号する方法であって、

マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウン  
ト ( P O C ) 値のための P O C リセットを示すデータを復号することと、ここにおいて、  
前記第1のピクチャがアクセスユニット中に含まれ、

前記第1のピクチャの前記 P O C 値のための前記 P O C リセットを示す前記データに基  
づいて、および前記第1のピクチャを復号するより前に、コーディング順序において前記  
第1のピクチャに先行する、および前記アクセスユニット中に含まれない、復号ピクチャ  
バッファ ( D P B ) 内に記憶されたすべてのピクチャを出力することと、  
を備える、方法。

10

## [ C 2 ]

前記 D P B に記憶されたすべてのピクチャを出力することが、出力のために必要とされ  
るとマークされた、前記 D P B 内のすべてのピクチャを出力することを備える、C 1 に記  
載の方法。

## [ C 3 ]

すべてのピクチャを出力することが、前記第1のピクチャのビデオデータの初期のスラ  
イスのスライスヘッダを復号した後、および前記スライスのペイロードを復号するより前  
に、すべてのピクチャを出力することを備える、C 1 に記載の方法。

20

## [ C 4 ]

前記 P O C リセットを示す前記データを復号することが、ビデオデータの前記第1のピ  
クチャのスライスのための P O C リセットインデックスを表すデータを復号することを備  
える、C 1 に記載の方法。

## [ C 5 ]

前記 P O C リセットを示す前記データが、前記第1のピクチャを包含する前記アクセス  
ユニットから開始する P O C リセット期間を示すデータを備える、C 1 に記載の方法。

## [ C 6 ]

前記 D P B のすべてのピクチャを出力した後、前記アクセスユニットの前記第1のピク  
チャおよび任意の他のピクチャのための P O C 値リセットを実行することをさらに備える  
、C 1 に記載の方法。

30

## [ C 7 ]

前記 D P B のすべてのピクチャを出力した後、前記アクセスユニットの各ピクチャを復  
号するための参照ピクチャリストを構築することをさらに備える、C 1 に記載の方法。

## [ C 8 ]

前記 D P B のすべてのピクチャを出力することが、最小のピクチャ順序カウン  
ト ( P O C ) 値を有する前記 D P B のピクチャから開始して、前記 D P B の前記ピクチャの前記 P  
O C 値の昇順で、前記 D P B の前記ピクチャを出力することと、レイヤ i d の昇順で、前  
記 D P B の同じ P O C 値を有するピクチャを出力することと、を備える、C 1 に記載の方  
法。

40

## [ C 9 ]

ビデオデータを符号化する方法であって、

ビデオエンコーダが、マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャ  
のピクチャ順序カウン  
ト ( P O C ) 値のための P O C リセットを示すデータを符号化する  
ことと、ここにおいて、前記第1のピクチャがアクセスユニット中に含まれ、

前記第1のピクチャの前記 P O C 値のための前記 P O C リセットを示す前記データに基  
づいて、および前記第1のピクチャを復号するより前に、コーディング順序において前記  
第1のピクチャに先行する、および前記アクセスユニット中に含まれない、復号ピクチャ

50



バッファ (DPB) 内に記憶されたすべてのピクチャを出力することとを備える、方法。

[C10]

前記DPBに記憶されたすべてのピクチャを出力することが、出力のために必要とされるとマークされた、前記DPB内のすべてのピクチャを出力することを備える、C9に記載の方法。

[C11]

すべてのピクチャを出力することが、前記第1のピクチャのビデオデータの初期のスライスのスライスヘッダを符号化した後、および前記スライスのペイロードを符号化するより前に、すべてのピクチャを出力することを備える、C9に記載の方法。

10

[C12]

前記POCリセットを示す前記データを符号化することが、ビデオデータの前記第1のピクチャのスライスのためのPOCリセットインデックスを表すデータを符号化することを備える、C9に記載の方法。

[C13]

前記POCリセットを示す前記データが、前記第1のピクチャを包含する前記アクセスユニットから開始するPOCリセット期間を示すデータを備える、C9に記載の方法。

[C14]

前記DPBのすべてのピクチャを出力した後、前記アクセスユニットの前記第1のピクチャおよび任意の他のピクチャのためのPOC値リセットを実行することをさらに備える、C9に記載の方法。

20

[C15]

前記DPBのすべてのピクチャを出力した後、前記アクセスユニットの各ピクチャを符号化するための参照ピクチャリストを構築することをさらに備える、C9に記載の方法。

[C16]

前記DPBのすべてのピクチャを出力することが、最小のピクチャ順序カウント (POC) 値を有する前記DPBのピクチャから開始して、前記DPBの前記ピクチャの前記POC値の昇順で、前記DPBの前記ピクチャを出力することと、レイヤidの昇順で、前記DPBの同じPOC値を有するピクチャを出力することと、を備える、C9に記載の方法。

30

[C17]

ビデオデータをコーディングするためのデバイスであって、

マルチレイヤビデオデータを記憶するように構成された復号ピクチャバッファ (DPB) を備えるメモリと、

前記マルチレイヤビデオデータの第1の受信レイヤの第1のピクチャのピクチャ順序カウント (POC) 値のためのPOCリセットを示すデータをコーディングすることと、  
ここにおいて、前記第1のピクチャがアクセスユニット中に含まれ、

前記第1のピクチャの前記POC値のための前記POCリセットを示す前記データに基づいて、および前記第1のピクチャを復号するより前に、コーディング順序において前記第1のピクチャに先行する、および前記アクセスユニット中に含まれない、前記DPB

40

内に記憶されたすべてのピクチャを出力することと、  
を行うように構成されたビデオコードと、  
を備える、デバイス。

[C18]

前記DPBに記憶されたすべてのピクチャを出力するために、前記ビデオコードが、出力のために必要とされるとマークされた、前記DPB内のすべてのピクチャを出力するように構成される、C17に記載のデバイス。

[C19]

すべてのピクチャを出力するために、前記ビデオコードが、前記第1のピクチャのビデオデータの初期のスライスのスライスヘッダを復号した後、および前記スライスのペイロ

50

ードを復号するより前に、すべてのピクチャを出力するように構成される、C 1 7に記載のデバイス。

[ C 2 0 ]

前記 P O C リセットを示す前記データをコーディングするために、前記ビデオコードが、ビデオデータの前記第 1 のピクチャのスライスのための P O C リセットインデックスを表すデータを復号するように構成される、C 1 7に記載のデバイス。

[ C 2 1 ]

前記 P O C リセットを示す前記データが、前記第 1 のピクチャを包含する前記アクセスユニットから開始する P O C リセット期間を示すデータを備える、C 1 7に記載のデバイス。

10

[ C 2 2 ]

前記ビデオコードが、前記 D P B のすべてのピクチャを出力した後、前記アクセスユニットの前記第 1 のピクチャおよび任意の他のピクチャのための P O C 値リセットを実行するようにさらに構成される、C 1 7に記載のデバイス。

[ C 2 3 ]

前記ビデオコードが、前記 D P B のすべてのピクチャを出力した後、前記アクセスユニットの各ピクチャを復号するための参照ピクチャリストを構築するようにさらに構成される、C 1 7に記載のデバイス。

[ C 2 4 ]

前記 D P B のすべてのピクチャを出力するために、前記ビデオコードが、最小のピクチャ順序カウンタ ( P O C ) 値を有する前記 D P B のピクチャから開始して、前記 D P B の前記ピクチャの前記 P O C 値の昇順で、前記 D P B の前記ピクチャを出力することと、レイヤ i d の昇順で、前記 D P B の同じ P O C 値を有するピクチャを出力することと、を行うように構成される、C 1 7に記載のデバイス。

20

[ C 2 5 ]

前記デバイスが、  
集積回路、  
マイクロプロセッサ、または  
ワイヤレス通信デバイス  
のうちの少なくとも 1 つを備える、C 1 7に記載のデバイス。

30

[ C 2 6 ]

ビデオデータをコーディングするためのデバイスであって、  
マルチレイヤビデオデータの第 1 の受信レイヤの第 1 のピクチャのピクチャ順序カウンタ ( P O C ) 値のための P O C リセットを示すデータをコーディングするための手段と、  
ここにおいて、前記第 1 のピクチャがアクセスユニット中に含まれ、  
前記第 1 のピクチャの前記 P O C 値のための前記 P O C リセットを示す前記データに基づいて、および前記第 1 のピクチャを復号するより前に、コーディング順序において前記第 1 のピクチャに先行する、および前記アクセスユニット中に含まれない、復号ピクチャバッファ ( D P B ) 内に記憶されたすべてのピクチャを出力するための手段と、  
を備える、デバイス。

40

[ C 2 7 ]

前記 D P B に記憶されたすべてのピクチャを出力するための前記手段が、出力のために必要とされるとマークされた、前記 D P B 内のすべてのピクチャを出力するための手段を備える、C 2 6に記載のデバイス。

[ C 2 8 ]

すべてのピクチャを出力するための前記手段が、前記第 1 のピクチャのビデオデータの初期のスライスのスライスヘッダを復号した後、および前記スライスのペイロードを復号するより前に、すべてのピクチャを出力するための手段を備える、C 2 6に記載のデバイス。

[ C 2 9 ]

50

前記 P O C リセットを示す前記データをコーディングするための前記手段が、ビデオデータの前記第 1 のピクチャのスライスのための P O C リセットインデックスを表すデータをコーディングするための手段を備える、C 2 6 に記載のデバイス。

[ C 3 0 ]

前記 D P B のすべてのピクチャを出力することが、最小のピクチャ順序カウント ( P O C ) 値を有する前記 D P B のピクチャから開始して、前記 D P B の前記ピクチャの前記 P O C 値の昇順で、前記 D P B の前記ピクチャを出力することと、レイヤ i d の昇順で、前記 D P B の同じ P O C 値を有するピクチャを出力するための手段と、を備える、C 2 6 に記載のデバイス。

[ C 3 1 ]

命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記命令が、実行されると、ビデオデータをコーディングするためのデバイスのプロセッサに、

マルチレイヤビデオデータの第 1 の受信レイヤの第 1 のピクチャのピクチャ順序カウント ( P O C ) 値のための P O C リセットを示すデータをコーディングすることと、ここにおいて、前記第 1 のピクチャがアクセスユニット中に含まれ、

前記第 1 のピクチャの前記 P O C 値のための前記 P O C リセットを示す前記データに基づいて、および前記第 1 のピクチャを復号するより前に、コーディング順序において前記第 1 のピクチャに先行する、および前記アクセスユニット中に含まれない、復号ピクチャバッファ ( D P B ) 内に記憶されたすべてのピクチャを出力することと、  
を行わせる、非一時的コンピュータ可読記憶媒体。

[ C 3 2 ]

前記 D P B に記憶されたすべてのピクチャを出力するために、前記命令が前記プロセッサに、出力のために必要とされるとマークされた、前記 D P B 内のすべてのピクチャを出力することを行わせる、C 3 1 に記載の非一時的コンピュータ可読記憶媒体。

[ C 3 3 ]

すべてのピクチャを出力するために、前記命令が前記プロセッサに、前記第 1 のピクチャのビデオデータの初期のスライスのスライスヘッダを復号した後、および前記スライスのペイロードを復号するより前に、すべてのピクチャを出力することを行わせる、C 3 1 に記載の非一時的コンピュータ可読記憶媒体。

[ C 3 4 ]

前記 P O C リセットを示す前記データをコーディングするために、前記命令が前記プロセッサに、ビデオデータの前記第 1 のピクチャのスライスのための P O C リセットインデックスを表すデータをコーディングすることを行わせる、C 3 1 に記載の非一時的コンピュータ可読記憶媒体。

[ C 3 5 ]

前記 D P B のすべてのピクチャを出力するために、前記命令が前記プロセッサに、最小のピクチャ順序カウント ( P O C ) 値を有する前記 D P B のピクチャから開始して、前記 D P B の前記ピクチャの前記 P O C 値の昇順で、前記 D P B の前記ピクチャを出力することと、レイヤ i d の昇順で、前記 D P B の同じ P O C 値を有するピクチャを出力することと、を行わせる、C 3 1 に記載の非一時的コンピュータ可読記憶媒体。

10

20

30

40

【図1】

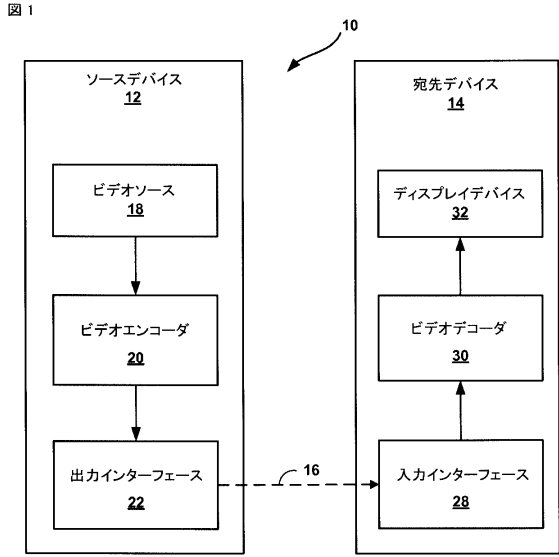


FIG. 1

【図2】

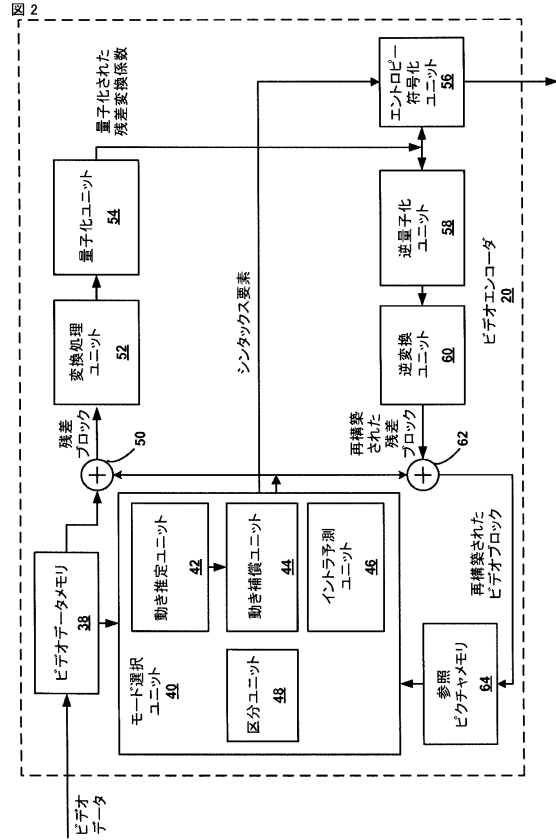


FIG. 2

【図3】

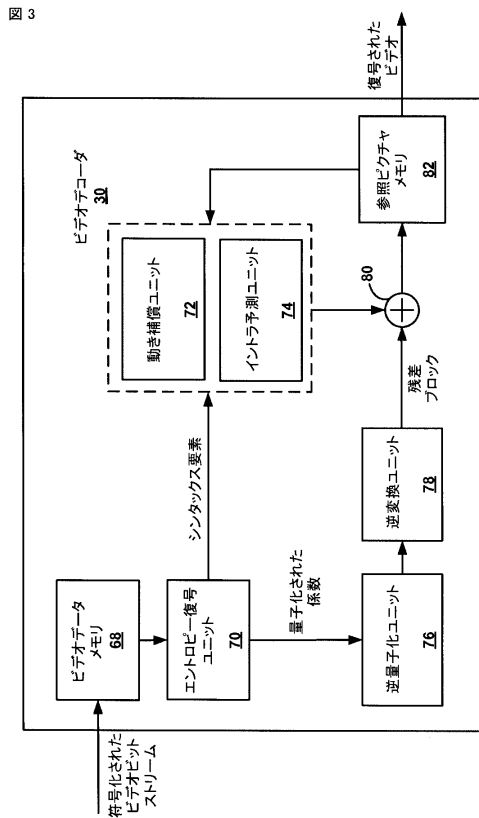


FIG. 3

【図4】

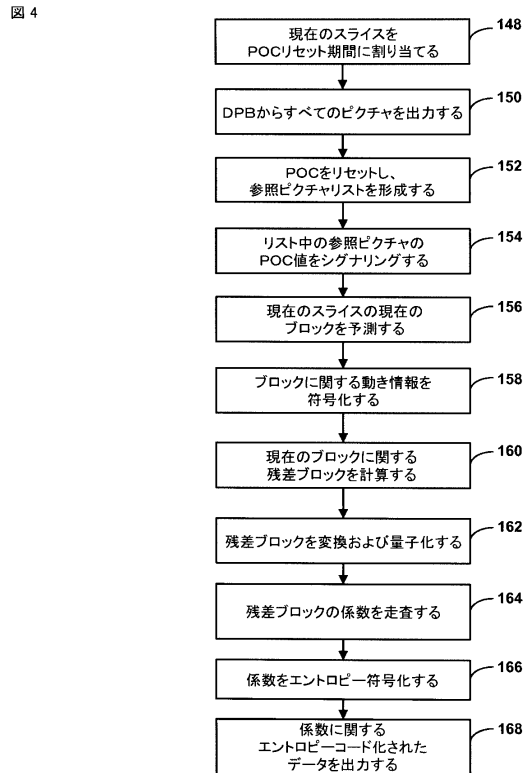


FIG. 4

【 図 5 】

図 5

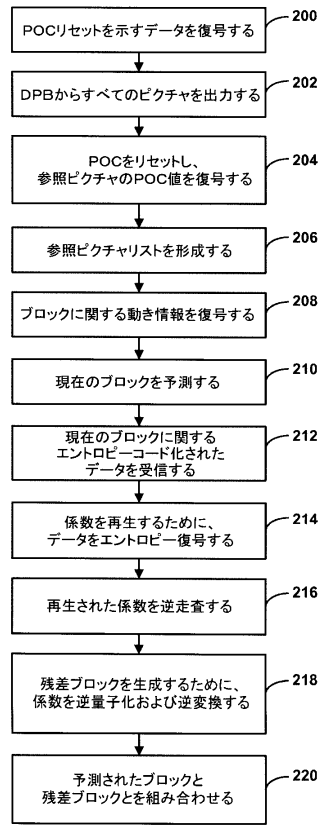


FIG. 5

【 図 6 】

図 6

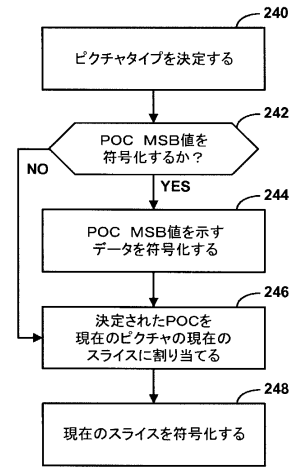


FIG. 6

【 図 7 】

図 7

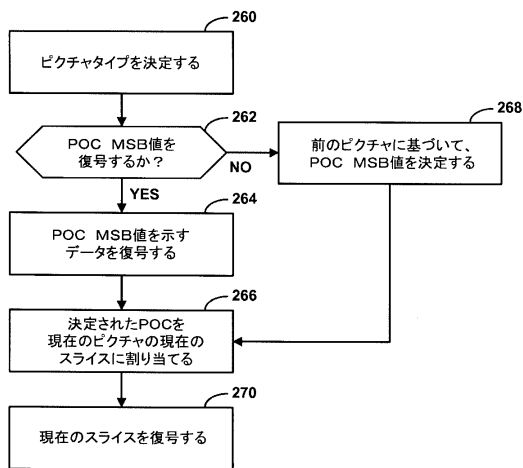


FIG. 7

## フロントページの続き

- (72)発明者 ヘンドライ、フヌ  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 ワン、イエ-クイ  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 ラマスブラモニアン、アダルシュ・クリシュナン  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

審査官 松永 隆志

- (56)参考文献 国際公開第 2 0 1 3 / 0 5 4 9 5 1 ( W O , A 1 )  
Ying Chen, et al. , MV-HEVC/SHVC HLS: Cross-layer POC alignment , Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 5th Meeting: Vienna, AT, 27 July - 2 Aug. 2013 , URL:http://phenix.it-sudparis.eu/jct2/doc\_end\_user/documents/5\_Vienna/wg11/JCT3V-E0075-v1.zip , 2 0 1 3 年 7 月 1 6 日 , JCT3V-E0075 , p.1~4  
Adarsh K. Ramasubramonian, et al. , MV-HEVC/SHVC HLS: On picture order count , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Geneva, CH, 23 Oct. - 1 Nov. 2013 , URL:http://phenix.it-sudparis.eu/jct/doc\_end\_user/documents/15\_Geneva/wg11/JCTVC-00213-v1.zip , 2 0 1 3 年 1 0 月 1 5 日 , JCTVC-00213 , p.1~10

- (58)調査した分野(Int.Cl. , D B 名)  
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8