

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7414976号  
(P7414976)

(45)発行日 令和6年1月16日(2024.1.16)

(24)登録日 令和6年1月5日(2024.1.5)

(51)国際特許分類

F I

H O 4 N 19/70 (2014.01)

H O 4 N 19/70

H O 4 N 19/50 (2014.01)

H O 4 N 19/50

請求項の数 13 (全67頁)

|                   |                               |          |                               |
|-------------------|-------------------------------|----------|-------------------------------|
| (21)出願番号          | 特願2022-521050(P2022-521050)   | (73)特許権者 | 503433420                     |
| (86)(22)出願日       | 令和2年9月30日(2020.9.30)          |          | 華為技術有限公司                      |
| (65)公表番号          | 特表2022-550989(P2022-550989 A) |          | HUAWEI TECHNOLOGIES CO., LTD. |
| (43)公表日           | 令和4年12月6日(2022.12.6)          |          | 中華人民共和國 5 1 8 1 2 9 広東省深      |
| (86)国際出願番号        | PCT/CN2020/119696             |          | ▲チェン▼市龍崗区坂田 華為總部▲ベ            |
| (87)国際公開番号        | WO2021/068854                 |          | ン▼公樓                          |
| (87)国際公開日         | 令和3年4月15日(2021.4.15)          |          | Huawei Administrat            |
| 審査請求日             | 令和4年6月6日(2022.6.6)            |          | ion Building, Banti           |
| (31)優先権主張番号       | 62/912,046                    |          | an, Longgang Distri           |
| (32)優先日           | 令和1年10月7日(2019.10.7)          |          | ct, Shenzhen, Guang           |
| (33)優先権主張国・地域又は機関 | 米国(US)                        | (74)代理人  | dong 5 1 8 1 2 9, P. R. C     |
|                   |                               |          | hina                          |
|                   |                               |          | 100107766                     |
|                   |                               |          | 弁理士 伊東 忠重                     |
|                   |                               |          | 最終頁に続く                        |

(54)【発明の名称】 エンコーダ、デコーダ、および、対応する方法

(57)【特許請求の範囲】

【請求項1】

コード化されたビデオビットストリームをデコーディングする方法であって、  
前記コード化されたビデオビットストリームから、インデックス*i*を有する第1レイヤ  
がインターレイヤ予測を使用するか否かを指定する、前記インデックス*i*を有する第1シ  
ンタックス要素を獲得するステップと、  
前記コード化されたビデオビットストリームから、1つ以上の第2レイヤに関連する1  
つ以上の第2シンタックス要素を獲得するステップであり、インデックス*i*およびインデ  
ックス*j*を有する各第2シンタックス要素は、前記インデックス*j*を有する前記第2レイ  
ヤが前記インデックス*i*を有する前記第1レイヤの直接的な参照レイヤであるか否かを指  
定し、  
前記第1シンタックス要素の値が、前記インデックス*i*を有する前記第1レイヤはイン  
ターレイヤ予測を使用することが許可されていることを指定する場合に、*j*は0から*i*−  
1までの範囲内にあり、前記インデックス*j*を有する第2シンタックス要素のうち少なく  
とも1つの第2シンタックス要素は、1であり、1に等しい前記少なくとも1つの第2シ  
ンタックス要素は、前記インデックス*j*を有する前記第2レイヤが、前記インデックス*i*  
を有する前記第1レイヤの直接的な参照レイヤであることを指定する、  
ステップと、  
前記少なくとも1つの第2シンタックス要素に関連する前記第2レイヤのピクチャを、  
参照ピクチャとして使用することにより、前記第1レイヤのピクチャについてインターレ

イヤ予測を実行するステップと、  
を含む、方法。

【請求項 2】

1 に等しい前記第 1 シンタックス要素は、前記第 1 レイヤがインターレイヤ予測を使用しないことを指定し、または、

0 に等しい前記第 1 シンタックス要素は、前記第 1 レイヤがインターレイヤ予測を使用することを許可されていることを指定する、

請求項 1 に記載の方法。

【請求項 3】

0 に等しい前記第 2 シンタックス要素は、前記第 2 シンタックス要素に関連する前記第 2 レイヤが前記第 1 レイヤの直接的な参照レイヤではないことを指定する、

請求項 1 または 2 に記載の方法。

【請求項 4】

1 つ以上の第 2 シンタックス要素を獲得する前記ステップは、前記第 1 シンタックス要素の値が、前記第 1 レイヤがインターレイヤ予測を使用することを許可されていることを指定する場合に、実行される、

請求項 1 乃至 3 いずれか一項に記載の方法。

【請求項 5】

前記方法は、さらに、

前記第 1 シンタックス要素の値が、前記第 1 レイヤがインターレイヤ予測を使用しないことを指定する場合に、前記少なくとも 1 つの第 2 シンタックス要素に関連する前記第 2 レイヤのピクチャを、参照ピクチャとして使用することなく、前記第 1 レイヤのピクチャについて予測を実行するステップ、

を含む、請求項 1 乃至 4 いずれか一項に記載の方法。

【請求項 6】

コード化されたビデオビットストリームをエンコーディングする方法であって、前記方法は、

少なくとも 1 つの第 2 レイヤが第 1 レイヤの直接的な参照レイヤであるか否かを決定するステップと、

インデックス i を有する第 1 シンタックス要素を前記コード化されたビデオビットストリームへとエンコーディングするステップであり、前記第 1 シンタックス要素は、前記インデックス i を有する第 1 レイヤがインターレイヤ予測を使用するか否かを指定する、ステップと、

前記少なくとも 1 つの第 2 レイヤに関連する 1 つ以上の第 2 シンタックス要素を前記コード化されたビデオビットストリームへとエンコーディングするステップであり、インデックス i およびインデックス j を有する各第 2 シンタックス要素は、前記インデックス j を有する第 2 レイヤが前記インデックス i を有する前記第 1 レイヤの直接的な参照レイヤであるか否かを指定し、1 に等しい前記第 2 シンタックス要素は、前記インデックス j を有する前記第 2 レイヤが、前記インデックス i を有する前記第 1 レイヤの直接的な参照レイヤであることを指定する、ステップと、を含み、

前記第 1 シンタックス要素の値が、前記インデックス i を有する前記第 1 レイヤはインターレイヤ予測を使用することが許可されていることを指定する場合に、j は 0 から i - 1 までの範囲内にあり、前記インデックス j を有する第 2 シンタックス要素のうち少なくとも 1 つの第 2 シンタックス要素は、1 である、

方法。

【請求項 7】

1 に等しい前記第 1 シンタックス要素は、前記第 1 レイヤがインターレイヤ予測を使用しないことを指定し、または、

0 に等しい前記第 1 シンタックス要素は、前記第 1 レイヤがインターレイヤ予測を使用することを許可されていることを指定する、

10

20

30

40

50

請求項 6 に記載の方法。

【請求項 8】

0 に等しい前記第 2 シンタックス要素は、前記第 2 シンタックス要素に関連する前記第 2 レイヤが前記第 1 レイヤの直接的な参照レイヤではないことを指定する、

請求項 6 または 7 に記載の方法。

【請求項 9】

前記第 1 シンタックス要素の値が、前記第 1 レイヤはインターレイヤ予測を使用することを許可されていることを指定する場合に、

前記少なくとも 1 つの第 2 レイヤに関連する 1 つ以上の第 2 シンタックス要素を前記コード化されたビデオビットストリームへとエンコーディングするステップが実行される、

請求項 6 乃至 8 いずれか一項に記載の方法。

【請求項 10】

プログラムコードを含むコンピュータプログラムであって、

前記プログラムコードがコンピュータのプロセッサによって実行されると、

請求項 1 乃至 7 いずれか一項に記載の方法を、前記コンピュータに実施させる、コンピュータプログラム。

【請求項 11】

デコーダであって、

1 つのプロセッサと、

前記プロセッサに結合され、かつ、前記プロセッサによって実行されるプログラミングを保管している、非一時的コンピュータ読取り可能な記憶媒体と、

を含み、

前記プログラミングは、前記プロセッサによって実行されると、請求項 1 乃至 5 いずれか一項に記載の方法を、前記デコーダに実施させる、ように構成されている、

デコーダ。

【請求項 12】

エンコーダであって、

1 つのプロセッサと、

前記プロセッサに結合され、かつ、前記プロセッサによって実行されるプログラミングを保管している、非一時的コンピュータ読取り可能な記憶媒体と、

を含み、

前記プログラミングは、前記プロセッサによって実行されると、請求項 6 乃至 9 いずれか一項に記載の方法を、前記エンコーダに実施させる、ように構成されている、

エンコーダ。

【請求項 13】

プログラムコードを搬送する非一時的コンピュータ読取り可能な記憶媒体であって、コンピュータ装置によって実行されると、請求項 1 乃至 7 いずれか一項に記載の方法を、前記コンピュータ装置に実施させる、

非一時的コンピュータ読取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本出願(開示)の実施形態は、一般的に、画像処理の分野に関する。そして、より詳細には、層間(inter-layer)予測に関する。

【背景技術】

【0002】

ビデオコーディング(ビデオエンコーダおよびデコーディング)は、広範囲のデジタルビデオアプリケーションにおいて使用されている。例えば、ブロードキャストデジタルTV、インターネットおよびモバイルネットワークを介したビデオ送信、ビデオチャットといったリアルタイムの会話アプリケーション、ビデオ会議、DVDおよびブルーレイ(登録商標

10

20

30

40

50

）ディスク、ビデオコンテンツの収集および編集システム、並びに、セキュリティアプリケーションのカムコーダ、である。

【0003】

比較的短いビデオを描写するために必要とされるビデオデータの量は、相当な量であり、それは、データがストリーム化される場合、または、そうでなければ、制限された帯域幅容量を有する通信ネットワークを介して通信される場合に、結果として困難を生じ得る。従って、ビデオデータは、一般的に、現代の電気通信ネットワークを介して通信される前に圧縮される。ビデオがストレージ装置に保管される場合には、また、ビデオのサイズも問題となり得るだろう。なぜなら、メモリリソースが制限され得るからである。ビデオ圧縮装置は、しばしば、伝送または保管の前に、ビデオデータをコーディングするために、送信元（source）においてソフトウェア及び／又はハードウェアを使用し、それによって、デジタルビデオ映像を表現するために必要とされるデータ量を減少させている。圧縮されたデータは、次いで、ビデオデータをデコーディングするビデオ解凍装置によって宛先（destination）において受信される。ネットワーク資源は限られており、かつ、より高いビデオ品質の要求が絶えず増加しており、画質における犠牲をほとんど払わずに圧縮比率を改善する、改良された圧縮および解凍技術が望ましい。

10

【発明の概要】

【0004】

本出願の実施形態は、独立請求項に従って、エンコーディングおよびデコーディングのための装置および方法を提供する。

20

【0005】

前記および他の目的は、独立クレームの技術的事項（subject matter）によって達成される。さらなる実施の形態は、従属請求項、明細書、および図面から明らかである。

【0006】

特定の実施形態は、添付の独立請求項において概説され、他の実施形態は、従属請求項で概説されている。

【0007】

第1態様に従って、本発明は、コード化されたビデオビットストリームをデコーディングする方法に関する。本法は、デコーディング装置によって実行される。本方法は、前記コード化されたビデオビットストリームから、第1レイヤがインターレイヤ予測を使用するか否かを指定する第1シンタックス要素を獲得するステップと、前記コード化されたビデオビットストリームから、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素を獲得するステップであり、各第2シンタックス要素は、第2レイヤが前記第1レイヤの直接的な参照レイヤであるか否かを指定し、ここで、前記第1シンタックス要素の値が、前記第1レイヤはインターレイヤ予測を使用することが許可されていることを指定する場合に、前記1つ以上の第2シンタックス要素のうち少なくとも1つの第2シンタックス要素は、第2レイヤが前記第1レイヤの直接的な参照レイヤであることを指定する値を有する、ステップと、前記少なくとも1つの第2シンタックス要素に関連する前記第2レイヤのピクチャを、参照ピクチャとして使用することにより、前記第1レイヤのピクチャについてインターレイヤ予測を実行するステップと、を含む。

30

40

【0008】

代替的に、第1シンタックス要素は、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素がコード化されたビデオビットストリーム中に存在するか否かを指定している。さらに、1に等しい第1シンタックス要素は、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素がコード化されたビデオビットストリームに存在しないことを指定するか、または、0に等しい第1シンタックス要素は、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素がコード化されたビデオビットストリームに存在することを指定する。

ここで、レイヤは、同一のレイヤインデックスを有するコード化ピクチャのシーケンスを含む。

50

ここで、1つ以上の第2レイヤのレイヤインデックスは、第1レイヤのレイヤインデックスよりも小さい。

ここで、異なる第2シンタックス要素に関連する第2レイヤは、異なるレイヤインデックスを伴う。

ここで、1つ以上の第2シンタックス要素は、1つ以上の第2レイヤと一対一の対応関係にある。

ここで、ビットストリームは1つ以上のコード化ビデオシーケンス(CVS)を形成するビットのシーケンスである。

ここで、コード化ビデオシーケンス(CVS)は、AUのシーケンスである。

ここで、コード化レイヤビデオシーケンス(CLVS)は、同じ値のnuh\_layer\_idを持つPUのシーケンスである。

ここで、アクセスユニット(AU)は、異なるレイヤに属し、かつ、DPBからの出力について同じ時間に関連付けられたコード化ピクチャを含む、PUの集合である。

ここで、ピクチャユニット(PU)は、指定された分類規則に従って互いに関連し、デコーディングの順番に連続しており、そして、正確に1つのコード化ピクチャを含む、NALユニットの集合である。

ここで、インターレイヤ参照ピクチャ(ILRP)は、現在ピクチャと同じAU内のピクチャであり、nuh\_layer\_idは現在ピクチャのnuh\_layer\_idより小さい。

ここで、SPSは、ゼロ以上の全体CLVSに対して適用するシンタックス要素を含んでいるシンタックス構造である。

ここで、レイヤAがレイヤBを参照レイヤとして使用する場合、レイヤBはレイヤAの直接的な参照レイヤである。レイヤAがレイヤBを参照レイヤとして使用し、レイヤBがレイヤCを参照レイヤとして使用するが、レイヤAがレイヤCを参照レイヤとして使用しない場合、レイヤCはレイヤAの直接的な参照レイヤではない。

【0009】

そうした第1態様に従った、方法の可能な実装形態において、ここでは、1に等しい前記第1シンタックス要素は、前記第1レイヤがインターレイヤ予測を使用しないことを指定し、または、0に等しい前記第1シンタックス要素は、前記第1レイヤがインターレイヤ予測を使用することを許可されていることを指定する。

【0010】

第1態様の任意の先行する実装又はそうした第1態様に従った、方法の可能な実装形態において、ここでは、0に等しい前記第2シンタックス要素は、前記第2シンタックス要素に関連する前記第2レイヤが前記第1レイヤの直接的な参照レイヤではないことを指定し、または、1に等しい前記第2シンタックス要素は、前記第2シンタックス要素に関連する前記第2レイヤが前記第1レイヤの直接的な参照レイヤであることを指定する。

【0011】

第1態様の任意の先行する実装又はそうした第1態様に従った、方法の可能な実装形態において、ここでは、1つ以上の第2シンタックス要素を獲得する前記ステップは、前記第1シンタックス要素の値が、前記第1レイヤがインターレイヤ予測を使用することを許可されていることを指定する場合に、実行される。

【0012】

第1態様の任意の先行する実装又はそうした第1態様に従った、方法の可能な実装形態において、ここでは、前記方法は、さらに、前記第1シンタックス要素の値が、前記第1レイヤがインターレイヤ予測を使用しないことを指定する場合に、前記少なくとも1つの第2シンタックス要素に関連する前記レイヤのピクチャを、参照ピクチャとして使用することなく、前記第1レイヤのピクチャについて予測を実行するステップ、を含む。

【0013】

第2態様に従って、本発明は、コード化されたビデオビットストリームをエンコーディングする方法を提供する。本方法は、エンコーダによって実行される。本方法は、少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを決定するステッ

10

20

30

40

50

プと、シンタックス要素を前記コード化されたビデオビットストリームへとエンコーディングするステップであり、前記シンタックス要素は、前記第1レイヤがインターレイヤ予測を使用するか否かを指定する、ステップと、を含み、ここで、前記少なくとも1つの第2レイヤのどれもが前記第1レイヤの直接的な参照レイヤでない場合に、前記シンタックス要素の値は、前記第1レイヤがインターレイヤ予測を使用しないことを規定する。

【0014】

少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを決定するステップは、第1速度歪みコスト (rate distortion cost) が第2速度歪みコスト以下であると決定することに基づいて、第2レイヤが第1レイヤの直接的な参照レイヤであると決定するステップと、第1速度歪みコストが第2速度歪みコスト以上であると決定することに基づいて、第2レイヤが第1レイヤの直接的な参照レイヤではないと決定するステップ、を含み、ここで、第1速度歪みコストは、第1レイヤの直接的な参照レイヤとして第2レイヤを使用することによるコストであり、第2速度歪みコストは、第1レイヤの直接的な参照レイヤとして第2レイヤを使用しないコストである。

【0015】

そうした第2態様に従った、方法の可能な実装形態において、ここでは、前記シンタックス要素の前記値は、前記少なくとも1つの第2レイヤが前記第1レイヤの直接的な参照レイヤである場合に、前記第1レイヤがインターレイヤ予測を使用することを許可されることを指定する。

【0016】

第3態様に従って、本発明は、コード化されたビデオビットストリームをデコーディングするための装置に関する。本装置は、獲得ユニットおよび予測ユニットを含む。獲得ユニットは、コード化されたビデオビットストリームから、第1レイヤがインターレイヤ予測を使用するか否かを指定している第1シンタックス要素を獲得するように構成されている。獲得ユニットは、さらに、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素を獲得するように構成されており、各第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを指定する。ここで、第1レイヤはインターレイヤ予測を使用することが許されると第1シンタックス要素の値が指定する場合に、1つ以上の第2シンタックス要素のうち少なくとも1つの第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであることを指定する値を有する。そして、予測ユニットは、少なくとも1つの第2シンタックス要素に関連する第2レイヤのピクチャを参照ピクチャとして使用することによって、第1レイヤのピクチャについてインターレイヤ予測を実行するように構成されている。

【0017】

そうした第3態様に従った、方法の可能な実装形態において、ここでは、1に等しい前記第1シンタックス要素は、前記第1レイヤがインターレイヤ予測を使用しないことを指定し、または、0に等しい前記第1シンタックス要素は、前記第1レイヤがインターレイヤ予測を使用することを許可されていることを指定する。

【0018】

第3態様の任意の先行する実装又はそうした第3態様に従った、方法の可能な実装形態において、ここでは、0に等しい前記第2シンタックス要素は、前記第2シンタックス要素に関連する前記第2レイヤが前記第1レイヤの直接的な参照レイヤではないことを指定し、または、1に等しい前記第2シンタックス要素は、前記第2シンタックス要素に関連する前記第2レイヤが前記第1レイヤの直接的な参照レイヤであることを指定する。

【0019】

第3態様の任意の先行する実装又はそうした第3態様に従った、方法の可能な実装形態において、ここでは、1つ以上の第2シンタックス要素を獲得するように構成されている予測ユニットは、前記第1シンタックス要素の値が、前記第1レイヤがインターレイヤ予測を使用することを許可されていることを指定する場合に、実行される。

【0020】

第4態様に従って、本発明は、コード化されたビデオビットストリームをエンコーディングするための装置に関する。本装置は、決定ユニットおよびエンコーディングユニットを含む。決定ユニットは、少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを決定するように構成されている。エンコーディングユニットは、コード化されたビデオビットストリームへとシンタックス要素をエンコーディングするように構成されている。ここで、シンタックス要素は、第1レイヤがインターレイヤ予測を使用するか否かを指定する。ここで、少なくとも1つの第2レイヤのいずれも第1レイヤの直接的な参照レイヤではない場合、シンタックス要素の値は、第1レイヤがインターレイヤ予測を使用しないことを指定する。

【0021】

10

そうした第4態様に従った、方法の可能な実装形態において、ここでは、少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤである場合、前記シンタックス要素の値は、前記第1レイヤがインターレイヤ予測を使用することを許可されていることを指定する。

【0022】

本発明の第1態様の方法は、本発明の第3態様の装置によって実施することができる。本発明の第3態様に従った方法のさらなる特徴および実装形態は、本発明の第1態様による装置の特徴および実装形態に対応する。

【0023】

本発明の第2態様の方法は、本発明の第4態様の装置によって実施することができる。本発明の第4態様に従った方法のさらなる特徴および実装形態は、本発明の第2態様による装置の特徴および実装形態に対応する。

20

【0024】

第2態様に従った方法は、第1態様に従った第1装置の実装形態に対応する実装形態へと拡張され得る。それ故に、方法の実装形態は、第1装置の対応する実装形態の特徴を含む。

【0025】

第2態様に従った方法の利点は、第1態様に従った第1装置の対応する実装形態の利点と同じである。

【0026】

30

第5態様に従って、本発明は、ビデオストリームをデコーディングするための装置に関し、プロセッサおよびメモリを含む。メモリは、第1態様に従った方法をプロセッサに実行させる命令を保管している。

【0027】

第6態様に従って、本発明は、ビデオストリームをエンコーディングするための装置に関し、プロセッサおよびメモリを含む。メモリは、第2態様に従った方法をプロセッサに実行させる命令を保管している。

【0028】

第7態様に従って、実行されると、1つ以上のプロセッサにビデオデータをコード化するように構成された命令を保管したコンピュータ読取り可能なストレージ媒体が提案される。命令は、1つ以上のプロセッサに、第1態様または第2態様、もしくは、第1態様または第2態様の任意の可能な実施形態に従って、方法を実行させる。

40

【0029】

第8態様に従って、本発明は、コンピュータ上で実行されると、第1態様または第2態様、もしくは、第1態様または第2態様の任意の可能な実施形態に従って、方法を実行するためのプログラムコードを含むコンピュータプログラムに関する。

【0030】

第9態様に従って、本発明は、装置によってデコーディングされるコード化されたビットストリームを含む、非一時なストレージ媒体に関する。前記ビットストリームは、ビデオ信号または画像信号のフレームを複数のブロックへと分割することによって生成されて

50

おり、かつ、複数のシンタックス要素を含んでいる。ここで、前記複数のシンタックス要素は、第1レイヤがインターレイヤ予測を使用するか否かを指定している第1シンタックス要素、および、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素を含み、各第2シンタックス要素は、第2レイヤが前記第1レイヤの直接的な参照レイヤであるか否かを指定する。ここで、第1レイヤはインターレイヤ予測を使用することが許されると第1シンタックス要素の値が指定する場合に、1つ以上の第2シンタックス要素のうち少なくとも1つの第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであることを指定する値を有する。

#### 【0031】

1つ以上の実施形態の詳細は、添付の図面および以下の説明において明らかにされる。他の特徴、目的、および利点は、明細書、図面、および特許請求の範囲から明らかであろう。

#### 【0032】

さらに、以下の実施形態が提供される。

#### 【0033】

一つの実施形態において、コード化されたビデオビットストリームをデコーディングする方法が提供される。本方法は、以下を含む。

インデックス*i*を有するレイヤがインターレイヤ予測を使用するか否かを指定している、第1シンタックス要素を解析するステップであり、*i*は整数であり、かつ、*i*は0より大きい、ステップ。

第1条件が満たされた場合、インデックス*j*を有するレイヤがインデックス*i*を有するレイヤの直接的な参照レイヤであるか否かを指定している、第2シンタックス要素を解析するステップであり、ここで、第1条件は、第1シンタックス要素が、インデックス*i*を有するレイヤはインターレイヤ予測を使用することができ、*j*は*i*-1に等しく、かつ、*j*より小さいインデックスを有するレイヤのうちいずれか1つは、インデックス*i*を有するレイヤの直接的な参照レイヤではないこと、を指定すること、を含むこと。

第2シンタックス要素の値に基づいて、インデックス*i*を有するレイヤのピクチャを予測するステップ。

#### 【0034】

一つの実施形態において、第1シンタックス要素が、インデックス*i*を有するレイヤはインターレイヤ予測を使用し得ることを指定した場合、`vps_direct_direct_direct_dependency_flag[i][k]`の合計は0より大きく、*k*は0から*i*-1までの範囲内の全ての整数である。ここで、1に等しい`vps_direct_direct_direct_dependency_flag`は、インデックス*k*を有するレイヤが、インデックス*i*を有するレイヤの直接的な参照レイヤであることを指定し、0に等しい`vps_direct_direct_direct_dependency_flag`は、インデックス*k*を有するレイヤが、インデックス*i*を有するレイヤの直接的な参照レイヤではないことを指定する。

#### 【0035】

一つの実施形態において、第1シンタックス要素が、インデックス*i*を有するレイヤはインターレイヤ予測を使用し得ることを指定した場合、`vps_direct_direct_direct_dependency_flag[i][k]`の少なくとも1つの値は1に等しい。ここで、*k*は整数であり、かつ、*k*は0から*i*-1の範囲内にある。ここで、1に等しい`vps_direct_direct_direct_dependency_flag`は、インデックス*k*を有するレイヤが、インデックス*i*を有するレイヤの直接的な参照レイヤであることを指定し、0に等しい`vps_direct_direct_direct_dependency_flag`は、インデックス*k*を有するレイヤが、インデックス*i*を有するレイヤの直接的な参照レイヤではないことを指定する。

#### 【0036】

一つの実施形態において、インデックス*i*を有するレイヤのピクチャは、インデックス*i*を有するレイヤ内のピクチャ、または、インデックス*i*を有するレイヤに関連するピクチャを含む。

#### 【0037】

一つの実施形態において、コード化されたビデオビットストリームをデコーディングする方法が提供される。本方法は、以下を含む。

インデックス $i$ を有するレイヤがインターレイヤ予測を使用するか否かを指定している、第1シンタックス要素を解析するステップであり、 $i$ は整数であり、かつ、 $i$ は0より大きい、ステップ。

条件が満たされた場合、インデックス $j$ を有するレイヤの画像を、インデックス $i$ を有するレイヤの直接的な参照レイヤとして使用して、インデックス $i$ を有するレイヤのピクチャを予測するステップであり、ここで、 $j$ は整数であり、かつ、 $j$ は $i-1$ に等しく、ここで、本条件は、インデックス $i$ を有するレイヤがインターレイヤ予測を使用し得ることを指定するシンタックス要素を含む、ステップ。

#### 【0038】

一つの実施形態において、インデックス $i$ を有するレイヤのピクチャは、インデックス $i$ を有するレイヤ内のピクチャ、またはインデックス $i$ を有するレイヤに関連するピクチャを含む。

#### 【0039】

一つの実施形態において、コード化されたビデオビットストリームをデコーディングする方法が提供される。本方法は、以下を含む。

少なくとも1つの長期参照ピクチャ(LTRP)が、コード化ビデオシーケンス(CVS)内のコード化ピクチャのインター予測のために使用されるか否かを指定しているシンタックス要素を解析するステップ。ここで、少なくとも1つのLTRPの各ピクチャは「長期参照用(“used for long-term reference”)」としてマーク付けされるが、インターレイヤ参照ピクチャ(ILRP)ではない。

シンタックス要素の値に基づいて、CVS内の1つ以上のコード化ピクチャを予測するステップ。

#### 【0040】

一つの実施形態において、コード化されたビデオビットストリームをデコーディングする方法が提供される。本方法は、以下を含む。

条件が満たされているか否かを決定するステップであり、ここで、全条件は、現在レイヤのレイヤインデックスがプリセット値よりも大きいことを含む、ステップ。

条件が満たされる場合、少なくとも1つのインターレイヤ参照ピクチャ(ILRP)が、コード化ビデオシーケンス(CVS)内の任意のコード化ピクチャのインター予測のために使用されるか否かを指定している第1シンタックス要素を解析するステップ。

第1シンタックス要素の値に基づいて、CVS内の1つ以上のコード化ピクチャを予測するステップ。

#### 【0041】

一つの実施形態において、プリセット値は0である。

#### 【0042】

一つの実施形態において、前記条件は、第2シンタックス要素(例えば、sps\_video\_parameter\_set\_id)が0より大きいことを、さらに、含む。

#### 【0043】

一つの実施形態において、コード化されたビデオビットストリームをデコーディングする方法が提供される。本方法は、以下を含む。

条件が満たされているか否かを決定するステップであり、ここで、全条件は、現在レイヤのレイヤインデックスが、プリセット値より大きく、かつ、参照ピクチャリスト構造内の現在エントリがILRPエントリであることを含む、ステップ。

条件が満たされる場合、現在レイヤの直接的な従属レイヤのリストに対するインデックスを指定しているシンタックス要素を解析するステップ。

直接的な従属レイヤのリストに対するインデックスを使用して現在のILRPが獲得される、現在エントリの参照ピクチャリスト構造に基づいて、CVS内の1つ以上のコード化ピクチャを予測するステップ。

10

20

30

40

50

## 【0044】

一つの実施形態において、プリセット値は1である。

## 【0045】

一つの実施形態においては、請求項1乃至12のいずれか一項に記載の方法を実行するための処理回路を備える、エンコーダ(20)が提供される。

## 【0046】

一つの実施形態においては、請求項1乃至12のいずれか一項に記載の方法を実行するための処理回路を備える、デコーダ(30)が提供される。

## 【0047】

一つの実施形態においては、コンピュータまたはプロセッサ上で実行されたときに、先行する請求項のいずれか一項に記載の方法を実行するためのプログラムコードを含む、コンピュータプログラム製品が提供される。

10

## 【0048】

一つの実施形態においては、以下を含む、デコーダが提供される。

1つ以上のプロセッサ、および、

前記プロセッサに結合され、かつ、前記プロセッサによる実行のためのプログラミングを保管している非一時的コンピュータ読取り可能な記憶媒体であり、前記プログラミングは、前記プロセッサによって実行されると、先行する請求項のいずれか一項に記載の方法を実行するように前記デコーダを構成する、非一時的コンピュータ読取り可能な記憶媒体。

## 【0049】

20

一つの実施形態においては、以下を含む、エンコーダが提供される。

1つ以上のプロセッサ、および、

前記プロセッサに結合され、かつ、前記プロセッサによる実行のためのプログラミングを保管している非一時的コンピュータ読取り可能な記憶媒体であり、前記プログラミングは、前記プロセッサによって実行されると、先行する請求項のいずれか一項に記載の方法を実行するように前記エンコーダを構成する、非一時的コンピュータ読取り可能な記憶媒体。

## 【0050】

一つの実施形態においては、プログラムコードを搬送している、非一時的コンピュータ読取り可能な媒体が提供される。プログラムコードは、コンピュータ装置によって実行されると、先行する請求項のいずれか一項に記載の方法を、コンピュータ装置に実行させる。

30

## 【図面の簡単な説明】

## 【0051】

本発明の以下の実施形態においては、添付の図および図面を参照して、より詳細に説明されている。

【図1A】図1Aは、本発明の実施形態を実施するように構成された、ビデオコーディングシステムに係る一つ例を示しているブロック図である。

【図1B】図1Bは、本発明の実施形態を実施するように構成された、ビデオコーディングシステムに係る別の例を示しているブロック図である。

【図2】図2は、本発明の実施形態を実施するように構成された、ビデオエンコーダに係る一つ例を示しているブロック図である。

40

【図3】図3は、本発明の実施形態を実施するように構成された、ビデオデコーダに係る一つの例示的な構成を示しているブロック図である。

【図4】図4は、エンコーディング装置またはデコーディング装置に係る一つの例を示しているブロック図である。

【図5】図5は、エンコーディング装置またはデコーディング装置に係る別の例を示しているブロック図である。

【図6】図6は、2層を伴うスケーラブルなコーディングを示しているブロック図である。

【図7】図7は、コンテンツ配信サービスを実現する、コンテンツ供給システム3100に係る例示的な構成を示しているブロック図である。

【図8】図8は、端末装置の一つの例に係る構成を示しているブロック図である。

50

【図 9】図9は、一つの実施形態に従った、デコーディング方法に係るフローチャートを示している。

【図 10】図10は、一つの実施形態に従った、エンコーディング方法に係るフローチャートを示している。

【図 11】図11は、一つの実施形態に従った、デコーダの概略図である。

【図 12】図12は、一つの実施形態に従った、エンコーダの概略図である。

【0052】

以下の同一の参照符号は、そうでないものと明示的に指定されていない場合には、同一の、または、少なくとも機能的に同等の特徴を参照する。

【発明を実施するための形態】

【0053】

以下の説明では、添付の図面を参照する。図面は、本開示の一部を形成し、そして、例示として、本発明の実施形態の特定の態様または本発明の実施形態が使用され得る特定の態様を示している。本発明の実施形態は、他の態様において使用することができ、そして、図に示されていない構造的または論理的な変形を含むことが理解される。以下の詳細な説明は、従って、限定的な意味で解釈されるべきではなく、そして、本発明の範囲は、添付の請求項によって定義されるものである。

【0054】

例えば、説明される方法に関連する開示は、本方法を実行するように構成された対応する装置またはシステムについても、また、真であり、そして、その逆もどうようであり獲得するステップが理解される。例えば、1つ又は複数の特定の方法ステップが説明されている場合、たとえそうした1つ又は複数のユニットが明示的に図面において明示的に説明または示されていなくても、説明される1つ又は複数の方法ステップを実行するために、対応するデバイスは、1つ又は複数のユニット(例えば、機能ユニット)を含み得る(例えば、1つ又は複数のステップを実行する1つのユニット、または、複数のステップそれぞれを実行する複数のユニット)。一方で、例えば、1つ又は複数のユニット、例えば、機能ユニット、に基づいて特定の装置が説明される場合、対応する方法は、たとえそうした1つ又は複数のユニットが明示的に図面において明示的に説明または示されていなくても、対応する方法は、1つ又は複数のユニットの機能性を実行するための1つのステップを含み得る(例えば、1つ又は複数のユニットの機能性を実行する1つのステップ、または、複数のユニットのうち1つ又は複数の機能性を実行する複数のステップ)。さらに、ここにおいて説明される種々の例示的な実施形態及び／又は態様の特徴は、特にそうでないものと注記されていない場合には、互いに組み合わせられ獲得するステップが理解される。

【0055】

ビデオコーディングは、典型的には、ビデオまたはビデオシーケンスを形成する、一連の画像を処理することを参照する。用語「画像(“picture”)」の代わりに、用語「フレーム(“frame”)」または「イメージ(“image”)」が、ビデオコーディングの分野において同義語として使用され得る。ビデオコーディング(または、一般的にコーディング)は、2つの部分のビデオコーディングおよびビデオデコーディングを含む。ビデオコーディングは、送信元側で実行され、典型的には、(より効率的な保管及び／又は伝送のために)ビデオ映像(video image)を表現するために必要とされるデータの量を減少させるために、元のビデオ映像を処理することを含む。ビデオデコーディングは、宛先側で実行され、そして、典型的には、ビデオ映像を再構成するためにエンコーダと比較して逆処理を含む。ビデオ映像(または、一般的にピクチャ)の「コーディング(“coding”)」を参照する実施形態は、ビデオ映像またはそれぞれのビデオシーケンスの「エンコーディング(“encoding”)」または「デコーディング(“decoding”)」に関連するものと理解される。エンコーディング部とデコーディングの組み合わせは、CODEC(Coding and Decoding)としても、また、参照される。

【0056】

損失のないビデオコーディングの場合、元のビデオ映像を再構成することができる。す

10

20

30

40

50

なわち、再構成されたビデオ映像は、元のビデオ映像と同じ品質を有する(保管または伝送の最中に伝送損失またはその他のデータ損失がないと仮定している)。不可逆(lossy)ビデオコーディングの場合、ビデオ映像を表すデータの量を減らすために、例えば量子化によって、さらなる圧縮が実行され、それは、デコーダで完全には再構成することができない。すなわち、再構成されたビデオ映像の品質は、元のビデオ映像の品質と比較して、より低いか、または、より悪い。

【0057】

いくつかのビデオコーディング標準は、「不可逆ハイブリッドビデオコーデック(“lossy hybrid video codecs”)」(すなわち、サンプル領域における空間的および時間的予測と、変換領域において量子化を適用するための2D変換コーディングを組み合わせる)のグループに属している。ビデオシーケンスの各ピクチャは、典型的に、重複しないブロックのセットへと分割され、そして、コーディングは、典型的に、ブロックレベルで実行される。別の言葉で言えば、エンコーダで、ビデオは、典型的には、ブロック(ビデオブロック)レベルで、処理、すなわち、エンコーディングされる。例えば、予測ブロックを生成するために空間的(イントラピクチャ)予測及び／又は時間的(インターピクチャ)予測を使用すること、残差ブロック(residual block)を獲得するために現在ブロック(現在処理されている／処理されるべきブロック)から予測ブロックを減算すること、送信されるべきデータの量を低減するために残差ブロックを変換し、かつ、変換領域内の残差ブロックを量子化すること、による。一方で、デコーダでは、表現のために現在ブロックを再構成するように、エンコーディングされたブロックまたは圧縮されたブロックに対して、エンコーダと比較して逆の処理が適用される。さらに、エンコーダは、両者が同一の予測(例えば、イントラ予測およびインター予測)を生成し、かつ／あるいは、処理、すなわち、コーディング、のために、後続のブロックを再構成するように、デコーダ処理ループを複製する。

【0058】

ビデオコーディングシステム10に係る以下の実施形態において、図1から図3までに基づいて、ビデオエンコーダ20およびビデオデコーダ30が説明される。

【0059】

図1Aは、例示的なコーディングシステム10、例えば、この本出願の技術を利用することができるビデオコーディングシステム10(または、ショートコーディングシステム10)を示している概略ブロック図である。ビデオコーディングシステム10のビデオエンコーダ20(または、ショートエンコーダ20)およびビデオデコーダ30(または、ショートデコーダ30)は、本出願において説明される種々の実施例に従って技術を実行するように構成され得る装置の例を表している。

【0060】

図1Aに示されるように、コーディングシステム10は、エンコーディングされた映像データ12を提供するように構成された送信元デバイス12を含み、例えば、エンコーディングされた映像データ21をデコーディングするために、宛先デバイス14に対して提供する。

【0061】

送信元デバイス12は、エンコーダ20を含み、そして、加えて、すなわち、任意的に、ピクチャソース16、プリプロセッサ(または、前処理ユニット)18、例えば、映像プリプロセッサ18、および、通信インターフェイスまたは通信ユニット22を含み得る。

【0062】

ピクチャソース16は、以下を含んでよく、または、任意の種類の映像キャプチャ装置、例えば現実世界映像をキャプチャするためのカメラ、及び／又は、任意の種類の映像生成装置、例えばコンピュータアニメーション映像を生成するためのコンピュータグラフィックスプロセッサ、または、現実世界映像、コンピュータ生成映像(例えば、スクリーンコンテンツ、仮想現実(VR)映像)、及び／又は、それらの任意の組み合わせ(例えば、拡張現実(AR)映像)を獲得かつ／あるいは提供するための任意の種類の他の装置、であってよい。ピクチャソースは、上記の映像のいずれかを保管する任意の種類のメモリまたはストレージ装置であってよい。

10

20

30

40

50

## 【0063】

プリプロセッサ18および前処理ユニット18によって実行される処理と区別して、映像または映像データ17は、生の（raw）映像または生の映像データ17としても、また、参照される。

## 【0064】

プリプロセッサ18は、（生の）映像データ17を受け取り、映像データ17上で前処理を行い、前処理された映像データ19または前処理された映像データ19を獲得するように構成されている。プリプロセッサ18によって実行される前処理は、例えば、トリミング、カラーフォーマット変換（例えば、RGBからYCbCrへ）、カラー補正、またはノイズ除去を含み得る。前処理ユニット18は、任意的なコンポーネントであってよいことが理解され得る。

10

## 【0065】

ビデオエンコーダ20は、前処理された映像データ19を受信し、そして、エンコーディングされた映像データ21を提供するように構成されている（例えば、図2に基づいて、さらなる詳細が以下で説明される）。送信元デバイス12の通信インターフェイス22は、エンコーディングされた映像データ21を受信し、そして、保管または直接的な再構成のために、通信チャンネル13を介して、エンコーディングされた映像データ21（または、その任意のさらに処理されたバージョン）を他の装置、例えば宛先デバイス14、または任意の他の装置に送信するように構成され得る。

## 【0066】

宛先デバイス14は、デコーダ30（例えば、ビデオデコーダ30）を含み、そして、加えて、通信インターフェイスまたは通信ユニット28、ポストプロセッサ32（または、後処理ユニット32）、および、表示装置34を含んでよい。

20

## 【0067】

宛先デバイス14の通信インターフェイス28は、エンコーディングされた映像データ21（または、その任意のさらに処理されたバージョン）を、例えば、送信元デバイス12から直接的に、または、他の任意のソース、例えばストレージ装置、例えばエンコーディングされた映像データストレージ装置、から受信し、そして、デコーダ30に対してエンコーディングされた映像データ21を提供するように構成されている。

## 【0068】

通信インターフェイス22および通信インターフェイス28は、送信元デバイス12と宛先デバイス14との間の直接的な通信リンク、例えば直接的な有線または無線接続、を介して、もしくは、任意の種類のネットワーク、例えば有線または無線ネットワーク、もしくは、それらの任意の組み合わせ、または任意の種類のプライベートおよびパブリックネットワーク、またはそれらの任意の種類の組み合わせを介して、エンコーディングされた映像データ21またはエンコーディングされたデータ13を送信または受信するように構成され得る。

30

## 【0069】

通信インターフェイス22は、例えば、エンコーディングされた映像データ21を適切なフォーマット、例えばパケットへとパッケージし、かつ／あるいは、任意の種類の送信エンコーディング、もしくは、通信リンクまたは通信ネットワークを介した送信のための処理を使用して、エンコーディングされた映像データを処理するように構成され得る。

40

## 【0070】

通信インターフェイス22のカウンタパートを形成している、通信インターフェイス28は、例えば、送信データを受信し、そして、任意の種類の対応する送信デコーディング、または、処理及び／又はパッケージ解除（de-packaging）を使用して送信データを処理するように構成することができ、エンコーディングされた映像データ21を獲得する。

## 【0071】

通信インターフェイス22および通信インターフェイス28、両方は、送信元デバイス12から宛先デバイス14に対して指し示す図1Aにおける通信チャンネル13の矢印によって示されるように、一方向通信インターフェイスそして、または、双方向通信インターフェイス

50

として構成され得る。そして、例えば、通信リンク、及び／又は、データ伝送、例えばエンコーディングされた映像データ伝送、に関連する任意の他の情報を確認応答(acknowledge)し、かつ、交換するために、例えば、メッセージを送し、かつ、受信するように構成され得る。

【0072】

デコーダ30は、エンコーディングされた映像データ21を受信し、そして、デコーディングされた映像データ31またはデコーディングされたピクチャ31を提供するように構成されている(例えば、図3または図5に基づいて、さらなる詳細が以下で説明される)。

【0073】

宛先デバイス14のポストプロセッサ32は、デコーディングされた映像データ31(再構成された映像データとしても、また、呼ばれるもの)、例えばデコーディングされたピクチャ31を後処理するように構成されており、後処理された映像データ33、例えば後処理された映像33、を獲得する。後処理ユニット32によって実行される後処理は、例えば、色フォーマット変換(例えば、YCbCrからRGBへの変換)、色補正、トリミング、または再サンプリング、もしくは、例えば、表示装置34による、表示のために、例えば、デコーディングされた映像データ31を準備するための任意の他の処理を含み得る。

【0074】

宛先デバイス14の表示装置34は、例えばユーザまたはビューアに対して、画像を表示するために、後処理された映像データ33を受信するように構成されている。表示装置34は、再構成された画像を表現するための任意の種類のディスプレイ、例えば、統合された又は外部のディスプレイもしくはモニタ、であってよく、または、含んでよい。ディスプレイは、例えば、液晶ディスプレイ(LCD)、有機発光ダイオードディスプレイ(OLED)、プラズマディスプレイ、プロジェクタ、マイクロLEDディスプレイ、シリコン上の液晶(LCoS)、デジタル光プロセッサ(DLP)、または、任意の種類の他のディスプレイ、を含み得る。

【0075】

図1Aは、送信元デバイス12および宛先デバイス14を、別個の装置として描いているが、装置の実施形態は、また、両方または両方の機能性、送信元デバイス12または対応する機能性、および、宛先デバイス14または対応する機能性を含んでもよい。そうした実施形態において、送信元デバイス12または対応する機能性、および、宛先デバイス14または対応する機能性は、同一のハードウェア及び／又はソフトウェアを使用して、もしくは、別個のハードウェア及び／又はソフトウェア、またはそれらの任意の組み合わせによって、実装することができる。

【0076】

本説明に基づいて当業者には明らかなように、図1Aに示されるような送信元デバイス12及び／又は宛先デバイス14内の異なるユニットの機能性または機能性に係る存在および(正確な)分割は、実際のデバイスおよびアプリケーションに依存して変化し得る。

【0077】

エンコーダ20(例えば、ビデオエンコーダ20)、デコーダ30(例えば、ビデオデコーダ30)、または、エンコーダ20およびデコーダ30の両方は、図1Bに示されるような処理回路を介して実施され得る。1つ以上のマイクロプロセッサ、デジタル信号プロセッサ(DSP)、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)、離散論理、ハードウェア、専用のビデオコーディング、または、それらの任意の組み合わせ、といったものである。エンコーダ20は、図2のエンコーダ20、及び／又は、ここにおいて説明される任意の他のエンコーダシステムまたはサブシステムに関して説明されるように、種々のモジュールを具体化するために、処理回路46を介して実施され得る。デコーダ30は、図3のデコーダ30、及び／又は、ここにおいて説明される任意の他のデコーダシステムまたはサブシステムに関して説明されるように、種々のモジュールを具体化するために、処理回路46を介して実施され得る。処理回路は、後述されるように、種々の動作を実行するように構成され得る。図5に示されるように、技術が部分的にソフトウェアで実装

される場合、デバイスは、適切な非一時的コンピュータで読取り可能なストレージ媒体にソフトウェアのための命令を保管することができ、そして、この開示の技術を実行するために1つ以上のプロセッサを使用して、ハードウェアで命令を実行することができる。ビデオエンコーダ20およびビデオデコーダ30のいずれかは、例えば、図1Bに示されるように、単一の装置において組み合わされたエンコーダ／デコーダ(CODEC)の一部として統合され得る。

#### 【0078】

送信元デバイス12および宛先デバイス14は、例えば、ノートブックまたはラップトップコンピュータ、携帯電話、スマートフォン、タブレットまたはタブレットコンピュータ、カメラ、デスクトップコンピュータ、セットトップボックス、テレビ、表示装置、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装置(コンテンツサービスサーバまたはコンテンツ配信サーバといったもの)、ブロードキャスト受信機デバイス、ブロードキャスト送信機デバイスなど、任意の種類のハンドヘルド装置または固定装置を含む、任意の広範囲の装置を含むことができ、そして、オペレーティングシステムを使用しないか、または、任意の種類のオペレーティングシステムを使用することができる。場合によっては、送信元デバイス12および宛先デバイス14は、無線通信のために装備されてよい。従って、送信元デバイス12および宛先デバイス14は、無線通信装置であり得る。

#### 【0079】

場合によっては、図1Aに示されたビデオコーディングシステム10は、単なる一つの例であり、そして、本出願の技術は、必ずしもエンコーディング装置とデコーディング装置との間のデータ通信を含まない、ビデオコーディング設定(例えば、ビデオエンコーディングまたはビデオデコーディング)に対して適用され得る。他の例において、データは、ローカルメモリから検索され、ネットワークを介してストリーミングされる、等である。ビデオコーディング装置は、データをエンコーディングし、かつ、メモリに保管することができ、かつ／あるいは、ビデオデコーディング装置は、メモリからデータを検索し、かつ、デコーディングすることができる。いくつかの例では、エンコーディングおよびデコーディングは、相互に通信しないが、単に、メモリにデータをエンコーディングし、かつ、メモリに保管し、かつ／あるいは、メモリからデータを検索し、かつ、デコーディングする装置によって実行される。

#### 【0080】

説明の便宜上、本発明の実施形態は、例えば、高効率ビデオコーディング(High-Efficiency Video Coding、HEVC)を参照することにより、または、バーサタイルビデオコーディング(Versatile Video Coding、VVC)の参照ソフトウェア、ITU-Tビデオコーディングエキスパートグループ(VCEG)のJoint Collaboration on Video Coding(JCT-VC)、および、ISO/IEC Motion Picture Experts Group(MPEG)によって開発された次世代ビデオコーディング規格を参照して、ここにおいて説明される。当業者の一人であれば、本発明の実施形態が、HEVCまたはVVCに限定されるものではないことを理解するだろう。

#### 【0081】

エンコーダおよびエンコーディング方法

図2は、本出願の技術を実施するように構成された、例示的なビデオエンコーダ20の概略ブロック図を示している。図2の例において、ビデオエンコーダ20は、入力201(または、入力インターフェイス201)、残差計算ユニット204、変換処理ユニット206、量子化ユニット208、逆量子化ユニット210、および、逆変換処理ユニット212、再構成ユニット214、ループフィルタユニット220、デコーディングされた映像バッファ230、モード選択ユニット260、エントロピーエンコーディングユニット270、および、出力ユニット272(または、出力インターフェイス272)を含んでいる。モード選択ユニット260は、インター予測ユニット244、イントラ予測ユニット254、およびパーティション分割ユニット262、を含んでよい。インター予測ユニット244は、動き推定ユニットおよび動き補償ユニ

ット(図示なし)を含んでよい。図2に示されるようなビデオエンコーダ20は、また、ハイブリッドビデオコーデックに応じた、ハイブリッドビデオエンコーダまたビデオエンコーダとしても参照される。

#### 【0082】

残差計算ユニット204、変換処理ユニット206、量子化ユニット208、モード選択ユニット260は、エンコーダ20の前方(forward)信号経路を形成するものとして参照され、一方で、逆量子化ユニット210、逆変換処理ユニット212、再構成ユニット214、バッファ216、ループフィルタ220、デコーディングされた映像バッファ(DPB)230、インター予測ユニット244、および、イントラ予測ユニット254は、ビデオエンコーダ20の後方(backward)信号経路を形成するものとして参照され得る。ここで、ビデオエンコーダ20の後方信号経路は、デコーダの信号経路に対応している(図3のビデオデコーダ30を参照のこと)。逆量子化ユニット210、逆変換処理ユニット212、再構成ユニット214、ループフィルタ220、デコーディングされた映像バッファ(DPB)230、インター予測ユニット244、およびイントラ予測ユニット254は、また、ビデオエンコーダ20の「内蔵デコーダ(“built-in decoder”)」を形成するものとしても参照される。

#### 【0083】

ピクチャおよびピクチャ分割(ピクチャとブロック)

エンコーダ20は、例えば入力201を介して、ピクチャ7(または、映像データ17)、例えばビデオまたはビデオシーケンスを形成する一連のピクチャに係る映像を受信するように構成され得る。受信されたピクチャまたは映像データは、また、前処理されたピクチャ19(または、前処理された映像データ19)であってもよい。簡潔のために、以下の説明は、図17を参照する。ピクチャ17は、また、現在ピクチャまたはコーディングされるピクチャとしても参照され得る(特に、ビデオコーディングにおいて、現在ピクチャを、他のピクチャ、例えば、同じビデオシーケンス、すなわち、現在ピクチャも、また、含むビデオシーケンス、に係る以前にエンコーディング及び／又はデコーディングされたピクチャから区別するため)。

#### 【0084】

(デジタル)映像は、輝度値(intensity value)を伴うサンプルの2次元アレイまたはマトリクスであり、または、そのようにみなされ得る。アレイにおけるサンプルは、また、ピクセル(ピクチャ要素の短い形態)またはペル(pel)としても参照され得る。アレイまたはピクチャの水平および垂直方向(または軸)におけるサンプルの数は、ピクチャのサイズ及び／又は解像度を定義する。色の表現について、典型的には、3つの色成分が使用される。すなわち、ピクチャは、3つのサンプルアレイを表現するか、または、含むことができる。RGBフォーマットまたは色空間において、ピクチャは、対応する赤、緑、および青のサンプルアレイを含んでいる。しかしながら、ビデオコーディングにおいて、各ピクセルは、典型的には、ルミナンス(luminance)およびクロミナンス(chrominance)フォーマット、または、色空間、例えば、Yで示されるルミナンス成分(ときどき、代わりに、Lも、また使用される)、および、CbとCrで示される2つのクロミナンス成分を含む、YCbCrで表現される。ルミナンス(または、短くルマ)成分Yは、明るさ(brightness)またはグレーレベル(grey level)強度(例えば、グレースケール画像内など)を表し、一方、2つのクロミナンス(または、短くクロマ)成分CbとCrは、色度(chromaticity)または色情報コンポーネントを表している。従って、YCbCrフォーマットにおけるピクチャは、ルマサンプル値(Y)のルマサンプルアレイ、および、クロミナンス値(CbとCr)の2つのクロマサンプルアレイを含んでいる。RGBフォーマットにおけるピクチャは、YCbCrフォーマットへと転換(convert)または変換することができ、そして、その逆も同様である。本方法は、また、色変換または転換としても知られている。ピクチャがモノクロである場合、ピクチャは輝度サンプルアレイのみを含んでもよい。従って、画像は、例えば、モノクロフォーマットにおけるルマサンプルのアレイ、または、ルマサンプルのアレイ、並びに、4:2:0、4:2:2、および4:4:4カラーフォーマットにおけるクロマサンプルの2つの対応するアレイであり得る。

## 【0085】

ビデオエンコーダ20の実施形態は、ピクチャ17を複数の(典型的には重複しない)ピクチャブロック203へと分割するように構成されている、映像パーティション分割(partitioning)ユニット(図2に示されていない)を含み得る。これらのブロックは、また、ルートブロック、マクロブロック(H.264/AVC)、またはコーディングツリーブロック(CTB)、もしくはコーディングツリーユニット(CTU)(H.265/HEVCおよびVVC)としても参照され得る。映像パーティション分割ユニットは、ビデオシーケンスの全てのピクチャ、および、ブロックサイズを定義する対応するグリッドに対して同じブロックサイズを使用するように、または、ピクチャ間、サブセット間、または、ピクチャのグループ間でブロックサイズを変更し、かつ、各ピクチャを対応するブロックへと分割するように構成され得る。

10

## 【0086】

さらなる実施形態において、ビデオエンコーダは、ピクチャ17のブロック203、例えば、ピクチャ17を形成している1つ、数個、または、全てのブロックを直接的に受信するように構成され得る。ピクチャブロック203は、また、現在ピクチャブロック、または、コーディングされるピクチャブロックとしても参照され得る。

## 【0087】

ピクチャ17と同様に、ピクチャブロック203は、ピクチャ17よりも小さい寸法ではあるが、再び、輝度値(サンプル値)を伴うサンプルの2次元アレイまたはマトリクスであり、または、見なされ得る。別の言葉で言えば、ブロック203は、例えば、1つのサンプルアレイ(例えば、モノクロ映像17の場合はルマアレイであり、または、カラー映像の場合はルマまたはクロマアレイ)、または、3つのサンプルアレイ(例えば、カラー映像17の場合はルマおよび2つのクロマアレイ)、もしくは、適用されるカラーフォーマットに依存して任意の他の数及び／又は種類のアレイを含み得る。ブロック203の水平方向および垂直方向(または軸)におけるサンプルの数は、ブロック203のサイズを定義する。従って、ブロックは、例えば、サンプルの $M \times N$ ( $M$ 列 $N$ 行)アレイ、または、変換係数の $M \times N$ アレイであり得る。

20

## 【0088】

図2に示されるように、ビデオエンコーダ20の実施形態は、ブロック毎にピクチャ17のブロックをエンコーディングするように構成され得る。例えば、エンコーディングおよび予測は、ブロック203毎に実行される。

30

## 【0089】

図2に示されるように、ビデオエンコーダ20の実施形態は、さらに、スライス(ビデオスライスとしても、また、参照される)を使用することによってピクチャをパーティション分割及び／又はエンコーディングするように構成され得る。ここで、ピクチャは、1つ以上のスライス(典型的には、重複していない)へとパーティション分割またはエンコーディングすることができ、そして、各スライスは、1つ以上のブロック(例えば、CTU)、または、1つ以上のブロックのグループ(例えば、タイル(H.265/HEVCおよびVC)またはブリック(VVC))を含むことができる。

## 【0090】

図2に示されるように、ビデオエンコーダ20の実施形態は、さらに、スライス／タイル・グループ(ビデオタイルグループとしても、また、参照される)及び／又はタイルグループ(ビデオタイルとしても、また、参照される)を使用することによって、ピクチャをパーティション分割及び／又はエンコーディングするように構成され得る。ここで、ピクチャは、1つ以上のスライス／タイル・グループ(典型的には、重複していない)へとパーティション分割またはエンコーディングすることができ、そして、各スライス／タイル・グループは、例えば、1つ以上のブロック(例えば、CTU)または1つ以上のタイルを含むことができる。ここで、各タイルは、例えば、矩形の形状であってもよく、かつ、1つ以上のブロック(例えば、CTU)、例えば、完全または分割ブロックを含み得る。

40

## 【0091】

残差計算(Residual Calculation)

50

残差計算ユニット204は、ピクチャブロック203および予測ブロック265に基づいて、残差ブロック205(残差ブロック205としても、また、参照される)を計算するように構成することができる(予測ブロック265についてのさらなる詳細は後述される)。例えば、ピクチャブロック203のサンプル値から予測ブロック265のサンプル値を差し引くことによって、サンプル毎に(ピクセル毎に)サンプル領域における残差ブロック205を獲得する。

【0092】

変換 (Transform)

変換処理ユニット206は、変換領域における変換係数207を獲得するために、残差ブロック205のサンプル値について、変換、例えば離散コサイン変換 (DCT) または離散サイン変換 (DST)、を適用するように構成され得る。変換係数207は、また、変換残差係数

10

【0093】

変換処理ユニット206は、H.265/HEVCについて指定された変換といった、DCT/DSTの整数近似 (integer approximation) を適用するように構成され得る。直交DCT変換と比較して、そうした整数近似は、典型的には、所定のファクタによってスケール化 (scaled) される。順 (forward) 変換および逆 (inverse) 変換によって処理される残差ブロックのノルム (norm) を保存するために、変換プロセスの一部として追加なスケール係数が適用される。スケール係数は、典型的には、シフト演算のための2のべき乗 (power of two) であるスケール係数、変換係数のビット深度 (bit depth)、精度と実装コストとの間のトレードオフ、等のような所定の制約に基づいて選択される。特定のスケール係数は、例として、逆変換について、例えば、逆変換処理ユニット212によって(および、対応する逆変換について、ビデオデコーダ30での逆変換処理ユニット312によって)によって、指定される。そして、順変換について対応するスケール係数が、例えば、変換処理ユニット206によって、エンコーダ20において、それに応じて指定され得る。

20

【0094】

ビデオエンコーダ20の実施形態(それぞれに変換処理ユニット206)は、例えば、エントロピーエンコーディングユニット270を介して直接的に、またはエンコーディングされ、または圧縮された変換パラメータ、例えば、変換のタイプ、を出力するように構成され得る。その結果、例えば、ビデオデコーダ30は、デコーディングのための変換パラメータを受信し、かつ、使用することができる。

30

【0095】

量子化 (quantization)

量子化ユニット208は、例えば、スカラー量子化またはベクトル量子化を適用することによって、量子化係数209を獲得するために、変換係数207を量子化するように構成され得る。量子化係数209は、また、量子化変換係数209または量子化残差係数209としても参照される。

【0096】

量子化プロセスは、変換係数207の一部または全てに関連するビット深度を低減することができる。例えば、nビット変換係数は、量子化の最中に、mビット変換係数まで丸められ (rounded) 得る。ここで、nは、mより大きい。量子化の程度は、量子化パラメータ (QP) を調整することによって修正され得る。例えば、スカラー量子化について、より微細な (finer)、または、より粗大な (coarser) 量子化を達成するために、異なるスケール係数が適用され得る。より小さい量子化ステップサイズは、より微細な量子化に対応し、一方で、より大きい量子化ステップサイズは、より粗大な量子化に対応している。適用可能な量子化ステップサイズは、量子化パラメータ (QP) によって示され得る。量子化パラメータは、例えば、適用可能な量子化ステップサイズの事前に定義されたセットに対するインデックスであり得る。例えば、小さい量子化パラメータは、微細な量子化(小さい量子化ステップサイズ)に対応し、かつ、大きい量子化パラメータは、粗大な量子化(大きい量子化ステップサイズ)に対応し得るか、または、その逆も同様である。量子化は、量子化ステ

40

50

ップサイズによる除算 (division) を含んでよく、そして、対応する、かつ／あるいは、逆の脱量子化は、例えば、逆量子化ユニット210によって、量子化ステップサイズによる乗算を含んでよい。いくつかの標準、例えばHEVC、に従った実施形態は、量子化ステップサイズを決定するために、量子化パラメータを使用するように構成され得る。一般的に、量子化ステップサイズは、除算を含む方程式の固定点近似を使用する量子化パラメータに基づいて、計算され得る。残差ブロックのノルムを回復するために、量子化および脱量子化 (dequantization) について追加的なスケーリング係数が導入されてよく、それは、量子化ステップサイズおよび量子化パラメータについて方程式の固定点近似において使用されるスケーリングのために修正され得る。一つの実施例においては、逆変換および脱量子化のスケーリングを組み合わせることができる。代替的に、カスタマイズされた量子化テーブルが使用され、そして、エンコーダからデコーダへ、例えばビットストリームにおいて、信号化され得る。量子化は、不可逆オペレーションであり、ここでは、量子化ステップサイズの増加に伴って損失が増加する。

【0097】

ビデオエンコーダ20の実施形態(それぞれに、量子化ユニット208)は、例えば、エン트로ピーエンコーディングユニット270を介して、直接的に、または、エンコーディングされた、量子化パラメータ (QP) を出力するように構成され得る。その結果、例えば、ビデオデコーダ30は、デコーディングのために量子化パラメータを受信し、かつ、適用することができる。

【0098】

逆量子化 (Inverse Quantization)

逆量子化ユニット210は、脱量子化 (dequantized) 係数211を獲得するために、例えば、量子化ユニット208と同じ量子化ステップサイズに基づいて、または、使用して、量子化ユニット208によって適用される量子化スキームの逆数を適用することによって、量子化係数に量子化ユニット208の逆量子化を適用するように構成されている。脱量子化係数211は、また、脱量子化残差係数211としても参照としても参照され、かつ、一典型的には、量子化による損失のために変換係数と同一ではないが一変換係数207に対応している。

【0099】

逆変換 (Inverse Transform)

逆変換処理ユニット212は、変換処理ユニット206によって適用された変換の逆変換、例えば逆離散コサイン変換 (DCT)、逆離散サイン変換 (DST)、または、他の逆変換を適用するように構成されており、サンプル領域において再構成された残差ブロック213(または、対応する脱量子化係数213)を獲得する。再構成された残差ブロック213は、また、変換ブロック213としても参照される。

【0100】

再構成 (Reconstruction)

再構成ユニット214(例えば、加算器または足し算器214)は、変換ブロック213(すなわち、再構成された残差ブロック213)を、予測ブロック265に追加するように構成されており、例えば、一サンプル毎に一再構成された残差ブロック213のサンプル値と予測ブロック265のサンプル値を加算することによって、サンプル領域における再構成ブロック215を獲得する。

【0101】

フィルタリング

ループフィルタユニット220(または、短く「ループフィルタ ("loop filter")」220)は、フィルタリングされたブロック221を獲得するために、再構成されたブロック215をフィルタリングするように、または、一般的に、フィルタリングされたサンプル値を獲得するために、再構成されたサンプルをフィルタリングするように構成されている。ループフィルタユニットは、例えば、ピクセルの遷移を滑らかにするように、または、そうでなければ、ビデオ品質を改善するように構成されている。ループフィルタユニット220は、

10

20

30

40

50

デブロッキング (deblocking) フィルタ、サンプル適応オフセット (SAO) フィルタ、または、1つ以上の他のフィルタ、例えば、適応ループフィルタ (ALF)、ノイズ抑制フィルタ (NSF)、または、それらの任意の組み合わせ、といった1つ以上のループフィルタを含んでもよい。一つの実施形態において、ループフィルタユニット220は、デブロッキングフィルタ、SAOフィルタ、および、ALFフィルタを含んでよい。フィルタリングプロセスの順序は、デブロッキング、SAO、そして、ALFであってよい。別の例においては、クロマスケーリングを用いたルママッピング(LMCS)と呼ばれるプロセス(すなわち、適応ループ内リシェイパ)が追加される。この処理は、デブロッキングの前に実行される。別の例において、デブロッキングプロセスは、また、内部サブブロックエッジ、例えば、アフィンサブブロックエッジ、ATMVPサブブロックエッジ、サブブロック変換 (SBT) エッジ、および、イントラサブパーティション (ISP) エッジに対して適用されてもよい。ループフィルタユニット220は、図2ではインループフィルタとして示されているが、他のコンフィグレーションにおいて、ループフィルタユニット220は、ポスト・ループフィルタとして実装されてよい。フィルタリングされたブロック221は、また、フィルタリングされた再構成ブロック221としても参照され得る。

#### 【0102】

ビデオエンコーダ20の実施形態(それぞれに、ループフィルタユニット220)は、例えば、直接的に、または、エン트로ピーエンコーディングユニット270を介してエンコードされて、ループフィルタパラメータ(SAOフィルタパラメータ、またはALFフィルタパラメータ、またはLMCSパラメータといったもの)を出力するように構成され得る。その結果、例えば、デコーダ30は、同じループフィルタパラメータ、または、それぞれのループフィルタを受信し、そして、適用し得る。

#### 【0103】

デコーディングされたピクチャバッファ

デコーディングされたピクチャバッファ (DPB) 230は、ビデオエンコーダ20によってビデオデータをエンコーディングするために、参照ピクチャを保管するメモリであってよく、または、一般的な参照映像データ内にあってよい。DPB 230は、シンクロナスDRAM (ADRAM)、磁気抵抗RAM (MRAM)、抵抗変化型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) といった、種々のメモリデバイスのいずれかによって形成され得る。デコーディングされたピクチャバッファ (DPB) 230は、1つ以上のフィルタリングされたブロック221を保管するように構成され得る。デコーディングされたピクチャバッファ230は、さらに、同じ現在ピクチャまたは異なるピクチャ、例えば以前に再構成されたピクチャ、に係る、他の以前にフィルタリングされたブロック、例えば以前に再構成およびフィルタリングされたブロック221)を保管するように構成され得る。そして、例えば、インター予測のために、完全な以前に再構成、すなわちデコーディング、されたピクチャ (および、対応する参照ブロックおよびサンプル)、及び／又は、部分的に再構成された現在ピクチャ(および、対応する参照ブロックおよびサンプル)を提供し得る。デコーディングされたピクチャバッファ (DPB) 230は、また、1つ以上の再構成されたフィルタリングされていないブロック215、または、例えば、再構成されたブロック215がループフィルタユニット220によってフィルタリングされない場合には、一般的にフィルタリングされていない再構成されたサンプル、もしくは、再構成されたブロックまたはサンプルの任意の他のさらに処理されたバージョン、を保管するようにも構成され得る。

#### 【0104】

モード選択(パーティション分割と予測)

モード選択ユニット260は、パーティション分割 (partitioning) ユニット262、インター予測ユニット244、およびイントラ予測ユニット254を含み、そして、元の映像データ、例えば元のブロック203(現在ピクチャ17の現在ブロック203)、および、再構成された映像データ、例えばフィルタリングされ、かつ／あるいは、フィルタリングされていない、同じ(現在)ピクチャの再構成されたサンプルまたはブロック、及び／又は、1つ又は

複数の以前にデコーディングされたピクチャから、例えば、デコーディングされたピクチャバッファ230または他のバッファ(例えばラインバッファ、図示なし)からのものを、受信または獲得するように構成されている。再構成された映像データは、予測ブロック265または予測器265を獲得するために、予測、例えばインター予測またはイントラ予測、のための参照ピクチャデータとして使用される。

#### 【0105】

モード選択ユニット260は、現在ブロック予測モード(パーティション分割なしを含む)および予測モード(例えば、イントラまたはインター予測モード)についてパーティション分割を決定または選択し、かつ、残差ブロック205の計算のため、および、再構成ブロック215の再構成のために使用される、対応する予測ブロック265を生成するように構成され得る。

10

#### 【0106】

モード選択ユニット260の実施形態は、パーティション分割および予測モードを(例えば、モード選択ユニット260によってサポートされ、または利用可能なものから)選択するように構成されてもよく、最良の整合、すなわち、別の言葉で言えば、最小の残差(最小の残差は伝送または保管のためのより良好な圧縮を意味する)、または、最小の信号化オーバーヘッド(最小の信号化オーバーヘッドは伝送または保管のためより良好な圧縮を意味する)、を提供し、もしくは、両方を考慮またはバランスさせる。モード選択ユニット260は、レート歪み最適化(RDO)に基づいて、パーティション分割および予測モードを決定する、つまり、最小レート歪みを提供する予測モードを選択する、ように構成され得る。このコンテキストにおける「最良(“best”)」、「最小(“minimum”)」、「最適(“optimum”)」等の用語は、必ずしも全体的な「最良」、「最小」、「最適」等を参照するのではないが、また、閾値を超えるか、または下回る値、もしくは、「サブ最適の選択(“sub-optimum”)」につながる可能性があるが、複雑性および処理時間を短縮するような、終了基準または選択基準の達成も参照し得る。

20

#### 【0107】

別の言葉で言えば、パーティション分割ユニット262は、映像シーケンスからのピクチャをコーディングツリーユニット(CTU)のシーケンスへとパーティション分割するように構成され得る。そして、CTU 203は、さらに、例えば、クワッドツリー・パーティション分割(QT)、バイナリ・パーティション分割(BT)またはトリプルツリー・パーティション分割(TT)、もしくは、それらの任意の組み合わせを反復的に使用して、より小さなブロックパーティションまたはサブブロック(再びブロックを形成するもの)へとパーティション分割され、そして、例えば、ブロックパーティションまたはサブブロックのそれぞれについて予測を実行することができる。ここで、モード選択は、パーティション分割されたブロック203のツリー構造の選択を含み、そして、予測モードは、ブロックパーティションまたはサブブロックのそれぞれに対して適用される。

30

#### 【0108】

以下に、一つの例示的なビデオエンコーダ20によって実行されるパーティション分割(例えば、パーティション分割ユニット260によるもの)、および、予測処理(インター予測ユニット244およびイントラ予測ユニット254によるもの)について、より詳細に説明される。

40

#### 【0109】

##### パーティション分割 (Partitioning)

パーティション分割ユニット262は、映像シーケンスからのピクチャをコーディングツリーユニット(CTU)のシーケンスへとパーティション分割するように構成され得る。そして、パーティション分割ユニット262は、コーディングツリーユニット(CTU)203を、より小さなパーティション、例えば正方形または長方形サイズのより小さいブロック、へとパーティション分割(またはスプリット)することができる。3つのサンプルアレイを有するピクチャについて、CTUは、ルマサンプルの $N \times N$ ブロックと、クロマサンプルの2つの対応するブロックと一緒に構成されている。CTUにおけるルマブロックの最大許容サ

50

イズは、開発中のバーサタイルビデオコーディング(VVC)では $128 \times 128$ であると指定されているが、将来には $128 \times 128$ ではない値、例えば $256 \times 256$ 、であると指定することができる。ピクチャのCTUは、スライス／タイル・グループ、タイル、またはブリックとして、クラスタ化／グループ化され得る。タイルは、ピクチャの矩形領域をカバーし、そして、タイルは、1つ又は複数のブリックへと分割することができる。ブリックは、タイル内の複数のCTU列から構成されている。複数のブリックへとパーティション分割されていないタイルは、ブリックとして参照され得る。しかしながら、ブリックは、タイルの真のサブセットであり、かつ、タイルとしては参照されない。VVCでサポートされるタイルグループには2つのモードが存在し、すなわち、ラスタスキャン(raster-scan)スライス／タイル・グループモードおよび矩形スライスモードである。ラスタスキャン・タイルグループモードにおいて、スライス/タイル・グループは、ピクチャのタイル・ラスタスキャンにおけるタイルのシーケンスを含む。矩形スライスモードにおいて、スライスは、ピクチャの矩形領域を集散的に形成するピクチャの多くのブリックを含む。矩形スライス内のブリックは、スライスのブリック・ラスタスキャンの順である。これらのより小さなブロック(サブブロックとしても、また、参照される)は、なおもより小さなパーティションへと、さらにパーティション分割され得る。これは、また、ツリーパーティション分割または階層ツリーパーティション分割としても参照され、ここで、ルートブロック、例えばルートツリーレベル0(階層レベル0、深度0)、は、再帰的に分割され得る。例えば、次の下位ツリーレベル、例としてツリーレベル1(階層レベル1、深度1)、のノードの2つ以上のブロックへとパーティション分割され、ここで、これらのブロックは、パーティション分割が終了するまで、次の下位レベル、例えばツリーレベル2(階層レベル2、深度2)等、の2つ以上のブロックへと再びパーティション分割され得る。終了は、例えば、最大ツリー深度または最小ブロックサイズに達するなどの、終了基準が満たされるからである。それ以上パーティション分割されないブロックは、また、ツリーのリーフブロックまたはリーフノードとしても参照される。2つのパーティションへの分割を使用するツリーはバイナリツリー(BT)として参照され、3つのパーティションへの分割を使用するツリーはターナリツリー(ternary tree、TT)として参照され、そして、4つのパーティションへの分割を使用するツリーはクワッドツリー(quad tree、QT)として参照される。

【0110】

例えば、コーディングツリーユニット(CTU)は、ルマサンプルのCTB、3つのサンプルアレイを有するピクチャのクロマサンプルに係る2つの対応するCTB、または、サンプルをコード化するために使用される3つの別々のカラープレーンおよびシンタックス構造を使用してコード化されるモノクロ映像またはピクチャのサンプルに係るCTBであってよく、または、これらを含んでよい。これに対応して、コーディングツリーブロック(CTB)は、CTBへのコンポーネントの分割がパーティション分割であるように、Nのいくつかの値についてサンプルの $N \times N$ ブロックであってよい。コーディングユニット(CU)は、ルマサンプルのコーディングブロック、3つのサンプルアレイを有するピクチャのクロマサンプルに係る2つの対応するコーディングブロック、または、サンプルをコード化するために使用される3つの別々のカラープレーンおよびシンタックス構造を使用してコード化されるモノクロ映像またはピクチャのサンプルに係るコーディングブロックであってよく、または、これらを含んでよい。これに対応して、コーディングブロックは、コーディングブロックへのCTBの分割がパーティション分割であるように、MおよびNのいくつかの値についてサンプルの $M \times N$ ブロックであってよい。

【0111】

実施形態において、例えば、HEVCに従って、コーディングツリーユニット(CTU)は、コーディングツリーとして示されるクワッドツリー構造を使用することによって、CUへとスプリット(split)され得る。インターピクチャ(時間的)またはイントラピクチャ(空間的)予測を使用してピクチャ領域をコード化するか否かの決定は、リーフCUレベルで行われる。各リーフCUは、PUスプリットタイプ(splitting type)に応じて、さらに、1つ、2つ、または4つのPUへとスプリットすることができる。1つのPU内では、同じ予測プ

ロセスが適用され、そして、関連情報がPUベースでデコーダに対して送信される。PUスプリットタイプに基づいて予測プロセスを適用することによって残差ブロックを獲得した後で、リーフCUは、CUのためのコーディングツリーに類似した別のクワッドツリー構造に従って、変換ユニット(TU)へとパーティション分割され得る。

#### 【0112】

実施形態においては、例えば、現在開発中の最新のビデオコーディング標準、バーサタイルビデオコーディング(VVC)として参照されるもの、に従って、例えば、コーディングツリーユニットをパーティション分割するために使用される、バイナリおよびターナリ・スプリットセグメンテーション構造を使用して、結合されたクワッドツリーは、マルチタイプツリーにネスト化される。コーディングツリーユニット内のコーディングツリー構造において、CUは、正方形または矩形の形状のいずれかを有することができる。例えば、コーディングツリーユニット(CTU)は、最初に、クワッドツリーによってパーティション分割される。次いで、クォータ(quaternary) ツリーリーフノードは、さらに、マルチタイプツリー構造によってパーティション分割することができる。マルチタイプツリー構造には、4つのスプリットタイプが存在する。垂直バイナリスプリット(SPLIT\_BT\_VER)、水平バイナリスプリット(SPLIT\_BT\_HOR)、垂直ターナリスプリット(SPLIT\_TT\_VER)、水平ターナリスプリット(SPLIT\_TT\_HOR)である。マルチタイプツリーリーフノードは、コーディングユニット(CU)と呼ばれ、そして、CUが最大変換長に対して大き過ぎない限り、このセグメンテーションは、任意のさらなるパーティション分割なしで、予測および変換処理のために使用される。これは、ほとんどの場合、CU、PU、およびTUは、ネスト化されたマルチタイプのツリーコーディングブロック構造を伴うクワッドツリーにおいて同じブロックサイズを有することを意味する。例外は、最大サポート変換長が、CUの色成分の幅または高さよりも小さい場合に発生する。VVCは、ネスト化されたマルチタイプのツリーコーディングツリー構造を伴うクワッドツリーにおけるパーティションスプリット情報のユニークな信号化メカニズムを開発している。信号化(signaling)メカニズムでは、コーディングツリーユニット(CTU)は、クォータツリー(quaternary tree)のルートとして扱われ、そして、最初に、クォータツリー構造によってパーティション分割される。各クォータツリーリーフノードは(それを許すように十分に大きい場合)、次いで、マルチタイプツリー構造によってさらにパーティション分割される。マルチタイプツリー構造において、第1フラグ(mtt\_split\_cu\_flag)は、ノードがさらにパーティション分割されるか否かを示すために信号化され、ノードがさらにパーティション分割される場合に、第2フラグ(mtt\_split\_cu\_vertical\_flag)は、スプリット方向を示すために信号化され、そして、次いで、第3フラグ(mtt\_split\_cu\_binary\_flag)は、スプリットがバイナリスプリットか、またはターナリスプリットかを示すために信号化される。mtt\_split\_cu\_vertical\_flagとmtt\_split\_cu\_binary\_flagの値に基づいて、CUのマルチタイプツリー・スプリットモード(MttSplitMode)は、事前に定義されたルールまたはテーブルに基づいて、デコーダによって導出され得る。所定の設計、例えば、VVCハードウェアデコーダにおける64×64ルマブロックおよび32×32クロマパイプライン設計について、図6に示されるように、ルマコーディングブロックの幅または高さのいずれかが64より大きい場合にTTスプリットが禁止される。TTスプリットは、また、クロマコーディングブロックの幅または高さが32より大きい場合にも禁止される。パイプライン設計は、ピクチャを、仮想パイプラインデータユニット(VPDU)へと分割し、ピクチャ内の重複しないユニットとして定義される。ハードウェアデコーダにおいては、連続するVPDUが、複数のパイプラインステージ(stage)によって同時に処理される。VPDUサイズは、大部分のパイプラインステージにおけるバッファサイズにほぼ比例するので、VPDUサイズを小さく保つことが重要である。大部分のハードウェアデコーダにおいて、VPDUサイズは、最大変換ブロック(TB)サイズに設定できる。しかしながら、VVCでは、ターナリツリー((TT)とバイナリツリー(BT)パーティションが、VPDUサイズの増加をもたらし得る。加えて、ツリーノードブロックの一部がピクチャ境界の底部または右側を超える場合、ツリーノードブロックは、それぞれコード化されたCUの全てのサンプルがピクチャ境界の内側に配置されるまで

10

20

30

40

50

、強制的にスプリットされることが留意されるべきである。

一つの例として、Intra Sub-Partitions (ISP) ツールは、ブロックサイズに応じて、ルミナンス内予測ブロックを縦方向または横方向に2つまたは4つのサブパーティションに分割することができる。

#### 【0113】

一つの例において、ビデオエンコーダ20のモード選択ユニット260は、ここにおいて説明されるパーティション分割技術の任意の組み合わせを実行するように構成され得る。

#### 【0114】

上述のように、ビデオエンコーダ20は、(例えば、事前に決定された)予測モードのセットから最良または最適な予測モードを決定または選択するように構成されている。予測モードのセットは、例えば、イントラ予測モード及び／又はインター予測モードを含み得る。

#### 【0115】

イントラ予測 (Intra-Prediction)

イントラ予測モードのセットは、35個の異なるイントラ予測モードを含み得る。例として、HEVCで定義されているように、例えば、DC(または平均)モードおよびプラナーモード (planar mode) のような非方向モード、または、方向モードである。もしくは、67個の異なるイントラ予測モードを含み得る。例として、VVCで定義されているように、例えば、DC(または平均)モードおよびプラナーモードのような非方向モード、または、方向モードである。一つの例として、いくつかの従来の角度イントラ予測モードが、例えばVVCで定義されるように、非正方形ブロックについて、広角イントラ予測モードと適応的に置き換えられる。別の例として、DC予測のための分割演算を回避するために、非正方形ブロックについて、平均を計算するために、より長い辺だけが使用される。そして、プレーナモードのイントラ予測の結果は、さらに、位置依存イントラ予測結合(PDPC)法によって修正され得る。

#### 【0116】

イントラ予測ユニット254は、イントラ予測モードのセットのイントラ予測モードに従って、イントラ予測ブロック265を生成するために、同一の現在ピクチャの隣接ブロックの再構成されたサンプルを使用するように構成されている。

#### 【0117】

イントラ予測ユニット254(または、一般的に、モード選択ユニット260)は、さらに、エンコーディングされた映像データ21の中へ含めるためのシンタックス要素266の形式で、エントロピーエンコーディングユニット270に対してイントラ予測パラメータ(または、一般的に、ブロックについて選択されたイントラ予測モードを示す情報)を出力するように構成されている。その結果、例えば、ビデオデコーダ30は、デコーディングのための予測パラメータを受信し、そして、使用することができる。

#### 【0118】

インター予測(層間 (inter-layer) 予測を含む)

インター予測モードの(または、可能な)セットは、利用可能な参照ピクチャ(すなわち、例えばDBP 230に保管されている、以前に少なくとも部分的にデコーディングされたピクチャ)および他のインター予測パラメータ、例えば、参照ピクチャの全体、または、参照ピクチャの一部、例として、現在ブロックの領域の周囲のサーチウィンドウ領域、が、最良のマッチング参照ブロックをサーチするために使用されるか否か、及び／又は、例えば、ピクセル補間 (interpolation) が適用されるか否か、例えば、ハーフ／セミ・ペル (half/4semi-pel)、1/4ペル (quarter-pel)、及び／又は、1/16ペルであるか、または、そうでないか、に依存する。

#### 【0119】

上記の予測モードに加えて、スキップモード、直接モード、及び／又は、他のインター予測モードが適用され得る。

#### 【0120】

例えば、拡張マージ予測 (Extended merge prediction)、そうしたモードのマー

10

20

30

40

50

ジ候補リストは、次の5つのタイプの候補を順番に含むことによって構築されている。空間的隣接CUからの空間的MVP、コロケーションされたCUからの時間的MVP、FIFOテーブルからの履歴ベースMVP、ペアでの（pairwise）平均MVP、およびゼロMV、である。そして、マージモードのMVの精度を高めるために、バイラテラルマッチングベースのデコーダ側動きベクトル精緻化(decoder side motion vector refinement、DMVR)が適用され得る。MVDを用いたマージモード(merge mode with MVD、MMDV)は、動きベクトルの差異を用いたマージモードからのものである。MMVDフラグは、MMVDモードがCUに使用されるか否かを指定するために、スキップフラグとマージフラグを送信した直後に信号化される。そして、CUレベルの適応動きベクトル分解(adaptive motion vector resolution、AMVR)スキャンが適用され得る。AMVRは、CUのMVDが異なる精度でコード化されることを可能にする。現在CUの予測モードに依存して、現在CUのMVDを適応的に選択することができる。CUがマージモードでコード化される場合、結合されたインター／イントラ予測(CIIP)モードが現在CUに適用され得る。CIIP予測を獲得するために、インターおよびイントラ予測信号の加重平均が実行される。アフィン（affine）動き補償予測、ブロックのアフィン動きフィールドは、2つの制御点(4パラメータ)または3つの制御点動きベクトル(6パラメータ)の運動情報によって記述される。サブブロックベースの時間的動きベクトル予測(SbTMVP)は、HEVCにおける時間的動きベクトル予測(TMVP)に類似しているが、現在CU内のサブCUの動きベクトルを予測する。双方向（bi-directional）オプティカルフロー(BDOF)、BIOとして以前は呼ばれていたものは、特に、乗算の数および乗算器のサイズに関して、はるかに少ない計算量を必要とするより簡潔なバージョンである。三角（triangle）パーティション分割モード、そうしたモードでは、対角分割または反対角分割のいずれかを使用して、CUが2つの三角形パーティションへと均等に分割される。加えて、双予測（bi-prediction）モードは、2つの予測信号の加重平均を可能にするために、単純な平均を越えて拡張される。

【0121】

インター予測ユニット244は、動き推定（ME）ユニットおよび動き補償（MC）ユニット(両方とも図2に示されていない)を含み得る。動き推定ユニットは、動き推定のために、ピクチャブロック203(現在ピクチャ17の現在ピクチャブロック203)およびデコーディングされたピクチャ231、または、少なくとも1つ又は複数の以前に再構成されたブロック、例えば、1つ又は複数の他の／異なる以前にデコーディングされたピクチャ231の再構成ブロックを、受信または獲得するように構成され得る。例えば、ビデオシーケンスは、現在ピクチャおよび以前にデコーディングされたピクチャ231を含み得る。または、別の言葉で言えば、現在ピクチャと以前にデコーディングされたピクチャ231は、ビデオシーケンスを形成しているピクチャのシーケンスの一部であってよく、または、形成し得る。

【0122】

エンコーダ20は、例えば、複数の他のピクチャの同じ又は異なるピクチャに係る複数の参照ブロックから参照ブロックを選択し、かつ、参照ピクチャ(または、参照ピクチャインデックス)、及び／又は、参照ブロックの位置(x、y座標)と現在ブロックの位置との間のオフセット(空間オフセット)を、動き推定ユニットへのインター予測パラメータとして、提供するように構成され得る。このオフセットは、また、動きベクトル(motion vector、MV)としても呼ばれる。

【0123】

動き補償ユニットは、受信インター予測パラメータを獲得する、例えば、受信し、そして、インター予測ブロック265を獲得するために、インター予測パラメータに基づいて、または、使用して、インター予測を実行するように構成されている。動き補償ユニットによって実行される、動き補償は、動き推定によって決定された動き／ブロックベクトルに基づいて予測ブロックをフェッチすること又は生成することを含んでよく、サブピクセル精度への補間を実行し得る。補間フィルタリングは、既知のピクセルサンプルから追加のピクセルサンプルを生成することができ、従って、ピクチャブロックをコード化するために使用され得る候補予測ブロックの数を潜在的に増加させる。現在ピクチャブロックのPU

10

20

30

40

50

に対する動きベクトルを受信すると、動き補償ユニットは、参照ピクチャリストのうちの1つにおいて動きベクトルが指し示す予測ブロックを配置し得る。

【0124】

動き補償ユニットは、また、ビデオスライスのピクチャブロックをデコーディングする際のビデオデコーダ30による使用のために、ブロックおよびビデオスライスに関連するシンタックス要素も生成し得る。スライス及びそれぞれのシンタックス要素に加えて、または、代替として、タイルグループ及び／又はタイル、および、それぞれのシンタックス要素が、生成され又は使用され得る。

【0125】

エントロピーコーディング (Entropy Coding)

エントロピーコーディングユニット270は、例えば、エントロピーエンコーディングアルゴリズムまたはスキーム(例えば、可変長コーディング(VLC)スキーム、コンテキスト適応VLCスキーム(CAVLC)、算術コーディングスキーム、バイナリ化、コンテキスト適応バイナリ算術コーディング(CABAC)、シンタックススペースのコンテキスト適応バイナリ算術コーディング(SBAC)、確率間隔パーティション分割エントロピー(PIPE)コーディング、または、別の他のエントロピーエンコーディング方法または技術)を、量子化係数209、インター予測パラメータ、イントラ予測パラメータ、ループフィルタパラメータ、及び／又は、他のシンタックス要素について、適用し、またはバイパス(非圧縮)するように構成されており、例えば、エンコーディングされたビットストリーム21の形式で、出力272を介して出力され得るエンコーディングされた映像データ21を獲得する。その結果、ビデオデコーダ30は、デコーディングされたパラメータを受信し、そして、使用し得る。エンコーディングされたビットストリーム21は、ビデオデコーダ30へ送信されてよく、または、ビデオデコーダ30による後の送信または検索のためにメモリ内に保管され得る。

【0126】

ビデオエンコーダ20の他の構造的変形は、ビデオストリームをエンコーディングするために使用することができる。例えば、非変換ベースのエンコーダ20は、所定のブロックまたはフレームについて、変換処理ユニット206なしで、直接的に残差信号を量子化することができる。別の実施形態において、エンコーダ20は、量子化ユニット208および逆量子化ユニット210を組み合わせて単一のユニットにすることができる。

【0127】

デコーダおよびデコーディング方法

図3は、この本出願の技術を実施するように構成されたビデオデコーダ30に係る一つの例を示している。ビデオデコーダ30は、デコーディングされたピクチャ331を獲得するために、エンコーディングされた映像データ21(例えばエンコーディングされたビットストリーム21)、例えばエンコーダ20によってエンコーディングされたもの、を受信するように構成されている。エンコーディングされた映像データまたはビットストリームは、エンコーディングされた映像データ、例えば、エンコーディングされたビデオスライスのピクチャブロック(及び／又は、タイルグループまたはタイル)および関連するシンタックス要素を表すデータ、をデコーディングするための情報を含む。

【0128】

図3の例において、デコーダ30は、エントロピーデコーディングユニット304、逆量子化ユニット310、逆変換処理ユニット312、再構成ユニット314(例えば、足し算器314)、ループフィルタ320、デコーディングされたピクチャバッファ(DBP)330、モード適用ユニット360、インター予測ユニット344、および、イントラ予測ユニット354を含んでいる。インター予測ユニット344は、動き補償ユニットであってよく、または、含み得る。ビデオデコーダ30は、いくつかの例において、図2からのビデオエンコーダ100に関して説明されたエンコーディングパスと概ね相反性(reciprocal)のデコーディングパスを実行することができる。

【0129】

エンコーダ20に関して説明したように、逆量子化ユニット210、逆変換処理ユニット2

10

20

30

40

50

12、再構成ユニット214、ループフィルタ220、デコーディングされたピクチャバッファ(DPB) 230、インター予測ユニット344、および、イントラ予測ユニット354は、また、ビデオエンコーダ20の「内蔵デコーダ(“built-in decoder”)」を形成するものとしても参照される。従って、逆量子化ユニット310は、逆量子化ユニット110と機能的に同一であってよく、逆変換処理ユニット312は、逆変換処理ユニット212と機能的に同一であってよく、再構成ユニット314は、再構成ユニット214と機能的に同一であってよく、ループフィルタ320は、ループフィルタ220と機能的に同一であってよく、そして、デコーディングされたピクチャバッファ330は、デコーディングされたピクチャバッファ230と機能的に同一であってよい。従って、ビデオデコーダ20に係るそれぞれのユニットおよび機能について提供された説明は、ビデオ30に係るそれぞれのユニットおよび機能に対応して適用される。

10

#### 【0130】

#### エントロピーデコーディング (Entropy Decoding)

エントロピーデコーディングユニット304は、ビットストリーム21(または、一般的にエンコーディングされた映像データ21)を解析し、そして、例えば、エンコーディングされた映像データ21に対してエントロピーデコーディングを実行するように構成されており、

例えば、量子化係数309及び／又はデコーディングされたコード化パラメータ(図3には示されていない)、例えば、インター予測パラメータ(例えば、参照ピクチャインデックスおよび動きベクトル)、イントラ予測パラメータ(例えば、イントラ予測モードまたはインデックス)、変換パラメータ、量子化パラメータ、ループフィルタパラメータ、及び／又は、他のシンタックス要素、の幾つか又は全てを獲得する。エントロピーデコーディングユニット304は、エンコーダ20のエントロピーコーディングユニット270に関して説明されたように、エンコーディングスキームに対応するデコーディングアルゴリズムまたはスキームを適用するように構成され得る。エントロピーデコーディングユニット304は、さらに、インター予測パラメータ、イントラ予測パラメータ、及び／又は、他のシンタックス要素を、モード適用ユニット360に対して、および、他のパラメータを、デコーダ30の他のユニットに対して提供するように構成され得る。ビデオデコーダ30は、ビデオスライスレベル及び／又はビデオブロックレベルで、シンタックス要素を受信することができる。スライス及びそれぞれのシンタックス要素に加えて、または、代替として、タイルグループ及び／又はタイル、および、それぞれのシンタックス要素が、受信され、かつ／あるいは、使用され得る。

20

30

#### 【0131】

#### 逆量子化 (Inverse Quantization)

逆量子化ユニット310は、エンコーディングされた映像データ21から(例えば、エントロピーデコーディングユニット304による、例えば、解析及び／又はデコーディングによって)、量子化パラメータ(QP)(または、一般的に逆量子化に関する情報)、および、量子化係数を受信し、かつ、量子化パラメータに基づいて、逆量子化をデコーディングされた量子化閾値309に適用するように構成することができ、変換係数311としても、また、参照され得る、脱量子化係数311を獲得する。逆量子化プロセスは、量子化の程度、および、同様に、適用されるべき逆量子化の程度を決定するために、ビデオスライス(または、タイル又はタイルグループ)内の各ビデオブロックについて、ビデオエンコーダ20によって決定される量子化パラメータを使用することを含み得る。

40

#### 【0132】

#### 逆変換 (Inverse Transform)

逆変換処理ユニット312は、変換係数311としても、また、参照される、脱量子化係数311を受信し、かつ、サンプル領域において再構成された残差ブロック213を獲得するために、脱量子化係数311に変換を適用するように構成され得る。再構成された残差ブロック213は、また、変換ブロック313としても参照され得る。変換は、逆変換、例えば、逆DCT、逆DST、逆整数変換、または、概念的に類似した逆変換プロセス、であり得る。逆

50

変換処理ユニット312は、さらに、エンコーディングされた映像データ21から(例えば、エントロピーデコーディングユニット304による、例えば、解析及び／又はデコーディングによって)、変換パラメータまたは対応する情報を受信するように構成されてよく、脱量子化係数311に対して適用される変換を決定する。

#### 【0133】

##### 再構成 (Reconstruction)

再構成ユニット314(例えば、加算器または足し算器314)は、サンプル領域内の再構成ブロック315を獲得するために、予測ブロック365に対して、再構成残差ブロック313を追加ように構成され得る。例えば、再構成残差ブロック313のサンプル値および予測ブロック365のサンプル値を追加することによる、ものである。

#### 【0134】

##### フィルタリング

ループフィルタユニット320は、(コーディンググループ内、または、コーディンググループ後、のいずれかで)再構成ブロック315をフィルタリングするように構成されており、フィルタリングされたブロック321を獲得し、例えば、ピクセルの遷移を滑らかにするか、または、そうでなければ、ビデオ品質を改善する。ループフィルタユニット320は、デブロッキング (de-blocking) フィルタ、サンプル適応オフセット (SAO) フィルタ、または、1つ以上の他のフィルタ、例えば、適応ループフィルタ (ALF)、ノイズ抑制フィルタ (NSF)、または、それらの任意の組み合わせ、といった、1つ以上のループフィルタを含み得る。一つの例において、ループフィルタユニット220は、デブロッキングフィルタ、SAOフィルタ、および、ALFフィルタを含み得る。フィルタリングプロセスの順序は、デブロッキングフィルタ、SAO、そして、ALFであってよい。別の例においては、クロマスケーリングを用いたルママッピング(LMCS)と呼ばれるプロセス(すなわち、適応ループ内リシェイパ)が追加される。この処理は、デブロッキングの前に実行される。別の例において、デブロッキングフィルタプロセスは、また、内部サブブロックエッジ、例えば、アフィンサブブロックエッジ、ATMVPサブブロックエッジ、サブブロック変換 (SBT) エッジ、および、イントラサブパーティション (ISP) エッジに対して適用されてもよい。ループフィルタユニット320は、図3ではインループフィルタとして示されているが、他のコンフィギュレーションにおいて、ループフィルタユニット320は、ポスト・ループフィルタとして実装されてよい。

#### 【0135】

##### デコーディングされたピクチャバッファ

ピクチャのデコーディングされたビデオブロック321は、次いで、デコーディングされたピクチャバッファ330内に保管される。バッファは、デコーディングされたピクチャ331を、他のピクチャのため、及び／又は、出力それぞれの表示のための、後に続く動き補償のための参照ピクチャとして保管する。

#### 【0136】

デコーダ30は、例えば出力312を介して、デコーディングされたピクチャ311を、ユーザに対して提示または表示するために、出力するように構成されている。

#### 【0137】

##### 予測 (Prediction)

インター予測ユニット344は、インター予測ユニット244(特には、動き補償ユニット)と同一であってよく、そして、イントラ予測ユニット354は、機能において、インター予測ユニット254と同一であってよく、そして、パーティション分割及び／又は予測パラメータ、または、エンコーディングされた映像データ21から受信したそれぞれの情報(例えば、エントロピーデコーディングユニット304による、例えば、解析及び／又はデコーディングによるもの)に基づいて、分割またはパーティション分割の決定および予測を実行する。モード適用ユニット360は、予測ブロック365を獲得するために、再構成されたピクチャ、ブロックまたはそれぞれのサンプル(フィルタリングされているか、または、フィルタリングされていない)に基づいて、ブロックごとに予測(イントラ予測またはインター予

10

20

30

40

50

測)を実行するように構成され得る。

#### 【0138】

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ド化されるとき、モード適用ユニット360のイントラ予測ユニット354は、信号化されたイントラ予測モードおよび現在ピクチャの以前にデコーディングされたブロックからのデータに基づいて、現在ビデオスライスのピクチャブロックについて予測ブロック365を生成するように構成されている。ビデオピクチャがインターコード化(すなわち、BまたはP)スライスとしてコード化されるとき、モード適用ユニット360のインター予測ユニット344(例えば、動き補償ユニット)は、エン트로ピーデコーディングユニット304から受信される動きベクトルおよび他のシンタックス要素に基づいて、現在ビデオスライスのビデオブロックについての予測ブロック365を生成するように構成されている。インター予測について、予測ブロックは、参照ピクチャリストのうち1つの中の参照ピクチャの1つから生成され得る。ビデオデコーダ30は、DPB 330内に保管された参照ピクチャに基づいて、デフォルトの構築技術を使用して、参照フレームリスト、リスト0およびリスト1を構築することができる。スライス(例えば、ビデオスライス)に加えて、または、代替的に、タイルグループ(例えば、ビデオタイルグループ)及び／又はタイル(例えば、ビデオタイル)を使用する実施形態について、同一または類似に適用され得る。例えば、ビデオは、I、PまたはBタイルグループ、及び／又は、タイルを使用してコード化され得る。

#### 【0139】

モード適用ユニット360は、動きベクトルまたは関連する情報、および、他のシンタックス要素を解析することによって、現在ビデオスライスのビデオブロックについて予測情報を決定するように構成されており、そして、デコーディングされる現在ビデオブロックの予測ブロックを生成するために、予測情報を使用する。例えば、モード適用ユニット360は、受信されたシンタックス要素のいくつかを使用して、ビデオスライスのビデオブロックをコード化するために使用される予測モード、インター予測のスライスタイプ(例えば、Bスライス、Pスライス、またはGPBスライス)、スライスに対する参照ピクチャリストのうち1つ以上についての構成情報、スライスの各インターエンコーディングされたビデオブロックについての動きベクトル、スライスの各インターコード化されたビデオブロックについてのインター予測ステータス、および、現在ビデオスライス内のビデオブロックをデコーディングするための他の情報を決定する。スライス(例えば、ビデオスライス)に加えて、または、代替的に、タイルグループ(例えば、ビデオタイルグループ)及び／又はタイル(例えば、ビデオタイル)を使用する実施形態について、同一または類似に適用され得る。例えば、ビデオは、I、PまたはBタイルグループ、及び／又は、タイルを使用してコード化され得る。

#### 【0140】

図3に示されるようにビデオデコーダ30の実施形態は、スライス(ビデオスライスとしても、また、参照されるもの)を使用することによって、ピクチャをパーティション分割し、かつ／あるいは、デコーディングするように構成され得る。ここで、ピクチャは、1つ以上のスライス(典型的には、重複していない)を使用して、パーティション分割され、または、デコーディングされ得る。そして、各スライスは、1つ以上のブロック(例えば、CTU)、または、1つ以上のブロックグループ(例えば、タイル(H.265/HEVCおよびVVC)またはブリック(VVC))を含み得る。

#### 【0141】

図3に示されるようにビデオデコーダ30の実施形態は、スライス／タイル・グループ(ビデオタイルグループとしても、また、参照されるもの)、及び／又は、タイル(ビデオタイルとしても、また、参照されるもの)を使用することによって、ピクチャをパーティション分割し、かつ／あるいは、デコーディングするように構成され得る。ここで、ピクチャは、1つ以上のスライス／タイル・グループ(典型的には、重複していない)を使用して、パーティション分割され、または、デコーディングされ得る。そして、各スライスは、各スライス／タイル・グループは、例えば、1つ以上のブロック(例えば、CTU)または1つ

以上のタイルを含み得る。ここで、各タイルは、例えば、矩形形状であってよく、かつ、1つ以上のブロック(例えば、CTU)、例として、完全または分割ブロックを含み得る。

#### 【0142】

エンコーディングされた映像データ21をデコーディングするために、ビデオデコーダ30の他のバリエーションを使用することができる。例えば、デコーダ30は、ループフィルタリングユニット320なしで、出力ビデオストリームを生成することができる。例えば、非変換ベースのデコーダ30は、所定のブロックまたはフレームについて、逆変換処理ユニット312なしで、直接的に残差信号を逆量子化することができる。別の実施形態において、ビデオデコーダ30は、逆量子化ユニット310および逆変換処理ユニット312を単一のユニットへと結合することができる。

10

#### 【0143】

エンコーダ20およびデコーダ30においては、現在ステップの処理結果がさらに処理され、そして、次いで、次のステップに対して出力され得ることが理解されるべきである。例えば、補間フィルタリング、動きベクトル導出、またはループフィルタリングの後で、クリップまたはシフトといった、さらなる操作が、補間フィルタリング、動きベクトル導出、またはループフィルタリングの処理結果について実行され得る。

#### 【0144】

さらなる操作が、現在ブロックの導出された動きベクトル(これらに限定されるわけではないが、アフィンモードの制御点動きベクトル、アフィン、プラナー、ATMVPモードにおけるサブブロック動きベクトル、時間的動きベクトル、などを含んでいる)に対して適用され得ることが、留意されるべきである。例えば、動きベクトルの値は、その表現ビットに従って事前に定義された範囲に制約される。動きベクトルの表現ビットがbitDepthである場合に、範囲は、 $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)}-1$ である。ここで、「 $\wedge$ 」は指数化(exponentiation)を意味している。例えば、bitDepthが16に等しく設定されている場合、範囲は-32768~32767であり、bitDepthが18に等しく設定されている場合、範囲は-131072~131071である。例えば、導出された動きベクトルの値(例えば、1つの8×8ブロックの中の4×4サブブロックのMV)は、4つの4×4サブブロックのMVの整数部の間の最大差が、1ピクセル以下といった、Nピクセル以下であるように制約されている。ここでは、ビット深度に従って、動きベクトルを制約するための2つの方法を提供する。

20

#### 【0145】

図4は、本開示の一つの実施形態に従った、ビデオコーディング装置400の概略図である。ビデオコーディング装置400は、ここにおいて説明されるように、開示される実施形態を実施するのに適している。一つの実施形態において、ビデオコーディング装置400は、図1Aのビデオデコーダ30といったデコーダ、または、図1Aのビデオエンコーダ20といったエンコーダであり得る。

30

#### 【0146】

ビデオコーディング装置400は、データを受信するための入口ポート410(または、入力ポート410)および受信機ユニット(Rx)420、データを処理するためのプロセッサ、論理ユニット、または中央処理装置(CPU)430、データを送信するための送信機ユニット(Tx)440および出口ポート450(または、出力ポート450)、および、データを保管するためのメモリ460、を含んでいる。ビデオコーディング装置400は、また、光信号または電気信号の入口または出口のために、入口ポート410、受信機ユニット420、送信機ユニット440、および、出口ポート450に対して結合された光-電気(optical-to-electrical、OE)コンポーネントおよび電気-光(electrical-to-optical、EO)コンポーネントも含み得る。

40

#### 【0147】

プロセッサ430は、ハードウェアおよびソフトウェアによって実装される。プロセッサ430は、1つ以上のCPUチップ、コア(例えば、マルチコアプロセッサとして)、FPGA、ASIC、およびDSPとして、実装され得る。プロセッサ430は、入口ポート410、受信機ユニット420、送信機ユニット440、出口ポート450、およびメモリ460と通信する。プロ

50

セッサ430は、コーディングモジュール470を含む。コーディングモジュール470は、上述の開示された実施形態を実施する。例えば、コーディングモジュール470は、種々のコーディング操作を実施し、処理し、準備し、または提供する。コーディングモジュール470を含むことは、従って、ビデオコーディング装置400の機能性に対して実質的な改善を提供し、かつ、異なる状態へのビデオコーディング装置400の変換をもたらす。代替的に、コーディングモジュール470は、メモリ460内に保管された命令として実装され、そして、プロセッサ430によって実行される。

#### 【0148】

メモリ460は、1つ以上のディスク、テープドライブ、およびソリッドステートドライブを含み、そうしたプログラムが実行のために選択されたときにプログラムを保管し、そして、プログラム実行の最中に読み出される命令およびデータを保管するために、オーバーフローデータストレージ装置として使用され得る。メモリ460は、例えば、揮発性及び／又は不揮発性であってよく、そして、リードオンリーメモリ（ROM）、ランダムアクセスメモリ（RAM）、三元連想メモリ（Ternary Content-Addressable Memory、TCAM）、及び／又は、スタティックランダムアクセスメモリ（SRAM）であってよい。

#### 【0149】

図5は、一つの例示的な実施形態に従った、図1の送信元デバイス12および宛先デバイス14のいずれかまたは両方として使用することができる装置500の概略ブロック図である。

装置500におけるプロセッサ502は、中央処理装置であり得る。代替的に、プロセッサ502は、現在存在している、または、今後開発される、情報を操作または処理することができる、任意の他のタイプのデバイスまたは複数のデバイスであり得る。開示される実施形態は、図示されるように単一のプロセッサ、例えば、プロセッサ502を用いて実施することができるが、1つより多いプロセッサを使用して、速度および効率における利点が達成され得る。

#### 【0150】

装置500におけるメモリ504は、実装においてリードオンリーメモリ（ROM）デバイスまたはランダムアクセスメモリ（RAM）デバイスであり得る。任意の他の適切なタイプのストレージ装置が、メモリ504として使用され得る。メモリ504は、バス512を使用してプロセッサ502によってアクセスされるコードおよびデータ506を含み得る。メモリ504は、さらに、オペレーティングシステム508およびアプリケーションプログラム510を含み得る。アプリケーションプログラム510は、プロセッサ502がここにおいて説明される方法を実行するのを可能にする少なくとも1つのプログラムを含んでいる。例えば、アプリケーションプログラム510は、アプリケーション1からNまでを含むことができ、さらに、ここにおいて説明される方法を実行するビデオコーディングアプリケーションを含み得る。

装置500は、また、ディスプレイ518といった、1つ以上の出力装置も含み得る。ディスプレイ518は、一つの例では、ディスプレイと、タッチ入力を感じ取るように動作可能なタッチ感応エレメントとを組み合わせたタッチ感応ディスプレイであり得る。ディスプレイ518は、バス512を介してプロセッサ502に結合され得る。

#### 【0151】

ここにおいては、単一のバスとして示されているが、装置500のバス512は、複数のバスで構成され得る。さらに、二次ストレージ装置514は、装置500の他のコンポーネントに直接的に結合され、または、ネットワークを介してアクセスされ得る。そして、メモリカードといった単一の集積ユニット、または、複数のメモリカードといった複数のユニットを含み得る。装置500は、従って、多種多様な構成で実装することができる。

#### 【0152】

スケーラブルコーディング（Scalable coding）

スケーラブルコーディングは、品質スケーラブル（PSNRスケーラブル）、空間スケーラブル、等を含んでいる。例えば、図6に示されるように、シーケンスは、低空間分解能バー

10

20

30

40

50

ジョンヘダウンサンプリングされ得る。低空間分解能バージョンおよび元の空間分解能(高空間分解能)バージョンの両方が、エンコーディングされる。そして、一般的に、低空間分解能が、最初にコード化され、そして、後にコード化される高空間分解能について参照のために使用される。

【0153】

レイヤの情報(番号、依存性、出力)を記述するために、以下のように定義されたVPS(Video Parameter Set)が存在している。

【表1】

|                                                   |      |    |
|---------------------------------------------------|------|----|
| video_parameter_set_rbsp() {                      | 記述子  | 10 |
| ...                                               |      |    |
| <b>vps_max_layers_minus1</b>                      | u(6) |    |
| if( vps_max_layers_minus1 > 0 )                   |      |    |
| <b>vps_all_independent_layers_flag</b>            | u(1) |    |
| for( i = 0; i <= vps_max_layers_minus1; i++ ) {   |      |    |
| <b>vps_layer_id[i]</b>                            | u(6) | 20 |
| if( i > 0 && !vps_all_independent_layers_flag ) { |      |    |
| <b>vps_independent_layer_flag[i]</b>              | u(1) |    |
| if( !vps_independent_layer_flag[i] )              |      |    |
| for( j = 0; j < i; j++ )                          |      |    |
| <b>vps_direct_dependency_flag[i][j]</b>           | u(1) |    |
| }                                                 |      |    |
| }                                                 |      |    |
| ...                                               |      | 30 |
| }                                                 |      |    |

**vps\_max\_layers\_minus1 plus1**は、VPSを参照する各CVSで許可されるレイヤの最大数を指定する。

1に等しい**vps\_all\_independent\_layers\_flag**は、CVS内の全てのレイヤがインターレイヤ予測を使用することなく独立してコード化されることを指定する。

0に等しい**vps\_all\_independent\_layers\_flag**は、CVS内の1つ以上のレイヤがインターレイヤ予測を使用し得ることを指定する。

存在しない場合、**vps\_all\_independent\_layers\_flag**の値は1に等しいものと推定される。

**vps\_all\_independent\_layers\_flag**が1に等しい場合、**vps\_independent\_layer\_flag[i]**の値は1に等しいものと推定される。

**vps\_all\_independent\_layers\_flag**が0に等しい場合、**vps\_independent\_layer\_flag[0]**の値は1と推定される。

**vps\_layer\_id[i]**は、i番目レイヤの**nuh\_layer\_id**値を指定する。mおよびnの2つの非負の整数値について、mがnより小さい場合、**vps\_layer\_id[m]**の値は**vps\_layer\_id[n]**より小さい。

1に等しい**vps\_independent\_layer\_flag[i]**は、インデックスiを有するレイヤがインターレイヤ予測を使用しないことを指定する。

10

20

30

40

50

0に等しいvps\_independent\_layer\_flag[i]は、インデックスiを有するレイヤがインターレイヤ予測を使用することができ、かつ、vps\_layer\_dependency\_flag[i]がVPS内に存在することを指定する。

0に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤについて直接的な参照レイヤではないことを指定する。

1に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤについて直接的な参照レイヤであることを指定する。vps\_direct\_dinercy\_dependency\_flag[i][j]が、0からvps\_max\_layers\_minus1の範囲内でiおよびjについて存在しない場合は、0に等しいものと推定される。

変数DirectDependentLayerIdx[i][j]は、i番目レイヤのj番目の直接的な依存性レイヤを指定しており、以下のように導出される。

【数 1】

```
for( i = 1; i < vps_max_layers_minus1; i-- )
    if( !vps_independent_layer_flag[i] )
        for( j = i, k = 0; j >= 0; j-- )
            if( vps_direct_dependency_flag[i][j] )
                DirectDependentLayerIdx[i][k++] = j
```

(7-2)

変数GeneralLayerIdx[i]は、vps\_layer\_id[i]と等しいnuh\_layer\_idを有するレイヤのレイヤインデックスを指定しており、以下のように導出される。

【数 2】

```
for( i = 0; i <= vps_max_layers_minus1; i++ )
    GeneralLayerIdx[vps_layer_id[i]] = i
```

(7-3)

簡単な説明は、以下のとおりである。

vps\_max\_layers\_minus1 add1は、レイヤの数を意味する。

vps\_all\_independent\_layers\_flagは、全てのレイヤが独立してコード化されているか否かを示す。

vps\_layer\_id[i]は、i番目レイヤのレイヤIDを示す。

vps\_independent\_layer\_flag[i]は、i番目レイヤが独立してコード化されているか否かを示す。

vps\_direct\_dinercy\_dependency\_flag[i][j]は、j番目レイヤがi番目レイヤについて参照のために使用されるか否かを示す。

【 0 1 5 4】

DPB管理と参照ピクチャマーキング

デコーディングプロセスにおけるこれらの参照ピクチャを管理するために、デコーディングされたピクチャは、以降のピクチャデコーディングについて参照使用のために、デコーディングピクチャバッファ（DPB）内に保持することが必要とされる。これらのピクチャを示すために、それらのピクチャオーダーカウンタ(POC)情報は、直接的または間接的にスライスヘッダにおいて信号化する必要がある。一般的には、2つの参照ピクチャリストが存在する。list0およびlist1である。そして、リスト内のピクチャを信号化するために、参照ピクチャインデックスも含めることが必要とされる。単（uni）予測については、1つの参照ピクチャリストから参照ピクチャがフェッチされ、双（bi）予測については、2つの参照ピクチャリストから参照ピクチャがフェッチされる。全ての参照ピクチャはDPB内に保管される。DPB内の全てのピクチャは、「長期参照用（“used for long-term reference”）」、「短期参照用（“used for short-term reference”）」、または

「参照不使用 (“unused for reference”)」としてマーク付けされており、そして、3つのステータスについて1つだけである。一旦、ピクチャが「参照不使用」としてマーク付けされると、それは、参照のために全く使用されない。また、出力のために保管する必要がない場合は、DPBから削除され得る。参照ピクチャのステータスは、スライスヘッダ内で信号化することができ、または、スライスヘッダ情報から導出することができる。RPL(reference picture list)法と呼ばれる、新しい参照ピクチャ管理方法が提案された。RPLは、現在コーディングピクチャについて全体の参照ピクチャセットまたは複数セットを提案し、参照ピクチャセット内の参照ピクチャは、現在ピクチャまたは将来(後の、または、次の)ピクチャデコーディングのために使用される。よって、RPLはDPB内のピクチャ情報を反映しており、現在ピクチャについて参照のために参照ピクチャが使用されていなくても、次のピクチャについて参照のために使用される場合は、RPL内に保存することが必要とされる。

10

ピクチャは、再構成された後で、DPB内に保存され、そして、デフォルトで「短期参照用」としてマーク付けされる。DPB管理操作は、スライスヘッダ内のRPL情報を解析した後で開始される。

#### 【0155】

##### 参照ピクチャリスト構成

参照ピクチャ情報は、スライスヘッダを介して信号化され得る。また、シーケンスパラメータセット(SPS)においていくつかのRPL候補が存在し得る。この場合に、スライスヘッダは、全体のRPLシンタックス構造を信号化することなく、必要とされるRPL情報を獲得するために、RPLインデックスを含み得る。もしくは、RPLシンタックス構造全体が、スライスヘッダ内で信号化され得る。

20

#### 【0156】

##### RPL方法の導入

RPL信号化のコストビットを節約するために、SPS内にいくつかのRPL候補が存在し得る。ピクチャは、SPSからRPL情報を獲得するために、RPLインデックス(ref\_pic\_list\_idx[i])を使用することができる。RPL候補は、以下のように信号化される。

#### 【表2】

|                                                          |       |
|----------------------------------------------------------|-------|
| seq_parameter_set_rbsp() {                               | 記述子   |
| ...                                                      |       |
| <b>rpl1_same_as_rpl0_flag</b>                            | u(1)  |
| for( i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++ ) { |       |
| <b>num_ref_pic_lists_in_sps[i]</b>                       | ue(v) |
| for( j = 0; j < num_ref_pic_lists_in_sps[i]; j++ )       |       |
| ref_pic_list_struct(i, j)                                |       |
| }                                                        |       |
| ...                                                      |       |

30

40

#### 【0157】

セマンティックは、以下のとおりである。

1に等しいrpl1\_sim\_as\_rpl0\_flagは、シンタックス構造num\_ref\_pic\_lists\_in\_sps[1]およびref\_pic\_list\_struct(1,rplsidx)が存在しないことを指定し、そして、以下を適用する。

- num\_ref\_pic\_lists\_in\_sps[1]の値は、num\_ref\_pic\_lists\_in\_sps[0]の値と等し

50

いものと推定される。

—ref\_pic\_list\_struct(1,rplsIdx)における各シンタックス要素の値は、0からnum\_ref\_pic\_lists\_in\_sps[0]-1の範囲のrplsIdxについてref\_pic\_list\_struct(0,rplsIdx)における対応するシンタックス要素の値と等しいものと推定される。

num\_ref\_pic\_lists\_in\_sps[i]は、SPSに含まれる1に等しいlistIdxを有するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造の番号を指定する。num\_ref\_pic\_lists\_in\_sps[i]の値は、0から64まで、を含む範囲とする。

【0 1 5 8】

SPSからのRPLインデックスに基づいてRPL情報を獲得する他に、RPL情報は、スライスヘッダにおいて信号化され得る。

【表 3】

|                                                                                                                                     |       |
|-------------------------------------------------------------------------------------------------------------------------------------|-------|
| slice_header() {                                                                                                                    | 記述子   |
| ...                                                                                                                                 |       |
| if( ( nal_unit_type != IDR_W_RADL && nal_unit_type !=<br>IDR_N_LP )   <br>sps_idr_rpl_present_flag ) {                              |       |
| for( i = 0; i < 2; i++ ) {                                                                                                          |       |
| if( num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i]<br>&&<br>( i == 0    ( i == 1 &&<br>rpl1_idx_present_flag ) ) ) |       |
| <b>ref_pic_list_sps_flag[i]</b>                                                                                                     | u(1)  |
| if( ref_pic_list_sps_flag[i] ) {                                                                                                    |       |
| if( num_ref_pic_lists_in_sps[i] > 1 &&<br>( i == 0    ( i == 1 &&<br>rpl1_idx_present_flag ) ) )                                    |       |
| <b>ref_pic_list_idx[i]</b>                                                                                                          | u(v)  |
| } else                                                                                                                              |       |
| ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])                                                                                 |       |
| for( j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++ ) {                                                                              |       |
| if( ltrp_in_slice_header_flag[i][RplsIdx[i]] )                                                                                      |       |
| <b>slice_poc_lsb_lt[i][j]</b>                                                                                                       | u(v)  |
| <b>delta_poc_msb_present_flag[i][j]</b>                                                                                             | u(1)  |
| if( delta_poc_msb_present_flag[i][j] )                                                                                              |       |
| <b>delta_poc_msb_cycle_lt[i][j]</b>                                                                                                 | ue(v) |
| }                                                                                                                                   |       |
| }                                                                                                                                   |       |
| ...                                                                                                                                 |       |

10

20

30

40

50

1に等しいref\_pic\_list\_sps\_flag[i]は、現在スライスの参照ピクチャリストiが、SPS内でiに等しいlistIdxを有するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造の1つに基づいて導出されることを指定する。

0に等しいref\_pic\_list\_sps\_flag[i]は、現在スライスの参照ピクチャリストiが、現在ピクチャのスライスヘッダ内に直接的に含まれるiに等しいlistIdxを有するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造に基づいて導出されることを指定する。

ref\_pic\_list\_sps\_flag[i]が存在しない場合は、以下が適用される。

— num\_ref\_pic\_lists\_in\_sps[i]が0に等しい場合、ref\_pic\_list\_sps\_flag[i]の値は0に等しいものと推定される。

— そうでなければ(num\_ref\_pic\_lists\_in\_sps[i]が0より大きい)、rpl1\_idx\_present\_flagが0に等しい場合、ref\_pic\_list\_sps\_flag[1]の値はref\_pic\_list\_sps\_flag[0]に等しいものと推定される。

— そうでなければ、ref\_pic\_list\_sps\_flag[i]の値は、pps\_ref\_pic\_list\_sps\_idc[i]-1に等しいものと推定される。

ref\_pic\_list\_idx[i]は、SPS内に含まれるiに等しいlistIdxを有するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造のリストの中へ、現在ピクチャの参照ピクチャリストiの導出のために使用されるiに等しいlistIdxを有するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造のインデックスを指定する。

シンタックス要素ref\_pic\_list\_idx[i]は、Ceil(Log2(num\_ref\_pic\_lists\_in\_sps[i]))ビットによって表される。

存在しない場合、ref\_pic\_list\_idx[i]の値は0に等しいものと推定される。ref\_pic\_list\_idx[i]の値は、0からnum\_ref\_pic\_lists\_in\_sps[i]-1まで、を含む範囲とする。

ref\_pic\_list\_sps\_flag[i]が1に等しく、かつ、num\_ref\_pic\_lists\_in\_sps[i]が1に等しい場合、ref\_pic\_list\_idx[i]の値は0に等しいものと推定される。

ref\_pic\_list\_sps\_flag[i]が1に等しく、かつ、rpl1\_idx\_present\_flagが0に等しい場合、ref\_pic\_list\_idx[1]の値はref\_pic\_list\_idx[0]に等しいものと推定される。

変数RplsIdx[i]は、以下のように導出される。

【数 3】

$$\begin{aligned} \text{RplsIdx}[i] &= \text{ref\_pic\_list\_sps\_flag}[i] ? \text{ref\_pic\_list\_idx}[i] : \\ &\quad \text{num\_ref\_pic\_lists\_in\_sps}[i] \end{aligned} \quad (7-95)$$

slice\_poc\_lsb\_lt[i][j]は、i番目の参照ピクチャリスト内のj番目のLTRPエントリのピクチャオーダークラウドモジュールMaxPicOrderCntLsbの値を指定する。slice\_poc\_lsb\_lt[i][j]シンタックス要素の長さは、log2\_max\_pic\_order\_cnt\_lsb\_minus4+4ビットである。

変数PocLsbLt[i][j]は、次のように導出される。

【数 4】

$$\begin{aligned} \text{PocLsbLt}[i][j] &= \text{ltrp\_in\_slice\_header\_flag}[i][\text{RplsIdx}[i]] ? \\ &\quad \text{slice\_poc\_lsb\_lt}[i][j] : \text{rpls\_poc\_lsb\_lt}[\text{listIdx}][\text{RplsIdx}[i]][j] \end{aligned} \quad (7-96)$$

1に等しいdelta\_poc\_msb\_present\_flag[i][j]は、delta\_poc\_msb\_cycle\_lt[i][j]が存在することを指定する。

0に等しいdelta\_poc\_msb\_present\_flag[i][j]は、delta\_poc\_msb\_cycle\_lt[i][j]が存在しないことを指定する。

prevTid0Picを、現在ピクチャと同じnuh\_layer\_idを有し、0に等しいTemporalIdを有し、かつ、RASLまたはRADLピクチャではない、デコーディング順序における以前のピクチャとする。

setOfPrevPocValsを、以下で構成される集合とする。

- － prevTid0PicのPicOrderCntVal、
- － prevTid0PicのRefPicList[0]またはRefPicList[1]内のエントリによって参照され、かつ、現在ピクチャと同じnuh\_layer\_idを有する、各ピクチャのPicOrderCntVal、
- － デコーディング順序においてprevTid0Picに続く、各ピクチャのPicOrderCntValは、現在ピクチャと同じnuh\_layer\_idを有し、かつ、デコーディング順序において現在ピクチャに先行する。

setOfPrevPocVals内に値モジュロMaxPicOrderCntLsbがPocLsbLt[i][j]に等しい値が1個より多く存在する場合、delta\_poc\_msb\_present\_flag[i][j]の値は1に等しい。  
delta\_poc\_msb\_cycle\_lt[i][j]は、変数FullPocLt[i][j]の値を以下のように指定する。

【数 5】

```

if( j == 0 )
    DeltaPocMsbCycleLt[i][j] = delta_poc_msb_cycle_lt[i][j]
else
    DeltaPocMsbCycleLt[i][j] = delta_poc_msb_cycle_lt[i][j] +
    DeltaPocMsbCycleLt[i][j-1]
FullPocLt[i][j] = PicOrderCntVal - DeltaPocMsbCycleLt[i][j] *
MaxPicOrderCntLsb -
(PicOrderCntVal & (MaxPicOrderCntLsb-1)) + PocLsbLt[i][j]

```

(7-97)

delta\_poc\_msb\_cycle\_lt[i][j]の値は、0から $2^{(32-\log_2 \max\_pic\_order\_cnt\_lsb\_minus4-4)}$ まで、を含む範囲とする。

存在しない場合、delta\_poc\_msb\_cycle\_lt[i][j]の値は0に等しいものと推定される。

【 0 1 5 9 】

RPLのシンタックス構造は、以下のとおりである。

10

20

30

40

50

【表 4】

|                                                                   |       |
|-------------------------------------------------------------------|-------|
| ref_pic_list_struct(listIdx, rplsIdx) {                           | 記述子   |
| <b>num_ref_entries</b> [listIdx][rplsIdx]                         | ue(v) |
| if( long_term_ref_pics_flag )                                     |       |
| <b>ltrp_in_slice_header_flag</b> [listIdx][rplsIdx]               | u(1)  |
| for( i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++ ) { |       |
| if( inter_layer_ref_pics_present_flag )                           |       |
| <b>inter_layer_ref_pic_flag</b> [listIdx][rplsIdx][i]             | u(1)  |
| if( !inter_layer_ref_pics_flag[listIdx][rplsIdx][i] ) {           |       |
| if( long_term_ref_pics_flag )                                     |       |
| <b>st_ref_pic_flag</b> [listIdx][rplsIdx][i]                      | u(1)  |
| if( st_ref_pic_flag[listIdx][rplsIdx][i] ) {                      |       |
| <b>abs_delta_poc_st</b> [listIdx][rplsIdx][i]                     | ue(v) |
| if( AbsDeltaPocSt[listIdx][rplsIdx][i] > 0 )                      |       |
| <b>strp_entry_sign_flag</b> [listIdx][rplsIdx][i]                 | u(1)  |
| } else if( !ltrp_in_slice_header_flag[listIdx][rplsIdx] )         |       |
| <b>rpls_poc_lsb_lt</b> [listIdx][rplsIdx][j++]                    | u(v)  |
| } else                                                            |       |
| <b>ilrp_idc</b> [listIdx][rplsIdx][i]                             | ue(v) |
| }                                                                 |       |
| }                                                                 |       |

num\_ref\_entries[listIdx][rplsIdx]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造のエントリ数を指定する。num\_ref\_entries[listIdx][rplsIdx]の値は、0からsps\_max\_dec\_pic\_buffering\_minus1+14まで、を含む範囲とする。

0に等しいltrp\_in\_slice\_header\_flag[listIdx][rplsIdx]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のLTRPエントリのPOC LSBがref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内に存在することを指定する。

1に等しいltrp\_in\_slice\_header\_flag[listIdx][rplsIdx]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のLTRPエントリのPOC LSBがref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内に存在しないことを指定する。

1に等しいinter\_layer\_ref\_pic\_flag[listIdx][rplsIdx][i]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のi番目のエントリがILRPエントリであることを指定する。

0に等しいinter\_layer\_ref\_pic\_flag[listIdx][rplsIdx][i]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のi番目のエントリがILRPエントリではないことを指定する。存在しない場合、inter\_layer\_ref\_pic\_flag[listIdx][rplsIdx][i]の値は0に等しいものと推定される。

1に等しいst\_ref\_pic\_flag[listIdx][rplsIdx][i]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のi番目のエントリがSTRPエントリであることを指定する。

0に等しいst\_ref\_pic\_flag[listIdx][rplsIdx][i]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造のi番目のエントリがLTRPエントリであることを指定する。

inter\_layer\_ref\_pic\_flag[listIdx][rplsIdx][i]が0に等しく、かつ、st\_ref\_pic\_flag[listIdx][rplsIdx][i]が存在しない場合、st\_ref\_pic\_flag[listIdx][rplsIdx][i]の値は1に等しいものと推定される。

変数NumLtrpEntries[listIdx][rplsIdx]は、以下のように導出される。

【数 6】

```
for( i = 0, NumLtrpEntries[listIdx][rplsIdx] = 0; i <
num_ref_entries[listIdx][rplsIdx]; i++ )
```

10

```
    if( !inter_layer_ref_pic_flag[listIdx][rplsIdx][i]&&!st_ref_pic_flag[listIdx][rplsIdx][i] )
        NumLtrpEntries[listIdx][rplsIdx]++
```

(7-120)

abs\_delta\_poc\_st[listIdx][rplsIdx][i]は、変数AbsDeltaPocSt[listIdx][rplsIdx][i]の値を、以下のように指定する。

【数 7】

```
if( sps_weighted_pred_flag || sps_weighted_bipred_flag )
```

20

```
    AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i]
```

```
(7-121)
```

```
else
```

```
    AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i]+1
```

abs\_delta\_poc\_st [listIdx][rplsIdx][i]の値は、0から $2^{15}-1$ まで、を含む範囲とする。1に等しいstrp\_entry\_entry\_sign\_flag[listIdx][rplsIdx][i]は、シンタックス構造ref\_pic\_list\_struct(listIdx,rplsIdx)内のi番目のエントリが0以上の値を有することを指定する。

30

0に等しいstrp\_entry\_sign\_flag[listIdx][rplsIdx][i]は、シンタックス構造ref\_pic\_list\_struct(listIdx,rplsIdx)内のi番目のエントリが0未満の値を有することを指定する。存在しない場合、strp\_entry\_sign\_flag[listIdx][rplsIdx][i]の値は1に等しいものと推定される。

リストDeltaPocValSt[listIdx][rplsIdx]は、以下のように導出される。

【数 8】

```
for( i = 0; i < num_ref_entries[listIdx][rplsIdx]; i++ )
```

```
    if( !inter_layer_ref_pic_flag[listIdx][rplsIdx][i]&&st_ref_pic_flag[listIdx][rplsIdx][i] )
```

40

```
(7-122)
```

```
        DeltaPocValSt[listIdx][rplsIdx][i] =
```

```
( strp_entry_sign_flag[listIdx][rplsIdx][i] ) ?
```

```
        AbsDeltaPocSt[listIdx][rplsIdx][i] : 0 -
```

```
AbsDeltaPocSt[listIdx][rplsIdx][i]
```

rpls\_poc\_lsb\_lt[listIdx][rplsIdx][i]は、ref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造内のi番目のエントリによって参照されるピクチャのピクチャオーダーカウントモジュールMaxPicOrderCntLsbの値を指定する。

50

`rpls_poc_lsb_lt[listIdx][rplsIdx][i]` シンタックス要素の長さは、`log2_max_pic_order_cnt_lsb_minus4+4`ビットである。

【0160】

RPL構造に係るいくつかの一般的な説明

各リストについて、RPL構造が存在する。最初に、リスト内の参照ピクチャの数を示すために`num_ref_entries[listIdx][rplsIdx]`が信号化される。`ltrp_in_slice_header_flag[listIdx][rplsIdx]`は、LSB(最下位ビット)情報がスライスヘッダ内で信号化されているか否かを示すために使用される。現在参照ピクチャがインターレイヤ参照ピクチャでない場合、`st_ref_pic_flag[listIdx][rplsIdx][i]`は、それが長期(long-term)参照ピクチャであるか否かを示す。それが短期(short-term)参照ピクチャである場合、POC情報(`abs_delta_poc_st`および`strp_entry_sign_flag`)が信号化される。`ltrp_in_in_slice_header_flag[listIdx][rplsIdx]`がゼロである場合、現在参照ピクチャのLSB情報を導出するために`rpls_poc_lsb_lt[listIdx][rplsIdx][j+++]`が使用される。MSB(最上位ビット)は、直接的に導出され得るか、または、スライスヘッダ内の情報(`delta_poc_msb_present_flag[i][j]`および`delta_poc_msb_cycle_lt[i][j]`)に基づいて導出され得る。

【0161】

参照ピクチャリスト構成のためのデコーディングプロセス

このプロセスは、非IDRピクチャの各スライスについて、デコーディングプロセスの開始において呼び出される。

参考ピクチャは、参考インデックスを通して扱われる。参照インデックスは、参照ピクチャリストのインデックスである。Iスライスをデコーディングするとき、参照ピクチャリストは、スライスデータのデコーディングにおいて使用されない。Pスライスをデコーディングするとき、参照ピクチャリスト0(すなわち、`RefPicList[0]`)のみが、スライスデータのデコーディングにおいて使用される。Bスライスをデコーディングするとき、参照ピクチャリスト0および参照ピクチャリスト1(すなわち、`RefPicList[1]`)の両方が、スライスデータのデコーディングにおいて使用される。

非IDRピクチャの各スライスに対するデコーディングプロセスの開始において、参照ピクチャリスト`RefPicList[0]`および`RefPicList[1]`が導出される。参照ピクチャリストは、8.3.3条項に規定される参照ピクチャのマーキングにおいて、または、スライスデータのデコーディングにおいて使用される。

注記1—それがピクチャの最初のスライスではない、非IDRピクチャのIスライスについて、ビットストリーム適合性チェックのために`RefPicList[0]`および`RefPicList[1]`が導出され得るが、それらの導出は、現在ピクチャ、または、デコーディング順序において現在ピクチャに続くピクチャをデコーディングするためには必要ではない。それがピクチャの最初のスライスではない、Pスライスについて、ビットストリーム適合性検査のために`RefPicList[1]`が導出され得るが、その導出は、現在ピクチャ、または、デコーディング順序において現在ピクチャに続くピクチャをデコーディングするためには必要ではない。

参照ピクチャリスト`RefPicList[0]`および`RefPicList[1]`は、以下のように構成されている。

【数9-1】

10

20

30

40

50

```

for( i = 0; i < 2; i++ ) {
    for( j = 0, k = 0, pocBase = PicOrderCntVal; j <
num_ref_entries[i][RplsIdx[i]]; j++ ) {
        if( !inter_layer_ref_pic_flag[i][RplsIdx[i]][j] ) {
            if( st_ref_pic_flag[i][RplsIdx[i]][j] ) {
                RefPicPocList[i][j] = pocBase-DeltaPocValSt[i][RplsIdx[i]][j]
                if( there is a reference picture picA in the DPB with the same
nuh_layer_id as the current picture
10
                    and PicOrderCntVal equal to RefPicPocList[i][j] )
                    RefPicList[i][j] = picA
            else
                RefPicList[i][j] = "no reference picture"
(8-5)
                pocBase = RefPicPocList[i][j]
            } else {
                if( !delta_poc_msb_cycle_lt[i][k] ) {
                    if( there is a reference picA in the DPB with the same nuh_layer_id
20
as the current picture and
                        PicOrderCntVal&(MaxPicOrderCntLsb-1) equal to
PocLsbLt[i][k] )
                            RefPicList[i][j] = picA
                        else
                            RefPicList[i][j] = "no reference picture"
                            RefPicLtPocList[i][j] = PocLsbLt[i][k]
                    } else {
                        if( there is a reference picA in the DPB with the same nuh_layer_id
30
as the current picture and
                            PicOrderCntVal equal to FullPocLt[i][k] )
                                RefPicList[i][j] = picA
                            else
                                RefPicList[i][j] = "no reference picture"
                                RefPicLtPocList[i][j] = FullPocLt[i][k]
                    }
                }
            k++
40

```

【数 9 - 2】

```

    }
  } else {
    layerIdx =
DirectDependentLayerIdx[GeneralLayerIdx[nuh_layer_id]][ilrp_idc[i]][RplsIdx][j
]]
    refPicLayerId = vps_layer_id[layerIdx]
    if( there is a reference picture picA in the DPB with nuh_layer_id equal
to refPicLayerId and
        the same PicOrderCntVal as the current picture )
      RefPicList[i][j] = picA
    else
      RefPicList[i][j] = "no reference picture"
  }
}
}

```

10

20

RPLが構築された後で、ここでは、refPicLayerIdがILRPのレイヤIDであり、PicOrderCntValがpicAのPOC値であり、マーキングプロセスは、いかのとおりである。

【0162】

参照ピクチャマーキングのためのデコーディングプロセス

このプロセスは、条項8.3.2に指定されているように、スライスヘッダのデコーディング、および、スライスについて参照ピクチャリスト構成のためのデコーディングプロセスの後であるが、スライスデータのデコーディングの以前に、ピクチャ毎に1回呼び出される。この処理は、「参照不使用」または「長期参照用」としてマーク付けされているDPB内の1つ以上の参照ピクチャを結果として生じ得る。

DPB内のデコーディングされたピクチャは、「参照不使用」、「短期参照用」、または「長期参照用」としてマーク付けされ得るが、デコーディングプロセスの動作中の任意所与の時点では、これら3つのうち1つのみである。これらのマーキングの1つをピクチャに割り当てることは、適用可能な場合は、これらのマーキングの別のものを暗黙のうちに削除する。ピクチャが「参照用」としてマーク付けされているものとして参照される場合、これは、「短期参照用」または「長期参照用」(両方ではない)としてマーク付けされているピクチャを集合的に参照する。

30

STRPおよびILRPは、それらのnuh\_layer\_idおよびPicOrderCntValの値によって識別される。LTRPは、nuh\_layer\_id値およびPそれらのPicOrderCntVal値のLog2(MaxLt PicOrderCntLsb)LSBによって識別される。

現在ピクチャがCLVSSピクチャの場合、現在ピクチャと同じnuh\_layer\_idを有するDPB内の現在の全ての参照ピクチャは（もしあれば）、「参照不使用」としてマーク付けされる。

40

そうでなければ、以下が適用される。

－ RefPicList[0]またはRefPicList[1]内の各LTRPエントリについて、参照されるピクチャが現在ピクチャと同じnuh\_layer\_idを有するSTRPである場合、そのピクチャは「長期参照用」としてマーク付けされる。

－ RefPicList[0]またはRefPicList[1]内の任意のエントリによって参照されないDPB内の現在ピクチャと同じnuh\_layer\_idを有する各参照ピクチャは、「参照不使用」としてマーク付けされる。

－ RefPicList[0]またはRefPicList[1]内の各ILRPエントリについて、参照されるピク

50

チャは「長期参照用」としてマーク付けされる。

【0163】

ここでは、ILRP(インターレイヤ参照ピクチャ)が「長期参照用」としてマーク付けされていることに注意すること。

【0164】

SPS内には、インターレイヤ参照情報に関連する2つのシンタックスが存在する。

【表5】

|                                   |      |
|-----------------------------------|------|
| seq_parameter_set_rbsp() {        | 記述子  |
| ...                               |      |
| sps_video_parameter_set_id        | u(4) |
| ...                               |      |
| long_term_ref_pics_flag           | u(1) |
| inter_layer_ref_pics_present_flag | u(1) |
| sps_idr_rpl_present_flag          | u(1) |
| rpl1_same_as_rpl0_flag            | u(1) |
|                                   |      |
|                                   |      |
| ...                               |      |

sps\_video\_parameter\_set\_idは、0より大きい場合、SPSによって参照されるVPSについてvps\_video\_parameter\_set\_idの値を指定する。sps\_video\_parameter\_set\_idが0に等しい場合、SPSはVPSを参照せず、そして、SPSを参照して各CVSをデコーディングするときに、VPSは参照されない。

0に等しいlong\_term\_ref\_pics\_flagは、CVS内の任意のコード化ピクチャのインター予測のためにLTRPが使用されないことを指定する。1に等しいlong\_term\_ref\_pics\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにLTRPが使用され得ることを指定する。

0に等しいinter\_layer\_ref\_pics\_present\_flagは、CVS内の任意のコード化ピクチャのインター予測のためにILRPが使用されないことを指定する。1に等しいinter\_layer\_ref\_pics\_present\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにILRPが使用され得ることを指定する。sps\_video\_parameter\_set\_idが0に等しい場合、inter\_layer\_ref\_pics\_present\_flagの値は0であるものと推定される。

【0165】

簡単な説明は、以下のとおりである。

long\_term\_ref\_pics\_flagは、デコーディングプロセスにおいてLTRPが使用され得るか否かを示すために使用される。

inter\_layer\_ref\_pics\_present\_flagは、デコーディングプロセスにおいてILRPが使用され得るか否かを示すために使用される。

【0166】

よって、inter\_layer\_ref\_pics\_present\_flagが1に等しい場合、デコーディングプロセスで使用されるILRPが存在してよく、「長期参照用」としてマーク付けされる。この場合には、たとえlong\_term\_ref\_pics\_flagが0に等しくても、デコーディングプロセスにおいて使用されるLTRPが存在する。よって、long\_term\_ref\_pics\_flagのセマンティックとの矛盾が存在する。

既存の方法において、インターレイヤ参照情報のためのいくつかのシンタックス要素は、現在レイヤのインデックスを考慮することなく、常に信号化されている。この発明は、信号化効率を改善するために、シンタックス要素に対していくつかの条件を加えることを提案する。

【0167】

long\_term\_ref\_pics\_flagは、ltrp\_in\_slice\_header\_flagおよびst\_ref\_pic\_flagの解析を制御するためだけに使用されるので、セマンティック (semantic) はRPLにおけるフラグ解析の解析を制御するために修正される。

インターレイヤ参照情報のためのシンタックス要素は、現在レイヤのインデックスを考慮して信号化される。情報が現在レイヤのインデックスによって導出できる場合、情報は信号化される必要がない。

10

【0168】

long\_term\_ref\_pics\_flagは、ltrp\_in\_slice\_header\_flagおよびst\_ref\_pic\_flagの解析を制御するためだけに使用されるので、セマンティックはRPLにおけるフラグ解析の解析を制御するために修正される。

インターレイヤ参照情報のためのシンタックス要素は、現在レイヤのインデックスを考慮して信号化される。情報が現在レイヤのインデックスによって導出できる場合、情報は信号化される必要がない。

【0169】

本発明の第1実施形態[セマンティック](LTRPおよびILRPの矛盾を除去するために、long\_term\_ref\_pics\_flagのセマンティックを修正する)

20

【0170】

long\_term\_ref\_pics\_flagは、ltrp\_in\_slice\_header\_flagおよびst\_ref\_pic\_flagの解析を制御するためだけに使用されるので、セマンティックが以下のように修正される。  
1に等しいlong\_term\_ref\_pics\_flagは、ltrp\_in\_slice\_header\_flag、st\_ref\_pic\_flagがシンタックス構造ref\_pic\_list\_struct(listIdx,rplIdx)内に存在することを指定する。  
0に等しいlong\_term\_ref\_pics\_flagは、これらのシンタックス要素がシンタックス構造ref\_pic\_list\_struct(listIdx,rplIdx)内に存在しないことを指定する。

また、セマンティックは、ILRPを除外するために、以下のようにも修正され得る。

0に等しいlong\_term\_ref\_pics\_flagは、CVS内のコード化ピクチャのインター予測のためにLTRPが使用されないことを指定する。1に等しいlong\_term\_ref\_pics\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにLTRPが使用され得ることを指定する。ここで、LTRPはILRP(インターレイヤ参照ピクチャ)を含まない。

30

【0171】

本発明の第2実施形態[VPS]

【0172】

提案1 : vps\_direct\_direct\_dependency\_flag[i][j]の条件付き信号化(インターレイヤ参照情報は、現在レイヤのインデックスを考慮して信号化され、冗長情報信号化を除去し、コーディング効率を改善する。)

【0173】

40

オプション1. A :

ここで、iが1に等しい場合、それはレイヤ1が他のレイヤを参照する必要があることに注意すること。一方、レイヤ0のみが参照レイヤであり得るので、vps\_direct\_direct\_dependency\_flag[i][j]は、信号化される必要はない。iが1より大きい場合のみ、vps\_direct\_direct\_dependency\_flag[i][j]は、信号化される必要がある。

50

【表 6】

|                                                   |      |
|---------------------------------------------------|------|
| video_parameter_set_rbsp() {                      | 記述子  |
| <b>vps_video_parameter_set_id</b>                 | u(4) |
| <b>vps_max_layers_minus1</b>                      | u(6) |
| if( vps_max_layers_minus1 > 0 )                   |      |
| <b>vps_all_independent_layers_flag</b>            | u(1) |
| for( i = 0; i <= vps_max_layers_minus1; i++ ) {   |      |
| <b>vps_layer_id[i]</b>                            | u(6) |
| if( i > 0 && !vps_all_independent_layers_flag ) { |      |
| <b>vps_independent_layer_flag[i]</b>              | u(1) |
| if( i > 1 && !vps_independent_layer_flag[i] )     |      |
| for( j = 0; j < i; j++ )                          |      |
| <b>vps_direct_dependency_flag[i][j]</b>           | u(1) |
| }                                                 |      |
| }                                                 |      |
| if( vps_max_layers_minus1 > 0 ) {                 |      |
| ...                                               |      |

0に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤではないことを指定する。  
 1に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤであることを指定する。

vps\_direct\_dependency\_flag[i][j]が、iおよびjについて、0からvps\_max\_layers\_minus1まで、を含む範囲内に存在しない場合、iが1に等しく、かつ、vps\_independent\_layer\_flag[i]が0に等しければ、vps\_direct\_dependency\_flag[i][j]は、1に等しいものと推定され、そうでなければ、0に等しいものと推定される。

【0174】

#### オプション1.B:

上記の実装方法(オプション1.A)の他に、オプション1.Bが存在する。これは、i、および0からi-1まで、を含む範囲内のjについて、および、vps\_independent\_layer\_flag[i]が0に等しく、かつ、0からi-2まで、を含む範囲のjについて、vps\_direct\_dependency\_flag[i][j]の全ての値が0に等しい場合、vps\_direct\_dependency\_flag[i][i-1]の値は信号化される必要がなく、そして、1に等しいものと推定される、ことを意味する。

10

20

30

40

50

【表 7】

|                                                      |      |
|------------------------------------------------------|------|
| video_parameter_set_rbsp() {                         | 記述子  |
| <b>vps_video_parameter_set_id</b>                    | u(4) |
| <b>vps_max_layers_minus1</b>                         | u(6) |
| if( vps_max_layers_minus1 > 0 )                      |      |
| <b>vps_all_independent_layers_flag</b>               | u(1) |
| for( i = 0; i <= vps_max_layers_minus1; i++ ) {      |      |
| <b>vps_layer_id[i]</b>                               | u(6) |
| if( i > 0 && !vps_all_independent_layers_flag ) {    |      |
| <b>vps_independent_layer_flag[i]</b>                 | u(1) |
| if(!vps_independent_layer_flag[i] )                  |      |
| for(j = 0, SumDependencyFlag=0; j < i; j++ ) {       |      |
| if( !(j == i-1 && SumDependencyFlag == 0) )          |      |
| <b>vps_direct_dependency_flag[i][j]</b>              | u(1) |
| SumDependencyFlag+= vps_direct_dependency_flag[i][j] |      |
| }                                                    |      |
| }                                                    |      |
| }                                                    |      |
| if( vps_max_layers_minus1 > 0 ) {                    |      |
| ...                                                  |      |

0に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤではないことを指定する。1に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤであることを指定する。

vps\_direct\_dependency\_flag[i][j]が、iおよびjについて、0からvps\_max\_layers\_minus1まで、を含む範囲内に存在しない場合、vps\_independent\_layer\_flag[i]が0に等しく、かつ、jが0に等しく、そして、SumDependencyFlagの値が0に等しければ、vps\_direct\_dependency\_flag[i][j]は、1に等しいものと推定され、そうでなければ、0に等しいものと推定される。

#### 【0175】

提案2：vps\_direct\_dependency\_flag[i][j]のセマンティックにおける制約

また、我々は、シンタックス信号化方法、またはシンタックステーブルを変更することなく、vps\_direct\_dependency\_flag[i][j]のセマンティックに制約を加えることができる。基本的に、iについて、インデックスiを有するレイヤが従属レイヤ(vps\_independent\_layer\_flag[i]が0に等しい)であれば、vps\_direct\_dependency\_flag[i][j]の少なくとも1つの値は、jが0からi-1の範囲で、1に等しい。代替的に、vps\_direct\_dependency\_flag[i][j]の合計は、jが0からi-1の範囲で、0に等しくないべきである。または、1以上(例えば、>=1)であるべきである。もしくは、0より大きい(例えば、>0)であるべきである。

【0176】

オプション2．A：

0に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤではないことを指定する。1に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤであることを指定する。

vps\_direct\_dinercy\_dependency\_flag[i][j]は、iおよびjについて、0からvps\_max\_layers\_minus1まで、を含む範囲内に存在しない場合、0に等しいものと推定される。ここで、i、および0からi-1まで、を含む範囲内のjについて、および、vps\_independent\_layer\_flag[i]が0に等しい場合、vps\_direct\_direct\_dependency\_flag[i][j]の合計は、0より大きい。

10

【0177】

オプション2．B：

0に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤではないことを指定する。1に等しいvps\_direct\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤであることを指定する。vps\_direct\_dinercy\_dependency\_flag[i][j]は、iおよびjについて、0からvps\_max\_layers\_minus1まで、を含む範囲の内に存在しない場合、0に等しいものと推定される。ここで、i、および0からi-1まで、を含む範囲内のjについて、かつ、vps\_independent\_layer\_flag[i]が0に等しい場合、vps\_direct\_direct\_dependency\_flag[i][j]の少なくとも1つの値は、1に等しい。

20

【0178】

提案3：提案1 + 提案2

【0179】

オプション3：

実際には、オプション1とオプション2を組み合わせる他の実装方法とすることができる。オペレーション1．B + オペレーション2．Bも同様である。

30

40

50

【表 8】

|                                                      |      |
|------------------------------------------------------|------|
| video_parameter_set_rbsp() {                         | 記述子  |
| <b>vps_video_parameter_set_id</b>                    | u(4) |
| <b>vps_max_layers_minus1</b>                         | u(6) |
| if( vps_max_layers_minus1 > 0 )                      |      |
| <b>vps_all_independent_layers_flag</b>               | u(1) |
| for( i = 0; i <= vps_max_layers_minus1; i++ ) {      |      |
| <b>vps_layer_id[i]</b>                               | u(6) |
| if( i > 0 && !vps_all_independent_layers_flag ) {    |      |
| <b>vps_independent_layer_flag[i]</b>                 | u(1) |
| if(!vps_independent_layer_flag[i] )                  |      |
| for(j = 0, SumDependencyFlag=0; j < i; j++ ) {       |      |
| if( !(j == i-1 && SumDependencyFlag == 0) )          |      |
| <b>vps_direct_dependency_flag[i][j]</b>              | u(1) |
| SumDependencyFlag+= vps_direct_dependency_flag[i][j] |      |
| }                                                    |      |
| }                                                    |      |
| }                                                    |      |
| if( vps_max_layers_minus1 > 0 ) {                    |      |
| ...                                                  |      |

0に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤではないことを指定する。1に等しいvps\_direct\_dependency\_flag[i][j]は、インデックスjを有するレイヤが、インデックスiを有するレイヤのための直接的な参照レイヤであることを指定する。

vps\_direct\_dependency\_flag[i][j]が、iおよびjについて、0からvps\_max\_layers\_minus1まで、を含む範囲の内に存在しない場合、vps\_independent\_layer\_flag[i]が0に等しく、かつ、jがi-1に等しく、そして、SumDependencyFlagの値が0に等しければ、vps\_direct\_dependency\_flag[i][j]は、1に等しいものと推測され、そうでなければ、0に等しいものと推測される。ここで、i、および0からi-1まで、を含む範囲内のjについて、および、vps\_independent\_layer\_flag[i]が0に等しい場合、vps\_direct\_dependency\_flag[i][j]の少なくとも1つの値は、1に等しい。

#### 【0180】

結合方法は、ここにおいては限定されず、以下のようにすることもできる。

オペレーション1．A + オペレーション2．Bも同様である。

オペレーション1．A + オペレーション2．Aも同様である。

オペレーション1．B + オペレーション2．Aも同様である。

#### 【0181】

本発明[sps]の第3実施形態[sps](インターレイヤ参照情報は、現在レイヤのインデックスを考慮して信号化され、冗長情報信号化を除去し、コーディング効率を改善する。)

ここでは、sps\_video\_parameter\_set\_idが0に等しい場合、それは、複数のレイヤが存在しないこと、よって、inter\_layer\_ref\_pics\_flagを信号化する必要は存在しないこと、そして、フラグは、デフォルトによって0であること、を意味することに注意すること。

【表 9】

|                                     |      |
|-------------------------------------|------|
| seq_parameter_set_rbsp() {          | 記述子  |
| ...                                 |      |
| sps_video_parameter_set_id          | u(4) |
| ...                                 |      |
| long_term_ref_pics_flag             | u(1) |
| if( sps_video_parameter_set_id > 0) |      |
| inter_layer_ref_pics_present_flag   | u(1) |
| sps_idr_rpl_present_flag            | u(1) |
| rpl1_same_as_rpl0_flag              | u(1) |
|                                     |      |
|                                     |      |
| ...                                 |      |

0に等しいinter\_layer\_ref\_pics\_present\_flagは、CVS内の任意のコード化ピクチャのインター予測のためにILRPが使用されないことを指定する。1に等しいinter\_layer\_ref\_pics\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにILRPが使用され得ることを指定する。inter\_layer\_ref\_pics\_flagが存在しない場合、それは0に等しいものと推定される。

ここでは、GeneralLayerIdx[nuh\_layer\_id]が0に等しい場合、現在レイヤは0番目のレイヤであり、あらゆる他のレイヤを参照できないことに注意する。よって、inter\_layer\_ref\_pics\_present\_flagを信号化する必要は存在せず、そして、値は、デフォルトによって0である。

【表 10】

|                                       |      |
|---------------------------------------|------|
| seq_parameter_set_rbsp() {            | 記述子  |
| ...                                   |      |
| sps_video_parameter_set_id            | u(4) |
| ...                                   |      |
| long_term_ref_pics_flag               | u(1) |
| if(GeneralLayerIdx[nuh_layer_id] > 0) |      |
| inter_layer_ref_pics_present_flag     | u(1) |
| sps_idr_rpl_present_flag              | u(1) |
| rpl1_same_as_rpl0_flag                | u(1) |

0に等しいinter\_layer\_ref\_pics\_present\_flagは、CVS内のコード化ピクチャのインター予測のためにILRPが使用されないことを指定する。1に等しいinter\_layer\_ref\_pics\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにILRPが使用され得ることを指定する。inter\_layer\_ref\_pics\_flagが存在しない場合、それは0に等しいものと推定される。

上記の両方のケースをコーディングして、別のアプリケーションの例を以下に示す。

【表 1 1】

|                                                                          |      |
|--------------------------------------------------------------------------|------|
| seq_parameter_set_rbsp() {                                               | 記述子  |
| ...                                                                      |      |
| sps_video_parameter_set_id                                               | u(4) |
| ...                                                                      |      |
| long_term_ref_pics_flag                                                  | u(1) |
| if( sps_video_parameter_set_id > 0 && GeneralLayerIdx[nuh_layer_id] > 0) |      |
| inter_layer_ref_pics_present_flag                                        | u(1) |
| sps_idr_rpl_present_flag                                                 | u(1) |
| rpl1_same_as_rpl0_flag                                                   | u(1) |
|                                                                          |      |
|                                                                          |      |
| ...                                                                      |      |

0に等しいinter\_layer\_ref\_pics\_present\_flagは、CVS内のコード化ピクチャのインター予測のためにILRPが使用されないことを指定する。1に等しいinter\_layer\_ref\_pics\_flagは、CVS内の1つ以上のコード化ピクチャのインター予測のためにILRPが使用され得ることを指定する。inter\_layer\_ref\_pics\_flagが存在しない場合、それは0に等しいものと推定される。

【 0 1 8 2 】

#### 本発明の第4実施形態[RPL]

ここでは、GeneralLayerIdx[nuh\_layer\_id]が1に等しい場合、現在レイヤはレイヤ1であり、かつ、レイヤ0のみを参照することでき、一方で、レイヤ0のilrp\_idcは0でなければならないこと、に注意する。よって、この場合には、irp\_idcを信号化する必要は存在しない。

【表 1 2】

|                                                                   |       |
|-------------------------------------------------------------------|-------|
| ref_pic_list_struct(listIdx, rplsIdx) {                           | 記述子   |
| <b>num_ref_entries</b> [listIdx][rplsIdx]                         | ue(v) |
| if( long_term_ref_pics_flag )                                     |       |
| <b>ltrp_in_slice_header_flag</b> [listIdx][rplsIdx]               | u(1)  |
| for( i = 0, j = 0; i < num_ref_entries[listIdx][rplsIdx]; i++ ) { |       |
| if( inter_layer_ref_pics_present_flag )                           |       |
| <b>inter_layer_ref_pic_flag</b> [listIdx][rplsIdx][i]             | u(1)  |
| if( !inter_layer_ref_pics_flag[listIdx][rplsIdx][i] ) {           |       |
| if( long_term_ref_pics_flag )                                     |       |
| <b>st_ref_pic_flag</b> [listIdx][rplsIdx][i]                      | u(1)  |
| if( st_ref_pic_flag[listIdx][rplsIdx][i] ) {                      |       |
| <b>abs_delta_poc_st</b> [listIdx][rplsIdx][i]                     | ue(v) |
| if( AbsDeltaPocSt[listIdx][rplsIdx][i] > 0 )                      |       |
| <b>strp_entry_sign_flag</b> [listIdx][rplsIdx][i]                 | u(1)  |
| } else if( !ltrp_in_slice_header_flag[listIdx][rplsIdx] )         |       |
| <b>rpls_poc_lsb_lt</b> [listIdx][rplsIdx][j++]                    | u(v)  |
| } else if( GeneralLayerIdx[nuh_layer_id] > 1 )                    |       |
| <b>ilrp_idc</b> [listIdx][rplsIdx][i]                             | ue(v) |
| }                                                                 |       |
| }                                                                 |       |
| }                                                                 |       |

ilrp\_idc[listIdx][rplsIdx][i]は、直接的に依存性レイヤのリストについて、直接的に依存性レイヤのリストに対するref\_pic\_list\_struct(listIdx,rplsIdx)シンタックス構造におけるi番目のエントリのILRPに係るインデックスを指定する。ilrp\_idc[listIdx][rplsIdx][i]の値は、0からGeneralLayerIdx[nuh\_layer\_id]-1まで、を含む範囲内にあるとする。GeneralLayerIdx[nuh\_layer\_id]が1に等しい場合、ilrp\_idc[listIdx][rplsIdx][i]の値は0に等しいものと推定される。

【0183】

本発明の第5実施形態[組み合わせ]

ここでは、実施形態1から実施形態4の実施形態の一部または全てを組み合わせ、新たな実施形態を形成することができることに注意する。

例えば、実施形態1＋実施形態2＋実施形態3＋実施形態4、または、実施形態2＋実施形態3＋実施形態4、もしくは、他の組み合わせ、である。

【0184】

以下は、上述の実施形態に示されるような、エンコーディング方法、並びに、デコーディング方法、および、それらを使用するシステムに係るアプリケーションの説明である。

【0185】

図7は、コンテンツ配信サービスを実現するためのコンテンツ供給システム3100を示しているブロック図である。このコンテンツ供給システム3100は、キャプチャ装置3102、

端末装置3106を含み、そして、任意的に、ディスプレイ3126を含んでいる。キャプチャ装置3102は、通信リンク3104を介して端末装置3106と通信する。通信リンクは、上述の通信チャンネル13を含み得る。通信リンク3104は、これらに限定されるわけではないが、WIFI（登録商標）、イーサネット（登録商標）、ケーブル、無線(3G/4G/5G)、USB、または、それらの任意の種類の組み合わせ、などを含む。

#### 【0186】

キャプチャ装置3102は、データを生成し、そして、上記の実施形態に示されるように、エンコーディング方法によってデータをエンコーディングし得る。代替的に、キャプチャ装置3102は、ストリーミングサーバ(図に示されていない)に対してデータを分配することができ、そして、サーバは、データをエンコーディングし、かつ、エンコーディングされたデータを端末装置3106へ送信する。キャプチャ装置3102は、これらに限定されるわけではないが、カメラ、スマートフォンまたはパッド、コンピュータまたはラップトップ、ビデオ会議システム、PDA、車載装置、または、それらの任意の組み合わせ、などを含む。例えば、キャプチャ装置3102は、上述のように、送信元デバイス12を含み得る。データがビデオを含む場合、キャプチャ装置3102に含まれるビデオエンコーダ20は、ビデオエンコーディング処理を実際に実行し得る。データがオーディオ(すなわち、音声)を含む場合、キャプチャ装置3102に含まれるオーディオエンコーダは、オーディオエンコーディング処理を実際に実行し得る。いくつかの実際的なシナリオについて、キャプチャ装置3102は、エンコーディングされたビデオおよび音声データを、それらを一緒に多重化(multiplexing)することによって分配する。他の実際的なシナリオについて、例えば、ビデオ会議システムにおいては、エンコーディングされた音声データおよびエンコーディングされたビデオデータは、多重化されない。キャプチャ装置3102は、エンコーディングされた音声データおよびエンコーディングされたビデオデータを、端末装置3106に別々に分配する。

#### 【0187】

コンテンツ供給システム3100において、端末装置310は、エンコーディングされたデータを受信し、そして、再生(reproduce)する。端末装置3106は、スマートフォンまたはパッド3108、コンピュータまたはラップトップ3110、ネットワークビデオレコーダ(NVR)／デジタルビデオレコーダ(DVR)3112、TV3114、セットトップボックス(STB)3116、ビデオ会議システム3118、ビデオ監視システム3120、パーソナルデジタルアシスタント(PDA)3122、車載装置3124、または、これらの任意の組み合わせといった、データ受信および回復(recovering)能力を有する装置、またはそのような、上記のエンコーディングされたデータをデコーディングすることができるものであってよい。例えば、端末装置3106は、上述のように宛先デバイス14を含み得る。エンコーディングされたデータがビデオを含む場合、端末装置に含まれるビデオデコーダ30は、ビデオデコーディングを実行するように優先付けされる。エンコーディングされたデータがオーディオを含む場合、端末装置に含まれるオーディオデコーダは、オーディオデコーディング処理を実行するように優先付けされる。

#### 【0188】

自身のディスプレイを有する端末装置、例えば、スマートフォンまたはパッド3108、コンピュータまたはラップトップ3110、ネットワークビデオレコーダ(NVR)／デジタルビデオレコーダ(DVR)3112、TV3114、パーソナルデジタルアシスタント(PDA)3122、または車載装置3124、について、端末装置は、デコーディングされたデータをそのディスプレイにフィード(feed)することができる。ディスプレイを備えない端末装置、例えば、STB3116、ビデオ会議システム3118、またはビデオ監視システム3120については、デコーディングされたデータを受信し、そして、表示するために外部ディスプレイ3126がそこに接続さる。

#### 【0189】

このシステムにおける各装置がエンコーディングまたはデコーディングを実行する場合、上述の実施形態に示されるように、映像エンコーディング装置または映像デコーディン

グ装置が使用され得る。

【0190】

図8は、端末装置3106の一つの例に係る構成を示されている図である。端末装置3106がキャプチャ装置3102からストリームを受信した後で、プロトコル処理ユニット3202は、ストリームの送信プロトコルを分析する。このプロトコルは、これらに限定されるわけではないが、リアルタイムストリーミングプロトコル(RTSP)、ハイパーテキスト転送プロトコル(HTTP)、HTTPライブストリーミングプロトコル(HLS)、MPEG-DASH、リアルタイムトランスポートプロトコル(RTP)、リアルタイムメッセージプロトコル(RTMP)、または、これらの任意の種類の組み合わせ、を含む。

【0191】

プロトコル処理ユニット3202がストリームを処理した後で、ストリームファイルが生成される。ファイルは、逆多重化(demultiplexing)ユニット3204に対して出力される。逆多重化ユニット3204は、多重化されたデータをエンコーディングされた音声データおよびエンコーディングされたビデオデータへと分離することができる。上述の、いくつかの実際的なシナリオについて、例えば、ビデオ会議システムにおいて、エンコーディングされた音声データおよびエンコーディングされたビデオデータは多重化されない。この状況において、エンコーディングされたデータは、逆多重化ユニット3204を介することなく、ビデオデコーダ3206およびオーディオデコーダ3208へ送信される。

【0192】

逆多重化処理を介して、ビデオエレメンタリーストリーム(ES)、オーディオES、および、任意的にサブタイトルが生成される。上述の実施形態において説明したようにビデオデコーダ30を含む、ビデオデコーダ3206は、上述の実施形態で示されたようなデコーディング方法によって、ビデオESをデコーディングして、ビデオフレームを生成し、そして、このデータを同期ユニット3212へフィードする。オーディオデコーダ3208は、オーディオフレームを生成するためにオーディオESをデコーディングし、そして、このデータを同期ユニット3212へフィードする。代替的に、ビデオフレームは、同期ユニット3212へそれを供給する前に、バッファ(図Yに示されていない)に保管してよい。同様に、オーディオフレームは、同期ユニット3212へそれを供給する前に、バッファ(図Yに示されていない)に保管してよい。

【0193】

同期ユニット3212は、ビデオフレームとオーディオフレームとを同期させ、そして、ビデオ／オーディオディスプレイ3214に対してビデオ／オーディオを供給する。例えば、同期ユニット3212は、ビデオとオーディオ情報の表現を同期させる。情報は、コード化されたオーディオおよびビジュアルデータの表現に関するタイムスタンプ、および、データストリーム自身の配送に関するタイムスタンプを使用して、シンタックスでコード化することができる。

【0194】

サブタイトルがストリームに含まれる場合、サブタイトルデコーダ3210は、サブタイトルをデコーディングし、そして、ビデオフレームおよびオーディオフレームとそれを同期させ、そして、ビデオ／オーディオ／サブタイトルをビデオ／オーディオ／サブタイトルディスプレイ3216へ供給する。

【0195】

本発明は、上述のシステムに限定されるものではなく、そして、上述の実施形態における映像エンコーディング装置または映像デコーディング装置のいずれかは、他のシステム、例えば、車両システムの中に組み込まれ得る。

【0196】

数学演算子(Mathematical Operators)

このアプリケーションで使用される数学演算子は、Cプログラミング言語で 사용되는演算子と同様である。しかしながら、整数除算演算と算術シフト演算の結果がより正確に定義されており、そして、指数化(exponentiation)および実数値除算(real-valued

10

20

30

40

50

division) といった追加的な演算が定義されている。番号付けとカウントの規定 (convention) は、一般的に0から始まる。例えば、「1番 (“the first”)」は0番目 (0-th) と同等であり、「2番 (“the second”)」は2番目と同等である、等。

【0197】

算術演算子 (Arithmetic operators)

以下の算術演算子は、次のように定義される。

+ 加算。

- 減算(2引数の演算子として)または否定(単項前置演算子として)。

\* 乗算、マトリクス乗算を含む。

x y 指数。xのy乗を指定する。他のコンテキストにおいて、そうした表記は、指数関数としての解釈を意図しない上付き文字として使用される。

10

/ 結果がゼロに向けて切り捨てられる整数除算。例えば、7/4および-7/-4は1に切り捨てられ、そして、-7/4および7/-4は-1に切り捨てられる。

÷ 切り捨て又は丸めが意図されていない、数式における除算を示すために使用される。

【数10】

$$\frac{x}{y}$$

20

切り捨て又は丸めが意図されていない、数式における除算を示すために使用される。

【数11】

$$\sum_{i=x}^y f(i)$$

f(i)の和であり、iは、xからyまでの全ての整数値をとる。

x % y モジュロ。xをyで割った値の余りであり、x ≥ 0かつy > 0である整数xおよびyに対してのみ定義される。

30

【0198】

論理演算子 (Logical operators)

以下の論理演算子は、次のように定義される。

x & y xおよびyに係るブール論理「and」。

x | y xおよびyに係るブール論理「or」。

! ブール論理「not」。

x ? y : z xが真 (TRUE) または0でない場合はyの値で評価する。そうでなければ、はzの値で評価する。

【0199】

40

関係演算子 (Relational operators)

以下の関係演算子は、次のように定義される。

> より大きい。

>= より大きいか、または、等しい。

< より小さい。

<= より小さいか、または、等しい。

= 等しい。

!= 等しくない。

【0200】

値「na」(not applicable、該当せず)が割り当てられたシンタックス要素または変数

50

に対して関係演算子が適用される場合、値「na」は、シンタックス要素または変数の別個の値 (distinct value) として扱われる。値「na」は、あらゆる他の値と等しくないものとみなされる。

#### 【0201】

ビット単位の演算子 (Bit-wise operators)

以下のビット単位の演算子は、次のように定義される。

& ビット単位の「and」。整数引数について操作する場合、整数値の2の補数表示について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合、より短い引数は、0に等しいより最上位のビットを加えることによって拡張される。

| ビット単位の「or」。整数引数について操作する場合、整数値の2の補数表示について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合、より短い引数は、0に等しいより最上位のビットを加えることによって拡張される。

^ ビット単位の「exclusive or」。整数引数について操作する場合、整数値の2の補数表示について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合、より短い引数は、0に等しいより最上位のビットを加えることによって拡張される。

x >> y xの2の補数整数表示をyの二進数だけ算術的に右シフト。この関数は、yの負でない整数値に対してのみ定義される。右シフトの結果として最上位ビット(MSB)へとシフトされたビットは、シフト演算の以前のxのMSBに等しい値を有する。

x << y xの2の補数整数表示をyの二進数だけ算術的に左シフト。この関数は、yの負でない整数値に対してのみ定義される。左シフトの結果として最下位ビット(LSB)にシフトされたビットは、0に等しい値を有する。

#### 【0202】

代入演算子 (Assignment operator)

以下の算術演算子は、次のように定義される。

= 代入演算子。

++ 増加、すなわちx++は、x=x+1と等価である。アレイ・インデックスにおいて使用される場合、増加演算の以前の変数の値で評価する。

-- 減少、すなわちx--は、x=x-1と等価であり、アレイ・インデックスにおいて使用される場合、減少演算の以前の変数の値で評価する。

+= 指定された量だけの増分、すなわちx+=3は、x=x+3と等価であり、かつ、x+=(-3)は、x=x+(-3)と等価である。

-= 指定された量だけの減少、すなわちx-=3は、x=x-3と等価であり、かつ、x-=(-3)は、x=x-(-3)と等価である。

#### 【0203】

範囲表記 (Range notation)

以下の表記が、値の範囲を指定するために使用される。

x = y . . z xはyからzまで、を含む整数値を取り、x、y、およびzは、整数であり、かつ、zはより大きい。

#### 【0204】

数学的関数

以下の数学的関数が定義される。

#### 【数12】

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

Asin(x) 三角法の逆正弦 (sine) 関数。-1.0から1.0まで、を含む範囲内にある引数xについて動作する。出力値は、 $-\pi \div 2$ から $\pi \div 2$ まで、を含む範囲内にあり、単位はラジアンである。

**Atan(x)** 三角法の逆正接 (tangent) 関数。引数xについて動作する。出力値は、 $-\pi \div 2$  から  $\pi \div 2$  まで、を含む範囲内にあり、単位はラジアンである。

【数 1 3】

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x == 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases}$$

10

**Ceil(x)** x以上の最小整数。

**Clip1<sub>y</sub>(x)**=Clip3(0,(1 < <BitDepth<sub>y</sub>)-1,x)

**Clip1<sub>c</sub>(x)**=Clip3(0,(1 < <BitDepth<sub>c</sub>)-1,x)

【数 1 4】

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

20

**cos(x)** 三角法の余弦 (cos) 関数。引数xについて動作する。

**Floor(x)** x以下の最大整数。

【数 1 5】

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d / 2 \\ c - d & ; \quad a - b > d / 2 \\ c & ; \quad \text{otherwise} \end{cases}$$

**Ln(x)** xの自然対数(基底-eの対数。ここで、eは自然対数基底定数2.718281828...である)。

30

**log2(x)** xに係る基底2の対数。

**log10(x)** xに係る基数10の対数。

【数 1 6】

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

【数 1 7】

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

40

**Round(x)**=Sign(x)\*Floor(Abs(x)+0.5)

【数 1 8】

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

**Sin(x)** 三角法の正弦 (sine) 関数。引数xについて動作し、単位はラジアンである。

50

Sqrt(x)= $\sqrt{x}$   
Swap(x,y)=(y,x)  
Tan(x) 三角法の正接 (tangent) 関数。引数xについて動作し、単位はラジアンである。  
【 0 2 0 5 】

演算の優先順位 (Order of operation precedence)

式における優先順位が、括弧を使用して明示的に示されていない場合、以下の規則が適用される。

- ー より高い優先順位の演算は、より低い優先順位の演算の前に評価される。
- ー 同じ優先順位の演算は、左から右へ連続的に評価される。

以下のテーブルは、最高から最低まで演算の優先順位を示しており、テーブルにおけるより高い位置は、より高い優先順位を示している。

Cプログラミング言語においても、また、使用される演算子について、この明細書で使用される優先順位は、Cプログラミング言語において使用される優先順位と同じである。

【表 1 3】

|                                                     |
|-----------------------------------------------------|
| 演算 (オペランドは x, y,および z)                              |
| "x++", "x--"                                        |
| "!x", "-x" (単項前置演算子として)                             |
| $x^y$                                               |
| "x*y", "x/y", "x÷y", " $\frac{x}{y}$ ", "x%y"       |
| "x+y", "x-y" (2 引数の演算子として), " $\sum_{i=x}^y f(i)$ " |
| "x<<y", "x>>y"                                      |
| "x<y", "x<=y", "x>y", "x>=y"                        |
| "x==y", "x!=y"                                      |
| "x&y"                                               |
| "x y"                                               |
| "x&& y"                                             |
| "x  y"                                              |
| "x?y:z"                                             |
| "x..y"                                              |
| "x=y", "x+=y", "x-=y"                               |

テーブル：最高(テーブル上部)から最低(テーブル下部)までの演算の優先順位

【 0 2 0 6 】

論理演算のテキスト記述

テキストにおいて、論理演算のステートメントは、以下の形式で数学的に記述される。

【数 1 9】

```

if( condition 0 )
    statement 0
else if( condition 1 )
    statement 1
...
else /* informative remark on remaining condition */
    statement n

```

10

上記は、以下の方法で記述され得る。

【表 1 4】

... as follows / ... the following applies:

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- ...
- Otherwise (informative remark on remaining condition), statement n

20

テキストにおける各 “If...otherwise, if...Otherwise, ...” ステートメントは、直後に “if ..” が続く “...as follows” または “...the following applies” と共に導入される。“If...otherwise, if...Otherwise, ...” の最後の条件は、常に “Otherwise, ...” である。インターリーブされたIf...otherwise, if...Otherwise, ...” ステートメントは、“...as follows” または “...the following applies” をエンディングの “Otherwise, ...” とマッチングすることによって特定され得る。

【0 2 0 7】

論理演算のテキスト記述

テキストにおいて、論理演算のステートメントは、以下の形式で数学的に記述される。

30

【数 2 0】

```

if( condition 0a  &&  condition 0b )
    statement 0
else if( condition 1a  ||  condition 1b )
    statement 1
...
else
    statement n

```

40

上記は、以下の方法で記述され得る。

## 【表 1 5】

... as follows / ... the following applies:

- If all of the following conditions are true, statement 0:
  - condition 0a
  - condition 0b
- Otherwise, if one or more of the following conditions are true, statement 1:
  - condition 1a
  - condition 1b
- ...
- Otherwise, statement n

10

## 【0 2 0 8】

テキストにおいて、論理演算のステートメントは、以下の形式で数学的に記述される。

```
if(condition 0)
    statement 0
if(condition 1)
    statement 1
```

20

上記は、以下の方法で記述され得る。

```
When condition 0, statement 0
When condition 1, statement 1
```

## 【0 2 0 9】

本発明の実施形態は、主にビデオコーディングに基づいて説明されてきたが、コーディングシステム10、エンコーダ20、およびデコーダ30(および対応するシステム10)の実施形態、並びに、ここにおいて説明される他の実施形態も、また、静止映像処理またはコーディング、すなわち、ビデオコーディングにおけるように、任意の先行する又は連続するピクチャから独立した個々のピクチャの処理またはコーディング、のために構成され得ることが留意されるべきである。一般的には、映像処理コーディングが単一のピクチャ17に限定される場合、インター予測ユニット244(エンコーダ)および344(デコーダ)のみが利用可能でないことがある。ビデオエンコーダ20およびビデオデコーダ30の他の全ての機能性(ツールまたはテクノロジーとしても、また、参照される)は、静止映像処理、例えば、残差計算204/304、変換206、量子化208、逆量子化210/310、(逆)変換212/312、パーティション分割262/362、イントラ予測254/354、及び／又は、ループフィルタリング220、320、および、エントロピーコーディング270とエントロピーデコーディング304、について等しく使用され得る。

30

## 【0 2 1 0】

例えば、エンコーダ20およびデコーダ30の実施形態、並びに、例えば、エンコーダ20およびデコーダ30に関連して、ここにおいて説明される機能は、ハードウェア、ソフトウェア、ファームウェア、または、それらの任意の組み合わせで実施され得る。ソフトウェアにおいて実施される場合、機能は、コンピュータ読取り可能な媒体に保管され、または、1つ以上の命令またはコードとして通信媒体を介して送信され、そして、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。コンピュータ読取り可能な媒体は、データストレージ媒体といった有形の媒体に対応する、コンピュータ読取り可能な記憶媒体、または、例えば、通信プロトコルに従って、1つの場所から他の場所へのコンピュータプログラムの転送を促進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、コンピュータ読取り可能な媒体は、一般的に、(1)非一時的である有形のコンピュータ読取り可能な記憶媒体、または、(2)信号または搬送波といった通信媒体、に対応し得る。データスト

40

50

レンジ媒体は、この開示において説明される技術の実施のための命令、コード、及び／又はデータ構造を検索するために、1つ以上のコンピュータまたは1つ以上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ってよい。コンピュータプログラム製品は、コンピュータ読取り可能な媒体を含み得る。

#### 【0211】

特には、図9で示されるように、デコーダにおいて実装されたコード化されたビデオビットストリームをデコーディングする方法が提供されている。本方法は、以下を含む。S901、コード化されたビデオビットストリームから、第1レイヤがインターレイヤ予測を使用するか否かを指定している第1シンタックス要素(すなわち、vps\_independent\_layer\_flag[i])を獲得すること。S902、コード化されたビデオビットストリームから、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素(すなわち、vps\_direct\_direct\_dependency\_flag[i][j])を獲得することであり、各第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを指定し、ここで、第1レイヤはインターレイヤ予測を使用することが許されると第1シンタックス要素の値が指定する場合に、1つ以上の第2シンタックス要素のうち少なくとも1つの第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであることを指定する値を有する。および、S903、少なくとも1つの第2シンタックス要素に関連する第2レイヤのピクチャを参照ピクチャとして使用することによって、第1レイヤのピクチャについてインターレイヤ予測を実行すること。

#### 【0212】

同様に、図10で示されるように、エンコーダにおいて実装されるコード化データを含むビデオビットストリームをエンコーディングする方法が提供される。本方法は、以下を含む。S1001、少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを決定すること。S1003、シンタックス要素をコード化されたビデオビットストリームへとエンコーディングすることであり、ここで、シンタックス要素は第1レイヤがインターレイヤ予測を使用するか否かを指定し、ここで、少なくとも1つの第2レイヤのいずれも第1レイヤの直接的な参照レイヤではない場合、シンタックス要素の値は、第1レイヤがインターレイヤ予測を使用しないことを指定する。

#### 【0213】

図11は、複数のピクチャについてコード化データを含むビデオビットストリームをデコーディングするために構成されたデコーダ1100を示す。示された例に従ったデコーダ1100は、獲得ユニット1110および予測ユニット1120を含む。獲得ユニット1110は、コード化されたビデオビットストリームから、第1レイヤがインターレイヤ予測を使用するか否かを指定している第1シンタックス要素を獲得するように構成されている。獲得ユニット1110は、さらに、1つ以上の第2レイヤに関連する1つ以上の第2シンタックス要素を獲得するように構成されており、各第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを指定する。ここで、第1レイヤはインターレイヤ予測を使用することが許されると第1シンタックス要素の値が指定する場合に、1つ以上の第2シンタックス要素のうち少なくとも1つの第2シンタックス要素は、第2レイヤが第1レイヤの直接的な参照レイヤであることを指定する値を有する。そして、予測ユニット1120は、少なくとも1つの第2シンタックス要素に関連する第2レイヤのピクチャを参照ピクチャとして使用することによって、第1レイヤのピクチャについてインターレイヤ予測を実行するように構成されている。

#### 【0214】

ここで、ユニットは、プロセッサによる実行のためのソフトウェアモジュール、または、処理回路であってよい。

#### 【0215】

ここで、獲得ユニット1110は、エントロピーデコーディングユニット304であってよい。予測ユニット1120は、インター予測ユニット344であってよい。デコーダ1100は、宛先デバイス14、デコーダ30、装置500、ビデオデコーダ3206、または、端末装置310

6であってよい。

【0216】

同様に、図12で示されるように、複数のピクチャについてコード化データを含むビデオビットストリームをエンコーディングするように構成されたエンコーダ1200が提供される。エンコーダ1200は、決定ユニット1210およびエンコーディングユニット1220を含む。決定ユニット1210は、少なくとも1つの第2レイヤが第1レイヤの直接的な参照レイヤであるか否かを決定するように構成されている。エンコーディングユニット1220は、コード化されたビデオビットストリームへとシンタックス要素をエンコーディングするように構成されている。ここで、シンタックス要素は、第1レイヤがインターレイヤ予測を使用するか否かを指定する。ここで、少なくとも1つの第2レイヤのいずれも第1レイヤの直接的な参照レイヤではない場合、シンタックス要素の値は、第1レイヤがインターレイヤ予測を使用しないことを指定する。

10

【0217】

ここで、ユニットは、プロセッサによる実行のためのソフトウェアモジュール、または、処理回路であってよい。

【0218】

第1エンコーディングユニット1210および第2エンコーディングユニット1220は、エントリピーエンコーディングユニット270であってよい。決定ユニットは、モード選択ユニット260であってよい。エンコーダ1200は、送信元デバイス12、エンコーダ20、または装置500であってよい。

20

【0219】

一つの例として、かつ、限定するものではなく、そうしたコンピュータ読取り可能なストレージ媒体は、RAM、ROM、EEPROM、CD-ROMまたは他の光ディスクストレージ装置、磁気ディスクストレージ装置、または他の磁気ストレージ装置、フラッシュメモリ、または、命令またはデータ構造の形態で所望のプログラムコードを保管するために使用され、かつ、コンピュータによってアクセスされ得る他の任意の媒体、を含むことができる。また、任意の接続は、適切にコンピュータ読取り可能な媒体と呼ばれる。例えば、同軸ケーブル、光ファイバケーブル、ツイストペア、デジタル加入者線（DSL）、または、赤外線、無線、およびマイクロ波といった無線技術を使用して、ウェブサイト、サーバ、または他のリモートソースから、命令が送信される場合に、同軸ケーブル、光ファイバケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波といった無線技術は、媒体の定義に含まれる。しかしながら、コンピュータ読取り可能なストレージ媒体およびデータストレージ媒体は、接続、搬送波、信号、または他の一時的な媒体を含むものではないが、代わりに、非一時的な、有形のストレージ媒体に向けられることが理解されるべきである。ディスク（disk and disc）は、ここにおいて使用されるように、コンパクトディスク（CD）、レーザディスク、光ディスク、デジタル多用途ディスク（DVD）、フロッピー（登録商標）ディスク、およびブルーレイディスクを含む。ここで、ディスクは、たいてい、磁気的にデータを再生し、一方で、ディスクは光学的にレーザーを用いてデータを再生する。上記の組み合わせも、また、コンピュータ読取り可能な媒体の範囲内に含まれるべきである。

30

【0220】

命令は、1つ以上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論理アレイ（FPGA）、または、他の同等な集積または個別論理回路といった、1つ以上のプロセッサによって実行され得る。従って、ここにおいて使用されるように、用語「プロセッサ（“processor”）」は、前述の構造のいずれか、または、ここにおいて説明される技術の実施に適した任意の他の構造を参照し得る。加えて、いくつかの態様において、ここにおいて説明される機能は、エンコーディングおよびデコーディングのために構成されている専用ハードウェア及び／又はソフトウェアモジュールの中で提供されてよく、または、組み合わされたコーデックに組み込まれてよい。また、本技術は、1つ以上の回路または論理素子においても完全に

40

50

実装され得る。

【0221】

この開示の技術は、ワイヤレスハンドセット、集積回路（IC）または一組のIC(例えば、チップセット)を含む、多種多様なデバイスまたは装置において実施され得る。種々のコンポーネント、モジュール、またはユニットが、開示された技術を実施するように構成されたデバイスの機能的な態様を強調するために、この開示において説明されているが、異なるハードウェアユニットによる実現を必ずしも必要とするものではない。むしろ、上述のように、種々のユニットは、コーデックハードウェアユニット内で組み合わせられてよく、または、適切なソフトウェア及び／又はファームウェアと併せて、上述のような1つ以上のプロセッサを含む、相互運用（interoperative）ハードウェアユニットのコレクションによって提供されてよい。

10

20

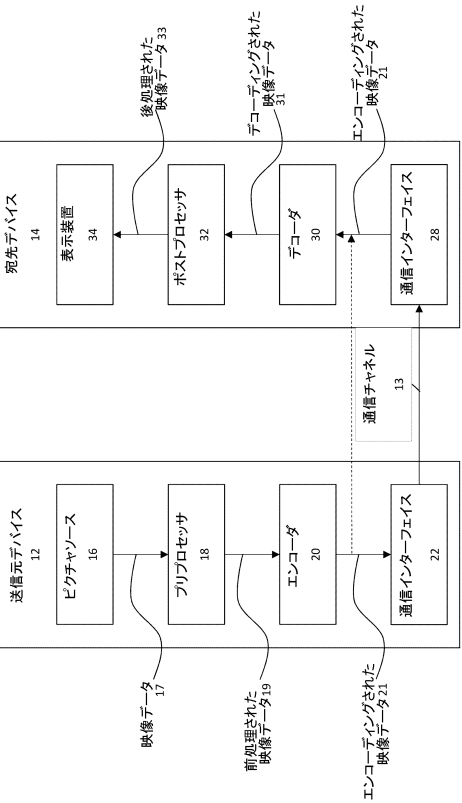
30

40

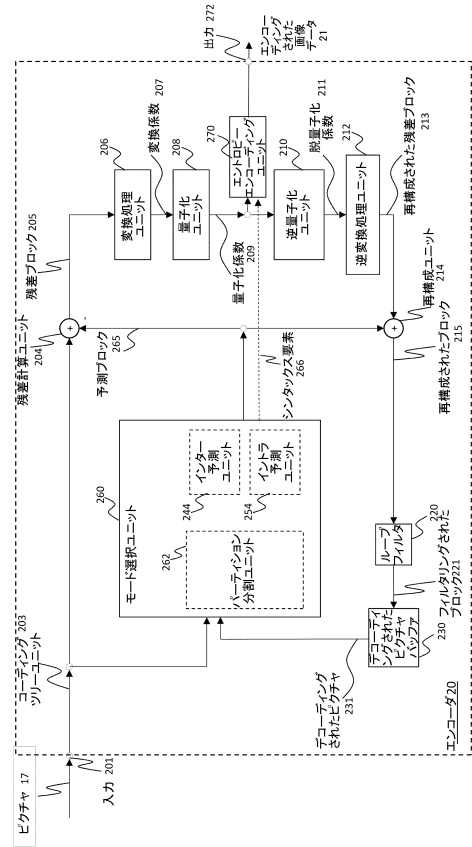
50

【図面】

【図 1 A】



【図 2】



【図 1 B】

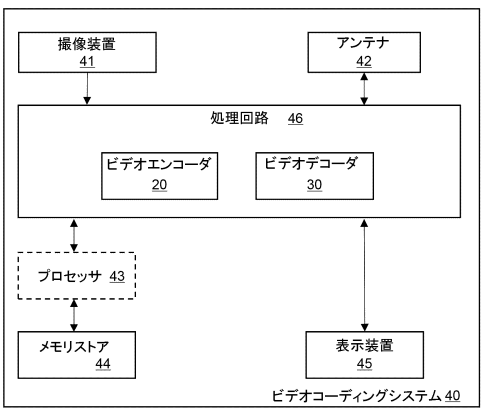


FIG. 1B

【図 3】

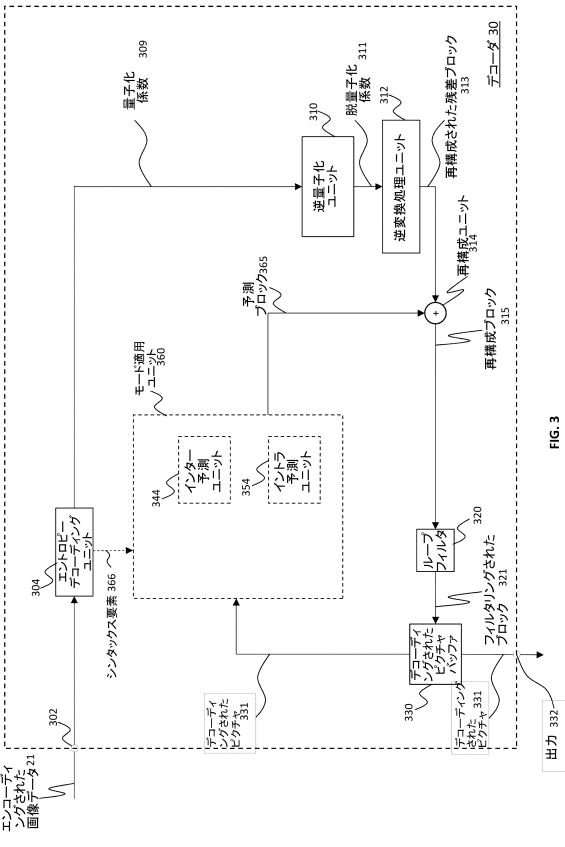


FIG. 3

【図 4】

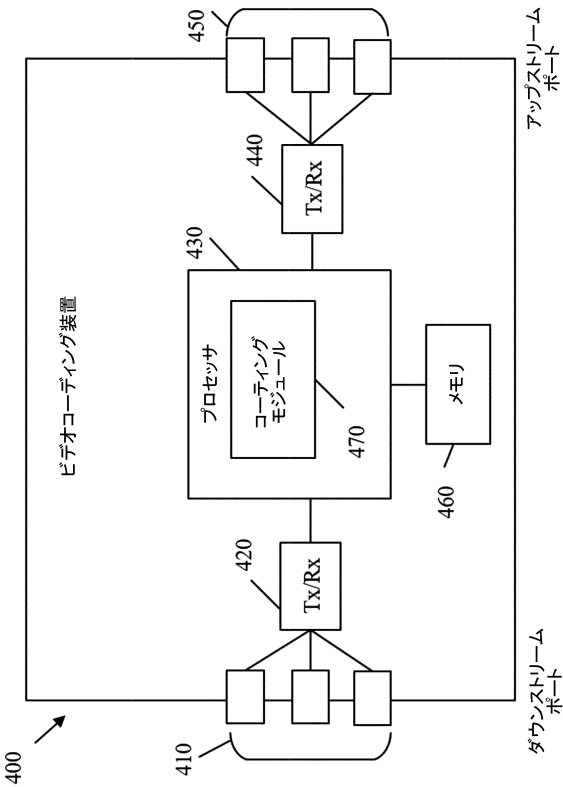


FIG. 4

【図 5】

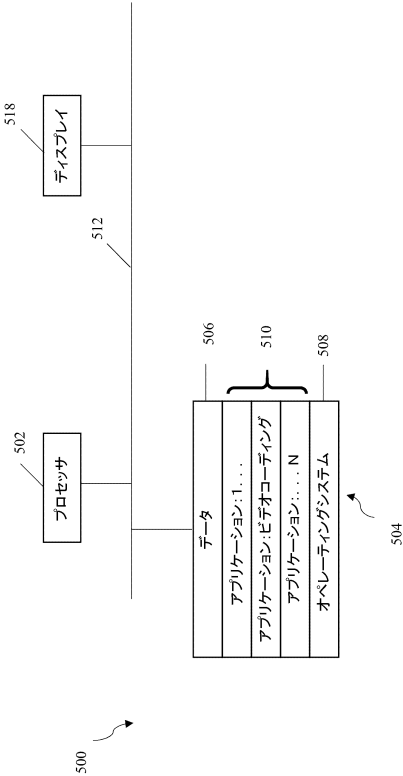


FIG. 5

【図 6】

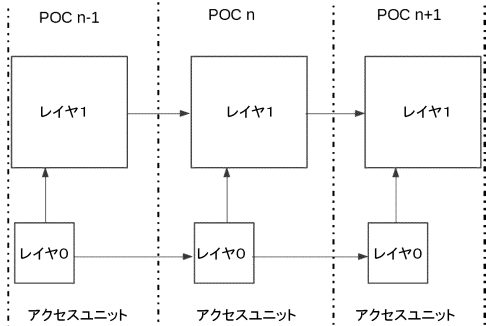


FIG. 6

【図 7】

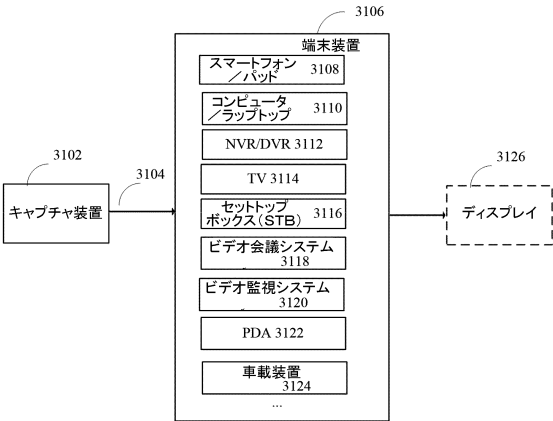


FIG. 7

10

20

30

40

50

【図 8】

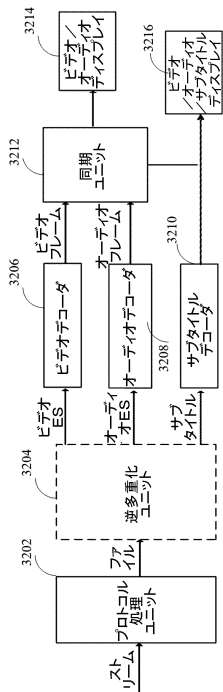


FIG. 8

【図 9】

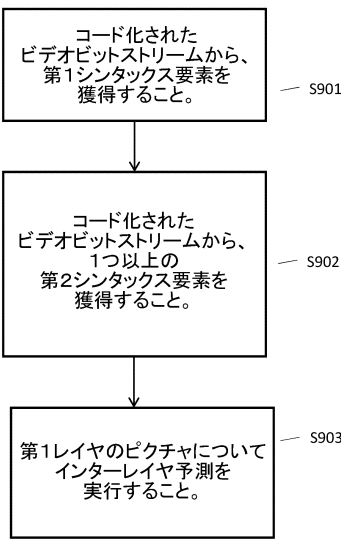


FIG. 9

【図 10】

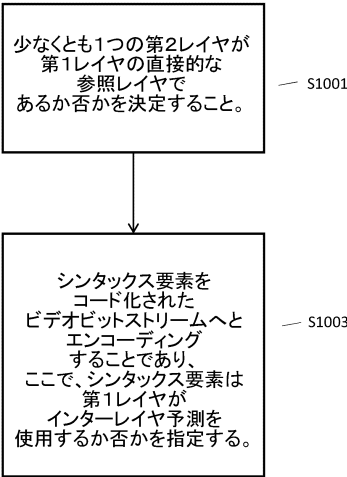


FIG. 10

【図 11】

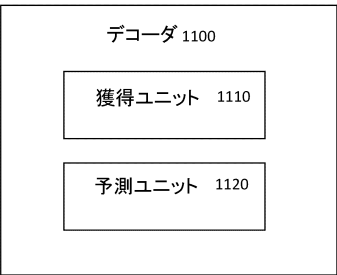


FIG. 11

10

20

30

40

50

【図 12】

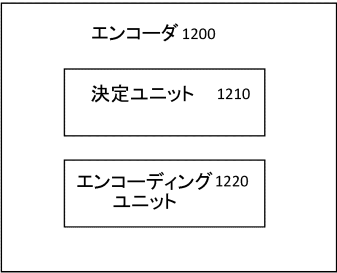


FIG. 12

10

20

30

40

50

## フロントページの続き

- (74)代理人 100070150  
弁理士 伊東 忠彦
- (74)代理人 100135079  
弁理士 宮崎 修
- (72)発明者 マー, シアーン  
中国 5 1 8 1 2 9 グァンドン シェンチェン ロンガン・ディストリクト バンティエン ホアウ  
エイ・アドミニストレーション・ビルディング
- (72)発明者 ヤーン, ハイタオ  
中国 5 1 8 1 2 9 グァンドン シェンチェン ロンガン・ディストリクト バンティエン ホアウ  
エイ・アドミニストレーション・ビルディング
- 審査官 田中 純一
- (56)参考文献 特表 2 0 1 7 - 5 2 2 7 9 2 ( J P , A )  
特表 2 0 2 2 - 5 2 1 5 1 9 ( J P , A )  
米国特許出願公開第 2 0 2 1 / 0 0 9 2 4 4 8 ( U S , A 1 )  
特開 2 0 1 8 - 1 5 2 8 7 4 ( J P , A )  
米国特許出願公開第 2 0 1 5 / 0 0 6 3 4 5 3 ( U S , A 1 )  
米国特許出願公開第 2 0 1 5 / 0 3 5 8 6 4 1 ( U S , A 1 )  
米国特許出願公開第 2 0 1 7 / 0 1 3 4 7 4 2 ( U S , A 1 )  
国際公開第 2 0 1 5 / 0 5 3 2 8 6 ( W O , A 1 )  
米国特許出願公開第 2 0 1 6 / 0 2 4 1 8 3 5 ( U S , A 1 )  
Vadim Seregin, Muhammed Coban, Adarsh K. Ramasubramonian, Marta Karczewicz, AHG8:  
On inter-layer reference for scalability support, JVET-O0243-v1, Joint Video Experts Tea  
m (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenbur  
g, 2019年06月26日, pp.1-5
- (58)調査した分野 (Int.Cl., D B 名)  
H 0 4 N 7 / 1 2  
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8