(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7658982号
(P7658982)

(45)発行日　令和7年4月8日(2025.4.8)

(24)登録日　令和7年3月31日(2025.3.31)

(51)国際特許分類

Ｈ０４Ｎ　19/70　(2014.01)		Ｈ０４Ｎ	19/70
Ｈ０４Ｎ　19/30　(2014.01)		Ｈ０４Ｎ	19/30

ＦＩ

請求項の数　22　（全60頁）

(21)出願番号　特願2022-556234(P2022-556234)
(86)(22)出願日　令和3年3月17日(2021.3.17)
(65)公表番号　特表2023-518453(P2023-518453A)
(43)公表日　令和5年5月1日(2023.5.1)
(86)国際出願番号　PCT/CN2021/081290
(87)国際公開番号　WO2021/185278
(87)国際公開日　令和3年9月23日(2021.9.23)
審査請求日　令和4年11月7日(2022.11.7)
(31)優先権主張番号　PCT/EP2020/057229
(32)優先日　令和2年3月17日(2020.3.17)
(33)優先権主張国・地域又は機関
欧州特許庁(EP)
前置審査

(73)特許権者　504161984
ホアウェイ・テクノロジーズ・カンパニー・リミテッド
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(74)代理人　100110364
弁理士　実広 信哉
(74)代理人　100133569
弁理士　野村 進
(72)発明者　ビャオ・ワン
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディス
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】

デコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをデコードする方法であって、前記ビットストリームは現在のビデオコーディング層を表すデータを含み、前記方法は、

前記ビットストリーム内にビデオパラメータセット(VPS)が存在しない場合、前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定するステップと、

前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定した場合、前記VPS以外で示される前記現在のビデオコーディング層のNALユニットの層識別子nuh_layer_idを、前記現在のビデオコーディング層の識別子vps_layer_idとして使用するステップと

を含む、方法。

【請求項２】

前記方法は、

前記現在のビデオコーディング層の前記識別子を使用して前記現在のビデオコーディング層の現在のスライスを予測するステップをさらに含む、請求項1に記載の方法。

【請求項３】

前記方法は、

前記ビットストリームは複数の層を含むと決定した場合、前記ビットストリームを解析

することによって前記現在のビデオコーディング層の前記識別子を取得するステップをさらに含む、請求項1または2に記載の方法。
【請求項４】
前記現在のビデオコーディング層の前記識別子は、前記現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築するために使用され、
前記現在のビデオコーディング層の前記識別子を使用して前記現在のビデオコーディング層の前記現在のスライスを予測するステップは、前記1つまたは複数の参照ピクチャリストを使用して前記現在のスライスを予測するステップを含む、請求項1から3のいずれか一項に記載の方法。
【請求項５】
前記現在のビデオコーディング層の前記NALユニットは、前記ビットストリーム内に存在する前記現在のビデオコーディング層のVCL NALユニットである、請求項1から4のいずれか一項に記載の方法。
【請求項６】
前記現在のビデオコーディング層の前記VCL NALユニットは、前記ビットストリーム内に存在する前記現在のビデオコーディング層のいずれか1つのVCL NALユニットである、請求項5に記載の方法。
【請求項７】
エンコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをエンコードする方法であって、前記ビットストリームは現在のビデオコーディング層を表すデータを含み、前記方法は、
前記ビットストリーム内にビデオパラメータセット(VPS)が存在しない場合、前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定するステップと、
前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定した場合、前記VPS以外で示される前記現在のビデオコーディング層のNALユニットの層識別子nuh_layer_idを、前記現在のビデオコーディング層の識別子vps_layer_idとして使用するステップと
を含む、方法。
【請求項８】
前記方法は、
前記現在のビデオコーディング層の前記識別子を使用して前記現在のビデオコーディング層の現在のスライスを予測するステップをさらに含む、請求項7に記載の方法。
【請求項９】
前記方法は、
前記ビットストリームは複数の層を含むと決定した場合、前記現在のビデオコーディング層の前記識別子を前記ビットストリームにエンコードするステップをさらに含む、請求項7または8に記載の方法。
【請求項１０】
前記現在のビデオコーディング層の前記識別子として前記現在のビデオコーディング層の前記NALユニットの前記層識別子を使用することは、前記現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築することであり、
前記現在のビデオコーディング層の前記識別子を使用して前記現在のビデオコーディング層の前記現在のスライスを予測するステップは、前記1つまたは複数の参照ピクチャリストを使用して前記現在のスライスを予測するステップを含む、請求項7から9のいずれか一項に記載の方法。
【請求項１１】
前記現在のビデオコーディング層の前記NALユニットは、前記ビットストリーム内に存在する前記現在のビデオコーディング層のVCL NALユニットである、請求項7から10のいずれか一項に記載の方法。

【請求項１２】
前記現在のビデオコーディング層の前記VCL NALユニットは、前記ビットストリーム内に存在する前記現在のビデオコーディング層のいずれか1つのVCL NALユニットである、請求項11に記載の方法。
【請求項１３】
請求項1から6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含む、コーディングデバイス。
【請求項１４】
請求項7から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含む、コーディングデバイス。
【請求項１５】
コンピュータまたはプロセッサ上で実行されると、請求項1から6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。
【請求項１６】
コンピュータまたはプロセッサ上で実行されると、請求項7から12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。
【請求項１７】
コーディングデバイスであって、
1つまたは複数のプロセッサと、
前記1つまたは複数のプロセッサに結合され、前記1つまたは複数のプロセッサによる実行のためのプログラミングを記憶する非一時的コンピュータ可読記憶媒体とを含み、
前記プログラミングは、前記1つまたは複数のプロセッサによって実行されると、請求項1から6のいずれか一項に記載の方法を実行するように前記コーディングデバイスを構成する、コーディングデバイス。
【請求項１８】
コーディングデバイスであって、
1つまたは複数のプロセッサと、
前記1つまたは複数のプロセッサに結合され、前記1つまたは複数のプロセッサによる実行のためのプログラミングを記憶する非一時的コンピュータ可読記憶媒体とを含み、
前記プログラミングは、前記1つまたは複数のプロセッサによって実行されると、請求項7から12のいずれか一項に記載の方法を実行するように前記コーディングデバイスを構成する、コーディングデバイス。
【請求項１９】
コンピュータデバイスによって実行されると、前記コンピュータデバイスに請求項1から6のいずれか一項に記載の方法を実行させるプログラムコードを担持する非一時的コンピュータ可読記憶媒体。
【請求項２０】
コンピュータデバイスによって実行されると、前記コンピュータデバイスに請求項7から12のいずれか一項に記載の方法を実行させるプログラムコードを担持する非一時的コンピュータ可読記憶媒体。
【請求項２１】
ビデオのビットストリームをデコードするための装置であって、
前記ビットストリーム内にビデオパラメータセット(VPS)が存在しない場合、現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定するように構成された決定ユニットと、
前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定した場合、前記VPS以外で示される前記現在のビデオコーディング層のNALユニットの層識別子nuh_layer_idを、前記現在のビデオコーディング層の識別子vps_layer_idとして使用するように構成された予測ユニットと
を含む、装置。

【請求項２２】
ビデオのビットストリームをエンコードするための装置であって、
前記ビットストリーム内にビデオパラメータセット(VPS)が存在しない場合、現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定するように構成された決定ユニットと、
前記現在のビデオコーディング層は前記ビットストリーム内の唯一の層であると決定した場合、前記VPS以外で示される前記現在のビデオコーディング層のNALユニットの層識別子nuh_layer_idを、前記現在のビデオコーディング層の識別子vps_layer_idとして使用するように構成された予測ユニットと
を含む、装置。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
本特許出願は、2020年3月17日に出願された国際特許出願PCT/EP2020/057229に基づく優先権を主張する。前述の特許出願の開示は、参照によりその全体が本明細書に組み込まれる。
【０００２】
本出願の実施形態は、一般にピクチャ処理の分野に関し、より具体的にはビデオコーディングに関する。
【背景技術】
【０００３】
ビデオコーディング(ビデオエンコーディングおよびデコーディング)は、たとえば、デジタルテレビの放送、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伝送、ビデオチャットなどのリアルタイムの会話型アプリケーション、ビデオ会議、DVDおよびBlu-ray(登録商標)ディスク、ビデオコンテンツの取得および編集システム、ならびにセキュリティアプリケーションのカムコーダなどの、幅広い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。
【発明の概要】
【発明が解決しようとする課題】
【０００４】
比較的短いビデオでも描写するために必要なビデオデータの量はかなりの量になる可能性があり、帯域幅容量が限られている通信ネットワークを介してデータをストリーミングする際、または他の方法で通信する場合に、困難をもたらす可能性がある。したがって、ビデオデータは一般に、最新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。メモリリソースが限られている可能性があるため、ビデオを記憶デバイスに記憶する場合、ビデオのサイズも問題になる可能性がある。ビデオ圧縮デバイスは、しばしば、伝送または記憶の前にビデオデータをコーディングするために、ソースにおいてソフトウェアおよび/またはハードウェアを使用し、それによって、デジタルビデオ画像を表現するために必要なデータ量が減少する。次いで、圧縮されたデータは、ビデオデータをデコードするビデオ圧縮解除デバイスによって送信先において受信される。ネットワークリソースが限られており、より高いビデオ品質に対する要求がますます高まっているため、ピクチャの質をほとんどまたはまったく犠牲にすることなく圧縮率を向上させる、改善された圧縮および圧縮解除技法が望まれている。
【課題を解決するための手段】
【０００５】
本出願の実施形態は、独立請求項による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。
【０００６】
上記および他の目的は、独立請求項の主題によって達成される。さらなる実施形態は、

従属請求項、明細書、および図面から明らかである。
【０００７】
第1の態様によれば、本発明は、デコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをデコードする方法に関し、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するステップと、現在のビデオコーディング層はビットストリーム内の唯一の層である(たとえば、sps_video_parameter_set_idが0に等しい)と決定した場合(たとえば、決定された場合のみ決定される)、現在のビデオコーディング層のNALユニットの識別子(たとえば、nuh_layer_id)を、現在のビデオコーディング層の識別子(たとえば、vps_layer_id)として使用するステップとを含む。
【０００８】
VPSは任意のパラメータセットである。たとえば、単一層のビットストリームの場合、ビットストリーム内にVPSがない可能性があり、したがって、VPSに存在する可能性がある現在のビデオコーディング層の識別子はビットストリーム内に存在しない可能性があり、現在のビデオコーディング層のNALユニットの識別子を単一層ビットストリームの現在のビデオコーディング層の識別子として使用すると、ビデオコーディングの実装形態が保証される。
【０００９】
ある実装形態によれば、本方法は、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するステップをさらに含みうる。
【００１０】
ある実装形態によれば、本方法は、ビットストリームは複数の層を含む(たとえば、sps_video_parameter_set_idが0より大きい)と決定した場合、ビットストリームを解析することによって現在のビデオコーディング層の識別子を取得するステップをさらに含みうる。
【００１１】
ビデオビットストリームをデコードする、そのように提供される方法は、現在のビデオコーディング層の識別子の効率的なシグナリングを保証し、ビットレートを節約する。
【００１２】
ある実装形態によれば、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築するために、現在のビデオコーディング層の識別子は使用され得、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測することは、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測することを含みうる。
【００１３】
ある実装形態によれば、現在のビデオコーディング層のNALユニットは、ビットストリーム内に存在する現在のビデオコーディング層のVCL NALユニットであり得る。
【００１４】
単一層ビットストリームがサブビットストリーム抽出から取得される場合、単一層ビットストリーム内の非VCLユニットは異なる層識別子を有する可能性があるため、現在のビデオコーディング層のVCL NALユニットの識別子を単一層ビットストリームの現在のビデオコーディング層の識別子として使用すると、より正確になる。
【００１５】
ある実装形態によれば、現在のビデオコーディング層のVCL NALユニットは、ビットストリーム内に存在する現在のビデオコーディング層の任意の1つのVCL NALユニット(または、すべてのVCL NALユニット)であり得る。
【００１６】
第2の態様によれば、本発明は、エンコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをエンコードする方法に関し、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオコーディング層

はビットストリーム内の唯一の層であるかどうかを決定するステップと、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合、現在のビデオコーディング層のNALユニットの識別子を、現在のビデオコーディング層の識別子として使用するステップとを含む。

【００１７】

ある実装形態によれば、本方法は、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するステップをさらに含みうる。

【００１８】

ある実装形態によれば、本方法は、ビットストリームは複数の層を含む(たとえば、sps_video_parameter_set_idが0より大きい)と決定した場合、現在のビデオコーディング層の識別子をビットストリームにエンコードするステップをさらに含みうる。

【００１９】

ある実装形態によれば、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築するために、現在のビデオコーディング層の識別子は使用され得、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測することは、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測することを含みうる。

【００２０】

ある実装形態によれば、現在のビデオコーディング層のNALユニットは、ビットストリーム内に存在する現在のビデオコーディング層のVCL NALユニットであり得る。

【００２１】

ある実装形態によれば、現在のビデオコーディング層のVCL NALユニットは、ビットストリーム内に存在する現在のビデオコーディング層の任意の1つのVCL NALユニット(または、すべてのVCL NALユニット)であり得る。

【００２２】

第3の態様によれば、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ドするための装置に関し、本装置は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するように構成された決定ユニットと、現在のビデオコーディング層はビットストリーム内の唯一の層である(たとえば、sps_video_parameter_set_idが0に等しい)と決定した場合(たとえば、決定された場合のみ決定される)、現在のビデオコーディング層のNALユニットの識別子(たとえば、nuh_layer_id)を、現在のビデオコーディング層の識別子(たとえば、vps_layer_id)として使用するように構成された予測ユニットとを含む。

【００２３】

ある実装形態によれば、予測ユニットは、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するようにさらに構成され得る。

【００２４】

ある実装形態によれば、ビットストリームは複数の層を含む(たとえば、sps_video_parameter_set_idが0より大きい)と決定した場合、本装置は、ビットストリームを解析することによって現在のビデオコーディング層の識別子を取得するように構成された取得ユニットをさらに含みうる。

【００２５】

ある実装形態によれば、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築するために、現在のビデオコーディング層の識別子は使用され得、予測ユニットは、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測するように構成され、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測するように構成されうる。

【００２６】

ある実装形態によれば、現在のビデオコーディング層のNALユニットは、ビットストリ

ーム内に存在する現在のビデオコーディング層のVCL NALユニットであり得る。
【００２７】
ある実装形態によれば、現在のビデオコーディング層のVCL NALユニットは、ビットストリーム内に存在する現在のビデオコーディング層の任意の1つのVCL NALユニット(または、すべてのVCL NALユニット)であり得る。
【００２８】
本発明の第4の態様によれば、本発明は、ビデオビットストリームをエンコードするための装置に関し、本装置は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するように構成された決定ユニットと、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合、現在のビデオコーディング層のNALユニットの識別子を現在のビデオコーディング層の識別子として使用するように構成された予測ユニットとを含む。
【００２９】
ある実装形態によれば、予測ユニットは、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するようにさらに構成され得る。
【００３０】
ある実装形態によれば、ビットストリームは複数の層を含む(たとえば、sps_video_parameter_set_idが0より大きい)と決定した場合、本装置は、現在のビデオコーディング層の識別子をビットストリームにエンコードするように構成されたエンコーディングユニットをさらに含みうる。
【００３１】
ある実装形態によれば、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築するために、現在のビデオコーディング層の識別子は使用され得、予測ユニットは、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測するように構成され得、また、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測するように構成され得る。
【００３２】
ある実装形態によれば、現在のビデオコーディング層のNALユニットは、ビットストリーム内に存在する現在のビデオコーディング層のVCL NALユニットである。
【００３３】
ある実装形態によれば、現在のビデオコーディング層のVCL NALユニットは、ビットストリーム内に存在する現在のビデオコーディング層の任意の1つのVCL NALユニット(または、すべてのVCL NALユニット)である。
【００３４】
本発明の第1の態様による方法は、本発明の第3の態様による装置によって行うことができる。本発明の第3の態様による装置のさらなる特徴および実装形態は、本発明の第1の態様による方法の特徴および実装形態に対応する。
【００３５】
本発明の第2の態様による方法は、本発明の第4の態様による装置によって行うことができる。本発明の第4の態様による装置のさらなる特徴および実装形態は、本発明の第2の態様による方法の特徴および実装形態に対応する。
【００３６】
上記の方法は、それぞれデコーディングデバイスまたはエンコーディングデバイスにおいて実装することができ、したがって、上記の例のうちのいずれか1つに従ってビデオビットストリームをエンコードする方法を実行するための処理回路を含むエンコーダが提供される。さらに、1つまたは複数のプロセッサと、プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する非一時的コンピュータ可読記憶媒体とを含むエンコーダが提供され、プログラミングは、プロセッサによって実行されると、上記の例のうちのいずれか1つに従ってビデオビットストリームをエンコードする方法を実行する

ようにエンコーダを構成する。同様に、上記の例のうちのいずれか1つに従ってビデオビットストリームをデコードする方法を実行するための処理回路を含むデコーダと、1つまたは複数のプロセッサと、プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する非一時的なコンピュータ可読記憶媒体を含むデコーダとが提供され、プログラミングは、プロセッサによって実行されると、上記の例のうちのいずれかに従ってビデオビットストリームをデコードする方法を実行するようにデコーダを構成する。

【００３７】

さらに、コンピュータまたはプロセッサ上で実行されると、上記の例のうちのいずれか1つによる方法を行う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。同様に、コンピュータデバイスによって実行されると、コンピュータデバイスに上記の例のうちのいずれか1つの方法を行わせるプログラムコードを担持する非一時的コンピュータ可読媒体が提供される。

【００３８】

1つまたは複数の実施形態の詳細は、添付の図面および以下の説明に記載されている。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になるであろう。

【００３９】

以下では、本発明の実施形態が、添付の図および図面を参照してより詳細に説明される。

【図面の簡単な説明】

【００４０】

【図１Ａ】本発明の実施形態を実装するように構成されたビデオコーディングシステムの一例を示すブロック図である。

【図１Ｂ】本発明の実施形態を実装するように構成されたビデオコーディングシステムの他の例を示すブロック図である。

【図２】本発明の実施形態を実装するように構成されたビデオエンコーダの例を示すブロック図である。

【図３】本発明の実施形態を実装するように構成されたビデオデコーダの例示的な構造を示すブロック図である。

【図４】エンコーディング装置またはデコーディング装置の一例を示すブロック図である。

【図５】エンコーディング装置またはデコーディング装置の他の例を示すブロック図である。

【図６】ラスタスキャン順の例を示す図である。

【図７】タイル、スライス、およびサブピクチャの例を示す図である。

【図８】スケーラブルなビデオコーディングの層とサブ層に関する例を示す図である。

【図９】一実施形態による、ビデオビットストリームをデコードする方法を示す図である。

【図１０】一実施形態による、ビデオビットストリームをエンコードする方法を示す図である。

【図１１】一実施形態による、ビデオビットストリームをデコードする方法を示す図である。

【図１２】一実施形態による、ビデオビットストリームをエンコードする方法を示す図である。

【図１３】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の例示的な構造を示すブロック図である。

【図１４】端末デバイスの一例の構造を示すブロック図である。

【発明を実施するための形態】

【００４１】

以下の説明では、本開示の一部を形成し、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例として示す添付の図面を参照する。本発明の実施形態は、他の態様において使用され得、図には示されていない構造的または論理的な変更を含むことが理解される。したがって、以下の詳細な説明は、限定的な意味で解釈され

るべきではなく、本発明の範囲は添付の特許請求の範囲によって定義される。
【００４２】
たとえば、説明した方法に関連する開示はまた、その方法を行うように構成された対応するデバイスまたはシステムにも当てはまり、その逆も同様であることが理解される。たとえば、1つまたは複数の特定の方法ステップが記載されている場合、対応するデバイスは、たとえそのような1つまたは複数のユニットが明示的に説明または図示されていない場合でも、記載された1つまたは複数の方法ステップを行うために1つまたは複数のユニット(たとえば、1つまたは複数のステップを行う1つのユニット、あるいは複数のステップのうちの1つまたは複数をそれぞれが行う複数のユニット)、たとえば機能ユニットを含み得る。一方、たとえば、特定の装置が1つまたは複数のユニット、たとえば機能ユニットに基づいて記述されている場合、対応する方法は、たとえそのような1つまたは複数のステップが明示的に説明または図示されていない場合でも、1つまたは複数のユニットの機能を行うための1つのステップ(たとえば、1つまたは複数のユニットの機能を行う1つのステップ、または複数のユニットのうちの1つまたは複数の機能をそれぞれが行う複数のステップ)を含み得る。さらに、特に断りのない限り、本明細書に記載の様々な例示的な実施形態および/または態様の特徴が互いに組み合わされてよいことを理解されたい。
【００４３】
ビデオコーディングは通常、ビデオまたはビデオシーケンスを形成するピクチャのシーケンスの処理を指す。「ピクチャ」という用語の代わりに、「フレーム」または「画像」という用語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、一般的なコーディング)は、ビデオエンコーディングとビデオデコーディングの2つの部分を含む。ビデオエンコーディングはソース側で行われ、ビデオピクチャを表現するために必要なデータ量を削減するため(より効率的な記憶および/または伝送のため)に、通常は元のビデオピクチャの処理(たとえば、圧縮による)を含む。ビデオデコーディングは宛先側で行われ、通常はビデオピクチャを再構築するエンコーダとは逆の処理を含む。ビデオピクチャ(または、一般的にピクチャ)の「コーディング」に言及する実施形態は、ビデオピクチャ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」または「デコーディング」に関連すると理解されるべきである。エンコーディング部分とデコーディング部分の組合せはまた、CODEC(Coding and Decoding)とも呼ばれる。
【００４４】
無損失ビデオコーディングの場合、元のビデオピクチャを再構築することができ、すなわち、再構築されたビデオピクチャは、元のビデオピクチャと同じ品質である(記憶中または伝送中に伝送損失または他のデータ損失がないことを前提とする)。非可逆ビデオコーディングの場合、ビデオピクチャを表すデータの量を減らすために、たとえば量子化によるさらなる圧縮が行われ、これは、デコーダにおいて完全に再構築することはできず、すなわち、再構築されたビデオピクチャの品質は、元のビデオピクチャの品質と比較して低いか、劣っている。
【００４５】
いくつかのビデオコーディング標準は、「非可逆ハイブリッドビデオコーデック」のグループに属する(すなわち、サンプルドメインにおいて空間予測と時間予測を組み合わせ、変換ドメイン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行う)。ビデオシーケンスの各ピクチャは、通常、重複しないブロックのセットにパーティショニングされ、コーディングは通常、ブロックレベルで実行される。言い換えれば、エンコーダにおいて、ビデオは通常、たとえば、予測ブロックを生成するために空間(イントラピクチャ)予測および/または時間(インターピクチャ)予測を使用することと、残差ブロックを取得するために現在のブロック(現在処理中/処理予定のブロック)から予測ブロックを減算することと、伝送されるデータの量を削減(圧縮)するために残差ブロックを変換して変換ドメインにおいて残差ブロックを量子化することとによって、ブロック(ビデオブロック)レベルで処理、つまりエンコードされるが、一方、デコーダにおいて、表現のために現在のブロックを再構築するために、エンコーダとは逆の処理がエンコードまたは圧縮されたブロックに

適用される。さらに、エンコーダは、デコーダの処理ループを複製して、両方が同一の予測(たとえば、イントラ予測およびインター予測)および/または後続のブロックの処理、つまりコーディングのための再構築を生成するようにする。

【００４６】

ビデオコーディングシステム10の以下の実施形態では、ビデオエンコーダ20およびビデオデコーダ30が、図1から図3に基づいて説明される。

【００４７】

図1Aは、本出願の技法を利用し得る例示的なコーディングシステム10、たとえば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)を示す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様々な例に従って技法を実行するように構成され得るデバイスの例を表す。

【００４８】

図1Aに示されるように、コーディングシステム10は、エンコードされたピクチャデータ21を、たとえばエンコードされたピクチャデータ21をデコードするための宛先デバイス14に提供するように構成されたソースデバイス12を含む。

【００４９】

ソースデバイス12はエンコーダ20を含み、追加的に、すなわち任意で、ピクチャソース16、プリプロセッサ(または、プリプロセッシングユニット)18、たとえば、ピクチャプリプロセッサ18、および通信インターフェースまたは通信ユニット22を含みうる。

【００５０】

ピクチャソース16は、任意の種類のピクチャキャプチャ装置、たとえば実世界のピクチャをキャプチャするためのカメラ、および/または任意の種類のピクチャ生成デバイス、たとえばコンピュータアニメーションピクチャを生成するためのコンピュータグラフィックプロセッサ、あるいは実世界のピクチャ、コンピュータで生成されたピクチャ(たとえば、画面コンテンツ、仮想現実(VR)ピクチャ)および/またはそれらの任意の組合せ(たとえば、拡張現実(AR)ピクチャ)を取得/提供するための任意の種類の他のデバイスを含むか、またはそれらであり得る。ピクチャソースは、前述のピクチャのいずれかを記憶する任意の種類のメモリまたはストレージであり得る。

【００５１】

プリプロセッサ18およびプリプロセッシングユニット18によって行われる処理と区別して、ピクチャまたはピクチャデータ17はまた、生ピクチャまたは生ピクチャデータ17と呼ばれ得る。

【００５２】

プリプロセッサ18は、前処理されたピクチャ19または前処理されたピクチャデータ19を取得するために、(生の)ピクチャデータ17を受信し、ピクチャデータ17に対して前処理を実行するように構成される。プリプロセッサ18によって実行される前処理は、たとえば、トリミング、色形式転換(たとえば、RGBからYCbCrへ)、色補正、またはノイズ除去を含みうる。プリプロセッシングユニット18は、オプションのコンポーネントであってもよいことを理解されたい。

【００５３】

ビデオエンコーダ20は、前処理されたピクチャデータ19を受信し、エンコードされたピクチャデータ21を提供するように構成される(さらなる詳細は、たとえば図2に基づいて以下で説明される)。

【００５４】

ソースデバイス12の通信インターフェース22は、エンコードされたピクチャデータ21を受信し、エンコードされたピクチャデータ21(または、そのさらに処理されたバージョン)を、通信チャネル13を介して他のデバイス、たとえば、記憶または直接再構築のために宛先デバイス14または任意の他のデバイスに伝送するように構成され得る。

【００５５】

宛先デバイス14は、デコーダ30(たとえば、ビデオデコーダ30)を含み、さらに、すなわち任意で、通信インターフェースまたは通信ユニット28、ポストプロセッサ32(または、ポストプロセッシングユニット32)、およびディスプレイデバイス34を含みうる。

【００５６】

宛先デバイス14の通信インターフェース28は、エンコードされたピクチャデータ21(または、そのさらに処理されたバージョン)を、たとえば、ソースデバイス12から直接、または任意の他のソース、たとえば記憶デバイス、たとえばエンコードされたピクチャデータ記憶デバイスから受信し、エンコードされたピクチャデータ21をデコーダ30に提供するように構成される。

【００５７】

通信インターフェース22および通信インターフェース28は、ソースデバイス12と宛先デバイス14との間の直接通信リンク、たとえば、直接のワイヤードまたはワイヤレス接続を介して、あるいは任意の種類のネットワーク、たとえば、ワイヤードまたはワイヤレスネットワークあるいはそれらの任意の組合せ、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合せを介して、エンコードされたピクチャデータ21またはエンコードされたデータを伝送または受信するように構成され得る。

【００５８】

通信インターフェース22は、たとえば、エンコードされたピクチャデータ21を適切な形式、たとえばパケットにパッケージ化するように、および/あるいは通信リンクまたは通信ネットワークを介した伝送のための任意の種類の伝送エンコーディングまたは処理を使用して、エンコードされたピクチャデータを処理するように構成され得る。

【００５９】

通信インターフェース22の対応物を形成する通信インターフェース28は、たとえば、伝送されたデータを受信し、エンコードされたピクチャデータ21を取得するために、任意の種類の対応する伝送デコーディングまたは処理および/あるいはデパッケージングを使用して伝送データを処理するように構成され得る。

【００６０】

通信インターフェース22と通信インターフェース28の両方は、ソースデバイス12から宛先デバイス14を指している図1Aの通信チャネル13の矢印によって示されるように単方向通信インターフェース、または双方向通信インターフェースとして構成されえ、通信リンクおよび/またはデータ伝送、たとえばエンコードされたピクチャデータ伝送に関連する任意の他の情報を確認および交換するために、たとえばメッセージを送受信するように、たとえば接続をセットアップするように構成されてもよい。

【００６１】

デコーダ30は、エンコードされたピクチャデータ21を受信し、デコードされたピクチャデータ31またはデコードされたピクチャ31を提供するように構成される(さらな詳細は、たとえば、図3または図5に基づいて以下で説明される)。

【００６２】

宛先デバイス14のポストプロセッサ32は、デコードされたピクチャデータ31(再構築されたピクチャデータとも呼ばれる)、たとえばデコードされたピクチャ31を、後処理されたピクチャデータ33、たとえば後処理されたピクチャ33を取得するために、後処理するように構成される。ポストプロセッシングユニット32によって実行される後処理は、デコードされたピクチャデータ31をたとえばディスプレイデバイス34による表示用に準備するための、たとえば、色形式転換(たとえば、YCbCrからRGBへ)、色補正、トリミング、または再サンプリング、あるいは他の任意の処理を含みうる。

【００６３】

宛先デバイス14のディスプレイデバイス34は、たとえばユーザまたは閲覧者にピクチャを表示するための後処理されたピクチャデータ33を受信するように構成される。ディスプレイデバイス34は、再構築されたピクチャを表すための任意の種類のディスプレイ、た

とえば、内蔵または外付けのディスプレイまたはモニタであってもよく、それを含んでもよい。ディスプレイは、たとえば、液晶ディスプレイ(LCD)、有機発光ダイオード(OLED)ディスプレイ、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)、デジタルライトプロセッサ(DLP)、または任意の種類の他のディスプレイを含みうる。

【００６４】

図1Aは、ソースデバイス12および宛先デバイス14を別個のデバイスとして示しているが、デバイスの実施形態はまた、ソースデバイス12または対応する機能、および宛先デバイス14または対応する機能の両方または両方の機能を含んでもよい。そのような実施形態では、ソースデバイス12または対応する機能、および宛先デバイス14または対応する機能は、同じハードウェアおよび/またはソフトウェアを使用して、あるいは別個のハードウェアおよび/またはソフトウェア、あるいはそれらの任意の組合せによって実装され得る。

【００６５】

本説明に基づいて当業者には明らかであるように、図1Aに示されるように、ソースデバイス12および/または宛先デバイス14内の異なるユニットまたは機能の存在および(正確な)スプリットは、実際のデバイスおよびアプリケーションによって異なる場合がある。

【００６６】

エンコーダ20(たとえば、ビデオエンコーダ20)またはデコーダ30(たとえば、ビデオデコーダ30)、あるいはエンコーダ20とデコーダ30の両方は、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ(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)、ディスクリートロジック、ハードウェア、ビデオコーディング専用、またはそれらの任意の組合せなどの、図1Bに示される処理回路を介して実装され得る。エンコーダ20は、図2のエンコーダ20および/または本明細書に記載の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した様々なモジュールを具現化するために、処理回路46を介して実装され得る。デコーダ30は、図3のデコーダ30および/または本明細書に記載の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した様々なモジュールを具現化するために、処理回路46を介して実装され得る。処理回路は、後述するように、様々な演算を実行するように構成され得る。図5に示されるように、技法がソフトウェアにおいて部分的に実装される場合、デバイスは、ソフトウェアの命令を適切な非一時的コンピュータ可読記憶媒体に記憶し得、本開示の技法を実行するために1つまたは複数のプロセッサを使用してハードウェアにおいて命令を実行し得る。ビデオエンコーダ20およびビデオデコーダ30のいずれかは、たとえば、図1Bに示されるように、単一のデバイス内の複合エンコーダ/デコーダ(CODEC)の一部として統合され得る。

【００６７】

ソースデバイス12および宛先デバイス14は、任意の種類のハンドヘルドデバイスまたは固定デバイス、たとえばノートブックまたはラップトップコンピュータ、モバイル電話、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ー、セットトップボックス、テレビ、ディスプレイデバイス、デジタルメディアプレイヤ、ビデオゲームコンソール、ビデオストリーミングデバイス(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など)、ブロードキャスト受信デバイス、ブロードキャストトランスミッタデバイスなどを含む、広範囲のデバイスのいずれかを含み得、オペレーティングシステムをまったく使用しなくてもよく、任意の種類のオペレーティングシステムを使用してもよい。場合によっては、ソースデバイス12および宛先デバイス14は、ワイヤレス通信のために装備され得る。したがって、ソースデバイス12および宛先デバイス14は、ワイヤレス通信デバイスであり得る。

【００６８】

場合によっては、図1Aに示されるビデオコーディングシステム10は単なる例であり、本出願の技法は、エンコーディングデバイスとデコーディングデバイスとの間のデータ通信を必ずしも含まないビデオコーディング設定(たとえば、ビデオエンコーディングまたはビデオデコーディング)に適用し得る。他の例では、データは、ローカルメモリから検索さ

れるか、ネットワークを介してストリーミングされる、などである。ビデオエンコーディングデバイスは、データをエンコードしてメモリに記憶し得、および/またはビデオデコーディングデバイスは、メモリからデータを検索してデコードし得る。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、互いに通信しないが、単にデータをメモリにエンコードする、および/またはメモリからデータを検索してデコードするデバイスによって実行される。

【００６９】

説明の都合上、本発明の実施形態は、たとえば、High-Efficiency Video Coding (HEVC)、またはITU-T Video Coding Experts Group (VCEG)のVideo Coding (JCT-VC)およびISO/IEC Motion Picture Experts Group (MPEG)に関するジョイントコラボレーションチームによって開発された次世代ビデオコーディング規格であるVersatile Video coding (VVC)の参照ソフトウェアを参照することによって、本明細書で説明される。当業者は、本発明の実施形態がHEVCまたはVVCに限定されないことを理解するであろう。

【００７０】

エンコーダおよびエンコード方法

図2は、本出願の技法を実装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の例では、ビデオエンコーダ20は、入力201(または、入力インターフェース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、再構築ユニット214、ループフィルタユニット220、デコードされたピクチャバッファ(DPB)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272(または、出力インターフェース272)を含む。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ニングユニット262を含み得る。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(図示せず)を含み得る。図2に示されるビデオエンコーダ20はまた、ハイブリッドビデオエンコーダまたはハイブリッドビデオコーデックによるビデオエンコーダと呼ばれ得る。

【００７１】

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の順方向信号経路を形成するといわれることもあり、一方、逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、バッファ216、ループフィルタ220、デコードされたピクチャバッファ(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ユニット254は、ビデオエンコーダ20の逆方向信号経路を形成するといわれることもあり、ビデオエンコーダ20の逆方向信号経路はデコーダの信号経路に対応する(図3のビデオデコーダ30を参照)。逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、ループフィルタ220、デコードされたピクチャバッファ(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」を形成するといわれることもある。

【００７２】

ピクチャおよびピクチャのパーティショニング(ピクチャおよびブロック)

エンコーダ20は、入力201を介して、ピクチャ17(または、ピクチャデータ17)、たとえば、ビデオまたはビデオシーケンスを形成するピクチャのシーケンスのピクチャ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ピクチャデータはまた、前処理されたピクチャ19(または前処理されたピクチャデータ19)であってもよい。簡単にするために、以下の説明ではピクチャ17を参照する。ピクチャ17はまた、現在のピクチャまたはコーディングされるピクチャと呼ばれることもある(特に、ビデオコーディングにおいて、現在のピクチャを他のピクチャ、たとえば、同じビデオシーケンスの以前にエンコーディングおよび/またはデコーディングされたピクチャ、すなわち、現在のピクチャも含むビデオシーケンスと区別するため)。

【００７３】

(デジタル)ピクチャは、強度値を持つサンプルの2次元配列またはマトリックスであるか

、またはそのように見なすことができる。配列内のサンプルはまた、ピクセル(ピクチャ要素の省略形)またはペル(pel)と呼ばれ得る。配列またはピクチャの水平および垂直方向(または、軸)のサンプル数は、ピクチャのサイズおよび/または解像度を定義する。色の表現には、通常、3つの色成分が使用され、すなわち、ピクチャは3つのサンプル配列で表現されてもよく、3つのサンプル配列を含んでもよい。RBG形式または色空間において、ピクチャは、対応する赤、緑、青のサンプル配列を含む。しかしながら、ビデオコーディングにおいては、各ピクセルは通常、輝度およびクロミナンス形式または色空間、たとえば、YCbCrで表され、これはYによって示される輝度成分(代わりにLが使用される場合もある)と、CbおよびCrによって示される2つのクロミナンス成分とを含む。輝度(または短い輝度)コンポーネントYは、明るさまたはグレーレベルの強度を表し(たとえば、グレースケールピクチャのように)、2つのクロミナンス(または短い彩度)コンポーネントCbおよびCrは、色度または色情報コンポーネントを表す。したがって、YCbCr形式のピクチャは、輝度サンプル値(Y)の輝度サンプル配列と、クロミナンス値(CbおよびCr)の2つのクロミナンスサンプル配列を含む。RGB形式のピクチャは、YCbCr形式に転換または変換される場合があり、その逆も同様であり、このプロセスは、色変換または転換としても知られる。ピクチャがモノクロの場合、ピクチャは輝度サンプル配列のみを含む場合がある。したがって、ピクチャは、たとえば、モノクロ形式の輝度サンプルの配列、または輝度サンプルの配列と、4:2:0、4:2:2、および4:4:4色形式の彩度サンプルの2つの対応する配列である可能性がある。

【００７４】

ビデオエンコーダ20の実施形態は、ピクチャ17を複数の(通常は重複しない)ピクチャブロック203にパーティショニングするように構成されたピクチャパーティショニングユニット(図2には図示されず)を含みうる。これらのブロックは、ルートブロック、マクロブロック(H.264/AVC)、あるいはコーディングツリーブロック(CTB)、またはコーディングツリーユニット(CTU)(H.265/HEVCおよびVVC)とも呼ばれ得る。ピクチャパーティショニングユニットは、ビデオシーケンスのすべてのピクチャと、ブロックサイズを定義する対応するグリッドに同じブロックサイズを使用するように、または、ピクチャあるいはピクチャのサブセットまたはピクチャのグループ間でブロックサイズを変更するように、および各ピクチャを対応するブロックにパーティショニングするように構成され得る。

【００７５】

さらなる実施形態では、ビデオエンコーダは、ピクチャ17のブロック203、たとえば、ピクチャ17を形成する1つ、複数、またはすべてのブロックを直接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、現在の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呼ばれ得る。

【００７６】

ピクチャ17と同様に、ピクチャブロック203も、ピクチャ17よりも小さい次元ではあるが、強度値(サンプル値)を有するサンプルの2次元配列またはマトリックスである、またはそのように見なすことができる。言い換えれば、ブロック203は、たとえば、1つのサンプル配列(たとえば、モノクロピクチャ17の場合の輝度配列、またはカラーピクチャの場合の輝度または彩度配列)または3つのサンプル配列(たとえば、カラーピクチャ17の場合は1つの輝度配列と2つの彩度配列)、または適用される色形式に応じて任意の他の数および/または種類の配列を含みうる。ブロック203の水平および垂直方向(または、軸)のサンプルの数は、ブロック203のサイズを定義する。したがって、ブロックは、たとえば、サンプルのＭ×Ｎ(Ｍ列×Ｎ行)配列、または変換係数のＭ×Ｎ配列であり得る。

【００７７】

図2に示されるビデオエンコーダ20の実施形態は、ピクチャ17をブロックごとにエンコードするように構成され得、たとえば、エンコーディングおよび予測はブロック203ごとに実行される。

【００７８】

図2に示されるビデオエンコーダ20の実施形態は、スライス(ビデオスライスとも呼ばれ

る)を使用することによってピクチャをパーティショニングおよび/またはエンコードするようにさらに構成され得、ピクチャは、1つまたは複数のスライス(通常は重複しない)を使用してパーティショニングまたはエンコードされ得、各スライスは1つまたは複数のブロック(たとえば、CTU)を含みうる。

【００７９】

図2に示されるビデオエンコーダ20の実施形態は、タイルグループ(ビデオタイルグループとも呼ばれる)および/またはタイル(ビデオタイルとも呼ばれる)を使用することによって、ピクチャをパーティショニングおよび/またはエンコードするようにさらに構成され得、ピクチャは、1つまたは複数のタイルグループ(通常は重複しない)を使用してパーティショニングまたはエンコードされ得、各タイルグループは、たとえば、1つまたは複数のブロック(たとえば、CTU)あるいは1つまたは複数のタイルを含み得、各タイルは、たとえば長方形であってもよく、1つまたは複数のブロック(たとえば、CTU)を含んでもよく、たとえば、完全ブロックまたは部分ブロックでもよい。

【００８０】

残差計算

残差計算ユニット204は、サンプル領域における残差ブロック205を取得するために、たとえば、ピクチャブロック203のサンプル値から予測ブロック265のサンプル値をサンプルごとに(ピクセルごとに)減算することによって、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205とも呼ばれる)を計算するように構成され得る(予測ブロック265についてのさらなる詳細は、後で提供される)。

【００８１】

変換

変換処理ユニット206は、変換領域における変換係数207を取得するために、変換、たとえば、残差ブロック205のサンプル値に対する離散コサイン変換(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、変換残差係数とも呼ばれ、変換領域における残差ブロック205を表す。

【００８２】

変換処理ユニット206は、H.265/HEVCに対して指定された変換など、DCT/DSTの整数近似を適用するように構成され得る。直交DCT変換と比較すると、そのような整数近似は通常、特定の係数によってスケーリングされる。順変換および逆変換によって処理される残差ブロックのノルムを保存するために、変換プロセスの一部として追加のスケーリング係数が適用される。スケーリング係数は通常、シフト演算の2の累乗であるスケーリング係数、変換係数のビット深度、精度と実装コストとの間のトレードオフなどの、特定の制約に基づいて選択される。特定のスケーリング係数は、たとえば、逆変換のために、たと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、たとえば、ビデオデコーダ30における逆変換処理ユニット312によって)指定され、エンコーダ20における、たとえば、変換処理ユニット206による順変換の対応するスケーリング係数は、それに従って特定され得る。

【００８３】

ビデオエンコーダ20(それぞれ変換処理ユニット206)の実施形態は、たとえば、ビデオデコーダ30がデコーディングのために変換パラメータを受信および使用し得るように、変換パラメータ、たとえば1つまたは複数の変換のタイプを、たとえば直接またはエントロピーエンコーディングユニット270を介してエンコーディングまたは圧縮されて出力するように構成され得る。

【００８４】

量子化

量子化ユニット208は、たとえばスカラ量子化またはベクトル量子化を適用することによって、量子化された係数209を取得するために、変換係数207を量子化するように構成され得る。量子化された係数209はまた、量子化された変換係数209または量子化された残差係数209と呼ばれ得る。

【００８５】
量子化プロセスは、変換係数207の一部または全部に関連付けられるビット深度を減少させることができる。たとえば、nビットの変換係数は、量子化中にmビットの変換係数に丸められ得、nはmよりも大きい。量子化パラメータ(QP)を調整することによって、量子化の程度が修正され得る。たとえば、スカラ量子化の場合、より細かい、またはより粗い量子化を実現するために、異なるスケーリングが適用され得る。より小さい量子化ステップサイズはより細かい量子化に対応し、より大きい量子化ステップサイズはより粗い量子化に対応する。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、たとえば、適用可能な量子化ステップサイズのあらかじめ定められたセットへのインデックスであり得る。たとえば、小さい量子化パラメータは細かい量子化(小さい量子化ステップサイズ)に対応し得、大きい量子化パラメータは粗い量子化(大きい量子化ステップサイズ)に対応し得、またはその逆も同様である。量子化は、量子化ステップサイズによる除算を含み得、また、たとえば逆量子化ユニット210による対応する非量子化および/または逆非量子化は、量子化ステップサイズによる乗算を含み得る。いくつかの標準による実施形態、たとえばHEVC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。一般に、量子化ステップサイズは、除算を含む式の固定小数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復元するために、量子化および非量子化に追加のスケーリング係数が導入され得、これは、量子化ステップサイズと量子化パラメータの式の固定小数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。1つの例示的な実装形態では、逆変換のスケーリングと非量子化を組み合わせることができる。あるいは、カスタマイズされた量子化テーブルが使用され、たとえばビットストリームにおいてエンコーダからデコーダにシグナリングされ得る。量子化は非可逆演算であり、損失は量子化ステップサイズの増加に伴って増加する。
【００８６】
ビデオエンコーダ20(それぞれ、量子化ユニット208)の実施形態は、たとえば、直接またはエントロ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、量子化パラメータ(QP)を出力するように構成され得、その結果、たと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および適用し得る。
【００８７】
逆量子化
逆量子化ユニット210は、たとえば、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、またはそれを使用して、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を適用することによって、非量子化された係数211を取得するために、量子化された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る。非量子化された係数211はまた、非量子化された残差係数211とも呼ばれ、量子化による損失のために変換係数とは典型的には同一ではないが、変換係数207に対応し得る。
【００８８】
逆変換
逆変換処理ユニット212は、サンプル領域において再構築された残差ブロック213(または、対応する非量子化された係数213)を取得するために、変換処理ユニット206によって適用される変換の逆変換、たとえば、逆離散コサイン変換(DCT)または逆離散サイン変換(DST)または他の逆変換を適用するように構成される。再構築された残差ブロック213は、変換ブロック213と呼ばれることもある。
【００８９】
再構築
再構築ユニット214(たとえば、加算器(adder)またはアナログ加算器(summer)214)は、サンプルドメインにおいて再構築されたブロック215を取得するために、サンプルごとに、たとえば、再構築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、変換ブロック213(すなわち、再構築された残差ブロック2

13)を予測ブロック265に加算するように構成される。
【００９０】
フィルタリング
ループフィルタユニット220(または、略して「ループフィルタ」220)は、フィルタリングされたブロック221を取得するために、再構築されたブロック215をフィルタリングするように、または一般に、フィルタリングされたサンプルを取得するために、再構築されたサンプルをフィルタリングするように構成される。ループフィルタユニットは、たとえば、ピクセル遷移を平滑化するか、そうでなければビデオ品質を改善するように構成される。ループフィルタユニット220は、デブロッキングフィルタ、サンプル適応オフセット(SAO)フィルタ、あるいは1つまたは複数の他のフィルタ、たとえば、バイラテラルフィルタ、適応ループフィルタ(ALF)、鮮鋭化、平滑化フィルタ、または協調フィルタなどの1つまたは複数のループフィルタ、あるいはそれらの任意の組合せを含みうる。図2において、ループフィルタユニット220がインループフィルタとして示されているが、他の構成では、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もよい。フィルタリングされたブロック221は、フィルタリングされた再構築ブロック221と呼ばれることもある。
【００９１】
ビデオエンコーダ20(それぞれ、ループフィルタユニット220)の実施形態は、たとえば、直接またはエントロ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(サンプル適応オフセット情報など)を出力するように構成され得、その結果、たとえばデコーダ30は、デコーディングのために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および適用し得る。
【００９２】
デコードされたピクチャバッファ
デコード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ドするために、参照ピクチャ、または一般的に参照ピクチャデータを記憶するメモリであり得る。DPB230は、シンクロナスDRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、または他のタイプのメモリデバイスなどの様々なメモリデバイス、のいずれかによって形成され得る。デコードされたピクチャバッファ(DPB)230は、1つまたは複数のフィルタリングされたブロック221を記憶するように構成され得る。デコードされたピクチャバッファ230は、他の以前にフィルタリングされたブロック、たとえば、同じ現在のピクチャまたは異なるピクチャ、たとえば以前に再構築されたピクチャの、以前に再構築およびフィルタリングされたブロック221を記憶するようにさらに構成され得、たとえばインター予測用に、たとえば、完全な以前に再構築された、すなわちデコードされたピクチャ(および、対応する参照ブロックとサンプル)および/または部分的に再構築された現在のピクチャ(および、対応する参照ブロックとサンプル)を提供し得る。デコードされたピクチャバッファ(DPB)230はまた、たとえば、再構築されたブロック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、1つまたは複数のフィルタリングされていない再構築されたブロック215、または一般にフィルタリングされていない再構築されたサンプルを、または再構築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記憶するように構成され得る。
【００９３】
モード選択(パーティショニングおよび予測)
モード選択ユニット260は、パーティショニングユニット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、元のピクチャデータ、たとえば、元のブロック203(現在のピクチャ17の現在のブロック203)、ならびに再構築されたピクチャデータ、たとえば、同じ(現在の)ピクチャのフィルタリングされた、および/またはフィルタリングされていない再構築されたサンプルまたはブロック、ならびに/あるいは、1つまたは複数の以前にデコードされたピクチャからの、たとえば、デコードされたピクチャバッ

ファ230または他のバッファ(たとえば、ラインバッファ、図示せず)からのフィルタリングされた、および/またはフィルタリングされていない再構築されたサンプルまたはブロックを受信または取得するように構成される。再構築されたピクチャデータは、予測ブロック265または予測子265を取得するために、予測用、たとえば、インター予測またはイントラ予測用の参照ピクチャデータとして使用される。

【００９４】

モード選択ユニット260は、現在のブロック予測モード(パーティショニングなしを含む)および予測モード(たとえば、イントラまたはインター予測モード)のパーティショニングを決定または選択し、対応する予測ブロック265を生成するように構成され得、これは残差ブロック205の計算、および再構築されたブロック215の再構築のために使用される。

【００９５】

モード選択ユニット260の実施形態は、パーティショニングおよび予測モードを(たとえば、モード選択ユニット260によってサポートされるか、またはモード選択ユニット260に利用可能なものから)選択するように構成され得、これは、最適な一致、または言い換えれば最小の残差(最小の残差は伝送または記憶のより良好な圧縮を意味する)、または最小のシグナリングオーバーヘッド(最小のシグナリングオーバーヘッドは伝送または記憶のより良好な圧縮を意味する)を提供するか、または両方を考慮するか、バランスを取る。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ニングおよび予測モードを決定する、すなわち、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。この文脈における「最良」、「最小」、「最適」などの用語は、必ずしも全体的な「最良」、「最小」、「最適」などを指すわけではなく、値がしきい値を超えたり下回ったりするなどの終了基準または選択基準を満たすこと、または「準最適な選択」につながる可能性があるが複雑さと処理時間を削減する他の制約を指すこともある。

【００９６】

言い換えれば、パーティショニングユニット262は、たとえば、四分木パーティショニング(QT)、二分木パーティショニング(BT)または三分木パーティショニング(TT)あるいはそれらの任意の組合せを繰り返し使用して、および、たとえば、ブロックパーティションまたはサブブロックの各々について予測を行うために、ブロック203をより小さい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)にパーティショニングするように構成され得、モード選択は、パーティショニングブロック203のツリー構造の選択を含み、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各々に適用される。

【００９７】

以下では、例示的なビデオエンコーダ20によって行われるパーティショニング(たとえば、パーティショニングユニット262による)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)をより詳細に説明する。

【００９８】

パーティショニング

パーティショニングユニット262は、現在のブロック203をより小さいパーティション、たとえば、正方形または長方形サイズの小さいブロックにパーティショニング(または、スプリット)し得る。これらのより小さいブロック(サブブロックとも呼ばれ得る)は、さらにより小さいパーティションにさらにパーティショニングされ得る。これはツリーパーティショニングまたは階層ツリーパーティショニングとも呼ばれ、たとえば、ルートツリーレベル0(階層レベル0、深さ0)におけるルートブロックは再帰的にパーティショニングされ得、たとえば、次の下位のツリーレベル、たとえばツリーレベル1におけるノード(階層レベル1、深さ1)の2つ以上のブロックにパーティショニングされ、これらのブロックは、たとえば、終了基準が満たされているため、たとえば、ツリーの深さの最大値またはブロックサイズの最小値に達したため、パーティショニングが終了するまで、次の下位レベル、たとえば、ツリーレベル2(階層レベル2、深さ2)の2つ以上のブロックなどに再度パーティショニングされ得る。さらにパーティショニングされていないブロックは、ツリーのリーフブロックまたはリーフノードとも呼ばれる。2つのパーティションへのパーティショ

ニングを使用するツリーは二分木(BT)と呼ばれ、3つのパーティションへのパーティショニングを使用するツリーは三分木(TT)と呼ばれ、4つのパーティションへのパーティショニングを使用するツリーは四分木(QT)と呼ばれる。

【００９９】

前述のように、本明細書で使用される「ブロック」という用語は、ピクチャの一部、特に正方形または長方形の部分であり得る。たとえば、HEVCおよびVVCを参照すると、ブロックは、コーディングツリーユニット(CTU)、コーディングユニット(CU)、予測ユニット(PU)、および変換ユニット(TU)、ならびに/あるいは対応するブロック、たとえば、コーディングツリーブロック(CTB)、コーディングブロック(CB)、変換ブロック(TB)、または予測ブロック(PB)であってもよく、それに対応してもよい。

【０１００】

たとえば、コーディングツリーユニット(CTU)は、輝度サンプルのCTB、3つのサンプル配列を有するピクチャの彩度サンプルの2つの対応するCTB、あるいはモノクロピクチャ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と構文構造を使用してコード化されたピクチャのサンプルのCTBであってもよく、またはそれらを含んでもよい。これに対応して、コンポーネントのCTBへの分割がパーティショニングであるように、コーディングツリーブロック(CTB)は、ある値のNに対するサンプルのN×Nブロックであり得る。コーディングユニット(CT)は、輝度サンプルのコーディングブロック、3つのサンプル配列を有するピクチャの彩度サンプルの2つの対応するコーディングブロック、あるいはモノクロピクチャ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と構文構造を使用してコード化されたピクチャのサンプルのコーディングブロックであってもよく、またはそれらを含んでもよい。これに対応して、CTBのコーディングブロックへの分割がパーティショニングであるように、コーディングブロック(CB)は、ある値のMおよびNに対するサンプルのM×Nブロックであり得る。

【０１０１】

実施形態では、たとえば、HEVCによれば、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四分木構造を使用することによってCUにスプリットされ得る。ピクチャ領域をインターピクチャ(時間)またはイントラピクチャ(空間)予測を使用してコーディングするかどうかの決定は、CUレベルにおいて行われる。各CUは、PUスプリットタイプに従って、さらに1つ、2つ、または4つのPUに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、関連情報がPUベースでデコーダに伝送される。PUスプリットタイプに基づく予測プロセスを適用することによって残差ブロックを取得した後、CUのコーディングツリーと同様の他の四分木構造に従って、CUを変換ユニット(TU)にパーティショニングすることができる。

【０１０２】

実施形態では、たとえば、Versatile Video Coding (VVC)と呼ばれる現在開発中の最新のビデオコーディング標準によれば、結合された四分木および二分木(QTBT)パーティショニングが、たとえばコーディングブロックをパーティショニングするために使用される。QTBTブロック構造では、CUは正方形または長方形のいずれかの形状を有することができる。たとえば、コーディングツリーユニット(CTU)は、最初に四分木構造によってパーティショニングされる。四分木リーフノードは、二分木ツリーまたは三分木(または、トリプル)ツリー構造によってさらにパーティショニングされる。パーティショニング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、そのセグメンテーションは、それ以上のパーティショニングなしで予測および変換処理に使用される。これは、CU、PU、およびTUがQTBTコーディングブロック構造において同じブロックサイズを有することを意味する。並行して、複数のパーティション、たとえば三分木パーティションがQTBTブロック構造とともに使用され得る。

【０１０３】

一例では、ビデオエンコーダ20のモード選択ユニット260は、本明細書で説明されるパーティショニング技法の任意の組合せを行うように構成され得る。

【０１０４】
上述のように、ビデオエンコーダ20は、(たとえば、あらかじめ定められた)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る。予測モードのセットは、たとえば、イントラ予測モードおよび/またはインター予測モードを含みうる。
【０１０５】
イントラ予測
イントラ予測モードのセットは、35の異なるイントラ予測モード、たとえば、DC(または、平均)モードおよび平面モードなどの無指向性モード、または、たとえばHEVCにおいて定義されている指向性モードを含んでもよく、67の異なるイントラ予測モード、たとえば、DC(または、平均)モードおよび平面モードなどの無指向性モード、または、たとえばVVCに定義されている指向性モードを含んでもよい。
【０１０６】
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、同じ現在のピクチャの隣接ブロックの再構築されたサンプルを使用するように構成される。
【０１０７】
イントラ予測ユニット254(または、一般にモード選択ユニット260)は、イントラ予測パラメータ(または、一般にブロックの選択されたイントラ予測モードを示す情報)を、ビデオデコーダ30がデコーディングのために予測パラメータを受信して使用することができるように、エンコードされたピクチャデータ21に含めるための構文要素266の形式でエントロピーエンコーディングユニット270に出力するようにさらに構成される。
【０１０８】
インター予測
インター予測モードのセット(または、可能なインター予測モード)は、利用可能な参照ピクチャ(すなわち、以前の少なくとも部分的にデコードされたピクチャ、たとえば、DBP230に記憶されたもの)および他のインター予測パラメータ、たとえば、参照ピクチャ全体か、一部のみ、たとえば、参照ピクチャの現在のブロックの領域の周りの検索ウィンドウ領域は、最も一致する参照ブロックを検索するために使用されるか、ならびに/あるいはピクセル補間が適用されるかどうか、たとえば、ハーフ/セミペル、および/またはクォーターペル補間か否かに依存する。
【０１０９】
上記の予測モードに加えて、スキップモードおよび/または直接モードが適用され得る。
【０１１０】
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には示されていない)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(現在のピクチャ17の現在のピクチャブロック203)およびデコードされたピクチャ231、または少なくとも1つまたは複数の以前に再構築されたブロック、たとえば、1つまたは複数の他の/異なる以前にデコードされたピクチャ231の再構築されたブロックを受信または取得するように構成され得る。たとえば、ビデオシーケンスは、現在のピクチャおよび以前にデコードされたピクチャ231を含み得、または、言い換えれば、現在のピクチャおよび以前にデコードされたピクチャ231は、ビデオシーケンスを形成するピクチャのシーケンスの一部であってもよく、それを形成してもよい。
【０１１１】
エンコーダ20は、たとえば、複数の他のピクチャの同じまたは異なるピクチャの複数の参照ブロックから参照ブロックを選択し、また、たとえば、参照ブロックの位置(x、y座標)と現在のブロックの位置との間の参照ピクチャ(または、参照ピクチャインデックス)および/またはオフセット(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、動きベクトル(MV)とも呼ばれる。
【０１１２】

動き補償ユニットは、インター予測ブロック265を取得するために、インター予測パラメータに基づいて、またはそれを使用して、インター予測パラメータを取得、たとえば受信することと、インター予測を行うこととを行うように構成される。動き補償ユニットによって行われる動き補償は、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または生成すること、場合によってはサブピクセル精度への補間を行うことを含み得る。補間フィルタリングは、知られている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可能性があり、したがって、ピクチャブロックをコーディングするために使用され得る候補予測ブロックの数が増える可能性がある。現在のピクチャブロックのPUの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突き止めることができる。

【０１１３】

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ドする際にビデオデコーダ30によって使用されるブロックおよびビデオスライスに関連付けられる構文要素を生成し得る。スライスおよびそれぞれの構文要素に加えて、またはそれらの代替として、タイルグループおよび/またはタイル、ならびにそれぞれの構文要素が生成または使用され得る。

【０１１４】

エントロピーコーディング

エントロピーエンコーディングユニット270は、たとえば、ビデオデコーダ30がパラメータを受信し、デコーディングのために使用することができるように、たとえばエンコードされたビットストリーム21の形式で、出力272を介して出力することができるエンコードされたピクチャデータ21を取得するために、たとえば、エントロピーエンコーディングアルゴリズムまたは方式(たとえば、可変長コーディング(VLC)方式、コンテキスト適応型VLC方式(CAVLC)、算術コーディング方式、二値化、コンテキスト適応型バイナリ算術コーディング(CABAC)、構文ベースのコンテキスト適応型バイナリ算術コーディング(SBAC)、確率間隔パーティショニングエントロピー(PIPE)コーディングまたは他のエントロピーエンコーディングの方法論または技法)、あるいは量子化された係数のバイパス(圧縮なし)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および/または他の構文要素を適用するように構成されている。エンコードされたビットストリーム21は、ビデオデコーダ30に伝送されてもよく、ビデオデコーダ30による後の伝送または検索のためにメモリに記憶されてもよい。

【０１１５】

ビデオストリームをエンコードするために、ビデオエンコーダ20の他の構造上の変形を使用することができる。たとえば、非変換ベースのエンコーダ20は、特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なしで残差信号を直接量子化することができる。他の実装形態では、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を単一のユニットに結合することができる。

【０１１６】

デコーダおよびデコーディング方法

図3は、本出願の技法を実装するように構成されたビデオデコーダ30の一例を示している。ビデオデコーダ30は、デコードされたピクチャ331を取得するために、たとえばエンコーダ20によってエンコードされた、エンコードされたピクチャデータ21(たとえば、エンコードされたビットストリーム21)を受信するように構成されている。エンコードされたピクチャデータまたはビットストリームは、エンコードされたピクチャデータ、たとえば、エンコードされたビデオスライス(および/あるいは、タイルグループまたはタイル)および関連付けられる構文要素のピクチャブロックを表すデータをデコードするための情報を含む。

【０１１７】

図3の例では、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、再構築ユニット314(たとえば、アナログ加算器3

14)、ループフィルタ320、デコード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を含む。インター予測ユニット344は、動き補償ユニットであってもよく、動き補償ユニットを含んでいてもよい。ビデオデコーダ30は、いくつかの例では、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逆のデコーディングパスを行い得る。

【０１１８】

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212、再構築ユニット214、ループフィルタ220、デコード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「ビルトインデコーダ」を形成するとも呼ばれる。したがって、逆量子化ユニット310は逆量子化ユニット210と機能において同一であってもよく、逆変換処理ユニット312は逆変換処理ユニット212と機能において同一であってもよく、再構築ユニット314は、再構築ユニット214と機能において同一であってもよく、ループフィルタ320は、ループフィルタ220と機能において同一であってもよく、デコードされたピクチャバッファ330は、デコードされたピクチャバッファ230と機能において同一であってもよい。したがって、ビデオエンコーダ20のそれぞれのユニットおよび機能について提供される説明は、ビデオデコーダ30のそれぞれのユニットおよび機能に対応して適用される。

【０１１９】

エントロピーデコーディング

エントロピーデコーディングユニット304は、たとえば、量子化された係数309および/またはデコードされたコーディングパラメータ(図3には示されていない)、たとえば、インター予測パラメータ(たとえば、参照ピクチャインデックスと動きベクトル)、イントラ予測パラメータ(たとえば、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、および/または他の構文要素の一部またはすべてを取得するために、ビットストリーム21(または、一般的にエンコードされたピクチャデータ21)を解析し、たとえば、エンコードされたピクチャデータ21に対してエントロピーデコーディングを行うように構成されている。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のエントロピーエンコーディングユニット270に関して説明したように、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、インター予測パラメータ、イントラ予測パラメータ、および/または他の構文要素をモード適用ユニット360に提供し、他のパラメータをデコーダ30の他のユニットに提供するようにさらに構成され得る。ビデオデコーダ30は、ビデオスライスレベルおよび/またはビデオブロックレベルにおいて構文要素を受信し得る。スライスおよびそれぞれの構文要素に加えて、またはそれらの代替として、タイルグループおよび/またはタイル、およびそれぞれの構文要素が受信および/または使用され得る。

【０１２０】

逆量子化

逆量子化ユニット310は、(たとえば、エントロピーデコーディングユニット304によって、たとえば構文解析および/またはデコーディングすることによって)エンコードされたピクチャデータ21から量子化パラメータ(QP)(または、一般的には逆量子化に関連する情報)および量子化された係数を受信することと、変換係数311とも呼ばれる非量子化係数311を取得するために、量子化パラメータに基づいて、デコードされた量子化係数309に逆量子化を適用することとを行うように構成され得る。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(あるいは、タイルまたはタイルグループ)内の各ビデオブロックに対してビデオエンコーダ20によって決定された量子化パラメータを使用することを含み得る。

【０１２１】

逆変換

逆変換処理ユニット312は、変換係数311とも呼ばれる非量子化係数311を受信するこ

と、サンプル領域において再構築された残差ブロック213を取得するために、非量子化係数311に変換を適用することとを行うように構成され得る。再構築された残差ブロック213は、変換ブロック313とも呼ばれ得る。変換は、逆変換、たとえば、逆DCT、逆DST、逆整数変換、または概念的に類似の逆変換プロセスであり得る。逆変換処理ユニット312は、非量子化係数311に適用される変換を決定するために、(たとえば、エントロピーデコーディングユニット304によって、たとえば構文解析および/またはデコーディングすることによって)エンコードされたピクチャデータ21から変換パラメータまたは対応する情報を受信するようにさらに構成され得る。

【０１２２】

再構築

再構築ユニット314(たとえば、加算器またはアナログ加算器314)は、サンプルドメインにおいて再構築されたブロック315を取得するために、たとえば、再構築された残差ブロック313のサンプル値と予測ブロック365のサンプル値を加算することによって、再構築された残差ブロック313を予測ブロック365に加算するように構成され得る。

【０１２３】

フィルタリング

(コーディングループ内、またはコーディングループ後のいずれかの)ループフィルタユニット320は、フィルタリングされたブロック321を取得するために、再構築されたブロック315をフィルタリングするように、たとえば、ピクセル遷移を平滑化するか、そうでなければビデオ品質を改善するように構成される。ループフィルタユニット320は、デブロッキングフィルタ、サンプル適応オフセット(SAO)フィルタ、あるいは1つまたは複数の他のフィルタ、たとえば、バイラテラルフィルタ、適応ループフィルタ(ALF)、鮮鋭化、平滑化フィルタ、または協調フィルタなどの1つまたは複数のループフィルタ、あるいはそれらの任意の組合せを含みうる。図3において、ループフィルタユニット320がインループフィルタとして示されているが、他の構成では、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もよい。

【０１２４】

デコードされたピクチャバッファ

次いで、ピクチャのデコードされたビデオブロック321は、デコードされたピクチャバッファ330に記憶され、デコードされたピクチャバッファ330は、デコードされたピクチャ331を、他のピクチャの後続の動き補償、および/またはそれぞれの出力表示用の参照ピクチャとして記憶する。

【０１２５】

デコーダ30は、デコードされたピクチャ331を、たとえば出力312を介して、ユーザに、提示または閲覧するために、出力するように構成される。

【０１２６】

予測

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、動き補償ユニット)と同一であり得、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と同一であり得、(たとえば、エントロピーデコーディングユニット304によって、たとえば構文解析および/またはデコーディングすることによって)エンコードされたピクチャデータ21から受信したパーティショニングおよび/または予測パラメータ、あるいはそれぞれの情報に基づいて、スプリットまたはパーティショニング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を取得するために、再構築されたピクチャ、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た、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測)を実行するように構成され得る。

【０１２７】

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場合、モード適用ユニット360のイントラ予測ユニット354は、シグナリングされたイントラ予測モードお

よび現在のピクチャの以前にデコードされたブロックからのデータに基づいて、現在のビデオスライスのピクチャブロックの予測ブロック365を生成するように構成される。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場合、モード適用ユニット360のインター予測ユニット344(たとえば、動き補償ユニット)は、エントロピーデコーディングユニット304から受信した動きベクトルおよび他の構文要素に基づいて、現在のビデオスライスのビデオブロックの予測ブロック365を生成するように構成される。インター予測の場合、予測ブロックは、参照ピクチャリストのうちの1つに含まれる参照ピクチャのうちの1つから生成され得る。ビデオデコーダ30は、DPB330に記憶された参照ピクチャに基づくデフォルト構築技法を使用して、参照フレームリスト、リスト0およびリスト1を構築し得る。タイルグループ(たとえば、ビデオタイルグループ)および/またはタイル(たとえば、ビデオタイル)をスライス(たとえば、ビデオスライス)に加えてまたはそれの代替として使用する実施形態に対して、または実施形態によって、同じまたは同様のことが適用され得、たとえば、ビデオは、I、P、またはBタイルグループおよび/またはタイルを使用してコード化され得る。

【０１２８】

モード適用ユニット360は、動きベクトルまたは関連情報および他の構文要素を構文解析することによって、現在のビデオスライスのビデオブロックの予測情報を決定するように構成され、デコードされている現在のビデオブロックの予測ブロックを生成するために、予測情報を使用する。たとえば、モード適用ユニット360は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード(たとえば、イントラまたはインター予測)、インター予測スライスタイプ(たとえば、Bスライス、Pスライス、またはGPBスライス)、スライスの参照ピクチャリストのうちの1つまたは複数の構築情報、スライスのインターエンコードされたビデオブロックごとの動きベクトル、スライスのインターコード化されたビデオブロックごとのインター予測ステータス、および現在のビデオスライスにおけるビデオブロックをデコードするための他の情報を決定するために、受信した構文要素のうちの一部を使用する。タイルグループ(たとえば、ビデオタイルグループ)および/またはタイル(たとえば、ビデオタイル)をスライス(たとえば、ビデオスライス)に加えてまたはそれの代替として使用する実施形態に対して、または実施形態によって、同じまたは同様のことが適用され得、たとえば、ビデオは、I、P、またはBタイルグループおよび/またはタイルを使用してコード化され得る。

【０１２９】

図3に示されるビデオデコーダ30の実施形態は、スライス(ビデオスライスとも呼ばれる)を使用することによって、ピクチャをパーティショニングおよび/またはデコードするように構成され得、ピクチャは1つまたは複数のスライス(通常は重複しない)を使用してパーティショニングまたはデコードされ得、各スライスは1つまたは複数のブロック(たとえば、CTU)を含みうる。

【０１３０】

図3に示されるビデオデコーダ30の実施形態は、タイルグループ(ビデオタイルグループとも呼ばれる)および/またはタイル(ビデオタイルとも呼ばれる)を使用することによって、ピクチャをパーティショニングおよび/またはデコードするように構成され得、ピクチャは、1つまたは複数のタイルグループ(通常は重複しない)を使用してパーティショニングまたはデコードされ得、各タイルグループは、たとえば、1つまたは複数のブロック(たとえば、CTU)あるいは1つまたは複数のタイルを含み得、各タイルは、たとえば長方形であってもよく、1つまたは複数のブロック(たとえば、CTU)を含んでもよく、たとえば、完全ブロックまたは部分ブロックでもよい。

【０１３１】

エンコードされたピクチャデータ21をデコードするために、ビデオデコーダ30の他の変形を使用することができる。たとえば、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。たとえば、非変換ベースのデコーダ30は、特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なしで残

差信号を直接逆量子化することができる。他の実装形態では、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を単一のユニットに結合することができる。

【０１３２】

エンコーダ20およびデコーダ30において、現在のステップの処理結果がさらに処理されて、次のステップに出力される場合があることを理解されたい。たとえば、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、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対して、クリップまたはシフトなどのさらなる演算が実行され得る。

【０１３３】

現在のブロックの導出された動きベクトルにさらなる演算(アフィンモードの制御ポイント動きベクトル、アフィン、平面、ATMVPモード、時間動きベクトルなどにおけるサブブロック動きベクトルを含むが、これらに限定されない)が適用される可能性があることに留意されたい。たとえば、動きベクトルの値は、その表現ビットに従ってあらかじめ定義された範囲に制限される。動きベクトルの代表ビットがbitDepthである場合、範囲は2^(bitDepth-1)～2^(bitDepth-1)-1であり、「^」は累乗を意味する。たとえば、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。たとえば、導出された動きベクトルの値(たとえば、1つの8×8ブロック内の4つの4×4サブブロックのMV)は、4つの4×4サブブロックMVの整数部分間の最大差が、1ピクセル以下などNピクセル以下になるように制約される。ここでは、bitDepthに従って動きベクトルを制限するための2つの方法を示す。

【０１３４】

方法1:フロー演算によってオーバーフローMSB (最上位ビット)を削除する

$$ux=(mvx+2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

$$mvx=(ux >= 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

$$uy=(mvy+2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

$$mvy=(uy >= 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

上式で、mvxは画像ブロックまたはサブブロックの動きベクトルの水平成分であり、mvyは画像ブロックまたはサブブロックの動きベクトルの垂直成分であり、uxとuyは中間値を示す。

【０１３５】

たとえば、mvxの値が-32769である場合、式(1)と(2)を適用した後、結果の値は32767になる。コンピュータシステムでは、10進数は2の補数として記憶される。-32769の2の補数は1、0111、1111、1111、1111(17ビット)であり、MSBが破棄されるため、結果の2の補数は0111、1111、1111、1111(10進数は32767)となり、式(1)と(2)を適用することによる出力と同じである。

$$ux=(mvpx + mvdx +2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

$$mvx=(ux >= 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

$$uy=(mvpy + mvdy +2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

$$mvy=(uy >= 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (8)$$

【０１３６】

演算は、式(5)から(8)に示されるように、mvpとmvdの合計の間に適用され得る。

【０１３７】

方法2:値をクリッピングすることによってオーバーフローMSBを削除する

$$vx=Clip3(-2^{bitDepth-1}, 2^{bitDepth-1}-1, vx)$$

$$vy=Clip3(-2^{bitDepth-1}, 2^{bitDepth-1}-1, vy)$$

上式で、vxは画像ブロックまたはサブブロックの動きベクトルの水平成分であり、vyは画像ブロックまたはサブブロックの動きベクトルの垂直成分であり、x、y、およびzはそれぞれMVクリッピングプロセスの3つの入力値に対応し、関数Clip3の定義は次のとおりで

ある。

【０１３８】

【数１】

$$\mathrm{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{そうではない場合} \end{cases}$$

【０１３９】

図4は、本開示の一実施形態によるビデオコーディングデバイス400の概略図である。ビデオコーディングデバイス400は、本明細書に記載の開示された実施形態を実装するために適している。一実施形態では、ビデオコーディングデバイス400は、図1Aのビデオデコーダ30などのデコーダであってもよく、図1Aのビデオエンコーダ20などのエンコーダであってもよい。

【０１４０】

ビデオコーディングデバイス400は、データを受信するための入力ポート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、データを伝送するためのトランスミッタユニット(Tx)440および出力ポート450(または、出力ポート450)ならびにデータを記憶するためのメモリ460を含む。ビデオコーディングデバイス400はまた、光信号または電気信号の出入り用に、入力ポート410、受信機ユニット420、トランスミッタユニット440、および出力ポート450に結合された光電気(OE)構成要素および電気光(EO)構成要素を含みうる。

【０１４１】

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つまたは複数のCPUチップ、コア(たとえば、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、入力ポート410、受信機ユニット420、トランスミッタユニット440、出力ポート450、およびメモリ460と通信する。プロセッサ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装する。たとえば、コーディングモジュール470は、様々なコーディング動作を実装、処理、準備、または提供する。したがって、コーディングモジュール470を含めることにより、ビデオコーディングデバイス400の機能が大幅に改善され、ビデオコーディングデバイス400が異なる状態に変換される。あるいは、コーディングモジュール470は、メモリ460に記憶され、プロセッサ430によって実行される命令として実装される。

【０１４２】

メモリ460は、1つまたは複数のディスク、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得、そのような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記憶し、プログラムの実行中に読み取られる命令とデータを記憶するために、オーバーフローデータ記憶デバイスとして使用され得る。メモリ460は、たとえば、揮発性および/または不揮発性であってもよく、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、三元連想メモリ(TCAM)、および/またはスタティックランダムメモリアクセスメモリ(SRAM)であってもよい。

【０１４３】

図5は、例示的な実施形態による、図1のソースデバイス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され得る装置500の簡略化されたブロック図である

【０１４４】

装置500におけるプロセッサ502は、中央処理装置とすることができる。代替として、プロセッサ502は、現在存在する、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１つまたは複数のデバイスであり得る。開示された実装形態は、図示されているように単一のプロセッサ、たとえばプロセッサ502を用いて実施することができるが、速度および効率の利点は、複数のプロセッサを使用して達成することができる。

【０１４５】

装置500内のメモリ504は、実装形態において、読取り専用メモリ(ROM)デバイスまたはランダムアクセスメモリ(RAM)デバイスとすることができる。メモリ504として、任意の他の適切なタイプの記憶デバイスを使用することができる。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を含むことができる。メモリ504は、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をさらに含むことができ、アプリケーションプログラム510は、プロセッサ502が本明細書に記載の方法を実行することを許可する少なくとも1つのプログラムを含む。たとえば、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を含むことができ、アプリケーション1からNは、本明細書に記載の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さらに含む。

【０１４６】

装置500はまた、ディスプレイ518などの1つまたは複数の出力デバイスを含むことができる。ディスプレイ518は、一例では、タッチ入力を感知するように動作可能なタッチセンシティブ要素とディスプレイを組み合わせたタッチセンシティブ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に結合することができる。

【０１４７】

本明細書では単一のバスとして示されているが、装置500のバス512は複数のバスから構成することができる。さらに、セカンダリストレージ514は、装置500の他の構成要素に直接結合することができ、またはネットワークを介してアクセスすることができ、メモリカードなどの単一の統合ユニットまたは複数のメモリカードなどの複数のユニットを含むことができる。したがって、装置500は、多種多様な構成において実装することができる。

【０１４８】

背景

パラメータセット

パラメータセットは基本的に類似しており、同じ基本的な設計目標、すなわち、ビットレート効率、エラー回復力、およびシステム層インターフェースの提供を共有している。HEVC(H.265)には、ビデオパラメータセット(VPS)、シーケンスパラメータセット(SPS)、およびピクチャパラメータセット(PPS)を含む、パラメータセットの階層があり、AVCおよびVVCにおける対応するものに似ている。各スライスは、スライスをデコーディングするために使用される情報にアクセスするために、単一のアクティブなPPS、SPS、およびVPSを参照する。PPSは、ピクチャ内のすべてのスライスに適用される情報を含んでいるため、ピクチャ内のすべてのスライスは同じPPSを参照する必要がある。異なるピクチャ内のスライスも同じPPSを参照することができる。同様に、SPSは、同じコード化されたビデオシーケンス内のすべてのピクチャに適用される情報を含んでいる。

【０１４９】

PPSは個別のピクチャによって異なる場合があるが、コード化されたビデオシーケンス内の多くのまたはすべてのピクチャが同じPPSを参照するのは一般的である。パラメータセットを再利用すると、共有情報を複数回送信する必要がなくなるため、ビットレートが効率的になる。また、パラメータセットの内容をより信頼性の高い外部通信リンクによって伝送したり、ビットストリーム内で頻繁に繰り返したりして、失われないようにすることができるため、損失に強い。

【０１５０】

シーケンスパラメータセット(SPS):各ピクチャヘッダにある構文要素によって参照されるPPSにある構文要素の内容によって決定される、0個以上のCLVS全体に適用される構文要素を含む構文構造。

【０１５１】

スケーラブルなビデオコーディング、層、およびビデオパラメータセット(VPS)

スケーラブルなビデオコーディングは、複数の層内のビデオをコーディングするためのメカニズムを提供し、各層は、同じビデオシーンの異なる品質表現を表す。ベース層は、最低品質の表現である。下位層を参照することによって1つまたは複数のエンハンスメント層(EL)がコード化され、改善されたビデオ品質を提供し得る。スケーラブルなコード化されたビデオビットストリームの層のサブセットをデコードすると、完全なビットストリームがデコードされた場合よりも品質が低くなるが、それでも許容できるビデオが得られる。これにより、通常、ビットレートを下げると、ビデオの品質が大幅に低下し、多くの場合、すぐに視聴に耐えられない品質になるスケーラブルでないビデオビットストリームと比較して、より穏やかな劣化が可能になる。

【０１５２】

スケーラブルなビデオシーケンスには、時間スケーラビリティ、空間スケーラビリティ、および品質スケーラビリティなどの、複数のタイプのスケーラビリティがある。図8は、空間スケーラビリティと時間スケーラビリティの両方を示す例を提供する。図8には、異なる解像度でコード化された2つの層がある。BLは解像度が低く、ELは解像度が高く、空間スケーラビリティは、BL、EL、またはその両方をデコードするデコーダを提供することによって実現される。

【０１５３】

空間スケーラビリティに加えて、コーディング層内で時間スケーラビリティが実現される。この例では、各コーディング層は2つの時間サブ層に分割され、それぞれ時間ID 0および1によってラベル付けされる。時間スケーラビリティは、時間サブ層0(時間IDが0に等しい)またはサブ層0と1の両方をデコードするためにデコーダを提供することによって実現される。

【０１５４】

異なる層内のピクチャには、異なる層ID、すなわち構文要素nuh_layer_idが割り当てられる。コード化層ビデオシーケンス(CLVS)は、同じ値のnuh_layer_idを有するピクチャのシーケンスであり、デコーディング順で、コーディングピクチャ(CLVSS、たとえばイントラピクチャ)を開始する特別なコーディング層ビデオシーケンスで構成され、その後に、CLVSSピクチャまでの後続のすべてのピクチャを含むが、CLVSSピクチャである後続のピクチャは含まない、CLVSSピクチャではない0個以上のピクチャが続く。

【０１５５】

VPSが適用され得るコード化されたビデオシーケンス(CVS)は、1つまたは複数のコード化層ビデオシーケンス(CLVS)で構成される。図8の例では、BLとELの最初のピクチャがCLVSSピクチャであり、他のすべてのピクチャがCLVSSピクチャでないと仮定すると、これは2つのCLVSから構成されるCVSである。

【０１５６】

同じ位置で層が異なるピクチャ(図8のAとBなど)は、アクセスユニット(AU)を形成する。アクセスユニットは、異なる層に属し、デコードされたピクチャバッファから出力される同じ時間に関連付けられるコード化されたピクチャを含む、ピクチャのセットで構成される。

【０１５７】

層は、独立層または依存層を含み、依存層は、その1つまたは複数の参照層を参照することによって層間予測を使用する必要があると予測し、ここで、参照層の層IDは、依存層の層IDよりも小さい場合がある。独立層は、層間予測を使用せずに予測される。

【０１５８】

次の表は、VPSにおいてシグナリングされる構文要素の例である。

【０１５９】
【表１Ａ】

Table 1:VPSにおいてシグナリングされる構文要素

video_parameter_set_rbsp( ) {	記述子
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(6)
vps_max_sublayers_minus1	u(3)
if(vps_max_layers_minus1 > 0 && vps_max_sublayers_minus1 > 0)	
vps_all_layers_same_num_sublayers_flag	u(1)
if(vps_max_layers_minus1 > 0)	
vps_all_independent_layers_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i]) {	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
max_tid_ref_present_flag[i]	u(1)
if(max_tid_ref_present_flag[i])	
max_tid_il_ref_pics_plus1[i]	u(3)
}	
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i ++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols_output_layer_flag[i][j]	u(1)
}	
}	
}	

【０１６０】

【表１Ｂ】

vps_num_ptls_minus1	u(8)
for(i = 0; i <= vps_num_ptls_minus1; i++) {	
if(i > 0)	
pt_present_flag[i]	u(1)
if(vps_max_sublayers_minus1 > 0 && !vps_all_layers_same_num_sublayers_flag)	
ptl_max_temporal_id[i]	u(3)
}	
while(!byte_aligned( ))	
vps_ptl_alignment_zero_bit /* equal to 0 */	f(1)
for(i = 0; i <= vps_num_ptls_minus1; i++)	
profile_tier_level(pt_present_flag[i], ptl_max_temporal_id[i])	
for(i = 0; i < TotalNumOlss; i++)	
if(vps_num_ptls_minus1 > 0)	
ols_ptl_idx[i]	u(8)
if(!vps_all_independent_layers_flag)	
vps_num_dpb_params	ue(v)
if(vps_num_dpb_params > 0 && vps_max_sublayers_minus1 > 0)	
vps_sublayer_dpb_params_present_flag	u(1)
for(i = 0; i < vps_num_dpb_params; i++) {	
if(vps_max_sublayers_minus1 > 0 && !vps_all_layers_same_num_sublayers_flag)	
dpb_max_temporal_id[i]	u(3)
dpb_parameters(dpb_max_temporal_id[i], vps_sublayer_dpb_params_present_flag)	
}	
for(i = 0; i < TotalNumOlss; i++) {	
if(NumLayersInOls[i] > 1) {	
ols_dpb_pic_width[i]	ue(v)
ols_dpb_pic_height[i]	ue(v)
if(vps_num_dpb_params > 1)	
ols_dpb_params_idx[i]	ue(v)
}	
}	
if(!each_layer_is_an_ols_flag)	
vps_general_hrd_params_present_flag	u(1)
if(vps_general_hrd_params_present_flag) {	
general_hrd_parameters( )	
if(vps_max_sublayers_minus1 > 0)	

【０１６１】

【表１Ｃ】

vps_sublayer_cpb_params_present_flag	u(1)
num_ols_hrd_params_minus1	ue(v)
for(i = 0; i <= num_ols_hrd_params_minus1; i++) {	
if(vps_max_sublayers_minus1 > 0 && !vps_all_layers_same_num_sublayers_flag)	
hrd_max_tid[i]	u(3)
firstSubLayer = vps_sublayer_cpb_params_present_flag ? 0 : hrd_max_tid[i]	
ols_hrd_parameters(firstSubLayer, hrd_max_tid[i])	
}	
if(num_ols_hrd_params_minus1 + 1 != TotalNumOlss && num_ols_hrd_params_minus1 > 0)	
for(i = 1; i < TotalNumOlss; i++)	
if(NumLayersInOls[i] > 1)	
ols_hrd_idx[i]	ue(v)
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data( ))	
vps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

【０１６２】
たとえば、vps_video_parameter_set_idは、次のセクションにおいて紹介するSPSによって参照することができるIDを表す。
【０１６３】
たとえば、vps_max_layers_minus1+1は、CVSにおいて許容される層の最大数を表す。
【０１６４】
たとえば、vps_max_sublayers_minus1+1は、CVにおいて許容される時間サブ層の最大数を表す。
【０１６５】
たとえば、vps_all_layers_same_num_sublayers_flagは、そのすべてのコーディング層が同じ数の時間サブ層を含むCVSを特徴とする。許容される層の最大数と許容される時間サブ層の最大数とが1より大きい場合にのみシグナリングされる。
【０１６６】
たとえば、vps_all_independent_layers_flagは、すべてのコーディング層が独立した層である、すなわち、各層に層間参照ピクチャが存在しないというCVSを特徴とする。層間参照ピクチャは、インター予測が同じ層ではなく他の層のピクチャを使用する場合に使用されることがある。
【０１６７】
たとえば、両端値を含む0からvps_max_layers_minus1までの範囲のiのvps_layer_id[i]は、層インデックス0からvps_max_layers_minus1まで、各層の層IDを表す。層インデックスと層IDとの間の違いに注意する価値がある。vps_layer_id[0]は0番目の層のIDを示し、0と等しくない可能性がある。

【０１６８】
簡単にするために、本明細書では残りの構文要素の紹介はスキップする。それらの説明については、VVCドラフト8を参照されたい。
【０１６９】
シーケンスパラメータセット(SPS)
SPSは、コード化されたビデオシーケンスの1つまたは複数の層に適用され、コード化されたビデオシーケンス内のピクチャごとに変化しないパラメータを含む。特に、SPSはサブピクチャをシグナリングする情報を含む。
【０１７０】
次の表の一部は、ITU JVET-Q2001-v13のSPSにおけるサブピクチャシグナリングの一部のスナップショットを示しており、ダウンロードリンクは次のとおりである。
http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/17_Brussels/wg11/JVET-Q2001-v13.zip.
本出願の残りの部分では、この先行技術文献は、簡単にするためにVVCドラフト8と命名される。
【０１７１】

【表２Ａ】

Table 2: SPS構文要素

seq_parameter_set_rbsp( ) {	記述子
sps_seq_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)
sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <= sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_width_max_in_luma_samples > CtbSizeY)	

【０１７２】

【表２Ｂ】

subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
…	
…	

【０１７３】
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のvps_video_parameter_set_idの値を指定する。
【０１７４】
sps_video_parameter_set_idが0と等しい場合、次が適用される。
-SPSはVPSを参照せず、SPSを参照して各CLVSをデコードする際にVPSを参照しない。
-vps_max_layers_minus1の値は0に等しいと推論される。
-CVSは1つの層のみを含むものとする(すなわち、CVS内のすべてのVCL NALユニットは同じ値のnuh_layer_idを有するものとする)。
【０１７５】
SPS内のいくつかの構文要素は、位置情報と各サブピクチャの制御フラグとをシグナリングする。i番目のサブピクチャの位置情報は次のものを含む。
・subpic_ctu_top_left_x[i]は、ピクチャ内のサブピクチャiの左上座標の水平成分を示す、または、
・subpic_ctu_top_left_y[i]は、ピクチャ内のサブピクチャiの左上座標の垂直成分を示す、または、
・subpic_width_minus1[i]は、ピクチャ内のサブピクチャiの幅を示す、または、
・subpic_height_minus1[i]は、ピクチャ内のサブピクチャiの高さを示す。
【０１７６】
いくつかの構文要素、たとえばsps_num_subpics_minus1は、ピクチャ内のサブピクチャの数を示す。
【０１７７】
CTU、スライス、タイル、およびサブピクチャへのピクチャのパーティショニング
【０１７８】
CTUへのピクチャのパーティショニング
ピクチャは、コーディングツリーユニット(CTU)のシーケンスに分割される。CTB(コーディングツリーブロック)という用語は、交換可能に使用されることがある。CTUの概念はHEVCの概念と同じである。3つのサンプル配列を有するピクチャの場合、CTUは輝度サンプルのN×Nブロックと、対応する2つの彩度サンプルブロックで構成される。図6は、ピクチャをCTUに分割した例を示している。フレーム内のCTUのサイズは、ピクチャの境界(不完全なCTUが存在する可能性がある場所)のものを除いて、同じでなければならない。
【０１７９】
タイルへのピクチャのパーティショニング
タイルが有効である場合、ピクチャは、垂直および/または水平境界で区切られたCTUの

長方形のグループに分割される。垂直および水平のタイル境界は、それぞれ、ピクチャを上から下へ、また左のピクチャの境界から右のピクチャの境界まで交差する。ビットストリームには、前述の水平および垂直タイル境界の位置に関連する指示が含まれている。

【０１８０】

スライスへのピクチャのパーティショニング

スライスの概念は、各スライスが同じピクチャの他のスライスから独立してデコードできるように、ピクチャのパーティショニングを提供し、デコーディングは、エントロピー、残差、および予測デコーディングを指す。タイルとの違いは、スライスはより任意の形状を有することができる(パーティショニングの可能性がより柔軟になる)ことと、スライスのパーティショニングの目的は並列処理ではなく、伝送環境におけるパケットサイズの一致とエラー回復力であることである。

【０１８１】

スライスは、完全なピクチャとその一部で構成され得る。HEVCにおいて、スライスは、処理順に複数の連続するピクチャのCTUで構成される。スライスは、スライスヘッダまたはピクチャパラメータセット、あるいは他のユニットにおいてシグナリングされるその開始CTUアドレスによって識別される。

【０１８２】

VVCのドラフト8では、スライスは整数個の完全なタイル、またはピクチャのタイル内の整数個の連続したCTU行を含む。したがって、各垂直スライス境界は常に垂直タイル境界でもある。スライスの水平境界がタイル境界ではなく、タイル内の水平CTU境界を含む可能性があり、これは、タイルが複数の長方形のスライスにスプリットされている場合に発生し、各スライスは、タイル内の整数の連続した完全なCTU行を含む。

【０１８３】

いくつかの例では、ラスタスキャンスライスモードと長方形スライスモードの2つのスライスモードがある。ラスタスキャンスライスモードでは、スライスは、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。長方形スライスモードでは、スライスはピクチャの長方形領域を集合的に形成する多数のタイルを含むか、または、スライスはピクチャの長方形領域を集合的に形成する1つのタイルの多数の連続するCTU行を含む。長方形のスライス内のタイルは、そのスライスに対応する長方形の領域内でタイルラスタスキャン順にスキャンされる。

【０１８４】

ピクチャのすべてのスライスがピクチャ全体を集合的に形成し、すなわち、ピクチャのすべてのCTUがピクチャのスライスのうちの1つに含まれている必要がある。タイルとサブピクチャにも同様のルールが適用される。

【０１８５】

サブピクチャへのピクチャのパーティショニング

ピクチャのサブピクチャ長方形パーティション。サブピクチャは、ピクチャ全体またはピクチャの一部であり得る。サブピクチャとは、各サブピクチャがビデオシーケンス全体の他のサブピクチャから独立してデコードできるようにピクチャをパーティショニングすることである。VVCドラフト8では、これはビットストリームにおいて示されている場合に当てはまりまる。すなわち、subpic_treated_as_pic_flag[i]指示がサブピクチャiに対して当てはまる場合、そのサブピクチャiは、ビデオシーケンス全体の他のサブピクチャから独立してデコード可能である。

【０１８６】

サブピクチャとタイルまたはスライスとの違いは、サブピクチャがビデオシーケンス内に独立してデコード可能なビデオシーケンスを作成することである。一方、タイルとスライスの場合、独立したデコーディングはビデオシーケンスの1つのピクチャ内でのみ保証される。

【０１８７】

VVCドラフト8では、サブピクチャは、ピクチャの長方形領域を集合的にカバーする1つ

または複数のスライスを含む。したがって、各サブピクチャ境界は常にスライス境界であり、各垂直サブピクチャ境界は常に垂直タイル境界である。

【０１８８】

図7は、タイル、スライス、およびサブピクチャの例を提供する。

【０１８９】

図7に示される一例では、ピクチャは、216個のCTU、4個のタイル、4個のスライス、および3個のサブピクチャにパーティショニングされる。sps_num_subpics_minus1の値は2であり、位置関連の構文要素は次の値を有する。

サブピクチャ0の場合

・subpic_ctu_top_left_x[0]はシグナリングされないが、0と推論される。

・subpic_ctu_top_left_y[0]はシグナリングされないが、0と推論される。

・subpic_width_minus1[0]、値は8である。

・subpic_height_minus1[0]、値は11である。

サブピクチャ1の場合

・subpic_ctu_top_left_x[1]、値は9である。

・subpic_ctu_top_left_y[1]、値は0である。

・subpic_width_minus1[1]、値は8である。

・subpic_height_minus1[1]、値は5である。

サブピクチャ2の場合

・subpic_ctu_top_left_x[2]、値は9である。

・subpic_ctu_top_left_y[2]、値は6である。

・subpic_width_minus1[2]はシグナリングされないが、8と推論される。

・subpic_height_minus1[2]はシグナリングされないが、5と推論される。

【０１９０】

コード化されたビットストリームは、ネットワーク抽象化層(NAL)ユニットで構成される。NALユニットは、必要に応じてエミュレーション防止バイトが散在する(Raw Byte Sequence Payload)RBSPの形式で、従うべきデータのタイプの指示とそのデータを含むバイトを含む構文構造である。

【０１９１】

RBSPは、NALユニットにカプセル化された整数のバイト数を含む構文構造であり、空であるか、または、構文要素を含むデータビットの文字列の形式であり、その後にRBSPストップビットと、0に等しい0個以上の後続ビットが続く。

エミュレーション防止バイトは、0×03に等しいバイトである。

【０１９２】

SPSまたはVPSなどのパラメータセットは、NALユニットにカプセル化される。SPSおよびVPSにおいて指定された構文要素情報を除いて、NALユニットごとにNALユニットヘッダがある。NALユニットヘッダは、層ID(すなわち、構文nuh_layer_id)、NALユニットタイプ(すなわち、構文nal_unit_type)などの情報を提供する。たとえば、VPSを含むNALユニットのNALユニットタイプはVPS_NUTであり、SPSを含むNALユニットのNALユニットタイプはSPS_NUTである。

【０１９３】

一般に、ビットストリーム内のNALユニットは、ビデオコーディング層(VCL)NALユニットと非VCL NALユニットの2つのクラスに分割することができる。ピクチャの本質的なコード化データを提供しないパラメータセット(VPSまたはSPSなど)を含むNALユニットは、非VCL NALユニットタイプに属する。対照的に、ピクチャの本質的なコード化データ(Iピクチャコード化ビデオデータなど)を含むNALユニットは、VCL NALユニットと呼ばれる。

【０１９４】

VVCドラフト8の表5では、NALユニットごとのNALユニットタイプとそれらの分類が定義されている。

【０１９５】
さらなる詳細については、VVCドラフト8を参照されたい。
【０１９６】
参照ピクチャリスト構築のためのデコーディングプロセス
このプロセスは、ピクチャのスライスごとのデコーディングプロセスの開始時に呼び出される。
【０１９７】
参照ピクチャは、参照インデックスを通じて扱われる。参照インデックスは、RPLへのインデックスである。Iスライスをデコードする場合、スライスデータをデコードする際にRPLは使用されない。Pスライスをデコードする場合、RPL0(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使用される。Bスライスをデコードする場合、RPL0とRPL1の両方(すなわち、RefPicList[1])がスライスデータのデコーディングに使用される。
【０１９８】
ピクチャの各スライスのデコード処理の開始時に、RPL RefPicList[0]およびRefPicList[1]が導出される。RPLは、8.3.3節において指定されている参照ピクチャのマーキング、またはスライスデータのデコードに使用される。
注1-ピクチャのIスライスの場合、RefPicList[0]およびRefPicList[1]は、ビットストリーム適合性チェックの目的で導出することができるが、それらの導出は、現在のピクチャまたはデコーディング順序で現在のピクチャに続くピクチャのデコーディングには必要ない。ピクチャのPスライスの場合、RefPicList[1]はビットストリーム適合性チェックの目的で導出することができるが、その導出は、現在のピクチャまたはデコーディング順序で現在のピクチャに続くピクチャのデコーディングには必要ない。
【０１９９】
sps_idr_rpl_present_flagが0に等しく、pps_rpl_info_in_ph_flagが0に等しく、nal_unit_typeがIDR_W_RADLまたはIDR_N_LPに等しい場合、RPL RefPicList[0]およびRefPicList[1]は両方とも空になるように、すなわち0個のエントリを含むように導出され、以下は、0または1に等しい各iに適用される。
RplsIdx[i]の値は、sps_num_ref_pic_lists[i]と等しいと推論される。
num_ref_entries[i][RplsIdx[i]]の値は、0に等しいと推論される。
NumRefIdxActive[i]の値は0に等しいと推論される。
【０２００】
そうでなければ、RPL RefPicList[0]およびRefPicList[1]、参照ピクチャのスケーリング比RefPicScale[i][j][0]およびRefPicScale[i][j][1]、ならびに参照ピクチャスケーリングフラグRprConstraintsActiveFlag[0][j]およびRprConstraintsActiveFlag[1][j]は次のように導出される。

```
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
            and PicOrderCntVal equal to RefPicPocList[i][j] )
          RefPicList[i][j] = picA
        else
          RefPicList[i][j] = "no reference picture"  (200)
        pocBase = RefPicPocList[i][j]
      }else{
```

```
            if( !delta_poc_msb_cycle_present_flag[i][k] ) {
               if(there is a reference picA in the DPB with the same nuh_layer
_id as the current picture and
                     PicOrderCntVal&(MaxPicOrderCntLsb-1) equal to PocLsbL
t[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_laye
r_id 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++
         }
      } else {
         layerIdx = DirectRefLayerIdx[GeneralLayerIdx[nuh_layer_id]][ilrp_idx
[i][RplsIdx[i]][j]]
         refPicLayerId = vps_layer_id[layerIdx]
         if( there is a reference picture picA in the DPB with nuh_layer_id eq
ual to refPicLayerId and
               the same PicOrderCntVal as the current picture )
            RefPicList[i][j] = picA
         else
            RefPicList[i][j] = "no reference picture"
      }
      fRefWidth is set equal to CurrPicScalWinWidthL of the reference pictur
e RefPicList[i][j]
      fRefHeight is set equal to CurrPicScalWinHeightL of the reference pictu
re RefPicList[i][j]
      refPicWidth, refPicHeight, refScalingWinLeftOffset, refScalingWinRight
Offset, refScalingWinTopOffset,
            and refScalingWinBottomOffset, are set equal to the values of pps
_pic_width_in_luma_samples,
            pps_pic_height_in_luma_samples, pps_scaling_win_left_offset, pp
s_scaling_win_right_offset,
            pps_scaling_win_top_offset, and pps_scaling_win_bottom_offset, r
espectively, of the reference
            picture RefPicList[i][j]
      fRefNumSubpics is set equal to sps_num_subpics_minus1 of the referen
ce picture RefPicList[i][j]
      RefPicScale[i][j][0] = ((fRefWidth 〈 〈14)+(CurrPicScalWinWidthL〉 〉 1))/
            CurrPicScalWinWidthL
      RefPicScale[i][j][1] = ((fRefHeight 〈 〈14)+(CurrPicScalWinHeightL〉 〉 1)
)/
```

```
            CurrPicScalWinHeightL
      RprConstraintsActiveFlag[i][j] = ( pps_pic_width_in_luma_samples != r
efPicWidth  ||
            pps_pic_height_in_luma_samples != refPicHeight  ||
            pps_scaling_win_left_offset != refScalingWinLeftOffset  ||
            pps_scaling_win_right_offset != refScalingWinRightOffset  ||
            pps_scaling_win_top_offset != refScalingWinTopOffset  ||
            pps_scaling_win_bottom_offset != refScalingWinBottomOffset  ||
            sps_num_subpics_minus1 != fRefNumSubpics )
   }
}
```

【０２０１】

０または１に等しい各ｉについて、RefPicList[i]における最初のNumRefIdxActive[i]エントリはRefPicList[i]におけるアクティブエントリと呼ばれ、RefPicList[i]における他のエントリは、RefPicList[i]における非アクティブエントリと呼ばれる。

【０２０２】

問題と解決策の説明

第１の問題

VPSは任意のパラメータセットである。たとえば、単一層のビットストリームの場合、ビットストリーム内にVPSがない可能性がある。VPS内の一部の構文要素のセマンティクスが不完全であることが確認されている。VPSがビットストリーム内にない場合、それらの定義はない。

【０２０３】

たとえば、VVCドラフト８では、構文要素vps_max_layers_minus1のセマンティクスは次のように指定されている。

vps_max_layers_minus1+1は、VPSを参照する各CVSにおいて許容される層の最大数を指定する。

【０２０４】

実施形態１

第１の実施形態によれば、VPSがビットストリームに存在しない場合を考慮して、vps_max_layers_minus1のセマンティックは次のように修正される。すなわち、vps_max_layers_minus1+1は、VPSを参照する各CVにおいて許容される層の最大数を指定する。存在しない場合、vps_max_layers_minus1の値は0に等しいと推論される。

【０２０５】

言い換えれば、VPSが存在しない場合。CVSには1つの層しかない。

【０２０６】

第２の問題

VPSにおける範囲0からvps_max_layers_minusの他の構文要素vps_layer_id[i]にも同じ問題があり、すなわち、VPSが存在しない場合に推論値を指定しない。vps_layer_id[i]のセマンティクスは、VVCドラフト８において次のように指定されている。すなわち、vps_layer_id[i]は、i番目の層のnuh_layer_id値を指定する。mとnの任意の2つの負でない整数値について、mがnより小さい場合、vps_layer_id[m]の値はvps_layer_id[n]より小さいものとする。

nuh_layer_idは層IDを表す。vps_layer_id[i]は、i番目の層の層IDを指定する。この定義は、より多くの層がある場合、上位層(層インデックスnを有する)の層IDが下位層(層インデックスmを有する)の層IDよりも大きくなければならないことも制約し、すなわち、vps_layer_id配列において指定された層IDは昇順である。

【０２０７】

vps_layer_id[i]の値は、VVCドラフト８におけるいくつかの中間変数を導出するために使用されることに注意されたい。たとえば、変数OutputLayerIdInOls[0][0]の導出は、

両端値を含む0からvps_max_layers_minus1までの範囲のiのvps_layer_id[i]の値に依存し、OutputLayerIdInOls[0][0]は、VVCドラフト8の条項8.1.2において指定されているように、デコーディングプロセスを決定するためにさらに使用される場合がある。VPSがビットストリームに存在しない場合、両端値を含む0からvps_max_layers_minus1の範囲のiのvps_layer_id[i]の推論値がないことによりOutputLayerIdInOls[0][0]の値が未定義であるため、デコーディングプロセスが未定義になる可能性がある。

【０２０８】

実施形態2

第2の実施形態によれば、VPSがビットストリームに存在しない場合を考慮して、vps_layer_id[i]のセマンティクスは次のように修正される。すなわち、vps_layer_id[i]は、i番目の層のnuh_layer_id値を指定する。mとnの任意の2つの負でない整数値について、mがnより小さい場合、vps_layer_id[m]の値はvps_layer_id[n]より小さいものとする。存在しない場合、vps_layer_id[0]は、シーケンスパラメータセット(SPS)のnuh_layer_idの値と等しいと推論される。

【０２０９】

言い換えれば、VPSが存在しない場合、ビデオコーディング層は1つしかない。構文要素vps_layer_id配列は1つのエントリ(すなわち、0番目の層)のみを有することができ、その値vps_layer_id[0]はSPSの層IDと等しいと推論される。

【０２１０】

実施形態3

第3の実施形態によれば、VPSがビットストリームに存在しない場合を考慮して、vps_layer_id[i]のセマンティクスは次のように修正される。すなわち、vps_layer_id[i]は、i番目の層のnuh_layer_id値を指定する。mとnの任意の2つの負でない整数値について、mがnより小さい場合、vps_layer_id[m]の値はvps_layer_id[n]より小さいものとする。存在しない場合、vps_layer_id[0]は、第1の非VCL NALユニットのnuh_layer_idの値と等しいと推論される。

【０２１１】

言い換えれば、VPSが存在しない場合、ビデオコーディング層は1つしかない。構文要素vps_layer_id配列は1つのエントリ(すなわち、0番目の層)のみを有することができ、その値vps_layer_id[0]は非VCL NALユニットの層IDの値と等しいと推論される。

【０２１２】

実施形態4

第4の実施形態によれば、VPSがビットストリームに存在しない場合を考慮して、vps_layer_id[i]のセマンティクスは次のように修正される。すなわち、vps_layer_id[i]は、i番目の層のnuh_layer_id値を指定する。mとnの任意の2つの負でない整数値について、mがnより小さい場合、vps_layer_id[m]の値はvps_layer_id[n]より小さいものとする。存在しない場合、vps_layer_id[0]は、第1のVCL NALユニットのnuh_layer_idの値と等しいと推論される。

【０２１３】

言い換えれば、VPSが存在しない場合、ビデオコーディング層は1つしかない。構文要素vps_layer_id配列は1つのエントリ(すなわち、0番目の層)のみを有することができ、その値vps_layer_id[0]はVCL NALユニットの層IDの値と等しいと推論される。

【０２１４】

実施形態5

第5の実施形態によれば、VPSがビットストリームに存在しない場合を考慮して、vps_layer_id[i]のセマンティクスは次のように修正される。すなわち、vps_layer_id[i]は、i番目の層のnuh_layer_id値を指定する。mとnの任意の2つの負でない整数値について、mがnより小さい場合、vps_layer_id[m]の値はvps_layer_id[n]より小さいものとする。存在しない場合、vps_layer_id[0]は、任意のNALユニットのnuh_layer_idの値と等しいと推論される。

【０２１５】
言い換えれば、VPSが存在しない場合、ビデオコーディング層は1つしかない、すなわち、すべてのNALユニットはそれらのNALユニットヘッダ中で同じ値のnuh_layer_idを有する。構文要素vps_layer_id配列は1つのエントリ(すなわち、0番目の層)のみを有することができ、その値vps_layer_id[0]は任意のNALユニットの層IDの値と等しいと推論される。
【０２１６】
上記の説明に従って、ビデオビットストリームをデコードする方法およびビデオビットストリームをエンコードする方法が本明細書で提供される。それに対応して、(コード化された)ビデオビットストリームをデコードするための装置、およびビデオビットストリームをデコードするための装置が本明細書で提供される。
【０２１７】
図9は、デコーディングデバイスによって実装されるビデオビットストリームをデコーディングする方法を示しており、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するステップ910と、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合(たとえば、決定された場合のみ決定される)(たとえば、sps_video_parameter_set_idが0に等しい)、現在のビデオコーディング層のNALユニットの識別子(たとえば、nuh_layer_id)を、現在のビデオコーディング層の識別子(たとえば、vps_layer_id)として使用するステップ920とを含む。
【０２１８】
ビットストリームは、ワイヤレスネットワークまたはワイヤードネットワークによって取得され得る点に留意されたい。ビットストリームは、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)、あるいは赤外線、無線、マイクロ波、WIFI、Bluetooth、LTE、または5Gなどのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから伝送され得る。
【０２１９】
ビットストリームは、ネットワーク抽象化層(NAL)ユニットストリームまたはバイトストリームの形式のビットシーケンスであり得、1つまたは複数のコード化ビデオシーケンス(CVS)を形成するアクセスユニット(AU)のシーケンスの表現を形成する。
【０２２０】
特定の例では、ビットストリーム形式は、いずれもビットストリームと呼ばれる、ネットワーク抽象化層(NAL)ユニットストリームとバイトストリームとの間の関係を指定する。
【０２２１】
ビットストリームは、NALユニットストリーム形式またはバイトストリーム形式の2つの形式のいずれかになる。NALユニットストリーム形式は、概念的にはより「基本的な」タイプである。NALユニットストリーム形式は、NALユニットと呼ばれる構文構造のシーケンスを含む。このシーケンスは、デコード順に並べられている。NALユニットストリーム内のNALユニットのデコード順(およびコンテンツ)には制約が課される。
【０２２２】
バイトストリーム形式は、NALユニットをデコード順に並べることと、バイトストリームを形成するために、各NALユニットの前に、開始コードプレフィックスと0個以上のゼロ値のバイトをプレフィックスとして付けることとを行うことによって、NALユニットストリーム形式から構築することができる。NALユニットストリーム形式は、このバイトストリーム内の一意の開始コードプレフィックスパターンの位置を検索することによって、バイトストリーム形式から抽出することができる。
【０２２３】
図10は、エンコーディングデバイスによって実装されるビデオビットストリームをエンコードする方法を示しており、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオコーディング層はビットストリーム内の唯一の層で

あるかどうかを決定するステップ1010と、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合、現在のビデオコーディング層のNALユニットの識別子を、現在のビデオコーディング層の識別子として使用するステップ1020とを含む。

【０２２４】

上述の方法は、以下に説明するように、ビデオデコーディング装置またはビデオエンコーディング装置(ビットストリームを生成する)にそれぞれ組み込むことができる。

【０２２５】

図11に示されるように、一実施形態に従って本明細書で提供されるビデオデコーディング装置1100は、決定ユニット1110と予測ユニット1120とを含む。図12に示されるように、一実施形態に従って本明細書で提供されるビデオエンコーディング装置1200は、決定ユニット1210と予測ユニット1220とを含む。

【０２２６】

図11に示されるビデオデコーディング装置1100に含まれる決定ユニット1110は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するように構成される。図11に示されるビデオデコーディング装置1100に含まれる予測ユニット1120は、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合(たとえば、決定された場合のみ決定される)(たとえば、sps_video_parameter_set_idが0に等しい)、現在のビデオコーディング層のNALユニットの識別子(たとえば、nuh_layer_id)を、現在のビデオコーディング層の識別子(たとえば、vps_layer_id)として使用するように構成されている。

【０２２７】

図12に示されるビデオエンコーディング装置1200に含まれる決定ユニット1210は、現在のビデオコーディング層はビットストリーム内の唯一の層であるかどうかを決定するように構成されている。図12に示されるビデオエンコーディング装置1200に含まれる予測ユニット1220は、現在のビデオコーディング層はビットストリーム内の唯一の層であると決定した場合、現在のビデオコーディング層のNALユニットの識別子を現在のビデオコーディング層の識別子として使用するように構成されている。

【０２２８】

図11に示されるビデオデコーディング装置1100は、図1A、図1B、および図3に示されるデコーダ30、ならびに図14に示されるビデオデコーダ3206であってもよいし、またはそれによって構成されてもよい。さらに、デコーディング装置1100は、図4に示されるビデオコーディングデバイス400、図5に示される装置500、および図13に示される端末デバイス3106によって構成されてもよい。図12に示されるエンコーディング装置1200は、図1A、図1B、および図3に示されるエンコーダ20であってもよいし、またはそれによって構成されてもよい。さらに、エンコーディング装置1200は、図4に示されるビデオコーディングデバイス400、図5に示される装置500、および図13に示されるキャプチャデバイス3102によって構成されてもよい。

【０２２９】

以下、上記の実施形態において示されるエンコーディング方法ならびにデコーディング方法の応用例と、それらを用いたシステムについて説明する。

【０２３０】

図13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を示すブロック図である。このコンテンツ供給システム3100は、キャプチャデバイス3102、端末デバイス3106を含み、オプションでディスプレイ3126を含む。キャプチャデバイス3102は、通信リンク3104を介して端末デバイス3106と通信する。通信リンクは、上述の通信チャネル13を含み得る。通信リンク3104は、WIFI、イーサネット、ケーブル、ワイヤレス(3G/4G/5G)、USB、またはそれらの任意の種類の組合せなどを含むが、これらに限定されない。

【０２３１】

キャプチャデバイス3102は、データを生成し、上記の実施形態において示されるよう

なエンコーディング方法によってデータをエンコードし得る。あるいは、キャプチャデバイス3102は、データをストリーミングサーバ(図示せず)に配信し得、サーバはデータをエンコードし、エンコードされたデータを端末デバイス3106に伝送する。キャプチャデバイス3102は、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、車載デバイス、あるいはそれらの任意の組合せなどを含むが、これらに限定されない。たとえば、キャプチャデバイス3102は、上述のソースデバイス12を含み得る。データがビデオを含む場合、キャプチャデバイス3102に含まれるビデオエンコーダ20が実際にビデオエンコーディング処理を行い得る。データがオーディオ(すなわち、音声)を含む場合、キャプチャデバイス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行い得る。いくつかの実際的なシナリオでは、キャプチャデバイス3102は、エンコードされたビデオデータとオーディオデータを一緒に多重化することによって配信する。他の実用的なシナリオでは、たとえばビデオ会議システムにおいて、エンコードされたオーディオデータとエンコードされたビデオデータは多重化されない。キャプチャデバイス3102は、エンコードされたオーディオデータとエンコードされたビデオデータを別々に端末デバイス3106に配信する。

【０２３２】

コンテンツ供給システム3100では、端末デバイス3106がエンコードされたデータを受信して再生する。端末デバイス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、テレビ3114、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、携帯情報端末(PDA)3122、車載デバイス3124、またはそれらのいずれかの組合せ、あるいは上記のエンコードされたデータをデコードすることができるものなど、データの受信および回復機能を備えたデバイスである可能性がある。たとえば、端末デバイス3106は、上述の宛先デバイス14を含み得る。エンコードされたデータがビデオを含む場合、ビデオデコーディングを実行するために、端末デバイスに含まれるビデオデコーダ30が優先される。エンコードされたデータがオーディオを含む場合、オーディオデコーディング処理を行うために、端末デバイスに含まれるオーディオデコーダが優先される。

【０２３３】

たとえば、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、テレビ3114、携帯情報端末(PDA)3122、または車載デバイス3124など、ディスプレイを備えた端末デバイスの場合、端末デバイスはデコードされたデータをそのディスプレイに供給することができる。ディスプレイを備えていない端末デバイス、たとえばSTB3116、ビデオ会議システム3118、またはビデオ監視システム3120の場合、デコードされたデータを受信して表示するために外部ディスプレイ3126に接触する。

【０２３４】

このシステムの各デバイス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、上記実施の形態に示されるように、ピクチャエンコーディングデバイスまたはピクチャデコーディングデバイスを使用することができる。

【０２３５】

図14は、端末デバイス3106の一例の構造を示す図である。端末デバイス3106がキャプチャデバイス3102からストリームを受信した後、プロトコル進行ユニット3202はストリームの伝送プロトコルを解析する。プロトコルは、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ングプロトコル(RTMP)、またはそれらの任意の種類の組合せなどを含むが、これらに限定されない。

【０２３６】

プロトコル進行ユニット3202がストリームを処理した後、ストリームファイルが生成

される。ファイルは逆多重化ユニット3204に出力される。逆多重化ユニット3204は、多重化されたデータをエンコードされたオーディオデータとエンコードされたビデオデータに分離することができる。上述のように、たとえばいくつかの実用的なシナリオでは、たとえばビデオ会議システムにおいて、エンコードされたオーディオデータとエンコードされたビデオデータは多重化されない。この状況では、エンコードされたデータは、逆多重化ユニット3204を介さずに、ビデオデコーダ3206およびオーディオデコーダ3208に伝送される。

【０２３７】

逆多重化処理を介して、ビデオエレメンタリストリーム(ES)、オーディオES、およびオプションでサブタイトルが生成される。上記の実施形態において説明したビデオデコーダ30を含むビデオデコーダ3206は、ビデオフレームを生成するために、上記の実施形態において示したようにビデオESをデコーディング方法によってデコードし、このデータを同期ユニット3212に供給する。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をデコードし、このデータを同期ユニット3212に供給する。あるいは、ビデオフレームは、それを同期ユニット3212に供給する前にバッファ(図14には示されていない)に記憶し得る。同様に、オーディオフレームは、それを同期ユニット3212に供給する前に、バッファ(図14には示されていない)に記憶し得る。

【０２３８】

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を同期させ、ビデオ/オーディオをビデオ/オーディオディスプレイ3216に供給する。たとえば、同期ユニット3212は、ビデオおよびオーディオ情報の提示を同期させる。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提示に関するタイムスタンプと、データストリーム自体の配信に関するタイムスタンプを使用して、構文においてコード化し得る。

【０２３９】

サブタイトルデコーダ3210は、ストリームにサブタイトルが含まれている場合、サブタイトルをデコードし、ビデオフレームおよびオーディオフレームと同期させ、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に供給する。

【０２４０】

なお、本発明は上記のシステムに限定されるものではなく、上記の実施形態におけるピクチャエンコーディングデバイスまたはピクチャデコーディングデバイスは、他のシステム、たとえば自動車システムに組み込むことができる。

【０２４１】

数学演算子

本願で使用される算術演算子は、Cプログラミング言語において使用されるものと類似している。しかしながら、整数除算と算術シフト演算の結果はより正確に定義され、累乗および実数値除算などの追加演算が定義される。番号付けおよびカウントの規則は一般に0から始まり、たとえば、「第1」は0番目に相当し、「第2」は1番目に相当する、などである。

【０２４２】

算術演算子

次の算術演算子は、次のように定義されている。

【０２４３】

【表３】

+	加算
−	減算(2 引数演算子として)または否定(単項前置演算子として)
*	行列乗算を含む乗算
x^y	べき乗。x の y 乗を指定する。他の文脈では、そのような表記法は、累乗として解釈されることを意図していない上付き文字に使用される。
/	結果をゼロに向けて切り捨てる整数除算。たとえば、7/4 と-7/-4 は 1 に切り捨てられ、-7/4 と 7/-4 は-1 に切り捨てられる。
÷	切り捨てや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。
$\frac{x}{y}$	切り捨てや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ます。
$\sum_{i=x}^{y} f(i)$	x から y までのすべての整数値を取る i による f(i)の合計。
x % y	係数。x を y で割った剰余であり、x>=0 および y>0 の整数 x および y に対してのみ定義される。

【０２４４】

論理演算子

次の論理演算子は、次のように定義されている。

x && y xとyのブール論理「and」

x ¦ ¦ y xとyのブール論理「or」

! ブール論理「not」

x ? y : z xがTRUEであるか、0に等しくない場合、yの値に評価され、それ以外の場合は、zの値に評価される。

【０２４５】

関係演算子

次の関係演算子は、次のように定義されている。

〉 より大きい

〉= 以上

〈 未満

〈= 以下

= = に等しい

!= に等しくない

【０２４６】

関係演算子が、値「na」(該当なし)が割り当てられた構文要素または変数に適用される場合、値「na」は、構文要素または変数の個別の値として扱われる。値「na」は、他のどの値とも等しくないと見なされる。

【０２４７】

ビット演算子

次のビット演算子は、次のように定義されている。

& ビット単位の「および(and)」。整数引数を演算する場合、整数値の2の補数表現を演算する。他の引数よりもビット数が少ないバイナリ引数を演算する場合、短い方の引数

は、0に等しい有効ビットを追加することによって拡張される。
　¦　ビット単位の「または(or)」。整数引数を演算する場合、整数値の2の補数表現を演算する。他の引数よりもビット数が少ないバイナリ引数を演算する場合、短い方の引数は、0に等しい有効ビットを追加することによって拡張される。
　^　ビット単位の「排他的論理和(exclusive or)」。整数引数を演算する場合、整数値の2の補数表現を演算する。他の引数よりもビット数が少ないバイナリ引数を演算する場合、短い方の引数は、0に等しい有効ビットを追加することによって拡張される。
　x 〉〉 y　yの2進数によるxの2の補数整数表現の算術右シフト。この関数は、yの非負の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)にシフトされたビットは、シフト演算前のxのMSBに等しい値を有する。
　x 〈〈 y　yの2進数によるxの2の補数整数表現の算術左シフト。この関数は、yの非負の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。

【０２４８】

代入演算子

　次の代入演算子は、次のように定義されている。
　=　代入演算子
　+ +　インクリメント、すなわち、x++はx=x+1と同等であり、配列インデックスにおいて使用されると、インクリメント演算前の変数の値を評価する。
　- -　デクリメント、すなわち、x- -はx=x-1と同等であり、配列インデックスにおいて使用されると、デクリメント演算前の変数の値を評価する。
　+=　指定した量だけインクリメントし、すなわち、x+=3はx=x+3と等しく、x+=(-3)はx=x+(-3)と等しい。
　-=　指定した量だけデクリメントし、すなわち、x-=3はx=x-3と等しく、x-=(-3)はx=x-(-3)と等しい。

【０２４９】

範囲表記

　値の範囲を指定するために、次の表記法が使用される。
　x= y..z　xは、両端値を含むyからzまでの整数値を取り、x、y、zは整数であり、zはyより大きい値である。

【０２５０】

数学関数

　次の数学関数が定義されている。

【０２５１】

【数２】

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

【０２５２】

　Asin(x)　両端値を含む-1.0から1.0までの範囲の引数xを演算する三角関数の逆正弦関数であり、ラジアン単位で両端値を含む-π÷2からπ÷2までの範囲の出力値を有する。
　Atan(x)　引数xを演算する三角関数の逆正接関数であり、ラジアン単位で両端値を含む-π÷2からπ÷2までの範囲の出力値を有する。

【０２５３】

【数３】

$$\mathrm{Atan2}(y,x)=\begin{cases}\mathrm{Atan}\left(\frac{y}{x}\right) & ; \quad x>0\\ \mathrm{Atan}\left(\frac{y}{x}\right)+\pi & ; \quad x<0 \ \&\& \ y>=0\\ \mathrm{Atan}\left(\frac{y}{x}\right)-\pi & ; \quad x<0 \ \&\& \ y<0\\ +\frac{\pi}{2} & ; \quad x==0 \ \&\& \ y>=0\\ -\frac{\pi}{2} & ; \quad \text{そうではない場合}\end{cases}$$

【０２５４】
Ceil(x) x以上の最小の整数。
$\mathrm{Clip1_Y}(x)=\mathrm{Clip3}(0,(1 \ll \mathrm{BitDepth_Y})-1,x)$
$\mathrm{Clip1_C}(x)=\mathrm{Clip3}(0,(1 \ll \mathrm{BitDepth_C})-1,x)$
【０２５５】
【数４】

$$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \quad z<x\\ y & ; \quad z>y\\ z & ; \quad \text{そうではない場合}\end{cases}$$

【０２５６】
Cos(x) 引数xをラジアン単位で演算する三角関数の余弦関数。
Floor(x) x以下の最大の整数。
【０２５７】
【数５】

$$\mathrm{GetCurrMsb}(a,b,c,d)=\begin{cases}c+d & ; \quad b-a>=d/2\\ c-d & ; \quad a-b>d/2\\ c & ; \quad \text{そうではない場合}\end{cases}$$

【０２５８】
Ln(x) xの自然対数(eを底とする対数であり、eは自然対数の底定数2.718 281 828...)。
Log2(x) xの2を底とする対数。
Log10(x) xの10を底とする対数。
【０２５９】
【数６】

$$\mathrm{Min}(x, y) = \begin{cases} x & ; \ x <= y \\ y & ; \ x > y \end{cases}$$

$$\mathrm{Max}(x, y) = \begin{cases} x & ; \ x >= y \\ y & ; \ x < y \end{cases}$$

【０２６０】
Round(x)=Sign(x)*Floor(Abs(x)+0.5)
【０２６１】
【数７】

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

【０２６２】
Sin(x) 引数xをラジアン単位で演算する三角正弦関数
【０２６３】
【数８】

$$\mathrm{Sqrt}(x) = \sqrt{x}$$

【０２６４】
Swap(x, y)=(y, x)
Tan(x) 引数xをラジアン単位で演算する三角正接関数
【０２６５】
動作の優先順位
式の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、次の規則が適用される。
- 優先順位の高い演算は、優先順位の低い演算よりも先に評価される。
- 同じ優先順位の演算は、左から右に順番に評価される。

【０２６６】
次の表は、演算の優先順位を最高から最低まで示しており、表内の位置が高いほど、優先順位が高いことを示す。
【０２６７】
Cプログラミング言語においても使用される演算子の場合、本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ものと同じである。
【０２６８】

【表４】

Table: 最高(表の上部)から最低(表の下部)までの演算の優先順位

演算(オペランド 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"

【０２６９】

論理演算のテキスト記述

テキストにおいて、次の形式で数学的に記述される論理演算のステートメント:

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

は、次のように記述され得る。

```
... as follows / ... the following applies:
-  If condition 0, statement 0
-  Otherwise, if condition 1, statement 1
-  ...
-  Otherwise (informative remark on remaining condition), statement n
```

テキスト内の各「If ... Otherwise, if ... Otherwise, ...」ステートメントは、「... as follows」または「... the following applies」で始まり、直後に「If ...」が続く。「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」と末尾の「Otherwise, ...」を照合することによって識別することができる。

【０２７０】

テキストにおいて、次の形式で数学的に記述される論理演算のステートメント:

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

は、次のように記述され得る。

```
... 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
```

【０２７１】

テキストにおいて、次の形式で数学的に記述される論理演算のステートメント:

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

は、次のように記述され得る。

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

【０２７２】

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、ならびに本明細書で説明される他の実施形態はまた、ビデオコーディングのように先行または連続するピクチャとは無関係に、静止画処理またはコーディング、すなわち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点に留意されたい。一般に、インター予測ユニット244(エンコーダ)および344(デコーダ)のみは、ピクチャ処理コーディングが単一のピクチャ17に限定されている場合に利用できない場合がある。ビデオエンコーダ20およびビデオデコーダ30の他のすべての機能（ツールまたは技術とも呼ばれる）は、静止画処理、たとえば、残差計算204/304、変換206、量子化208、逆量子化210/310、(逆)変換212/312、パーティショニング262/362、イントラ予測254/354、および/またはループフィルタリング220、320、ならびにエントロピーコーディング270およびエントロピーデコーディング304のために等しく使用され得る。

【０２７３】

実施形態、たとえばエンコーダ20およびデコーダ30の実施形態、ならびに、たとえば、エンコーダ20およびデコーダ30に関して本明細書で説明される機能は、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合せにおいて実装することができる。ソフトウェアにおいて実装される場合、機能は、コンピュータ可読媒体に記憶さ

れてもよく、または、通信媒体を介して1つまたは複数の命令またはコードとして伝送され、ハードウェアベースの処理ユニットによって実行されてもよい。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体、またはたとえば通信プロトコルに従って、ある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容易にする任意の媒体を含む通信媒体を含み得る。このように、コンピュータ可読媒体は一般に、(1)非一時的である有形のコンピュータ可読記憶媒体、または(2)信号または搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明する技法を実装するための命令、コード、および/またはデータ構造を取り出すために、1つまたは複数のコンピュータあるいは1つまたは複数のプロセッサによってアクセスできる任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【０２７４】

限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMまたは他の光ディスクストレージ、磁気ディスクストレージ、または他の磁気記憶デバイス、フラッシュメモリ、あるいは所望のプログラムコードを命令またはデータ構造の形式で記憶するために使用することができ、コンピュータによってアクセスすることができる任意の他の媒体を含むことができる。さらに、すべての接続は、コンピュータ可読媒体と適切に呼ばれる。たとえば、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、マイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから指示が伝送される場合、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、マイクロ波などのワイヤレス技術は、媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的な媒体を含まないが、代わりに非一時的な有形の記憶媒体を対象としていることを理解されたい。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザディスク(disc)、光ディスク(disc)、デジタルバーサタイルディスク(disc)(DVD)、フロッピーディスク(disk)、およびBlu-ray(登録商標)ディスク(disc)を含み、ディスク(disk)は通常、データを磁気的に再生するが、ディスク(disc)はレーザを使用して光学的にデータを再生する。上記の組合せも、コンピュータ可読媒体の範囲内に含める必要がある。

【０２７５】

命令は、1つまたは複数のデジタルシグナルプロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、または他の同等の集積または個別論理回路などの1つまたは複数のプロセッサによって実行され得る。したがって、「プロセッサ」という用語は、本明細書で使用される場合、前述の構造のいずれか、または本明細書で説明する技法の実装に適した任意の他の構造を指し得る。さらに、いくつかの態様では、本明細書で説明する機能は、エンコーディングおよびデコーディング用に構成された専用のハードウェアおよび/またはソフトウェアモジュール内で提供されてもよく、または、組み合わせたコーデックに組み込まれてもよい。さらに、これらの技法は、1つまたは複数の回路または論理要素において完全に実装することができる。

【０２７６】

本開示の技法は、ワイヤレスハンドセット、集積回路(IC)、またはICのセット（たとえば、チップセット）を含む、多種多様なデバイスまたは装置において実装され得る。本開示では、開示された技法を実行するように構成されたデバイスの機能的側面を強調するために、様々なコンポーネント、モジュール、またはユニットが説明されているが、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しない。むしろ、上記のように、様々なユニットがコーデックハードウェアユニットに組み合わされてもよく、適切なソフトウェアおよび/またはファームウェアとともに、上記の1つまたは複数のプロセッサを含む相互運用可能なハードウェアユニットの集合によって提供されてもよい。

【０２７７】

特に、以下のさらなる10の態様も本明細書で提供され、態様1～10として列挙する。

【０２７８】

1．デコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをデコードする方法の態様であって、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオコーディング層の識別子(たとえば、vps_layer_id[0])を指定する構文要素がビットストリームに存在しない(たとえば、VPSが存在しない)場合、現在のビデオコーディング層の識別子としてビットストリーム内のシーケンスパラメータセットの識別子を使用するステップと、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するステップと、を含む、態様。

【０２７９】

2．態様1の方法の一態様であって、現在のビデオコーディング層の識別子としてビットストリーム内のシーケンスパラメータセットの識別子を使用するステップは、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築することであり、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測するステップは、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測するステップを含む、態様。

【０２８０】

3．デコーディングデバイスによって実装されるビデオまたはピクチャのビットストリームをデコードする方法の態様であって、ビットストリームは現在のビデオコーディング層を表すデータを含み、本方法は、現在のビデオ コーディング層の識別子(たとえば、vps_layer_id[0])を指定する構文要素がビットストリームに存在しない(たとえば、VPSが存在しない)場合、現在のビデオコーディング層のNALユニットの識別子を、現在のビデオコーディング層の識別子として使用するステップと、現在のビデオコーディング層の識別子を使用して、現在のビデオコーディング層の現在のスライスを予測するステップと、を含む、態様。

【０２８１】

4．態様3の方法の一態様であって、現在のビデオコーディング層の識別子として現在のビデオコーディング層のNALユニットの識別子を使用するステップは、現在のビデオコーディング層の現在のスライスの1つまたは複数の参照ピクチャリストを構築することであり、現在のビデオコーディング層の識別子を使用して現在のビデオコーディング層の現在のスライスを予測するステップは、1つまたは複数の参照ピクチャリストを使用して現在のスライスを予測するステップを含む、態様。

【０２８２】

5．態様3または4の方法の一態様であって、現在のビデオコーディング層のNALユニットは、ビットストリームに存在する現在のビデオコーディング層の第1の非VCL NALユニットまたは第1のVCL NALユニットである、態様。

【０２８３】

6．態様3または4の方法の一態様であって、現在のビデオコーディング層のNALユニットは、ビットストリームに存在する現在のビデオコーディング層のいずれか1つのNALユニットである、態様。

【０２８４】

7．態様1から6のいずれか1つによる方法を実行するための処理回路を含む、デコーダ(30)の態様。

【０２８５】

8．コンピュータまたはプロセッサ上で実行されると、前述の態様1から6のうちのいずれか1つによる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の態様。

【０２８６】

9．1つまたは複数のプロセッサと、プロセッサに結合され、プロセッサによる実行のた

めのプログラミングを記憶する非一時的コンピュータ可読記憶媒体とを含み、プログラミングは、プロセッサによって実行されると、前述の態様1から6のいずれか1つによる方法を行うようにデコーダを構成する、デコーダの一態様。

【０２８７】

10．コンピュータデバイスによって実行されると、コンピュータデバイスに前述の態様1から6のうちのいずれか1つの方法を実行させるプログラムコードを担持する非一時的コンピュータ可読媒体の態様。

【符号の説明】

【０２８８】

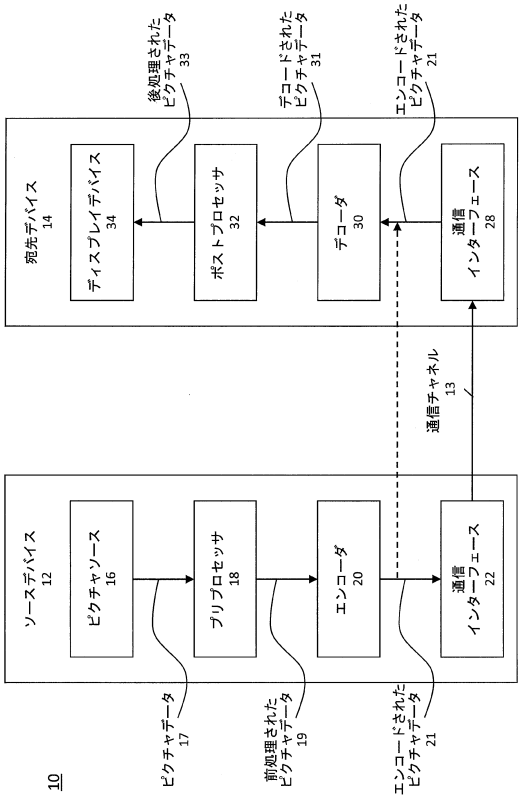
10 コーディングシステム
10 ビデオコーディングシステム
10 ショートコーディングシステム
12 ソースデバイス
13 通信チャネル
14 宛先デバイス
16 ピクチャソース
17 ピクチャまたはピクチャデータ
17 生ピクチャまたは生ピクチャデータ
18 プリプロセッサ(または、プリプロセッシングユニット)
18 ピクチャプリプロセッサ
19 前処理されたピクチャ
19 前処理されたピクチャデータ
20 ビデオエンコーダ
20 ショートエンコーダ
21 エンコードされたピクチャデータ
21 エンコードされたビットストリーム
22 通信インターフェースまたは通信ユニット
28 通信インターフェースまたは通信ユニット
30 デコーダ
30 ビデオデコーダ
30 ショートデコーダ
31 デコードされたピクチャデータ
31 デコードされたピクチャ
32 ポストプロセッサ
32 ポストプロセッシングユニット
33 後処理されたピクチャデータ
33 後処理されたピクチャ
34 ディスプレイデバイス
201 入力
201 入力インターフェース
203 ピクチャブロック
204 残差計算ユニット
205 残差ブロック
205 残差
206 変換処理ユニット
207 変換係数
208 量子化ユニット
209 量子化された係数
209 量子化された変換係数
209 量子化された残差係数

210 逆量子化ユニット
211 非量子化された係数
212 逆変換処理ユニット
213 再構築された残差ブロック
213 変換ブロック
214 再構築ユニット
214 加算器(adder)またはアナログ加算器(summer)
215 再構築されたブロック
216 バッファ
220 ループフィルタユニット
220 ループフィルタ
221 フィルタリングされたブロック
221 フィルタリングされた再構築ブロック
230 デコードされたピクチャバッファ(DPB)
231 デコードされたピクチャ
244 インター予測ユニット
254 イントラ予測ユニット
260 モード選択ユニット
262 パーティショニングユニット
265 予測ブロック
265 イントラ予測ブロック
265 予測子
266 構文要素
270 エントロピーエンコーディングユニット
272 出力
272 出力インターフェース
304 エントロピーデコーディングユニット
309 量子化された係数
310 逆量子化ユニット
311 変換係数
311 非量子化係数
312 逆変換処理ユニット
313 変換ブロック
313 再構築された残差ブロック
314 再構築ユニット
314 アナログ加算器
315 再構築されたブロック
320 ループフィルタ
320 ループフィルタユニット
321 フィルタリングされたブロック
321 デコードされたビデオブロック
330 デコードされたピクチャバッファ(DBP)
331 デコードされたピクチャ
344 インター予測ユニット
354 イントラ予測ユニット
360 モード適用ユニット
365 予測ブロック
400 ビデオコーディングデバイス
410 入力ポート
420 受信機ユニット(Rx)

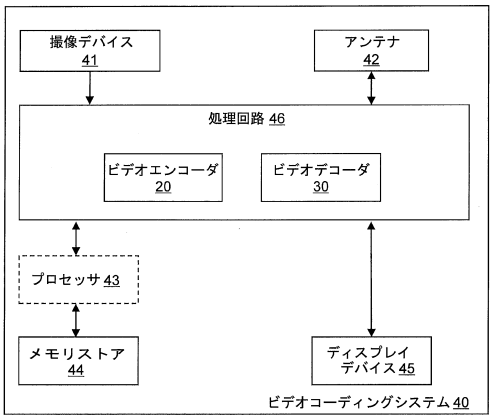
430 中央処理装置(CPU)
430 プロセッサ
440 トランスミッタユニット(Tx)
450 出力ポート
460 メモリ
470 コーディングモジュール
500 装置
502 プロセッサ
504 メモリ
506 コードおよびデータ
508 オペレーティングシステム
510 アプリケーションプログラム
512 バス
514 セカンダリストレージ
518 ディスプレイ
1100 ビデオデコーディング装置
1110 決定ユニット
1120 予測ユニット
1200 ビデオエンコーディング装置
1210 決定ユニット
1220 予測ユニット
3100 コンテンツ供給システム
3102 キャプチャデバイス
3104 通信リンク
3106 端末デバイス
3108 スマートフォンまたはパッド
3110 コンピュータまたはラップトップ
3112 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)
3114 テレビ
3116 セットトップボックス(STB)
3118 ビデオ会議システム
3120 ビデオ監視システム
3122 携帯情報端末(PDA)
3124 車載デバイス
3126 ディスプレイ
3126 外部ディスプレイ
3202 プロトコル進行ユニット
3204 逆多重化ユニット
3206 ビデオデコーダ
3208 オーディオデコーダ
3210 サブタイトルデコーダ
3212 同期ユニット
3214 ビデオ/オーディオディスプレイ
3216 ビデオ/オーディオ/サブタイトルディスプレイ

【図面】

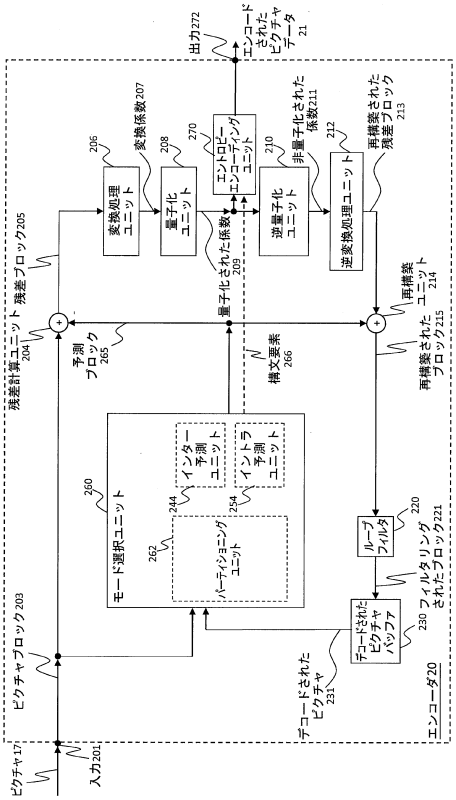
【図１Ａ】



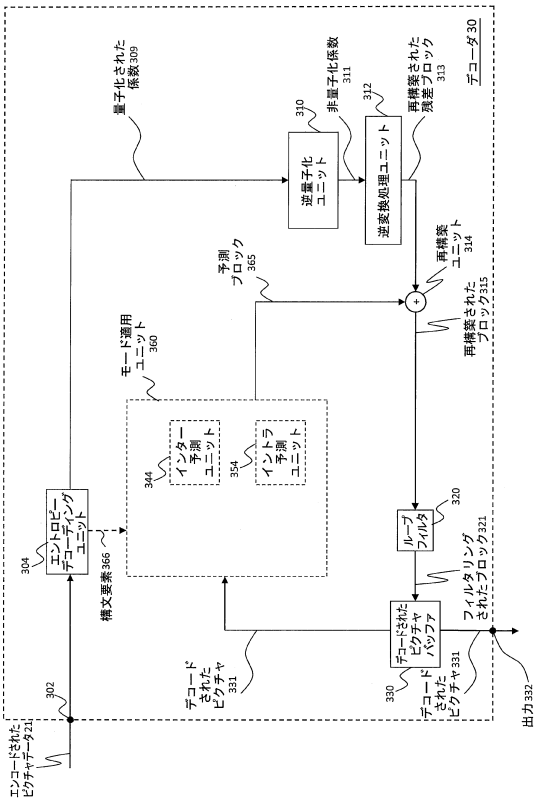
【図１Ｂ】



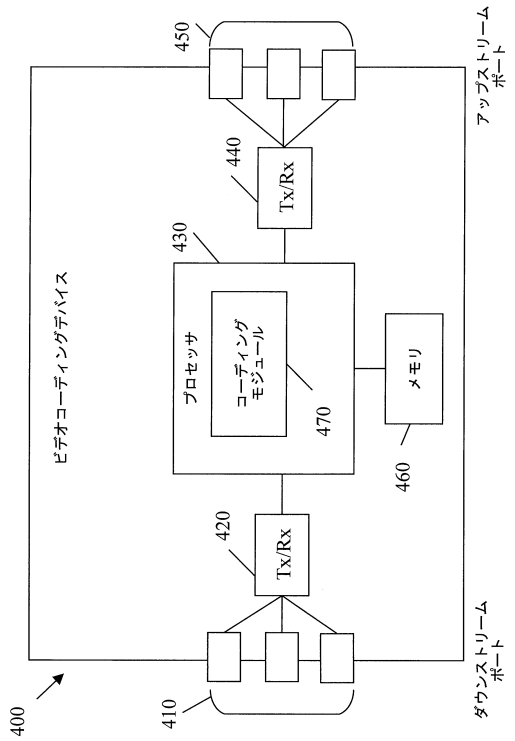
【図２】



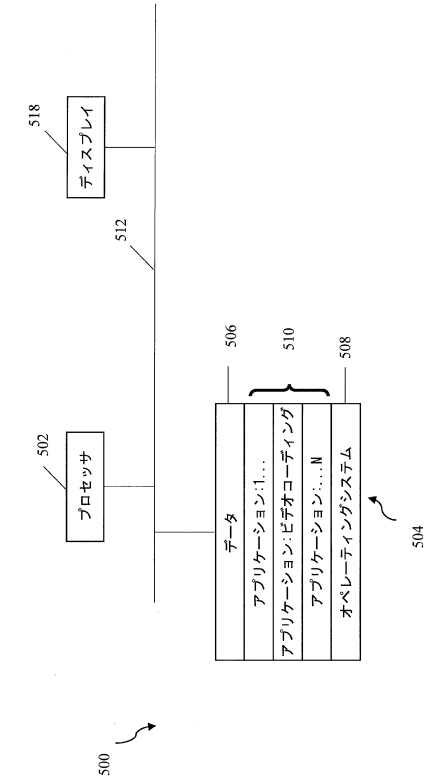
【図３】



【図４】



【図５】



【図６】

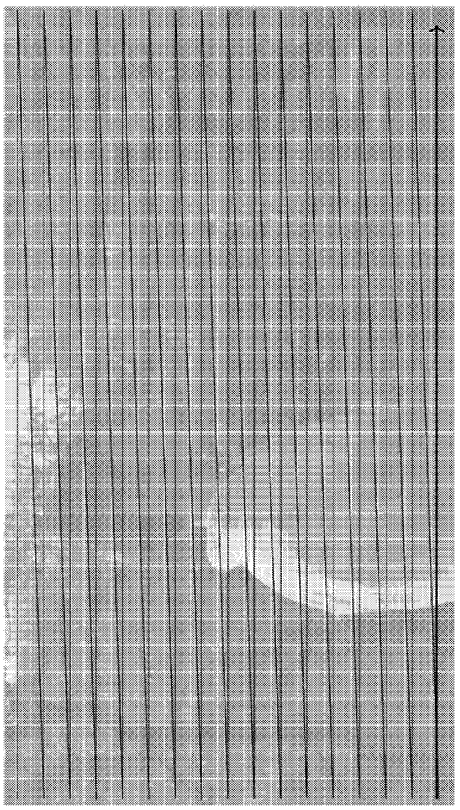
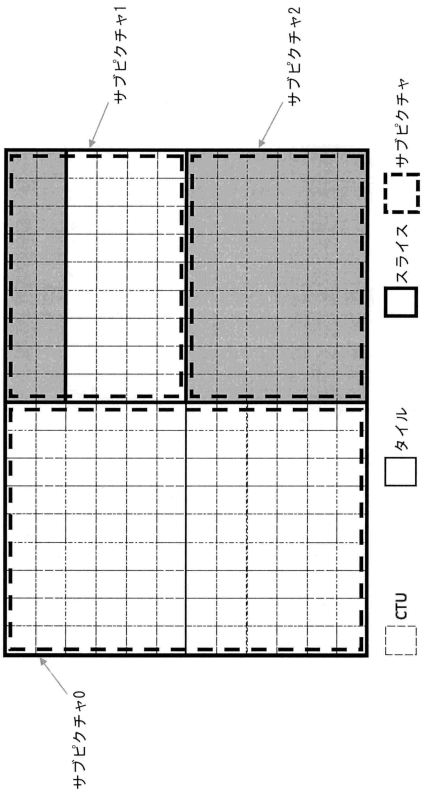
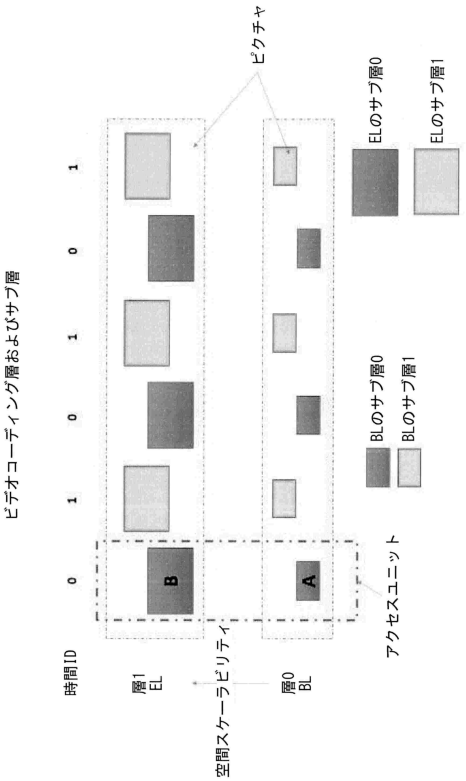


FIG. 6

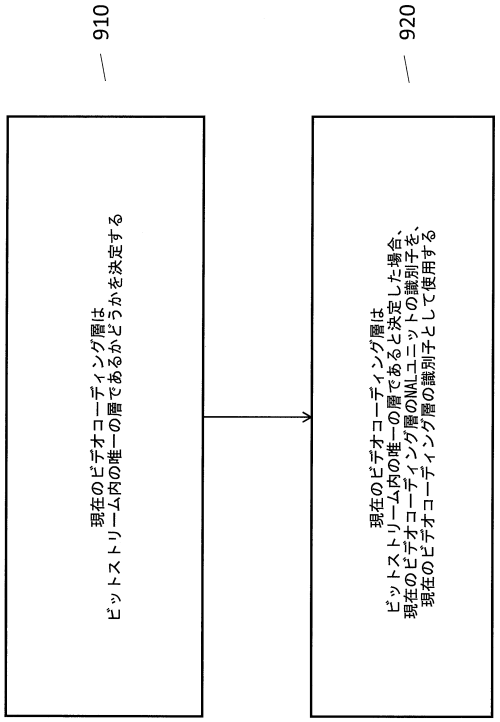
【図７】



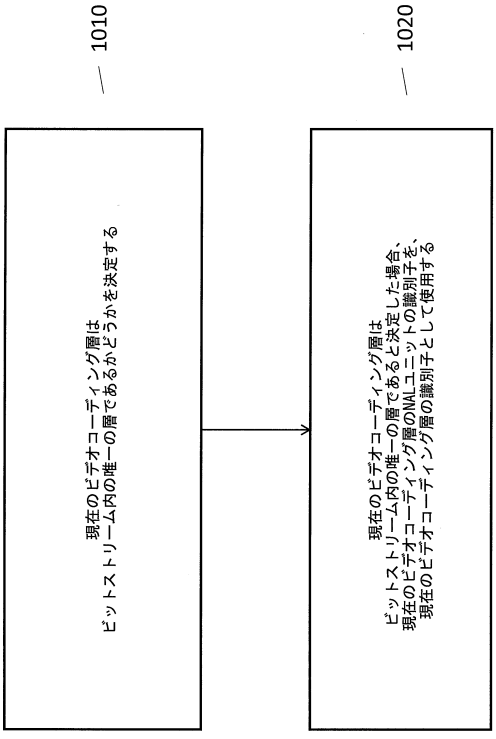
【図８】


ピクチャ
ELのサブ層0
ELのサブ層1
BLのサブ層0
BLのサブ層1
アクセスユニット
ビデオコーディング層およびサブ層
時間ID
0
1
0
1
0
1
層1
EL
層0
BL
空間スケーラビリティ
B
A

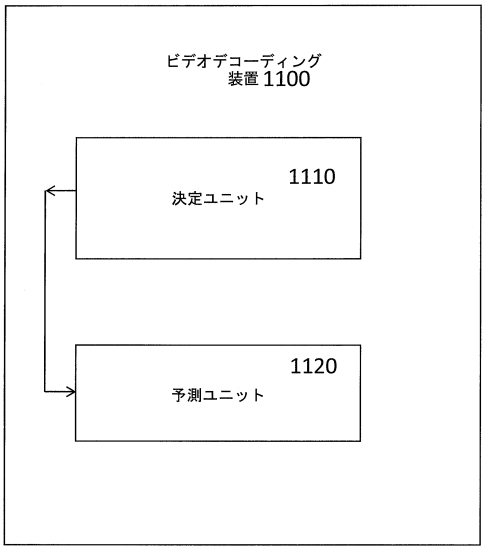
【図９】


910
920
現在のビデオコーディング層は
ビットストリーム内の唯一の層であるかどうかを決定する
現在のビデオコーディング層は
ビットストリーム内の唯一の層であると決定した場合、
現在のビデオコーディング層のNALユニットの識別子を、
現在のビデオコーディング層の識別子として使用する

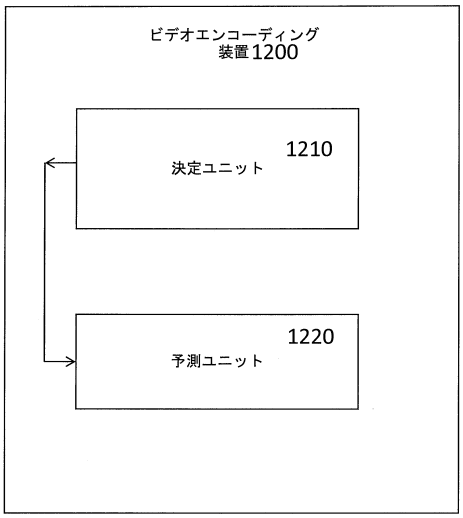
【図１０】


1010
1020
現在のビデオコーディング層は
ビットストリーム内の唯一の層であるかどうかを決定する
現在のビデオコーディング層は
ビットストリーム内の唯一の層であると決定した場合、
現在のビデオコーディング層のNALユニットの識別子を、
現在のビデオコーディング層の識別子として使用する

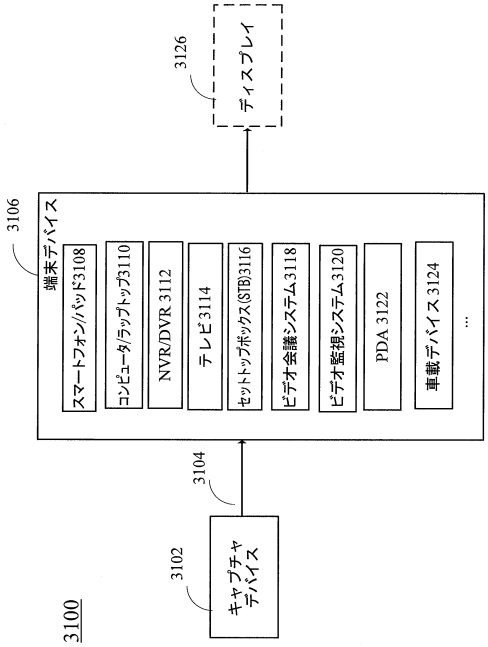
【図１１】


ビデオデコーディング
装置 1100
1110
決定ユニット
1120
予測ユニット

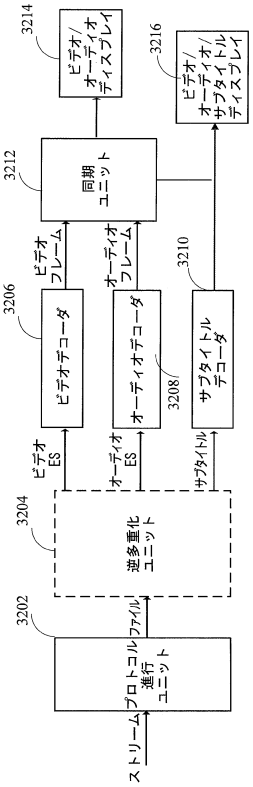
【図１２】


ビデオエンコーディング
装置1200
1210
決定ユニット
1220
予測ユニット

【図１３】


3126
ディスプレイ
3106
端末デバイス
スマートフォン/パッド3108
コンピュータ/ラップトップ3110
NVR/DVR 3112
テレビ3114
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
PDA 3122
車載デバイス3124
...
3104
3102
キャプチャデバイス
3100

【図１４】


3214
ビデオ/オーディオディスプレイ
3216
ビデオ/オーディオ/サブタイトルディスプレイ
3212
同期ユニット
ビデオフレーム
オーディオフレーム
3210
3206
ビデオデコーダ
オーディオデコーダ
3208
サブタイトルデコーダ
ビデオES
オーディオES
サブタイトル
3204
逆多重化ユニット
3202
プロトコル進行ユニット
ファイル
ストリーム

フロントページの続き

トリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(72)発明者　セミフ・エセンリク
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(72)発明者　エレナ・アレクサンドロブナ・アルシナ
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(72)発明者　アナンド・メヘル・コトラ
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング
(72)発明者　ハン・ガオ
中華人民共和国・５１８１２９・グァンドン・シェンツェン・ロンガン・ディストリクト・バンティアン・（番地なし）・ホアウェイ・アドミニストレーション・ビルディング

審査官　坂東 大五郎

(56)参考文献　Benjamin Bross et al.，Versatile Video Coding (Draft 8)，Joint Video Experts Team (JVET)，2020年03月12日，pp.37,99,175-178，[JVET-Q2001-vE] (version 15)
Sachin Deshpande，On Video Parameter Set and Highest Temporal Sub-layer，Joint Video Experts Team (JVET)，2019年09月24日，[JVET-P0185-v1] (version 1)

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｎ　１９／００－１９／９８