

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7675231号
(P7675231)

(45)発行日 令和7年5月12日(2025.5.12)

(24)登録日 令和7年4月30日(2025.4.30)

(51)国際特許分類 F I
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70
H 0 4 N 19/30 (2014.01) H 0 4 N 19/30

請求項の数 11 外国語出願 (全84頁)

(21)出願番号	特願2024-8764(P2024-8764)	(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・グアン 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
(22)出願日	令和6年1月24日(2024.1.24)	(74)代理人	100110364 弁理士 実広 信哉
(62)分割の表示	特願2022-521009(P2022-521009))の分割	(74)代理人	100133569 弁理士 野村 進
原出願日	令和2年10月6日(2020.10.6)	(72)発明者	イエ・クイ・ワン アメリカ合衆国・カリフォルニア・92 130・サン・ディエゴ・サンローズ・ 最終頁に続く
(65)公開番号	特開2024-36399(P2024-36399A)		
(43)公開日	令和6年3月15日(2024.3.15)		
審査請求日	令和6年2月5日(2024.2.5)		
(31)優先権主張番号	62/911,808		
(32)優先日	令和1年10月7日(2019.10.7)		
(33)優先権主張国・地域又は機関	米国(US)		

(54)【発明の名称】 サブビットストリーム抽出におけるSPSエラー回避

(57)【特許請求の範囲】

【請求項1】

エンコーダによって実装される方法であって、
前記エンコーダによって、ビットストリーム内のレイヤに対するコーディング済みレイヤビデオシーケンス(CLVIS)をエンコードするステップと、
前記エンコーダによって、前記ビットストリーム内に、前記CLVISによって参照されるシーケンスパラメータセット(SPS)をエンコードするステップであって、前記SPSは、前記レイヤがレイヤ間予測を使用しないとき前記CLVISのnuh_layer_id値に等しいネットワーク抽象化レイヤ(NAL)ユニットヘッダレイヤ識別子(nuh_layer_id)値を有するように制約され、前記SPSは、前記SPSによって参照されるビデオパラメータセット(VPS)に対する識別子(ID)の値を指定するSPS VPS識別子(sps_video_parameter_set_id)を含み、前記sps_video_parameter_set_idが0に等しいときに、nuh_layer_idに対応する一般レイヤインデックス(GeneralLayerIdx[nuh_layer_id])は0に設定される、ステップと、
前記エンコーダによって、デコーダに向けた通信のために前記ビットストリームを記憶するステップとを含む方法。

【請求項2】

GeneralLayerIdx[nuh_layer_id]は、現在のレイヤインデックスに等しい、請求項1に記載の方法。

【請求項3】

前記SPSによって参照される前記VPSが、対応するレイヤがレイヤ間予測を使用するか

どうかを指定するVPS独立レイヤフラグ(vps_independent_layer_flag)を有する、請求項1または2に記載の方法。

【請求項4】

前記レイヤは、前記vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき、レイヤ間予測を使用しない、請求項3に記載の方法。

【請求項5】

sps_video_parameter_set_idが0に等しいとき、前記vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]は1に等しいと推論される、請求項4に記載の方法。

【請求項6】

前記SPSは、前記sps_video_parameter_set_idが0に等しいとき、VPSを参照しない、請求項1に記載の方法。

10

【請求項7】

前記CLVSは、同じnuh_layer_id値を有するコーディング済みピクチャのシーケンスである、請求項1に記載の方法。

【請求項8】

ビットストリーム内のレイヤに対するコーディング済みレイヤビデオシーケンス(CLVS)をエンコードし、

前記ビットストリーム内に、前記CLVSによって参照されるシーケンスパラメータセット(SPS)をエンコードし、前記SPSは、前記レイヤがレイヤ間予測を使用しないとき前記CLVSのnuh_layer_id値に等しいネットワーク抽象化レイヤ(NAL)ユニットヘッダレイヤ識別子(nuh_layer_id)値を有するように制約され、前記SPSは、前記SPSによって参照されるビデオパラメータセット(VPS)に対する識別子(ID)の値を指定するSPS VPS識別子(sps_video_parameter_set_id)を含み、前記sps_video_parameter_set_idが0に等しいときに、nuh_layer_idに対応する一般レイヤインデックス(GeneralLayerIdx[nuh_layer_id])は0に設定される、エンコード手段と、

20

デコーダに向けた通信のために前記ビットストリームを記憶するための記憶手段とを備えるエンコーダ。

【請求項9】

請求項1から7のいずれか一項に記載の方法を実行するようにさらに構成される請求項8に記載のエンコーダ。

30

【請求項10】

ビットストリームを記憶するためのデバイスであって、前記ビットストリームを受信するよう構成された少なくとも1つの受信器と、前記ビットストリームを記憶するよう構成された少なくとも1つのメモリと、を含み、

前記ビットストリームが、

レイヤに対するコーディング済みレイヤビデオシーケンス(CLVS)と、前記CLVSによって参照されるシーケンスパラメータセット(SPS)とを含み、前記SPSは、前記レイヤがレイヤ間予測を使用しないとき前記CLVSのnuh_layer_id値に等しいネットワーク抽象化レイヤ(NAL)ユニットヘッダレイヤ識別子(nuh_layer_id)値を有し、前記SPSは、前記SPSによって参照されるビデオパラメータセット(VPS)に対する識別子(ID)の値を指定するSPS VPS識別子(sps_video_parameter_set_id)を含み、

40

前記sps_video_parameter_set_idが0に等しいときに、nuh_layer_idに対応する一般レイヤインデックス(GeneralLayerIdx[nuh_layer_id])が0に等しい、デバイス。

【請求項11】

ビットストリームを記憶するための方法であって、少なくとも1つの受信器によって前記ビットストリームを受信するステップと、少なくとも1つのメモリによって前記ビットストリームを記憶するステップと、を含み、前記ビットストリームが、

レイヤに対するコーディング済みレイヤビデオシーケンス(CLVS)と、前記CLVSによって参照されるシーケンスパラメータセット(SPS)とを含み、前記SPSは、前記レイヤがレイヤ

50

間予測を使用しないとき前記CLVSのnuh_layer_id値に等しいネットワーク抽象化レイヤ(NAL)ユニットヘッダレイヤ識別子(nuh_layer_id)値を有し、前記SPSは、前記SPSによって参照されるビデオパラメータセット(VPS)に対する識別子(ID)の値を指定するSPS VPS識別子(sps_video_parameter_set_id)を含み、前記sps_video_parameter_set_idが0に等しいときに、nuh_layer_idに対応する一般レイヤインデックス(GeneralLayerIdx[nuh_layer_id])が0に等しい、方法。

【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、参照により本明細書に組み込まれるYe-Kui Wangによって2019年10月7日に出願された「Scalability in Video Coding」なる名称の米国特許仮出願第62/911,808号に基づく優先権を主張する。

【0002】

本開示は、概して、ビデオコーディングに関係し、具体的には、サブビットストリーム抽出がマルチレイヤビットストリーム上で実行される場合にエラーを防止するためのメカニズムに関する。

【背景技術】

【0003】

比較的短いビデオであっても描画するために必要なビデオデータの量は実質的に大量になり得、その結果、データがストリーミングされるか、または帯域幅容量が限られている通信ネットワークを介した他の何らかの方法で伝達されるときに面倒なことが生じ得る。したがって、今日の電気通信ネットワークでは、ビデオデータは、伝達される前に圧縮されるのが一般的である。ビデオのサイズも、メモリリソースが限られていることもあり得るのでビデオがストレージデバイスに記憶されるとき、問題になる可能性がある。ビデオ圧縮デバイスは、伝送元でソフトウェアおよび/またはハードウェアを使用して伝送または記憶の前にビデオデータをコーディングし、それによってデジタルビデオイメージを表現するために必要なデータ量を削減することが多い。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伸張デバイスによって伝送先に受信される。ネットワークリソースに限度があり、求められる映像品質が高まる一方であることから、画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改善する圧縮および伸張技術の改善が望ましい。

【発明の概要】

【課題を解決するための手段】

【0004】

一実施形態において、本開示は、デコーダによって実装される方法を含み、この方法はデコーダによって、レイヤに対するコーディング済みレイヤビデオシーケンス(CLVS)と、CLVSによって参照されるシーケンスパラメータセット(SPS)とを含むビットストリームを受信することであって、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer

er_id値に等しいネットワーク抽象化レイヤ(NAL)ユニットヘッダレイヤ識別子(nuh_layer_id)値を有する、受信することと、デコーダによって、SPSに基づきCLVSからのコーディング済みピクチャをデコードして、デコード済みピクチャを生成することを含む。

【0005】

いくつかのビデオコーディングシステムは、ビデオシーケンスをピクチャのレイヤにコーディングする。異なるレイヤ内のピクチャは、異なる特性を有する。したがって、エンコーダは、デコーダ側の制約条件に応じて異なるレイヤをデコーダに伝送することができる。この機能を実行するために、エンコーダは、すべてのレイヤを単一のビットストリームにエンコードすることができる。要求に応じて、エンコーダは、ビットストリームから無関係な情報を除去するためにサブビットストリーム抽出プロセスを実行することができ

10

20

30

40

50

る。この結果は、デコーダによって要求されるレイヤ内のデータのみを含む抽出済みビットストリームである。レイヤがどのように関係付けられているかの記述は、ビデオパラメータセット(VPS)の中に含まれ得る。サイマルキャストレイヤは、他のレイヤを参照することなしに表示するように設定されたレイヤである。サイマルキャストレイヤがデコーダに伝送されるとき、サブビットストリーム抽出プロセスは、レイヤ関係がサイマルキャストレイヤをデコードするのに必要ないのでVPSを除去するものとしてよい。残念ながら、他のパラメータセット内のいくつかの変数は、VPSを参照し得る。それゆえに、サイマルキャストレイヤが伝送されるときVPSを除去することで、コーディング効率を高め得るが、その結果、エラーも引き起こし得る。さらに、SPSを正しく識別できなかった場合、結果として、SPSをVPSと一緒に誤って除去する可能性がある。これは、SPSが欠落している場合にデコーダにおいてレイヤが正しくデコードされ得ないので問題になることがある。本発明の例は、VPSがサブビットストリーム抽出中に除去されるときのエラーを回避するためのメカニズムを含む。具体的には、サブビットストリーム抽出プロセスは、nuh_layer_idsに基づきNALユニットを除去する。SPSは、CLVSがレイヤ間予測を使用しないレイヤ

10

に含まれるとき、SPSを参照するCLVSのnuh_layer_idに等しいnuh_layer_idを有するよう

に制約される。レイヤ間予測を使用しないレイヤは、サイマルキャストレイヤである。それゆえに、SPSは、VPSが除去されたときのレイヤと同じnuh_layer_idを有する。この方法

20

で、SPSは、サブビットストリーム抽出プロセスによって誤って除去され

ることがない。レイヤ間予測の使用は、VPS独立レイヤフラグ(vps_independent_layer_fl

ag)によってシグナリングされる。しかしながら、vps_independent_layer_flagは、VPSが

除去されると除去される。したがって、SPSがVPSを参照しないとき、vps_independent_la

yer_flag[GeneralLayerIdx[nuh_layer_id]]として表される、現在のレイヤに対するvp

s_independent_layer_flagは1であると推論される。SPSは、SPS VPS識別子(sps_video_param

30

eter_set_id)が0に設定されたとき、VPSを参照しない。さらに、vps_independent_layer_

flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき(たとえば、VPSが存在しないとき

)、デコーダおよび/または仮想参照デコーダ(HRD)は、現在のレイヤ/CLVSがレイヤ間予測

を使用しないことを推論することができる。この一連の推論を採用することによって、SP

Sは、VPSおよび対応するパラメータがビットストリームから除去されたときであってもサブ

40

ビットストリーム抽出プロセスによる抽出を回避するために適切なnuh_layer_idを含むように制約され得る。その結果、エンコーダおよびデコーダの機能が向上する。さらに、コーディング効率は、サイマルキャストレイヤのみを含むビットストリームから不要なVPSを首尾よく除去することによって高められ、これにより、エンコーダとデコーダの両方においてプロセッサ、メモリ、および/またはネットワークシグナリングリソースの使用量を低減する。

【0006】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、GeneralLayerIdx[nuh_layer_id]が現在のレイヤインデックスに等しい、ことを規定する。

【0007】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、vps_independe

50

n

t_layer_flagが対応するレイヤがレイヤ間予測を使用するかどうかを指定する、ことを規定する。

【0008】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、vps_independent

t_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき、レイヤがレイヤ間予測

を使用しない、ことを規定する。

【0009】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、SPSが、SPSによって参照されるVPSに対する識別子(ID)の値を指定するsps_video_parameter_set_idを含

み、sps_video_parameter_set_idが0に等しいとき、vps_independent_layer_flag[General

LayerIdx[nuh_layer_id]]は1に等しいと推論される、ことを規定する。

【0010】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、sps_video_parameter_set_idが0に等しいとき、SPSはVPSを参照しない、ことを規定する。

【0011】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、CLVSが同じnuh_layer_id値を有するコーディング済みピクチャのシーケンスである、ことを規定する。

【0012】

一実施形態において、本開示は、エンコーダによって実装される方法を含み、この方法はエンコーダによって、ビットストリーム内のレイヤに対するCLVSをエンコードすることと、エンコーダによって、ビットストリーム内に、CLVSによって参照されるSPSをエンコードすることとあって、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer

_id値に等しいnuh_layer_id値を有するよう制約される、エンコードすることと、エンコーダによって、デコーダに向けた通信のためにビットストリームを記憶することを含む。

【0013】

いくつかのビデオコーディングシステムは、ビデオシーケンスをピクチャのレイヤにコーディングする。異なるレイヤ内のピクチャは、異なる特性を有する。したがって、エンコーダは、デコーダ側の制約条件に応じて異なるレイヤをデコーダに伝送することができる。この機能を実行するために、エンコーダは、すべてのレイヤを単一のビットストリームにエンコードすることができる。要求に応じて、エンコーダは、ビットストリームから無関係な情報を除去するためにサブビットストリーム抽出プロセスを実行することができる。この結果は、デコーダによって要求されるレイヤ内のデータのみを含む抽出済みビットストリームである。レイヤがどのように関係付けられているかの記述は、ビデオパラメータセット(VPS)の中に含まれ得る。サイマルキャストレイヤは、他のレイヤを参照することなしに表示するように設定されたレイヤである。サイマルキャストレイヤがデコーダに伝送されるとき、サブビットストリーム抽出プロセスは、レイヤ関係がサイマルキャストレイヤをデコードするのに必要ないのでVPSを除去するものとしてよい。残念ながら、他のパラメータセット内のいくつかの変数は、VPSを参照し得る。それゆえに、サイマルキャストレイヤが伝送されるときVPSを除去することで、コーディング効率を高め得るが、その結果、エラーも引き起こし得る。さらに、SPSを正しく識別できなかった場合、結果として、SPSをVPSと一緒に誤って除去する可能性がある。これは、SPSが欠落している場合にデコーダにおいてレイヤが正しくデコードされ得ないので問題になることがある。

10

20

30

40

50

本発明の例は、VPSがサブビットストリーム抽出中に除去されるときのエラーを回避するためのメカニズムを含む。具体的には、サブビットストリーム抽出プロセスは、nuh_layer_idsに基づきNALユニットを除去する。SPSは、CLVSがレイヤ間予測を使用しないレイヤ

に含まれるとき、SPSを参照するCLVSのnuh_layer_idに等しいnuh_layer_idを有するよう

に制約される。レイヤ間予測を使用しないレイヤは、サイマルキャストレイヤである。それゆえに、SPSは、VPSが除去されたときのレイヤと同じnuh_layer_idを有する。この方法

で、SPSは、サブビットストリーム抽出プロセスによって誤って除去され

10

ることがない。レイヤ間予測の使用は、vps_independent_layer_flagによってシグナリ

ン

グされる。しかしながら、vps_independent_layer_flagは、VPSが除去されると除去され

る。したがって、SPSがVPSを参照しないとき、vps_independent_layer_flag[General Layer

rIdx[nuh_layer_id]]として表される、現在のレイヤに対するvps_independent_layer_

20

flagは1であると推論される。SPSは、sps_video_parameter_set_idが0に設定されたとき、VP

Sを参照しない。さらに、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]

が1に等しいとき(たとえば、VPSが存在しないとき)、デコーダおよび/またはHRDは、現在の

レイヤ/CLVSがレイヤ間予測を使用しないことを推論することができる。この一連の推

論を採用することによって、SPSは、VPSおよび対応するパラメータがビットストリームから

除去されたときであってもサブビットストリーム抽出プロセスによる抽出を回避するために

30

【0014】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、GeneralLayerIdx[nuh_layer_id]が現在のレイヤインデックスに等しい、ことを規定する。

【0015】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、vps_independent

layer_flagが対応するレイヤがレイヤ間予測を使用するかどうかを指定する、ことを規定する。

40

【0016】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、vps_independent

layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき、レイヤがレイヤ間

予測

【0017】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、SPSが、SPSによって参照されるVPSに対するIDの値を指定するsps_video_parameter_set_idを含み、s

ps_v

50

ideo_parameter_set_idが0に等しいとき、vps_independent_layer_flag[GeneralLayerIdx

[nuh_layer_id]]は1に等しいと推論される、ことを規定する。

【0018】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、sps_video_parameter_set_idが0に等しいとき、SPSはVPSを参照しない、ことを規定する。

【0019】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、CLVSが同じnuh_layer_id値を有するコーディング済みピクチャのシーケンスである、ことを規定する。

【0020】

一実施形態において、本開示は、ビデオコーディングデバイスを含み、これはプロセッサと、プロセッサに結合された受信器と、プロセッサに結合されたメモリと、プロセッサに結合された送信器とを備え、プロセッサ、受信器、メモリ、および送信器は、前述の態様のいずれかの方法を実行するように構成される。

【0021】

一実施形態において、本開示は、ビデオコーディングデバイスによって使用するためのコンピュータプログラム製品を含む非一時的コンピュータ可読媒体を含み、コンピュータプログラム製品は、プロセッサによって実行される時、ビデオコーディングデバイスに前述の態様のいずれかの方法を実行させるように非一時的コンピュータ可読媒体に記憶されているコンピュータ実行可能命令を含む。

【0022】

一実施形態において、本開示は、デコーダを含み、これはレイヤに対するCLVSおよびCLVSによって参照されるSPSを含むビットストリームを受信するための受信手段であって、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer_id値に等しいnuh_layer_id値を有する、受信手段と、SPSに基づきCLVSからのコーディング済みピクチャをデコードして、デコード済みピクチャを生成するためのデコード手段と、デコード済みビデオシーケンスの一部として表示するためにデコード済みピクチャを転送するための転送手段とを備える。

【0023】

いくつかのビデオコーディングシステムは、ビデオシーケンスをピクチャのレイヤにコーディングする。異なるレイヤ内のピクチャは、異なる特性を有する。したがって、エンコーダは、デコーダ側の制約条件に応じて異なるレイヤをデコーダに伝送することができる。この機能を実行するために、エンコーダは、すべてのレイヤを単一のビットストリームにエンコードすることができる。要求に応じて、エンコーダは、ビットストリームから無関係な情報を除去するためにサブビットストリーム抽出プロセスを実行することができる。この結果は、デコーダによって要求されるレイヤ内のデータのみを含む抽出済みビットストリームである。レイヤがどのように関係付けられているかの記述は、ビデオパラメータセット(VPS)の中に含まれ得る。サイマルキャストレイヤは、他のレイヤを参照することなしに表示するように設定されたレイヤである。サイマルキャストレイヤがデコーダに伝送されるとき、サブビットストリーム抽出プロセスは、レイヤ関係がサイマルキャストレイヤをデコードするのに必要ないのでVPSを除去するものとしてよい。残念ながら、他のパラメータセット内のいくつかの変数は、VPSを参照し得る。それゆえに、サイマルキャストレイヤが伝送されるときVPSを除去することで、コーディング効率を高め得るが、その結果、エラーも引き起こし得る。さらに、SPSを正しく識別できなかった場合、結果として、SPSをVPSと一緒に誤って除去する可能性がある。これは、SPSが欠落している場合にデコーダにおいてレイヤが正しくデコードされ得ないので問題になることがある。本発明の例は、VPSがサブビットストリーム抽出中に除去されるときエラーを回避するためのメカニズムを含む。具体的には、サブビットストリーム抽出プロセスは、nuh_lay

10

20

30

40

50

r_idsに基づきNALユニットを除去する。SPSは、CLVSがレイヤ間予測を使用しないレイヤ

に含まれるとき、SPSを参照するCLVSのnuh_layer_idに等しいnuh_layer_idを有するよう

に制約される。レイヤ間予測を使用しないレイヤは、サイマルキャストレイヤである。それゆえに、SPSは、VPSが除去されたときのレイヤと同じnuh_layer_idを有する。この方法

で、SPSは、サブビットストリーム抽出プロセスによって誤って除去され

ることがない。レイヤ間予測の使用は、vps_independent_layer_flagによってシグナリン

グされる。しかしながら、vps_independent_layer_flagは、VPSが除去されると除去され

る。したがって、SPSがVPSを参照しないとき、vps_independent_layer_flag[General Layer

rlIdx[nuh_layer_id]]として表される、現在のレイヤに対するvps_independent_layer fla

gは1であると推論される。SPSは、sps_video_parameter_set_idが0に設定されたとき、VP

Sを参照しない。さらに、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer _id]]

が1に等しいとき(たとえば、VPSが存在しないとき)、デコーダおよび/またはHRDは、現在

のレイヤ/CLVSがレイヤ間予測を使用しないことを推論することができる。この一連の推論を採用することによって、SPSは、VPSおよび対応するパラメータがビットストリームから除去されたときであってもサブビットストリーム抽出プロセスによる抽出を回避するために適切なnuh_layer_idを含むように制約され得る。その結果、エンコーダおよびデコーダの機能性が向上する。さらに、コーディング効率は、サイマルキャストレイヤのみを含むビットストリームから不要なVPSを首尾よく除去することによって高められ、これにより、エンコーダとデコーダの両方においてプロセッサ、メモリ、および/またはネットワークシグナリングリソースの使用量を低減する。

【0024】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、デコーダが前述の態様のいずれかの方法を実行するようにさらに構成される、ことを規定する。

【0025】

一実施形態において、本開示は、エンコーダを含み、これはビットストリーム内のレイヤに対するCLVSをエンコードし、ビットストリーム内に、CLVSによって参照されるSPSを

エンコードするためのエンコード手段であって、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer_id値に等しいnuh_layer_id値を有するよう制約される、エンコ

ード手段と、デコーダに向けた通信のためにビットストリームを記憶するための記憶手段とを備える。

【0026】

いくつかのビデオコーディングシステムは、ビデオシーケンスをピクチャのレイヤにコーディングする。異なるレイヤ内のピクチャは、異なる特性を有する。したがって、エンコーダは、デコーダ側の制約条件に応じて異なるレイヤをデコーダに伝送することができる。この機能を実行するために、エンコーダは、すべてのレイヤを単一のビットストリームにエンコードすることができる。要求に応じて、エンコーダは、ビットストリームから無関係な情報を除去するためにサブビットストリーム抽出プロセスを実行することができる。この結果は、デコーダによって要求されるレイヤ内のデータのみを含む抽出済みピッ

10

20

30

40

50

トストリームである。レイヤがどのように関係付けられているかの記述は、ビデオパラメータセット(VPS)の中に含まれ得る。サイマルキャストレイヤは、他のレイヤを参照することなしに表示するように設定されたレイヤである。サイマルキャストレイヤがデコーダに伝送されるとき、サブビットストリーム抽出プロセスは、レイヤ関係がサイマルキャストレイヤをデコードするのに必要ないのでVPSを除去するものとしてよい。残念ながら、他のパラメータセット内のいくつかの変数は、VPSを参照し得る。それゆえに、サイマルキャストレイヤが伝送されるときVPSを除去することで、コーディング効率を高め得るが、その結果、エラーも引き起こし得る。さらに、SPSを正しく識別できなかった場合、結果として、SPSをVPSと一緒に誤って除去する可能性がある。これは、SPSが欠落している場合にデコーダにおいてレイヤが正しくデコードされ得ないので問題になることがある。

本発明の例は、VPSがサブビットストリーム抽出中に除去されるときのエラーを回避するためのメカニズムを含む。具体的には、サブビットストリーム抽出プロセスは、`nuh_layer_ids`に基づきNALユニットを除去する。SPSは、CLVSがレイヤ間予測を使用しないレイヤに含まれるとき、SPSを参照するCLVSの`nuh_layer_id`に等しい`nuh_layer_id`を有するよう

に制約される。レイヤ間予測を使用しないレイヤは、サイマルキャストレイヤである。それゆえに、SPSは、VPSが除去されたときのレイヤと同じ`nuh_layer_id`を有する。この方法

で、SPSは、サブビットストリーム抽出プロセスによって誤って除去されることがない。レイヤ間予測の使用は、`vps_independent_layer_flag`によってシグナリングされる。しかしながら、`vps_independent_layer_flag`は、VPSが除去されると除去される。したがって、SPSがVPSを参照しないとき、`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]`として表される、現在のレイヤに対する`vps_independent_layer_flag`は1であると推論される。SPSは、`sps_video_parameter_set_id`が0に設定されたとき、VP

Sを参照しない。さらに、`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]`が1に等しいとき(たとえば、VPSが存在しないとき)、デコーダおよび/またはHRDは、現在のレイヤ/CLVSがレイヤ間予測を使用しないことを推論することができる。この一連の推論を採用することによって、SPSは、VPSおよび対応するパラメータがビットストリームから除去されたときであってもサブビットストリーム抽出プロセスによる抽出を回避するために適切な`nuh_layer_id`を含むように制約され得る。その結果、エンコーダおよびデコーダの機能が向上する。さらに、コーディング効率は、サイマルキャストレイヤのみを含むビットストリームから不要なVPSを首尾よく除去することによって高められ、これにより、エンコーダとデコーダの両方においてプロセッサ、メモリ、および/またはネットワークシグナリングリソースの使用量を低減する。

【0027】

任意選択で、前述の態様のいずれかにおいて、態様の別の実装形態は、エンコーダが前述の態様のいずれかの方法を実行するようにさらに構成される、ことを規定する。

【0028】

わかりやすくするために、前述の実施形態のうちのいずれか1つが他の前述の実施形態のうちのいずれか1つまたは複数と組み合わせられて、本開示の範囲内の新しい実施形態を形成するものとしてよい。

【0029】

10

20

30

40

50

これらおよび他の特徴は、添付図面および請求項と併せて次の詳細な説明からより明確に理解されるであろう。

【0030】

本開示をより完全に理解できるように、類似の番号は類似の部分を表す、添付図面および詳細な説明に関して以下の簡単な説明が参照される。

【図面の簡単な説明】

【0031】

【図1】ビデオ信号をコーディングする例示的な一方法のフローチャートである。

【図2】ビデオコーディングのための例示的なコーディングおよびデコーディング(コーデック)システムの概略図である。

【図3】例示的なビデオエンコーダを例示する概略図である。

【図4】例示的なビデオデコーダを例示する概略図である。

【図5】例示的な仮想参照デコーダ(HRD)を例示する概略図である。

【図6】レイヤ間予測を行うように構成されている例示的なマルチレイヤビデオシーケンスを例示する概略図である。

【図7】例示的なビットストリームを例示する概略図である。

【図8】例示的なビデオコーディングデバイスの概略図である。

【図9】サイマルキャストレイヤに対するサブビットストリーム抽出時にシーケンスパラメータセット(SPS)を保持することをサポートするためにマルチレイヤビデオシーケンスをビットストリーム内にエンコードする例示的な方法のフローチャートである。

【図10】サブビットストリーム抽出時にSPSが保持されているマルチレイヤビットストリームから抽出されたサイマルキャストレイヤを含むビットストリームからビデオシーケンスをデコードする例示的な方法のフローチャートである。

【図11】サイマルキャストレイヤに対するサブビットストリーム抽出時にSPSを保持することをサポートするためにマルチレイヤビデオシーケンスをビットストリーム内にコーディングするための例示的なシステムの概略図である。

【発明を実施するための形態】

【0032】

最初に、1つまたは複数の実施形態の例示的な実装が以下に提示されているが、開示されているシステムおよび/または方法は、現在知られているか、または存在している、いくつかの技術を使用することで実装され得ることは理解されるべきである。本開示は、本明細書において例示され、説明されている例示的な設計および実装を含む、以下に例示されている例示的な実装、図面、および技術に決して限定されるべきでないが、等価物の全範囲とともに付属の請求項の範囲内で修正され得る。

【0033】

次の用語は、本明細書において反対の文脈で使用されていない限り次のように定義される。具体的には、次の定義は、本開示をさらに明確にすることを意図されている。しかしながら、用語は、異なる文脈において異なる形で説明され得る。したがって、次の定義は、補足として考慮されるべきであり、本明細書においてそのような用語に対して提供される説明のいかなる他の定義を制限するものとして考慮されるべきではない。

【0034】

ビットストリームは、エンコーダとデコーダとの間の伝送のために圧縮されるビデオデータを含むビットのシーケンスである。エンコーダは、エンコーディングプロセスを使用してビデオデータをビットストリームに圧縮するように構成されるデバイスである。デコーダは、デコーディングプロセスを使用してビデオデータを表示のためにビットストリームに再構成するように構成されるデバイスである。ピクチャは、フレームまたはそのフィールドを作成するルマサンプルの配列および/またはクロマサンプルの配列である。エンコードされているか、またはデコードされているピクチャは、説明を明確にするために現在のピクチャとして参照され得る。コーディング済みピクチャは、アクセスユニット(AU)内のNALユニットヘッダレイヤ識別子(nuh_layer_id)の特定の値を有するビデオコーデ

10

20

30

40

50

ィングレイヤ(VCL)ネットワーク抽象化レイヤ(NAL)ユニットを含み、ピクチャのすべてのコーディングツリーユニット(CTU)を含むピクチャのコーディング済み表現である。デコード済みピクチャは、コーディング済みピクチャにデコーディングプロセスを適用することによって生成されるピクチャである。NALユニットは、望ましい場合にエミュレーション防止バイトを散在させた、データの型の指示である、ローバイトシーケンスペイロード(RBSP)の形態のデータを含むシンタックス構造である。VCL NALユニットは、ピクチャのコーディング済みスライスなどの、ビデオデータを含むようにコーディングされたNALユニットである。非VCL NALユニットは、ビデオデータをデコードすること、適合性チェックの実行、または他の動作をサポートするシンタックスおよび/またはパラメータなどの非ビデオデータを含むNALユニットである。レイヤは、レイヤId(識別子)によって示されているような指定された特性(たとえば、共通の解像度、フレームレート、画像サイズなど)を共有するVCL NALユニットと、関連付けられている非VCL NALユニットとのセットである。NALユニットヘッダレイヤ識別子(nuh_layer_id)は、NALユニットを含むレイヤの識別子を指定するシンタックス要素である。

【 0 0 3 5 】

仮想参照デコーダ(HRD)は、エンコーディングプロセスによって生成されたビットストリームの変動性をチェックして、指定された制約条件への適合性を検証するエンコーダ上で動作するデコーダモデルである。ビットストリームの適合性テストは、エンコード済みビットストリームがバーサタイル・ビデオ・コーディング(VVC)などの規格に準拠しているかどうかを決定するためのテストである。ビデオパラメータセット(VPS)は、ビデオ全体に関するパラメータを含むシンタックス構造である。シーケンスパラメータセット(SPS)は、0個またはそれ以上のコーディング済みレイヤビデオシーケンス(CLVS)全体に適用されるシンタックス要素を含むシンタックス構造である。CLVSは、同じnuh_layer_id値を有するコーディング済みピクチャのシーケンスである。SPSビデオパラメータセット識別子(sps_video_parameter_set_id)は、SPSによるVPS参照の識別子(ID)を指定するシンタックス要素である。一般レイヤインデックス(GeneralLayerIdx[i])は、対応するレイヤiのインデックスを指定する導出変数である。それゆえに、nuh_layer_idのレイヤIDを有する現在のレイヤは、GeneralLayerIdx[nuh_layer_id]によって指定されるインデックスを有する。現在のレイヤインデックスは、エンコードまたはデコードされているレイヤに対応するレイヤインデックスである。VPS独立レイヤフラグ(vps_independent_layer_flag[i])は、対応するレイヤiがレイヤ間予測を使用するかどうかを指定するシンタックス要素である。それゆえに、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]は、現在のレイヤがレイヤ間予測を使用するかどうかを指定する。レイヤ間予測は、異なるレイヤ(たとえば、同じアクセスユニット)からの参照ピクチャに基づき現在のレイヤの現在のピクチャのサンプル値のブロックをコーディングするメカニズムである。アクセスユニット(AU)は、同じ出力時間にすべて関連付けられている異なるレイヤ内のコーディング済みピクチャのセットである。VPSパラメータセット識別子(vps_video_parameter_set_id)は、他のシンタックス要素/構造による参照のためにVPSに対するIDを提供するシンタックス要素である。コーディング済みビデオシーケンスは、1つまたは複数のコーディング済みピクチャのセットである。デコード済みビデオシーケンスは、1つまたは複数のデコード済みピクチャのセットである。

【 0 0 3 6 】

本明細書において使用される頭字語は、アクセスユニット(AU)、コーディングツリーブロック(CTB)、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)、コーディ

10

20

30

40

50

ング済みレイヤビデオシーケンス(CLVSS)、コーディング済みレイヤビデオシーケンス開始(CLVSS)、コーディング済みビデオシーケンス(CVSS)、コーディング済みビデオシーケンス開始(CVSS)、ジョイントビデオエキスパートチーム(JVET)、仮想参照デコーダ(HRD)、動き制約タイルセット(MCTS)、最大転送ユニット(MTU)、ネットワーク抽象化レイヤ(NAL)、出力レイヤセット(OLS)、オペレーションポイント(OP)、ピクチャ順序カウンタ(POC)、ランダムアクセスポイント(RAP)、ローバイトシーケンスペイロード(RBSP)、シーケンスパラメータセット(PPS)、ビデオパラメータセット(VPS)、パーサタイル・ビデオ・コーディング(VVC)である。

【 0 0 3 7 】

多くのビデオ圧縮技術は、データの損失を最小限度に抑えてビデオファイルのサイズを縮小するために採用され得る。たとえば、ビデオ圧縮技術は、空間(たとえば、ピクチャ内)予測および/または時間(たとえば、ピクチャ間)予測を実行して、ビデオシーケンス内のデータ冗長性を低減するか、または除去することを含むことができる。ブロックベースのビデオコーディングのために、ビデオスライス(たとえば、ビデオピクチャ、またはビデオピクチャの一部)は、いくつかのビデオブロックに区別化されてよく、これは、ツリーブロック、コーディングツリーブロック(CTB)、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)、および/またはコーディングノードと呼ばれることもある。ピクチャのイントラコーディング済み(I)スライス中のビデオブロックは、同じピクチャ中の近傍ブロック中の参照サンプルに対する空間予測を使用してコーディングされる。ピクチャのインターコーディング済み一方向予測(P)または双方向予測(B)スライス中のビデオブロックは、同じピクチャ中の近傍ブロック中の参照サンプルに関する空間予測、または他の参照ピクチャ中の参照サンプルに関する時間予測を採用することによってコーディングされ得る。ピクチャは、フレームおよび/または画像と称されてよく、参照ピクチャは、参照フレームおよび/または参照画像と称されてよい。空間予測または時間予測は、結果として、画像ブロックを表す予測ブロックをもたらす。残差データは、元の画像ブロックと予測ブロックとの間のピクセル差分を表す。したがって、インターコーディング済みブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトル、およびコーディング済みブロックと予測ブロックとの間の差分を示す残差データに従ってエンコードされる。イントラコーディング済みブロックは、イントラコーディングモードおよび残差データに従ってエンコードされる。さらに圧縮するために、残差データは、ピクセル領域から変換領域に変換され得る。これらの結果、残差変換係数が得られ、残差変換係数が量子化され得る。量子化済み変換係数は、最初に2次元配列に配列され得る。量子化済み変換係数は、変換係数の1次元ベクトルを生成するために走査され得る。エントロピーコーディングが、なおいっそうの圧縮を行うために適用され得る。そのようなビデオ圧縮技術は、以下でより詳しく説明される。

【 0 0 3 8 】

エンコード済みビデオが正確にデコードされ得ることを確実にするために、ビデオは、対応するビデオコーディング規格に従ってエンコードされ、デコードされる。ビデオコーディング規格は、国際電気通信連合(ITU)標準化部門(ITU-T)H.261、国際標準化機構/国際電気標準会議(ISO/IEC)モーションピクチャエキスパートグループ(MPEG)-1 Part 2、ITU-T H.262またはISO/IEC MPEG-2 Part 2、ITU-T H.263、ISO/IEC MPEG-4 Part 2、ITU-T H.264またはISO/IEC MPEG-4 Part 10としても知られている高度ビデオコーディング(AVC)、およびITU-T H.265またはMPEG-H Part 2としても知られている高効率ビデオコーディング

10

20

30

40

50

(HEVC)を含む。AVCは、スケーラブルビデオコーディング(SVC)、マルチビュービデオコーディング(MVC)、マルチビュービデオコーディングプラス深さ(MVC+D)、および3次元(3D)AVC(3D-AVC)などの拡張機能を含む。HEVCは、スケーラブルHEVC(SHVC)、マルチビューHEVC(MV-HEVC)、および3D HEVC(3D-HEVC)などの拡張機能を含む。ITU-TおよびISO/IECのジョイントビデオエキスパートチーム(JVET)は、バーサタイル・ビデオ・コーディング(VVC)と称されるビデオコーディング規格の開発を開始している。VVCは、JVET-O2001-v14を含む、作業草案(WD)に含まれる。

【 0 0 3 9 】

いくつかのビデオコーディングシステムは、ビデオシーケンスをピクチャのレイヤにコーディングする。異なるレイヤ内のピクチャは、異なる特性を有する。したがって、エンコーダは、デコーダ側の制約条件に応じて異なるレイヤをデコーダに伝送することができる。この機能を実行するために、エンコーダは、すべてのレイヤを単一のビットストリームにエンコードすることができる。要求に応じて、エンコーダは、ビットストリームから無関係な情報を除去するためにサブビットストリーム抽出プロセスを実行することができる。この結果は、デコーダによって要求されるレイヤ内のデータのみを含む抽出済みビットストリームである。レイヤがどのように関係付けられているかの記述は、ビデオパラメータセット(VPS)の中に含まれ得る。サイマルキャストレイヤは、他のレイヤを参照することなしに表示するように設定されたレイヤである。サイマルキャストレイヤがデコーダに伝送されるとき、サブビットストリーム抽出プロセスは、レイヤ関係がサイマルキャストレイヤをデコードするのに必要ないのでVPSを除去するものとしてよい。残念ながら、他のパラメータセット内のいくつかの変数は、VPSを参照し得る。それゆえに、サイマルキャストレイヤが伝送されるときVPSを除去することで、コーディング効率を高め得るが、その結果、エラーも引き起こし得る。さらに、シーケンスパラメータセット(SPS)を正しく識別できなかった場合、結果として、SPSをVPSと一緒に誤って除去する可能性がある。これは、SPSが欠落している場合にデコーダにおいてレイヤが正しくデコードされ得ないので問題になることがある。

【 0 0 4 0 】

本明細書において開示されているのは、VPSがサブビットストリーム抽出中に除去されるときエラーを回避するためのメカニズムである。具体的には、サブビットストリーム抽出プロセスは、NALユニットレイヤ識別子(nuh_layer_ids)に基づきネットワーク抽象化レイヤ(NAL)ユニットを除去する。SPSは、CLVSがレイヤ間予測を使用しないレイヤに含まれるとき、SPSを参照するコーディング済みレイヤビデオシーケンス(CLVS)のnuh_layer_idに等しいnuh_layer_idを有するように制約される。レイヤ間予測を使用しないレイヤは、サイマルキャストレイヤである。それゆえに、SPSは、VPSが除去されたときのレイヤと同じnuh_layer_idを有する。この方法で、SPSは、サブビットストリーム抽出プロセスによって誤って除去されることがない。レイヤ間予測の使用は、VPS独立レイヤフラグ(vps_independent_layer_flag)によってシグナリングされる。しかしながら、vps_independent_layer_flagは、VPSが除去されると除去される。したがって、SPSがVPSを参照しないとき、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]として表される、現在

10

20

30

40

50

のレイヤに対するvps_independent_layer_flagは1であると推論される。SPSは、SPS VPS

識別子(sps_video_parameter_set_id)が0に設定されたとき、VPSを参照しない。さらに、

vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき(たとえ

ば、VPSが存在しないとき)、デコーダおよび/または仮想参照デコーダ(HRD)は、現在のレイヤ/CLVSがレイヤ間予測を使用しないことを推論することができる。この一連の推論を採用することによって、SPSは、VPSおよび対応するパラメータがビットストリームから除去されたときであってもサブビットストリーム抽出プロセスによる抽出を回避するために適切なnuh_layer_idを含むように制約され得る。その結果、エンコーダおよびデコーダの機能性が向上する。さらに、コーディング効率は、サイマルキャストレイヤのみを含むビットストリームから不要なVPSを首尾よく除去することによって高められ、これにより、エンコーダとデコーダの両方においてプロセッサ、メモリ、および/またはネットワークシグナリングリソースの使用量を低減する。

【0041】

図1は、ビデオ信号をコーディングする例示的な動作方法100のフローチャートである。具体的には、ビデオ信号は、エンコーダにおいてエンコードされる。エンコーディングプロセスは、様々なメカニズムを使用することによってビデオ信号を圧縮し、ビデオファイルサイズを縮小する。より小さいファイルサイズは、関連する帯域幅オーバーヘッドを低減しながら、圧縮済みビデオファイルがユーザへ伝送されることを可能にする。次に、エンドユーザに表示するために、デコーダは圧縮済みビデオファイルをデコードし、元のビデオ信号を再構成する。デコーディングプロセスは、一般的に、デコーダが一貫してビデオ信号を再構成することを可能にするようにエンコーディングプロセスをミラーリングする。

【0042】

ステップ101で、ビデオ信号は、エンコーダに入力される。たとえば、ビデオ信号は、メモリに記憶されている非圧縮ビデオファイルであってもよい。別の例として、ビデオファイルは、ビデオカメラなどのビデオキャプチャデバイスによってキャプチャされ、ビデオのライブストリーミングをサポートするようにエンコードされ得る。ビデオファイルは、音声コンポーネントとビデオコンポーネントの両方を含み得る。ビデオコンポーネントは、順に視聴されたとき、動きの視覚的な印象を与える一連の画像フレームを含む。フレームは、本明細書においてルーマ成分(またはルーマサンプル)と称される光、およびクロマ成分(または色サンプル)と呼ばれる色に関して表現されるピクセルを含む。いくつかの例において、フレームは、3次元表示をサポートするために深度値も含み得る。

【0043】

ステップ103で、ビデオはいくつかのブロックに区分化される。区分化することは、圧縮のために各フレーム内のピクセルを正方形および/または長方形のブロックに細分することを含む。たとえば、高効率ビデオコーディング(HEVC)(H.265およびMPEG-H Part 2と

しても知られている)において、フレームは、最初に、事前定義されたサイズ(たとえば、64ピクセル×64ピクセル)のブロックである、コーディングツリーユニット(CTU)に分割さ

れ得る。CTUは、ルーマサンプルとクロマサンプルの両方を含む。コーディングツリーは、CTUをいくつかのブロックに分割し、次いでさらなるエンコーディングをサポートする構成が達成されるまでブロックを再帰的に細分するために採用され得る。たとえば、フレームのルーマ成分は、個々のブロックが比較的均質な照明値を含むまで細分化され得る。さらに、フレームのクロマ成分は、個々のブロックが比較的均質な色値を含むまで細分化され得る。したがって、区分化メカニズムは、ビデオフレームのコンテンツによって異なる。

10

20

30

40

50

【 0 0 4 4 】

ステップ105で、各様々な圧縮メカニズムが採用され、ステップ103で区分化された画像

ブロックを圧縮する。たとえば、インター予測および/またはイントラ予測が採用され得る。インター予測は、共通のシーン内のオブジェクトが連続するフレーム内に出現する傾向があるという事実を利用するように設計されている。したがって、参照フレーム内のオブジェクトを描写するブロックは、隣接するフレーム内に繰り返し記述される必要はない。具体的には、テーブルなどのオブジェクトは、複数のフレームにわたって一定の位置に留まり得る。したがって、テーブルは、一度記述されると、隣接するフレームが参照フレームを再度参照することができる。複数のフレームにまたがるオブジェクトのマッチングには、パターンマッチングメカニズムが採用され得る。さらに、オブジェクトの移動やカメラの移動などにより、複数のフレームにまたがって移動するオブジェクトが表現され得る。特定の例として、ビデオは、複数のフレームにわたって画面上を移動する自動車を示し得る。動きベクトルは、そのような移動を記述するために使用することができる。動きベクトルは、フレーム内のオブジェクトの座標から参照フレーム内のオブジェクトの座標へのオフセットを提供する2次元ベクトルである。それゆえに、インター予測は、現在のフレーム内の画像ブロックを、参照フレーム内の対応するブロックからのオフセットを示す動きベクトルのセットとしてエンコードすることができる。

10

【 0 0 4 5 】

イントラ予測は、共通フレーム内のブロックをエンコードする。イントラ予測は、ルーマ成分およびクロマ成分がフレーム内でクラスタ化する傾向があるという事実を利用する。たとえば、木の一部分における緑色のパッチは、緑色の類似のパッチに隣接して位置決めされる傾向がある。イントラ予測では、複数の方向予測モード(たとえば、HEVCでは33種類)、平面モード、および直流(DC)モードを使用する。これらの方向モードは、現在のブロックが、対応する方向の近傍ブロックのサンプルと類似している/同じであることを示す。平面モードは、行/列(たとえば、平面)に沿った一連のブロックが、行の端にある近傍ブロックに基づき補間され得ることを示す。平面モードは、実際には、変化する値における比較的一定の勾配を使用することによって行/列にまたがる光/色の滑らかな遷移を示す。DCモードは、境界平滑化に使用され、ブロックが方向予測モードの角度方向に関連するすべての近傍ブロックのサンプルに関連付けられている平均値に類似する/同じであることを示す。したがって、イントラ予測ブロックは、実際の値ではなく様々な関係予測モード値として画像ブロックを表現することができる。さらに、インター予測ブロックは、実際の値ではなく動きベクトル値として画像ブロックを表現することができる。いずれの場合も、予測ブロックは、いくつかの場合において画像ブロックを正確に表さないことがある。任意の差分が、残差ブロック内に収められる。ファイルをさらに圧縮するために、変換が残差ブロックに適用され得る。

20

30

【 0 0 4 6 】

ステップ107で、様々なフィルタリング技術が適用され得る。HEVCでは、フィルタは、ループ内フィルタリングスキームに従って適用される。上で説明されているブロックベースの予測の結果、デコーダにおいてブロック状画像が作成され得る。さらに、ブロックベースの予測スキームは、ブロックをエンコードし、次いで、参照ブロックとして後で使用するためにエンコード済みブロックを再構成し得る。ループ内フィルタリングスキームは、ノイズ抑制フィルタ、デブロックングフィルタ、適応ループフィルタ、およびサンプル適応オフセット(SAO)フィルタをブロック/フレームに反復適用する。これらのフィルタは、そのようなブロックングアーチファクトを軽減し、それによりエンコード済みファイルが正確に再構成され得る。さらに、これらのフィルタは、再構成済み参照ブロック内のアーチファクトを軽減し、それにより、アーチファクトが再構成済み参照ブロックに基づきエンコードされる後続のブロック内にさらなるアーチファクトを生じさせる可能性が低くなる。

40

【 0 0 4 7 】

50

ビデオ信号が区分化され、圧縮され、フィルタリングされた後、結果として得られたデータは、ステップ109でビットストリームにエンコードされる。ビットストリームは、上で説明されているデータ、さらにはデコーダにおいて適切なビデオ信号再構成をサポートするために望ましい任意のシグナリングデータを含む。たとえば、そのようなデータは、区分データ、予測データ、残差ブロック、およびコーディング命令をデコーダに送る様々なフラグを含み得る。ビットストリームは、要求に応じてデコーダに向けて伝送するためにメモリに記憶され得る。ビットストリームは、また、複数のデコーダに向けてブロードキャストされ、および/またはマルチキャストされてもよい。ビットストリームの作成は、反復プロセスである。したがって、ステップ101、103、105、107、および109は、多数

10

のフレームおよびブロックにわたって連続的に、および/または同時に実行され得る。図1に示されている順序は、説明を明確にし、容易にするために提示されており、ビデオコーディングプロセスを特定の順序に制限することを意図されていない。

【0048】

デコーダは、ビットストリームを受信し、ステップ111でデコーディングプロセスを開始する。具体的には、デコーダは、エントロピーデコーディングスキームを使用して、ビットストリームに対応するシンタックスおよびビデオデータに変換する。デコーダは、ステップ111で、ビットストリームからのシンタックスデータを使用して、フレームに対する区分を決定する。区分化は、ステップ103におけるブロック区分化の結果と一致するべきである。次に、ステップ111で採用されているようなエントロピーエンコーディング/デコーディングについて説明する。エンコーダは、入力画像における値の空間的位置決めに基づきいくつかの可能な選択肢からブロック区分化スキームを選択するなど、圧縮プロセスにおいて多くの選択を行う。正確な選択肢をシグナリングすることでは、多数のピンが使用され得る。本明細書において使用されているように、ピンは、変数として扱われる2値(たとえば、文脈に応じて変化し得るビット値)である。エントロピーコーディングは、エンコーダが特定のケースに対して明らかに実行不可能である任意のオプションを破棄し、許容可能なオプションのセットを残すことを可能にする。次いで、各許容可能なオプションは、コードワードを割り当てられる。コードワードの長さは、許容可能なオプションの数に基づく(たとえば、2つのオプションには1つのピン、3から4つのオプションには2つのピンなど)。次いで、エンコーダは、選択されたオプションに対してコードワードをエンコードする。このスキームでは、コードワードは可能なすべてのオプションの潜在的に大きいセットからの選択を一意的に示すこととは反対に、許容可能なオプションの小さいサブセットからの選択を一意的に示すために望ましい大きさであるので、コードワードのサイズを縮小する。次いで、デコーダは、エンコーダと似た方式で、許容可能なオプションのセットを決定することによって選択をデコードする。許容可能なオプションのセットを決定することによって、デコーダは、コードワードを読み出し、エンコーダによって行われた選択を決定することができる。

20

30

【0049】

ステップ113で、デコーダは、ブロックデコーディングを実行する。具体的には、デコーダは、逆変換を採用して、残差ブロックを生成する。次いで、デコーダは、区分化に従って画像ブロックを再構成するために残差ブロックおよび対応する予測ブロックを使用する。予測ブロックは、ステップ105においてエンコーダで生成されるようなイントラ予測ブロックとインター予測ブロックの両方を含み得る。次いで、再構成済み画像ブロックは、ステップ111で決定された区分化データに従って再構成済みビデオ信号のフレーム内に位置決めされる。ステップ113に対するシンタックスは、また、上で説明されているようにエントロピーコーディングを介してビットストリーム内でシグナリングされ得る。

40

【0050】

ステップ115で、フィルタリングは、エンコーダでステップ107に類似する方式により再構成済みビデオ信号のフレーム上で実行される。たとえば、ノイズ抑制フィルタ、デブロ

50

ッキングフィルタ、適応ループフィルタ、およびSAOフィルタが、ブロッキングアーチファクトを除去するためにフレームに適用され得る。フレームがフィルタリングされた後、ビデオ信号は、エンドユーザによる視聴のためにステップ117でディスプレイに出力され得る。

【0051】

図2は、ビデオコーディングのための例示的なコーディングおよびデコーディング(コーデック)システム200の概略図である。具体的には、コーデックシステム200は、動作方法

100の実装形態をサポートするための機能性を提供する。コーデックシステム200は、エンコーダとデコーダの両方において採用されるコンポーネントを描くように一般化されている。コーデックシステム200は、動作方法100のステップ101および103に関して説明されて

いるようにビデオ信号を受信し、区分化し、その結果、区分化済みビデオ信号201が得られる。次いで、コーデックシステム200は、方法100におけるステップ105、107、および10

9に関して説明されているようにエンコーダとして動作するとき、区分化済みビデオ信号201をコーディング済みビットストリームに圧縮する。コーデックシステム200は、デコーダとして動作するとき、動作方法100のステップ111、113、115、および117に関して説明

されているようにビットストリームから出力ビデオ信号を生成する。コーデックシステム200は、一般コーダ制御コンポーネント211、変換スケーリングおよび量子化コンポーネン

ト213、ピクチャ内推定コンポーネント215、ピクチャ内予測コンポーネント217、動き補

償コンポーネント219、動き推定コンポーネント221、スケーリングおよび逆変換コンポ

ネント229、フィルタ制御分析コンポーネント227、ループ内フィルタコンポーネント22

5、デコード済みピクチャバッファコンポーネント223、ならびにヘッダフォーマット

ングおよびコンテキスト適応2値算術コーディング(CABAC)コンポーネント231を含む。そのようなコンポーネントは、図示されているように結合される。図2において、黒線はエン

コード/デコードされるべきデータの動きを示し、破線は他のコンポーネントの動作を制御する制御データの動きを示している。コーデックシステム200のコンポーネントは、すべて、エンコーダ内に存在してもよい。デコーダは、コーデックシステム200のコンポーネントのサブセットを含み得る。たとえば、デコーダは、ピクチャ内予測コンポーネント

【0052】

217、動き補償コンポーネント219、スケーリングおよび逆変換コンポーネント229、ル

ープ内フィルタコンポーネント225、およびデコード済みピクチャバッファコンポーネント2

23を含み得る。次に、これらのコンポーネントについて説明される。

区分化済みビデオ信号201は、コーディングツリーによってピクセルのいくつかのブロックに区分化されたキャプチャ済みビデオシーケンスである。コーディングツリーは、ピクセルのブロックをピクセルのより小さいブロックに細分するために様々な分割モードを採用する。次いで、これらのブロックは、より小さいブロックにさらに細分され得る。ブロックは、コーディングツリーのノードと称されてもよい。より大きい親ノードは、より小さい子ノードに分割される。ノードが細分される回数は、ノード/コーディングツリーの深さと称される。分割済みブロックは、いくつかの場合において、コーディングユニット(CU)に含まれ得る。たとえば、CUは、CUに対する対応するシンタックス命令とともに、

ルーマブロック、赤色差クロマ(Cr)ブロック、および青色差クロマ(Cb)ブロックを含むCT

10

20

30

40

50

Uのサブ部分であってよい。分割モードは、ノードを採用された分割モードに応じて様々な形状の2つ、3つ、または4つの子ノードにそれぞれ区分化するために採用される二分木(BT)、三分木(TT)、および四分木(QT)を含み得る。区分化済みビデオ信号201は、圧縮のた

めに一般コーダ制御コンポーネント211、変換スケーリングおよび量子化コンポーネント213、ピクチャ内推定コンポーネント215、フィルタ制御分析コンポーネント227、および動

き推定コンポーネント221に転送される。

【0053】

一般コーダ制御コンポーネント211は、アプリケーション制約に従って、ビットストリームへのビデオシーケンスの画像のコーディングに関係する決定を行うように構成される。たとえば、一般コーダ制御コンポーネント211は、再構成品質に対するビットレート/ビットストリームサイズの最適化を管理する。そのような決定は、ストレージスペース/帯域幅の利用可能性および画像解像度要求に基づき行われ得る。一般コーダ制御コンポーネント211は、また、バッファアンダーランおよびバッファオーバーランの問題を軽減するために伝送速度に照らしてバッファ利用を管理する。これらの問題を管理するために、一般コーダ制御コンポーネント211は、他のコンポーネントによる区分化、予測、およびフィルタリングを管理する。たとえば、一般コーダ制御コンポーネント211は、解像度を上げ、帯域幅使用量を増やすために圧縮複雑度を動的に上げるか、または解像度および帯域幅使用量を減らすために圧縮複雑度を下げるものとしてよい。したがって、一般コーダ制御コンポーネント211は、コーデックシステム200の他のコンポーネントを制御して、ビデ

オ信号再構成品質とビットレートの問題点とのバランスをとる。一般コーダ制御コンポーネント211は、他のコンポーネントの動作を制御する、制御データを作成する。制御データは、ヘッダフォーマットおよびCABACコンポーネント231にも転送され、ビットストリーム内にエンコードされ、デコーダでのデコーディングのためにパラメータをシグナリングする。

【0054】

区分化済みビデオ信号201はまた、インター予測のために、動き推定コンポーネント221および動き補償コンポーネント219に送信される。区分化済みビデオ信号201のフレームまたはスライスは、複数のビデオブロックに分割され得る。動き推定コンポーネント221および動き補償コンポーネント219は、時間予測を行うため1つまたは複数の参照フレーム内の1つまたは複数のブロックに関して受信されたビデオブロックのインター予測コーディングを実行する。コーデックシステム200は、複数のコーディングパスを実行して、たとえば、ビデオデータの各ブロックに対する適切なコーディングモードを選択し得る。

【0055】

動き推定コンポーネント221および動き補償コンポーネント219は、高度に統合されても

よいが、概念的な目的のために別々に例示されている。動き推定コンポーネント221によって実行される動き推定は、ビデオブロックに関する動きを推定する、動きベクトルを生成するプロセスである。動きベクトルは、たとえば、予測ブロックに関するコーディング済みオブジェクトの変位を示し得る。予測ブロックは、ピクセル差分に関して、コーディングされるべきブロックとよくマッチすることが判明しているブロックである。予測ブロックは、参照ブロックとも称され得る。そのようなピクセル差分は、差分絶対値和(SAD)、差分二乗和(SSD)、または他の差分メトリックによって決定され得る。HEVCは、CTU、コ

ーディングツリーブロック(CTB)、およびCUを含むいくつかのコーディング済みオブジェ

10

20

30

40

50

クトを採用する。たとえば、CTUは、CTBに分割され、次いでこれはCUに含まれるようにCB

に分割され得る。CUは、予測データを含む予測ユニット(PU)および/またはCUに対する変換済み残差データを含む変換ユニット(TU)としてエンコードされ得る。動き推定コンポーネント221は、レート歪み最適化プロセスの一部としてレート歪み分析を使用することによって、動きベクトル、PU、およびTUを生成する。たとえば、動き推定コンポーネント2

1は、現在のブロック/フレームに対して複数の参照ブロック、複数の動きベクトルなどを決定し、最良のレート歪み特性を有する参照ブロック、動きベクトルなどを選択し得る。最良のレート歪み特性は、ビデオ再構成の品質(たとえば、圧縮によるデータ損失の量)とコーディング効率(たとえば、最終的エンコーディングのサイズ)の両方のバランスをとる。

10

【0056】

いくつかの例において、コーデックシステム200は、デコード済みピクチャバッファコンポーネント223内に記憶されている参照ピクチャのサブ整数ピクセル位置に対する値を計算し得る。たとえば、ビデオコーデックシステム200は、参照ピクチャの1/4ピクセル位置、1/8ピクセル位置、または他の分数ピクセル位置の値を補間し得る。したがって、動き推定コンポーネント221は、フルピクセル位置および分数ピクセル位置に対する動き探索を実行し、分数ピクセル精度で動きベクトルを出力してもよい。動き推定コンポーネント221は、PUの位置と参照ピクチャの予測ブロックの位置とを比較することによって、インターコーディング済みスライス内のビデオブロックのPUに対する動きベクトルを計算する。動き推定コンポーネント221は、動き補償コンポーネント219へのエンコーディングお

20

および動きに対するヘッダフォーマットおよびCABACコンポーネント231への動きデータとして計算済み動きベクトルとして出力する。

【0057】

動き補償コンポーネント219によって実行される動き補償は、動き推定コンポーネント221によって決定された動きベクトルに基づき予測ブロックをフェッチするか、または生成することを伴い得る。ここでもまた、動き推定コンポーネント221および動き補償コンポーネント219は、いくつかの例において、機能的に一体化され得る。現在のビデオブロックのPUのための動きベクトルを受信すると、動き補償コンポーネント219は、動きベクトルが指す予測ブロックの位置を特定し得る。次いで、残差ビデオブロックは、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって形成される。一般に、動き推定コンポーネント221は、ルーマ成分に関して動き推定を実行し、動き補償コンポーネント219は、クロマ成分とルーマ成分の両方に対してルーマ成分に基づき計算された動きベクトルを使用する。予測ブロックおよび残差ブロックは、変換スケーリングおよび量子化コンポーネント213に転送される。

30

40

【0058】

区分化済みビデオ信号201は、また、ピクチャ内推定コンポーネント215およびピクチャ内予測コンポーネント217に送信される。動き推定コンポーネント221および動き補償コンポーネント219と同様に、ピクチャ内推定コンポーネント215およびピクチャ内予測コンポーネント217は、高度に統合され得るが、概念的な目的のために別々に例示されている。ピクチャ内推定コンポーネント215およびピクチャ内予測コンポーネント217は、上で説明

50

されているように、フレーム間で、動き推定コンポーネント221および動き補償コンポーネント219によって実行されるインター予測の代替として、現在のフレーム内のブロックに関して現在のブロックをイントラ予測する。特に、ピクチャ内推定コンポーネント215は、現在のブロックをエンコードするために使用するイントラ予測モードを決定する。いくつかの例では、ピクチャ内推定コンポーネント215は、複数のテスト済みイントラ予測モードから現在のブロックをエンコードするために適切なイントラ予測モードを選択する。次いで、選択されたイントラ予測モードは、エンコードのためにヘッダフォーマティングおよびCABACコンポーネント231に転送される。

【0059】

たとえば、ピクチャ内推定コンポーネント215は、様々なテスト済みイントラ予測モードに対してレート歪み分析を使用してレート歪み値を計算し、テスト済みモードの中から最良のレート歪み特性を有するイントラ予測モードを選択する。レート歪み分析では、一般的に、エンコード済みブロックとエンコード済みブロックを生成するためにエンコードされた元の未エンコードブロックとの間の歪み(またはエラー)の量、さらにはエンコード済みブロックを生成するために使用されるビットレート(たとえば、ビットの数)を決定する。ピクチャ内推定コンポーネント215は、どのイントラ予測モードがブロックに対して最良のレート歪み値を示すのかを決定するために様々なエンコード済みブロックに対する歪みおよびレートから比率を計算する。それに加えて、ピクチャ内推定コンポーネント215は、レート歪み最適化(RDO)に基づき深度モデリングモード(DMM)を使用して深度マップ

の深度ブロックをコーディングするように構成され得る。

【0060】

ピクチャ内予測コンポーネント217は、エンコーダで実装されたとき、ピクチャ内推定コンポーネント215によって決定された選択済みイントラ予測モードに基づき予測ブロックから残差ブロックを生成するか、またはデコーダで実装されたとき、ビットストリームから残差ブロックを読み出すものとしてよい。残差ブロックは、行列として表される、予測ブロックと元のブロックとの間の値の差分を含む。次いで、残差ブロックは、変換スケールリングおよび量子化コンポーネント213に転送される。ピクチャ内推定コンポーネント215およびピクチャ内予測コンポーネント217は、ルーマ成分とクロマ成分の両方で動作し得る。

【0061】

変換スケールリングおよび量子化コンポーネント213は、残差ブロックをさらに圧縮するように構成される。変換スケールリングおよび量子化コンポーネント213は、離散コサイン変換(DCT)、離散サイン変換(DST)、または概念的に類似する変換などの、変換を残差ブロックに適用し、残差変換係数値を含むビデオブロックを生成する。ウェーブレット変換、整数変換、サブバンド変換、または他のタイプの変換が使用されことも可能であろう。この変換は、ピクセル値領域からの残差情報を周波数領域などの変換領域に変換し得る。変換スケールリングおよび量子化コンポーネント213は、たとえば周波数に基づき、変換済み残差情報をスケールリングするようにも構成される。そのようなスケールリングは、異なる周波数情報が異なる粒度で量子化されるように、残差情報にスケール係数を適用することを伴い、これは、再構成済みビデオの最終的な視覚的品質に影響を及ぼし得る。変換スケールリングおよび量子化コンポーネント213は、ビットレートをさらに低減するために変換係数を量子化するようにも構成される。量子化プロセスは、係数の一部または全部に関連付けられているビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって変更され得る。いくつかの例では、変換スケールリングおよび量子化コンポーネント213は、次いで、量子化済み変換係数を含む行列の走査を実行し得る。量子化済み変換係数は、ヘッダフォーマティングおよびCABACコンポーネント231に転送され、ビット

ストリーム内にエンコードされる。

【0062】

10

20

30

40

50

スケーリングおよび逆変換コンポーネント229は、動き推定をサポートするように変換スケーリングおよび量子化コンポーネント213の逆演算を適用する。スケーリングおよび逆変換コンポーネント229は、たとえば、別の現在のブロックに対する予測ブロックになり得る参照ブロックとして後で使用するために、ピクセル領域内に残差ブロックを再構成するように逆スケーリング、変換、および/または量子化を適用する。動き推定コンポーネント221および/または動き補償コンポーネント219は、後のブロック/フレームの動き推

定において使用するために対応する予測ブロックに残差ブロックを追加して戻すことによって、参照ブロックを計算し得る。フィルタが再構成済み参照ブロックに適用され、それによりスケーリング、量子化、および変換時に作成されたアーチファクトを軽減する。そのようなアーチファクトは、さもなければ、後続のブロックが予測されるとき、不正確な予測を引き起こす(そして、追加のアーチファクトを作成する)可能性もある。

【0063】

フィルタ制御分析コンポーネント227およびループ内フィルタコンポーネント225は、残差ブロックおよび/または再構成済み画像ブロックにフィルタを適用する。たとえば、スケーリングおよび逆変換コンポーネント229からの変換済み残差ブロックは、ピクチャ内予測コンポーネント217および/または動き補償コンポーネント219からの対応する予測ブロックと組み合わせられて、元の画像ブロックを再構成し得る。次いで、フィルタが、再構成済み画像ブロックに適用され得る。いくつかの例では、フィルタは、代わりに残差ブロックに適用されてもよい。図2の他のコンポーネントと同様に、フィルタ制御分析コンポーネント227およびループ内フィルタコンポーネント225は、高度に統合されており、一

緒に実装され得るが、概念的な目的のために別々に描かれている。再構成済み参照ブロックに適用されるフィルタは、特定の空間領域に適用され、そのようなフィルタがどのように適用されるかを調整するための複数のパラメータを含む。フィルタ制御分析コンポーネント227は、再構成済み参照ブロックを解析して、そのようなフィルタが適用されるべき場所を決定し、対応するパラメータを設定する。そのようなデータは、エンコーディングのためのフィルタ制御データとしてヘッダフォーマットおよびCABACコンポーネント2

31に転送される。ループ内フィルタコンポーネント225は、フィルタ制御データに基づき、そのようなフィルタを適用する。フィルタは、デブロッキングフィルタ、ノイズ抑制フィルタ、SAOフィルタ、および適応ループフィルタを含んでもよい。そのようなフィルタは、例に応じて、空間/ピクセル領域(たとえば、再構成済みピクセルブロック上)または周波数領域内で適用され得る。

【0064】

エンコーダとして動作するとき、フィルタリングされた再構成済み画像ブロック、残差ブロック、および/または予測ブロックは、上で説明されているように動き推定で後から使用するためにデコード済みピクチャバッファコンポーネント223に記憶される。デコーダとして動作するとき、デコード済みピクチャバッファコンポーネント223は、再構成済みおよびフィルタリング済みブロックを記憶し、出力ビデオ信号の一部としてディスプレイへ転送する。デコード済みピクチャバッファコンポーネント223は、予測ブロック、残留ブロック、および/または再構成済み画像ブロックを記憶することができる任意のメモリデバイスであってよい。

【0065】

ヘッダフォーマットおよびCABACコンポーネント231は、コーデックシステム200の様々なコンポーネントからデータを受信し、そのようなデータを、デコーダに向けて伝送するためにコーディング済みビットストリーム内にエンコードする。具体的には、ヘッダフォーマットおよびCABACコンポーネント231は、一般制御データおよびフィル

10

20

30

40

50

タ

制御データなどの制御データをエンコードするために、様々なヘッダを生成する。さらに、イントラ予測データおよび動きデータを含む予測データ、さらには量子化済み変換係数データの形態の残差データは、すべてビットストリーム内にエンコードされる。最終的なビットストリームは、元の区分化済みビデオ信号201を再構成するためにデコーダに望ましいすべての情報を含む。そのような情報は、イントラ予測モードインデックステーブル(コードワードマッピングテーブルとも称される)、様々なブロックに対するエンコーディングコンテキストの定義、最もあり得そうなイントラ予測モードの指示、区分情報の指示なども含み得る。そのようなデータは、エントロピーコーディングを採用することによってエンコードされ得る。たとえば、情報は、コンテキスト適応可変長コーディング(CAVLC)、CABAC、シンタクスペースコンテキスト適応2値算術コーディング(SBAC)、確率区間区分化エントロピー(PIPE)コーディング、または別のエントロピーコーディング技術を採用することによってエンコードされ得る。エントロピーコーディングに従って、コーディング済みビットストリームは、別のデバイス(たとえば、ビデオデコーダ)に伝送されるか、または後から伝送するか、もしくは除去するためにアーカイブされ得る。

【0066】

図3は、例示的なビデオエンコーダ300を例示するブロック図である。ビデオエンコーダ300は、コーデックシステム200のエンコーディング機能を実装し、および/または動作方法100のステップ101、103、105、107、および/または109を実装するために採用され得る

。エンコーダ300は、入力ビデオ信号を区分化し、その結果、区分化済みビデオ信号301が得られ、これは区分化済みビデオ信号201に実質的に類似している。次いで、区分化済みビデオ信号301は、エンコーダ300のコンポーネントによってビットストリーム内に圧縮され、エンコードされる。

【0067】

具体的には、区分化済みビデオ信号301は、イントラ予測のためにピクチャ内予測コンポーネント317に転送される。ピクチャ内予測コンポーネント317は、ピクチャ内推定コンポーネント215およびピクチャ内予測コンポーネント217と実質的に類似しているものとしてよい。

区分化済みビデオ信号301は、また、デコード済みピクチャバッファコンポーネント323内の参照ブロックに基づくインター予測のために動き補償コンポーネント321に転送される。

動き補償コンポーネント321は、動き推定コンポーネント221および動き補償コンポーネント219と実質的に類似しているものとしてよい。

ピクチャ内予測コンポーネント317および動き補償コンポーネント321からの予測ブロックおよび残差ブロックは、残差

ブロックの変換および量子化のために、変換および量子化コンポーネント313に転送される。変換および量子化コンポーネント313は、変換スケールリングおよび量子化コンポーネント213と実質的に類似しているものとしてよい。変換済みおよび量子化済み残差ブロックならびに対応する予測ブロックは(関連する制御データとともに)、ビットストリーム内にコーディングするためにエントロピーコーディングコンポーネント331に転送される。エントロピーコーディングコンポーネント331は、ヘッダフォーマットおよびCABAC

コンポーネント231と実質的に類似しているものとしてよい。

【0068】

変換済みおよび量子化済み残差ブロックならびに/または対応する予測ブロックは、参

照ブロックに再構成し動き補償コンポーネント321で使用するために、変換および量子化コンポーネント313から逆変換および量子化コンポーネント329に転送される。逆変換および量子化コンポーネント329は、スケーリングおよび逆変換コンポーネント229と実質的に

類似しているものとしてよい。ループ内フィルタコンポーネント325内のループ内フィルタも、例に応じて、残差ブロックおよび/または再構成済み参照ブロックに適用される。

ループ内フィルタコンポーネント325は、フィルタ制御分析コンポーネント227およびル

ープ内フィルタコンポーネント225と実質的に類似しているものとしてよい。ループ内フィルタコンポーネント325は、ループ内フィルタコンポーネント225に関して説明されている

ように複数のフィルタを備え得る。次いで、フィルタリング済みブロックは、動き補償コンポーネント321により参照ブロックとして使用するためにデコード済みピクチャバッファコンポーネント323に記憶される。デコード済みピクチャバッファコンポーネント323は

、デコード済みピクチャバッファコンポーネント223と実質的に類似しているものとしてよい。

【0069】

図4は、例示的なビデオデコーダ400を例示するブロック図である。ビデオデコーダ400は、コーデックシステム200のデコーディング機能を実装し、および/または動作方法100のステップ111、113、115、および/または117を実装するために採用され得る。デコーダ4

00は、たとえばエンコーダ300からビットストリームを受信し、エンドユーザへの表示のためにビットストリームに基づき再構成済み出力ビデオ信号を生成する。

【0070】

ビットストリームは、エントロピーデコーディングコンポーネント433によって受信される。エントロピーデコーディングコンポーネント433は、CAVLC、CABAC、SBAC、PIPEコ

ーディング、または他のエントロピーコーディング技術などのエントロピーデコーディングスキームを実装するように構成される。たとえば、エントロピーデコーディングコンポーネント433は、ビットストリーム内のコードワードとしてエンコードされた追加のデータを解釈するためのコンテキストを提供するためにヘッダ情報を採用し得る。デコード済み情報は、一般制御データ、フィルタ制御データ、区分情報、動きデータ、予測データ、および残差ブロックからの量子化変換係数などの、ビデオ信号をデコードするための任意の望ましい情報を含む。量子化済み変換係数は、残差ブロック内への再構成のために逆変換および量子化コンポーネント429に転送される。逆変換および量子化コンポーネント429

は、逆変換および量子化コンポーネント329と類似しているものとしてよい。

【0071】

再構成済み残差ブロックおよび/または予測ブロックは、イントラ予測演算に基づき画像ブロックに再構成するためにピクチャ内予測コンポーネント417に転送される。ピクチャ内予測コンポーネント417は、ピクチャ内推定コンポーネント215およびピクチャ内予測

コンポーネント217と類似しているものとしてよい。具体的には、ピクチャ内予測コンポーネント417は、フレーム内の参照ブロックの位置を特定するために予測モードを採用し、残差ブロックをその結果に適用してイントラ予測画像ブロックを再構成する。再構成済みイントラ予測画像ブロックおよび/または残差ブロックならびに対応するインター予測データは、それぞれデコード済みピクチャバッファコンポーネント223およびループ内フィルタコンポーネント225に実質的に類似しているものとしてよい、ループ内フィルタコ

10

20

30

40

50

ンポーネント425を介してデコード済みピクチャバッファコンポーネント423に転送される

。ループ内フィルタコンポーネント425は、再構成済み画像ブロック、残差ブロックおよび/または予測ブロックをフィルタリングし、そのような情報は、デコード済みピクチャバッファコンポーネント423に記憶される。デコード済みピクチャバッファコンポーネント423からの再構成済み画像ブロックは、インター予測のために動き補償コンポーネント421に転送される。動き補償コンポーネント421は、動き推定コンポーネント221および/または動き補償コンポーネント219と実質的に類似しているものとしてよい。具体的には、動き補償コンポーネント421は、予測ブロックを生成するために参照ブロックからの動きベクトルを採用し、残差ブロックをその結果に適用して、画像ブロックを再構成する。また、結果として得られる再構成済みブロックは、ループ内フィルタコンポーネント425を介してデコード済みピクチャバッファコンポーネント423に転送され得る。デコード済みピクチャバッファコンポーネント423は、区分情報を介してフレーム内に再構成され得る、追加の再構成済み画像ブロックを記憶することを続ける。そのようなフレームは、シーケンスの中に置かれてもよい。シーケンスは、再構成済み出力ビデオ信号としてディスプレイへ出力される。

【0072】

図5は、例示的なHRD500を例示する概略図である。HRD500は、コーデックシステム200お

よび/またはエンコーダ300などのエンコーダにおいて採用され得る。HRD500は、方法100

のステップ109で作成されたビットストリームを、ビットストリームがデコーダ400などの

デコーダに転送される前にチェックするものとしてよい。いくつかの例において、ビットストリームは、ビットストリームがエンコードされる時、HRD500を通じて連続的に転送

され得る。ビットストリームの一部が関連付けられている制約条件に適合することに失敗した場合、HRD500はそのような失敗をエンコーダに知らせて、エンコーダに異なるメカニ

ズムによるビットストリームの対応するセクションの再エンコードを行わせることができる。

【0073】

HRD500は、仮想ストリームスケジューラ(HSS)541を備える。HSS541は、仮想配信メカニ

ズムを実行するように構成されているコンポーネントである。仮想配信メカニズムは、HRD500に入力されるビットストリーム551のタイミングおよびデータフローに関してビットストリームまたはデコーダの適合性をチェックするために使用される。たとえば、HSS541は、エンコーダから出力されたビットストリーム551を受信し、ビットストリーム551上の

適合性テストプロセスを管理するものとしてよい。特定の例において、HSS541は、コーディング済みピクチャがHRD500を通過して移動するレートを制御し、ビットストリーム551が

非適合データを含まないことを検証することができる。

【0074】

HSS541は、ビットストリーム551を事前定義されたレートでCPB543に転送することができる。

HRD500は、デコーディングユニット(DU)553においてデータを管理し得る。DU553は

、アクセスユニット(AU)またはAUのサブセット、および関連付けられている非ビデオコーディングレイヤ(VCL)ネットワーク抽象化レイヤ(NAL)ユニットである。具体的には、AU

10

20

30

40

50

は

、出力時間に関連付けられている1つまたは複数のピクチャを含む。たとえば、AUは、単一レイヤビットストリーム内に単一ピクチャを含み、マルチレイヤビットストリーム内に各レイヤに対するピクチャを含み得る。AUの各ピクチャは、各々対応するVCL NALユニットに含まれるスライスに分割され得る。したがって、DU553は、1つまたは複数のピクチャ

、ピクチャの1つまたは複数のスライス、またはそれらの組合せを含み得る。また、AU/DU

、ピクチャ、および/またはスライスをデコードするために使用されるパラメータは、非VCL NALユニットに含まれ得る。それゆえに、DU553は、DU553内でVCL NALユニットをデコ

10

ードすることをサポートするために必要なデータを含む非VCL NALユニットを含む。CPB54

3は、HRD500内の先入れ先出しバッファである。CPB543は、デコーディング順序でビデオ

データを含むDU553を収容する。CPB543は、ビットストリーム適合性検証時に使用するた

めのビデオデータを記憶する。

【0075】

CPB543は、DU553をデコーディングプロセスコンポーネント545に転送する。デコーディ

20

ングプロセスコンポーネント545は、VVC規格に適合するコンポーネントである。たとえば

、デコーディングプロセスコンポーネント545は、エンドユーザによって採用されるデコーダ400をエミュレートし得る。デコーディングプロセスコンポーネント545は、例示的な

エンドユーザのデコーダによって達成され得るレートでDU553をデコードする。デコーディングプロセスコンポーネント545が、CPB543のオーバーフローを防ぐ(またはバッファの

アンダーランを防ぐ)ために十分に速くDU553をデコードできない場合、ビットストリーム551は規格に適合していないので、再エンコードされるべきである。

30

【0076】

デコーディングプロセスコンポーネント545は、DU553をデコードし、デコード済みDU55

5を作成する。デコード済みDU555は、デコード済みピクチャを収容する。デコード済みDU

555は、DPB547に転送される。DPB547は、デコード済みピクチャバッファコンポーネント2

23、323、および/または423と実質的に類似しているものとしてよい。インター予測をサポートするために、デコード済みDU555から得られる参照ピクチャ556としての使用のマ

40

ークを付けられたピクチャは、さらなるデコードをサポートするためにデコーディングプロセスコンポーネント545に返される。DPB547は、デコード済みビデオシーケンスを一連の

ピクチャ557として出力する。ピクチャ557は、エンコーダによってビットストリーム551

にエンコードされたピクチャを一般的にミラーリングする再構成済みピクチャである。

【0077】

ピクチャ557は、出力クロッピングコンポーネント549に転送される。出力クロッピング

50

コンポーネント549は、適合性クロッピングウィンドウをピクチャ557に適用するように構成

成される。この結果、出力クロッピング済みピクチャ559が得られる。出力クロッピング済みピクチャ559は、完全に再構成されたピクチャである。したがって、出力クロッピング済みピクチャ559は、ビットストリーム551をデコードした後にエンドユーザに見える

である。それゆえに、エンコーダ側では出力クロッピング済みピクチャ559をレビューしてエンコーディングが満足のいくものであることを確認することができる。

【0078】

HRD500は、ビットストリーム551内のHRDパラメータに基づき初期化される。たとえば、

HRD500は、VPS、SPS、および/またはSEIメッセージからHRDパラメータを読み取るものと

してよい。次いで、HRD500は、そのようなHRDパラメータ内の情報に基づいて、ビットス

トリーム551に対して適合性テスト動作を実行し得る。具体例として、HRD500は、HRDパラ

メータから1つまたは複数のCPB配信スケジュールを決定し得る。配信スケジュールは、CPBおよび/またはDPBなどのメモリロケーションへのおよび/またはメモリロケーションからのビデオデータの配信に対するタイミングを指定する。したがって、CPB配信スケジュールは、AU、DU553、および/またはピクチャの、CPB543への/からの配信に対するタイ

ミングを指定する。HRD500は、CPB配信スケジュールに類似しているDPB547にDPB配信スケジュー

ールを採用し得ることに留意されたい。

【0079】

ビデオは、様々なレベルのハードウェア能力を有するデコーダによる使用のために、さらには様々なネットワーク条件に関して、異なるレイヤおよび/またはOLSにコーディングされ得る。CPB配信スケジュールは、これらの問題を反映するように選択される。したがって、上位レイヤサブビットストリームは、最適なハードウェアおよびネットワーク条件について指定され、したがって、上位レイヤは、CPB543内の大量のメモリおよびDPB547に

向かうDU553の転送に対する短い遅延を採用する1または複数のCPB配信スケジュールを受

信し得る。同様に、下位レイヤサブビットストリームは、限られたデコーダハードウェア能力および/または劣悪なネットワーク条件について指定される。したがって、下位レイヤは、CPB543内の少量のメモリおよびDPB547に向かうDU553の転送のためのより長い遅延

を採用する1つまたは複数のCPB配信スケジュールを受信し得る。次いで、OLS、レイヤ、サブレイヤ、またはそれらの組合せは、対応する配信スケジュールに従ってテストされ、その結果得られるサブビットストリームが、サブビットストリームに対して期待される条件下で正しくデコードされ得ることを確実にすることができる。したがって、ビットストリーム551内のHRDパラメータは、CPB配信スケジュールを示し、さらにはHRD500がCPB配信

スケジュールを決定し、CPB配信スケジュールに対応するOLS、レイヤ、および/またはサブレイヤに相関させることを可能にする十分なデータを含むことができる。

【0080】

図6は、レイヤ間予測621を行うように構成されている例示的なマルチレイヤビデオシーケンス600を例示する概略図である。マルチレイヤビデオシーケンス600は、たとえば方

10

20

30

40

50

法

100に従って、コーデックシステム200および/またはエンコーダ300などのエンコーダによ

ってエンコードされ、コーデックシステム200および/またはデコーダ400などのデコーダによってデコードされ得る。さらに、マルチレイヤビデオシーケンス600は、HRD500な

どのHRDによって規格適合性に関してチェックされ得る。マルチレイヤビデオシーケンス600は、コーディング済みビデオシーケンス内のレイヤに対する例示的なアプリケーションを示すために含まれている。マルチレイヤビデオシーケンス600は、レイヤN 631およびレイヤN+1 632などの、複数のレイヤを採用する任意のビデオシーケンスである。

【0081】

一例において、マルチレイヤビデオシーケンス600は、レイヤ間予測621を採用し得る。

レイヤ間予測621は、異なるレイヤ内のピクチャ611、612、613、および614とピクチャ615

、616、617、および618との間に適用される。図示されている例では、ピクチャ611、612

、613、および614は、レイヤN+1 632の一部であり、ピクチャ615、616、617、および618

は、レイヤN 631の一部である。レイヤN 631および/またはレイヤN+1 632などのレイヤは

、類似のサイズ、品質、解像度、信号対雑音比、能力などの特性の類似の値にすべて関連付けられるピクチャのグループである。レイヤは、VCL NALユニットおよび関連付けられている非VCL NALユニットのセットとして正式に定義され得る。VCL NALユニットは、ピク

チャのコーディング済みスライスなどの、ビデオデータを含むようにコーディングされたNALユニットである。非VCL NALユニットは、ビデオデータをデコードすること、適合性

チェックの実行、または他の動作をサポートするシンタックスおよび/またはパラメータなどの非ビデオデータを含むNALユニットである。

【0082】

図示されている例では、レイヤN+1 632は、レイヤN 631よりも大きい画像サイズに関連

付けられている。したがって、レイヤN+1 632内のピクチャ611、612、613、および614は

、この例では、レイヤN 631内のピクチャ615、616、617、および618よりも大きいピクチャ

サイズ(たとえば、より大きい高さおよび幅、したがって、より多くのサンプル)を有している。しかしながら、そのようなピクチャは、他の特性によってレイヤN+1 632とレイヤN 631との間で分離され得る。レイヤN+1 632およびレイヤN 631の2つのレイヤのみが図

示されているが、ピクチャのセットは、関連する特性に基づき任意の数のレイヤに分離され得る。レイヤN+1 632およびレイヤN 631は、レイヤIdによっても示され得る。レイヤId

は、ピクチャに関連付けられているデータの項目であり、ピクチャが示されているレイヤの一部であることを示す。したがって、各ピクチャ611~618は、対応するレイヤIdに関連

付けられており、それによりどのレイヤN+1 632またはレイヤN 631が対応するピクチャを

10

20

30

40

50

含むかを示し得る。たとえば、レイヤIdは、NALユニットヘッダレイヤ識別子(nuh_layer_

id)を含んでもよく、これは、NALユニット(たとえば、レイヤ内のピクチャのスライスおよび/またはパラメータを含む)を含むレイヤの識別子を指定するシンタックス要素である。レイヤN 631などの、より低い品質/ビットストリームサイズに関連付けられているレイヤは、一般的に、より低いレイヤIdを割り当てられ、下位レイヤと称される。さらに、レイヤN+1 632などの、より高い品質/ビットストリームサイズに関連付けられているレイヤは、一般的に、より高いレイヤIdを割り当てられ、上位レイヤと称される。

【0083】

異なるレイヤ631~632のピクチャ611~618は、代替的手段で表示されるように構成され

10

る。具体例として、デコーダは、より小さいピクチャが望ましい場合に現在の表示時間にピクチャ615をデコードして表示し得るか、またはデコーダは、より大きいピクチャが望ましい場合に現在の表示時間にピクチャ611をデコードして表示し得る。それゆえに、上位レイヤN+1 632におけるピクチャ611~614は、(ピクチャサイズの差にかかわらず)下位

レイヤN 631における対応するピクチャ615~618と実質的に同じ画像データを含む。具体的には、ピクチャ611は、ピクチャ615と実質的に同じ画像データを含み、ピクチャ612

はピクチャ616と実質的に同じ画像データを含む、などである。

20

【0084】

ピクチャ611~618は、同じレイヤN 631またはN+1 632内の他のピクチャ611~618を参照

してコーディングされ得る。同じレイヤ内の別のピクチャを参照してピクチャをコーディングすると、その結果、インター予測623が得られる。インター予測623は、実線の矢印で

示されている。たとえば、ピクチャ613は、レイヤN+1 632内のピクチャ611、612、および

/または614のうちの1つまたは2つを参照として使用してインター予測623を採用することによってコーディングされるものとしてよく、1つのピクチャは、一方向インター予測のために参照され、および/または2つのピクチャは、双方向インター予測のために参照される。たとえば、ピクチャ617は、レイヤN 631内のピクチャ615、616、および/または618の

30

うちの1つまたは2つを参照として使用してインター予測623を採用することによってコーディングされるものとしてよく、1つのピクチャは、一方向インター予測のために参照され、および/または2つのピクチャは、双方向インター予測のために参照される。ピクチャが、インター予測623を実行するとき、同じレイヤ内の別のピクチャに対する参照として使用されるとき、ピクチャは、参照ピクチャと称され得る。たとえば、ピクチャ612は、インター予測623に従ってピクチャ613をコーディングするために使用される参照ピクチャ

40

であってよい。インター予測623は、マルチレイヤコンテキストにおけるレイヤ内予測とも称され得る。それゆえに、インター予測623は、参照ピクチャおよび現在のピクチャが同じレイヤ内にある場合の現在のピクチャとは異なる参照ピクチャ内の示されたサンプルを参照することによって現在のピクチャのサンプルをコーディングするメカニズムである。

【0085】

ピクチャ611~618は、異なるレイヤ内の他のピクチャ611~618を参照することによって

もコーディングされ得る。このプロセスは、レイヤ間予測621として知られており、破線の矢印によって示されている。レイヤ間予測621は、現在のピクチャと参照ピクチャとが

50

異なるレイヤ内にあり、したがって異なるレイヤIDを有する参照ピクチャ内の示されているサンプルを参照することによって現在のピクチャのサンプルをコーディングするメカニズムである。たとえば、下位レイヤN 631内のピクチャは、上位レイヤN+1 632における対

応するピクチャをコーディングするための参照ピクチャとして使用され得る。具体例として、ピクチャ611は、レイヤ間予測621に従ってピクチャ615を参照することによってコー

ディングされ得る。そのような場合に、ピクチャ615は、レイヤ間参照ピクチャとして使用される。レイヤ間参照ピクチャは、レイヤ間予測621に使用される参照ピクチャである。ほとんどの場合において、レイヤ間予測621は、ピクチャ611などの現在のピクチャが

10

同じAUに含まれ、ピクチャ615などの下位レイヤにあるレイヤ間参照ピクチャのみを使用できるように制約される。複数のレイヤ(たとえば、2つよりも多い)が利用可能であるとき、レイヤ間予測621は、現在のピクチャよりも低いレベルの複数のレイヤ間参照ピクチャに基づき現在のピクチャをエンコード/デコードすることができる。

【0086】

ビデオエンコーダは、インター予測623およびレイヤ間予測621の多くの異なる組合せお

よび/または順列を介してピクチャ611~618をエンコードするためにマルチレイヤビデオシーケンス600を採用することができる。たとえば、ピクチャ615は、イントラ予測に従っ

20

てコーディングされ得る。次いで、ピクチャ616~618は、ピクチャ615を参照ピクチャと

して使用することによってインター予測623に従ってコーディングされ得る。さらに、ピクチャ611は、ピクチャ615をレイヤ間参照ピクチャとして使用することによってレイヤ間

予測621に従ってコーディングされ得る。次いで、ピクチャ612~614は、ピクチャ611を参

照ピクチャとして使用することによってインター予測623に従ってコーディングされ得る。それゆえに、参照ピクチャは、異なるコーディングメカニズムのための単一のレイヤ参照ピクチャとレイヤ間参照ピクチャの両方として働き得る。下位レイヤN 631ピクチャに基づき上位レイヤN+1 632ピクチャをコーディングすることによって、上位レイヤN+1 632

30

は、インター予測623およびレイヤ間予測621よりもかなり低いコーディング効率を有する

、イントラ予測を採用することを回避することができる。それゆえに、イントラ予測のコーディング効率の悪さは、最小/最低品質のピクチャに限定することができ、したがって、最小量のビデオデータをコーディングすることに限定することができる。参照ピクチャおよび/またはレイヤ間参照ピクチャとして使用されるピクチャは、参照ピクチャリスト構造に含まれる参照ピクチャリストのエントリにおいて示され得る。

40

【0087】

レイヤN+1 632およびレイヤN 631などの、レイヤは、出力レイヤセット(OLS)に含まれ得ることに留意されたい。OLSは、少なくとも1つのレイヤが出力レイヤである、1つまたは複数のレイヤのセットである。たとえば、レイヤN 631は、第1のOLSに含まれ、レイヤN

631およびレイヤN-1 632は、両方とも、第2のOLSに含まれ得る。これは、デコーダ側の条件に応じて、異なるOLSが異なるデコーダに送信されることを可能にする。たとえば、サブビットストリーム抽出プロセスは、ターゲットOLSがデコーダに送信される前に、マルチレイヤビデオシーケンス600からターゲットOLSに無関係なデータを除去することができる。それゆえに、マルチレイヤビデオシーケンス600のエンコード済みコピーは、エン

50

コーダ(または対応するコンテンツサーバ)に記憶され、様々なOLSが、要求があったときに抽出され異なるデコーダに送信され得る。

【0088】

サイマルキャストレイヤは、レイヤ間予測621を採用しないレイヤである。たとえば、レイヤN+1 632は、レイヤ間予測621に基づきレイヤN 631を参照することによってコーデ

ィングされる。しかしながら、レイヤ631は、別のレイヤを参照することによってコーディングされることはない。それゆえに、レイヤ631は、サイマルキャストレイヤである。マルチレイヤビデオシーケンス600などの、スケーラブルビデオシーケンスは、一般的に、ベースレイヤと、ベースレイヤの何らかの特性を強化する1つまたは複数のエンハンスメントレイヤとを採用する。図6において、レイヤN 631はベースレイヤである。ベースレイヤは、一般的に、サイマルキャストレイヤとしてコーディングされる。図6は例示的であり、非限定的であるが、それは、複数のレイヤを有するビデオシーケンスは、依存関係の多くの異なる組合せ/順列を使用し得るからであることにも留意されたい。ビットストリームは、任意の数のレイヤを含み、任意の数のそのようなレイヤは、サイマルキャストレイヤであり得る。たとえば、レイヤ間予測621は、完全に省略することができ、その場合、すべてのレイヤはサイマルキャストレイヤである。別の例として、マルチビューアプリケーションは、2つまたはそれ以上の出力レイヤを表示する。それゆえに、マルチビューアプリケーションは、一般的に、サイマルキャストレイヤである2つまたはそれ以上のベースレイヤを含み、各ベースレイヤに対応するエンハンスメントレイヤを含み得る。

【0089】

サイマルキャストレイヤは、レイヤ間予測621を使用するレイヤと異なる方式で処理され得る。たとえば、レイヤ間予測621を使用するレイヤをコーディングするとき、エンコーダは、デコードをサポートするために、レイヤの数さらにはレイヤ間の依存関係も示すべきである。しかしながら、サイマルキャストレイヤについては、そのような情報は省略できる。たとえば、レイヤN+1 632およびレイヤN 631の構成は、以下でより詳細に説明さ

れるようにVPSにおいて示され得る。しかしながら、レイヤN 631は、そのような情報なしでデコードされ得る。それゆえに、VPSは、レイヤN 631のみがデコーダに伝送されるべきとき、対応するビットストリームから除去され得る。しかしながら、これは、ビットストリーム内に残っているパラメータがVPSを参照している場合には、エラーを生じさせる可能性がある。さらに、各レイヤは、SPSによって記述され得る。サイマルキャストレイヤに対するSPSは、SPSがVPSとともに誤って除去されないようにマルチレイヤビットストリーム内にエンコードされるべきである。これらおよび他の問題は、以下でより詳しく説明される。

【0090】

図7は、例示的なビットストリーム700を例示する概略図である。たとえば、ビットストリーム700は、方法100に従ってコーデックシステム200および/またはデコーダ400によっ

てデコードするために、コーデックシステム200および/またはエンコーダ300によって生成され得る。さらに、ビットストリーム700は、マルチレイヤビデオシーケンス600を含み

得る。それに加えて、ビットストリーム700は、HRD500などの、HRDの動作を制御するため

の様々なパラメータを含み得る。このようなパラメータに基づき、HRD500は、デコードす

るためにデコーダに向けて伝送する前に、規格への適合性についてビットストリーム700をチェックすることができる。

【0091】

ビットストリーム700は、VPS711、1つまたは複数のSPS713、複数のピクチャパラメ

10

20

30

40

50

ータ

セット(PPS)715、複数のスライスヘッダ717、および画像データ720を含む。VPS711は、ピ

ットストリーム700全体に関係するデータを含む。たとえば、VPS711は、ビットストリーム700で使用されるOLS、レイヤ、および/またはサブレイヤに関係するデータを含んでもよい。SPS713は、ビットストリーム700に含まれるコーディング済みビデオシーケンス内のすべてのピクチャに共通するシーケンスデータを含む。たとえば、各レイヤは、1つまたは複数のコーディング済みビデオシーケンスを含むものとしてよく、各コーディング済みビデオシーケンスは、対応するパラメータについてSPS713を参照し得る。SPS713に

10

おけるパラメータは、ピクチャのサイズ設定、ビット深度、コーディングツールパラメータ、ビットレート制限などを含むことができる。各シーケンスがSPS713を参照する一方で、いくつかの例では単一のSPS713は複数のシーケンスに対するデータを含むことができることに留意されたい。PPS715は、ピクチャ全体に適用されるパラメータを含む。したがって、ビデオシーケンス内の各ピクチャは、PPS715を参照し得る。各シーケンスがPPS715を

参照する一方で、いくつかの例では単一のPPS715は複数のピクチャに対するデータを含むことができることに留意されたい。たとえば、複数の類似のピクチャは、類似のパラメータに従ってコーディングされ得る。そのような場合、単一のPPS715は、そのような類似のピクチャに対するデータを含み得る。PPS715は、対応するピクチャにおけるスライス、量子化

20

【0092】

スライスヘッダ717は、ピクチャ内の各スライスに特有のパラメータを含む。したがって、ビデオシーケンス内のスライス毎に1つのスライスヘッダ717があり得る。スライスヘッダ717は、スライスタイプ情報、ピクチャ順序カウント(POC)、参照ピクチャリスト、予測重み、タイルエントリポイント、またはデブロッキングパラメータなどを含み得る。いくつかの例では、ビットストリーム700は、ピクチャヘッダも含むものとしてよく、これは、単一のピクチャ内のすべてのスライスに適用されるパラメータを含むシンタックス構造であることに留意されたい。このような理由から、ピクチャヘッダおよびスライスヘッダ717は、いくつかの文脈において入れ換えて使用され得る。たとえば、いくつかのパラメータは、そのようなパラメータがピクチャ内のすべてのスライスに共通であるかどうかに応じてスライスヘッダ717とピクチャヘッダとの間で移動され得る。

30

【0093】

画像データ720は、インター予測、レイヤ間予測、および/またはイントラ予測に従ってエンコードされたビデオデータ、さらには対応する変換済みおよび量子化済み残差データを含む。たとえば、画像データ720は、レイヤ723および724、ピクチャ725および726、お

よび/またはスライス727および728を含み得る。レイヤ723および724は、nuh_layer_id 73

2などのレイヤIDによって示されているような指定された特性(たとえば、共通の解像度、フレームレート、画像サイズなど)を共有するVCL NALユニット741と、関連付けられている非VCL NALユニット742とのセットである。たとえば、レイヤ723は、同じnuh_layer_id

40

732を共有するピクチャのセット725を含んでもよい。同様に、レイヤ724は、同じnuh_layer_id

er_id 732を共有するピクチャのセット726を含んでもよい。レイヤ723および724は、実質

的に類似しているものとしてよいが、異なるコンテンツを含んでいてもよい。たとえば、レイヤ723および724は、それぞれ、図6からのレイヤN 631およびレイヤN+1 632を含んで

50

もよい。それゆえに、コーディング済みビットストリーム700は、複数のレイヤ723および

724を含んでもよい。説明を明確にするために2つのレイヤ723および724のみが示されてい

るが、任意の数のレイヤ723および724がビットストリーム700に含まれ得る。

【0094】

nuh_layer_id 732は、少なくとも1つのNALユニットを含むレイヤ723および/または724

の識別子を指定するシンタックス要素である。たとえば、ベースレイヤとして知られる最低品質のレイヤは、より高い品質のレイヤに対してnuh_layer_id 732の値を高くした最低

10

値のnuh_layer_id 732を含み得る。したがって、下位レイヤは、nuh_layer_id 732のより

小さい値を有するレイヤ723または724であり、上位レイヤは、nuh_layer_id 732のより大

きい値を有するレイヤ723または724である。レイヤ723および724のデータは、nuh_layer_

id 732に基づき関連する。たとえば、パラメータセットおよびビデオデータは、そのようなパラメータセット/ビデオデータを含む最下位レイヤ723または724に対応するnuh_layer_

20

id 732の値に関連付けられ得る。それゆえに、VCL NALユニット741のセットは、VCL NAL

ユニット741のセットがすべてnuh_layer_id 732の特定の値を有するとき、レイヤ723

および/または724の一部である。

【0095】

ピクチャ725および726のセットからのピクチャは、フレームまたはそのフィールドを

作成するルーマサンプルの配列および/またはクロマサンプルの配列である。たとえば、ピクチャ725および726のセットからのピクチャは、表示のために出力され得るか、または出

30

力する他のピクチャのコーディングをサポートするために使用されるコーディング済み画像である。ピクチャ725および726のセットからのピクチャは、実質的に類似しているが、

ピクチャ725のセットはレイヤ723に含まれ、ピクチャ726のセットはレイヤ724に含まれる

。ピクチャ725および726のセットからのピクチャは、それぞれ、1つまたは複数のスライス727および728を含む。スライス727/728は、VCL NALユニット741などの、単一のNALユニ

ットに排他的に含まれるピクチャの整数個数の完全なタイルまたは(たとえば、タイル内の)整数個数の連続する完全なコーディングツリーユニット(CTU)行として定義され得る。スライス727とスライス728とは、スライス727がピクチャ725およびレイヤ723に含まれ、

40

スライス728がピクチャ726およびレイヤ724に含まれることを除き、実質的に類似してい

る。スライス727/728は、CTUおよび/またはコーディングツリーブロック(CTB)にさらに分

割される。CTUは、コーディングツリーによって区分化され得る事前定義されたサイズのサンプルのグループである。CTBは、CTUのサブセットであり、CTUのルーマ成分またはク

ロマ成分を含む。CTU/CTBは、コーディングツリーに基づきコーディングブロックにさら

50

に分割される。次いで、コーディングブロックは、予測メカニズムに従ってエンコード/デコードされ得る。

【0096】

コーディング済みレイヤビデオシーケンス(CLVS)743およびCLVS744は、それぞれ、同じnuh_layer_id 732値を有する、コーディング済みピクチャ725およびピクチャ725のシーケンスである。たとえば、CLVS743および/または744は、それぞれ、単一のレイヤ723および/または724に含まれるピクチャ725および/または726のシーケンスであってよい。したがって、それぞれ、CLVS743はレイヤ723のピクチャ725のすべてを含み、CLVS744はレイヤ724のピクチャ726のすべてを含んでいる。CLVS743および744の各々は、対応するSPS713を参照している。例に応じて、CLVS743および744は同じSPS 713を参照するか、またはCLVS743および744は各々異なるSPS713を参照し得る。

10

【0097】

ビットストリーム700は、NALユニットのシーケンスとしてコーディングされうる。NAL

ユニットは、ビデオデータおよび/またはサポートしているシンタックスのためのコンテナである。NALユニットは、VCL NALユニット741または非VCL NALユニット742とすることができる。

VCL NALユニット741は、画像データ720および関連付けられているスライスヘ

ッダ717などの、ビデオデータを含むようにコーディングされたNALユニットである。具

20

体例として、各スライス727、728および関連付けられているスライスヘッダ717は、単一のV

CL NALユニット741にエンコードされ得る。非VCL NALユニット742は、ビデオデータをデ

コードすること、適合性チェックの実行、または他の動作をサポートするシンタックスおよび/またはパラメータなどの非ビデオデータを含むNALユニットである。たとえば、非VCL NALユニット742は、VPS711、SPS713、PPS715、ピクチャヘッダ、または他のサポートし

ているシンタックスを含むことができる。それゆえに、ビットストリーム700は、一連のVCL NALユニット741および非VCL NALユニット742である。各NALユニットは、nuh_layer_id

30

732を含み、これは、エンコーダまたはデコーダが、どのレイヤ723または724が対応する

NALユニットを含むかを決定することを可能にする。

【0098】

複数のレイヤ723および724を含むビットストリーム700は、エンコードされ、デコーダ

によって要求されるまで記憶されうる。たとえば、デコーダは、レイヤ723、レイヤ724、

40

ならびに/または複数のレイヤ723および724を含むOLSを要求することができる。特定の例

では、レイヤ723はベースレイヤであり、レイヤ724はエンハンスメントレイヤである。追

加のレイヤもまた、ビットストリーム700において採用され得る。エンコーダおよび/またはコンテンツサーバは、要求された出力レイヤをデコードするために必要なレイヤ723および/または724のみをデコーダに送信すべきである。たとえば、レイヤが異なるピクチャサイズに使用されるとき、最大のピクチャサイズを要求するデコーダは、レイヤ723と724

の両方を有するビットストリーム700全体を受信し得る。最小のピクチャサイズを要求す

50

るデコーダは、レイヤ723のみを受信し得る。中間のピクチャサイズを要求するデコーダは、レイヤ723および他の中間レイヤを受信し得るが、最上位レイヤ724を受信せず、したがって、ビットストリーム全体を受信し得ない。同じアプローチは、フレームレート、ピクチャ解像度などの他のレイヤ特性にも使用され得る。

【0099】

サブビットストリーム抽出プロセス729は、ビットストリーム700からサブビットストリーム701を抽出して、上で説明されている機能をサポートするために採用される。サブビットストリーム701は、ビットストリーム700からのNALユニット(たとえば、非VCL NAL

10

ユニット742およびVCL NALユニット741)のサブセットである。具体的には、サブビットスト

リーム701は、1つまたは複数のレイヤに関するデータを含み得るが、他のレイヤに関するデータを含み得ない。図示されている例では、サブビットストリーム701は、レイヤ723に関するデータを含むが、レイヤ724に関するデータを含まない。したがって、サブビットストリーム701は、SPS713、PPS715、スライスヘッダ717、およびレイヤ723、CLV

S743、ピクチャ725、およびスライス727を含む画像データ720を含む。サブビットスト

20

リーム抽出プロセス729は、nuh_layer_id 732に基づきNALユニットを除去する。たとえば、

上位レイヤ724のみに関連付けられているVCL NALユニット741および非VCL NALユニット74

2は、より高いnuh_layer_id 732値を含み、したがって、より高いnuh_layer_id 732値を

有するすべてのNALユニットを除去することで、下位レイヤ723および関連するパラメータ

を抽出する。各NALユニットは、サブビットストリーム抽出プロセス729をサポートするた

30

めに、NALユニットを含む最下位レイヤのnuh_layer_id 732以下であるnuh_layer_id 732

値を含む。ビットストリーム700およびサブビットストリーム701は各々、一般的にビット

ストリームと称され得ることに留意されたい。

【0100】

図示されている例では、サブビットストリーム701は、サイマルキャストレイヤ(たとえば、ベースレイヤ)を含む。上で指摘されているように、サイマルキャストレイヤは、レイヤ間予測を使用しない任意のレイヤである。VPS711は、レイヤ723および724の構成を記

40

述するデータを含む。しかしながら、このデータは、レイヤ723などの、サイマルキャストレイヤをデコードするためには必要ない。それゆえに、サブビットストリーム抽出プロセス729は、サイマルキャストレイヤを抽出するときに高められたコーディング効率をサポートするためにVPS711を除去する。これは、いくつかのビデオコーディングシステムに問題を引き起こし得る。具体的には、SPS713のいくつかのパラメータは、VPS711を参照し

得る。VPS711が除去されたとき、デコーダおよび/またはHRDは、そのようなパラメータに

よって参照されるデータがもはや存在しないので、そのようなパラメータを解決することができない場合がある。この結果、HRDでサイマルキャストレイヤに対する適合性テスト

50

を実行するときにエラーが発生する可能性がある。代替的に、この結果、デコーダで表示するためにサイマルキャストレイヤが伝送されるときにデコーダにおいて予測不可能なエラーが生じ得る。さらに、各SPS713を正しく識別できない場合、結果として、サブビットストリーム抽出プロセス729によってサイマルキャストレイヤに対してVPS711が除去されるときにSPS713を不適切に除去することになり得る。

【 0 1 0 1 】

本開示では、これらのエラーに対処する。具体的には、SPS713は、CLVS743を含むレイ

ヤ723がレイヤ間予測を使用しない(サイマルキャストレイヤである)ときにSPS713を参照するCLVS743と同じnuh_layer_id 732を含むように制約される。さらに、SPS713は、

10

CLVS744を含むレイヤ724がレイヤ間予測を使用する(サイマルキャストレイヤではない)ときにCLVS744のnuh_layer_id 732以下であるnuh_layer_id 732を有し得る。この方法で、CLVS74

3および743の両方が、任意選択で、同じSPS713を参照し得る。さらに、SPS713は、SPS713

がCLVS743/レイヤ723と同じnuh_layer_id 732を含むので、サイマルキャストレイヤ(たと

えば、レイヤ723)に対してサブビットストリーム抽出プロセス729によって除去されない。

20

【 0 1 0 2 】

さらに、SPS713は、sps_video_parameter_set_id 731を含む。sps_video_parameter_set

id 731は、SPS713によって参照されるVPS711のIDを指定するシンタックス要素である。

具体的には、VPS711は、vps_video_parameter_set_id 735を含み、これは、他のシンタッ

クス要素/構造によって参照されるVPS711のIDを提供するシンタックス要素である。VPS71

1が存在するとき、sps_video_parameter_set_id 731は、vps_video_parameter_set

30

id 735の値に設定される。しかしながら、SPS713がサイマルキャストレイヤに使用されるとき、sps_video_parameter_set_id 731は、0に設定される。別の言い方をすれば、sps_v

ideo_parameter_set_id 731は、0より大きいとき、SPS713によって参照されるVPS711について

、vps_video_parameter_set_id 735の値を指定する。sps_video_parameter_set_id 731が

0に等しいとき、SPS713は、VPS711を参照せず、SPS713を参照する任意のコーディング済

40

みレイヤビデオシーケンスをデコードするときどのVPS711も参照されることはない。これは、異なるレイヤ(たとえば、サイマルキャストレイヤに対する1つのSPSおよび非サイマルキャストレイヤに対する別のSPS)に別のSPS713を使用するか、またはサブビットスト

リーム抽出プロセス729中、sps_video_parameter_set_id 731の値を変更することによっ

て達成され得る。この方法で、sps_video_parameter_set_id 731は、サブビットストリー

ム抽出プロセス729中にVPS711が除去されるときに利用不可能であるIDを誤って参照する

50

ことがない。

【 0 1 0 3 】

さらに、HRDおよび/またはデコーダによって導出される様々な変数も、VPS711のパラメータを参照する。したがって、そのような変数は、sps_video_parameter_set_id 731が0

に設定されたときに既定値に設定される。これは、マルチレイヤビットストリームに対して正しく動作している間に、そのような変数が、VPS711がサイマルキャストレイヤに対して抽出されるときに使用可能値に適切に解決され得ることを確実にする。たとえば、デコーダおよび/またはHRDは、ビットストリーム700および/またはサブビットストリーム701

に基づきGeneralLayerIdx[i]を導出することができる。GeneralLayerIdx[i]は、対応する

レイヤiのインデックスを指定する導出変数である。それゆえに、GeneralLayerIdx[i]は、現在のレイヤのnuh_layer_id732をGeneralLayerIdx[i]におけるレイヤiとして含めるこ

とによって現在のレイヤのレイヤインデックスを決定するために採用され得る。これは、nuh_layer_idに対応する一般レイヤインデックス(GeneralLayerIdx[nuh_layer_id])とし

て表され得る。したがって、GeneralLayerIdx[nuh_layer_id]は、対応するレイヤに対する現在のレイヤインデックスを示す。このプロセスは、レイヤ724などの、非サイマルキャストレイヤに対しては正しく動作するが、サイマルキャストレイヤ723に対してはエラーを引き起こす可能性がある。したがって、sps_video_parameter_set_id 731が0(サイマ

ルキャストレイヤを示す)であるときにGeneralLayerIdx[nuh_layer_id]は0に設定され、

および/または0であると推論される。

【 0 1 0 4 】

別の例として、VPS711は、VPS独立レイヤフラグ(vps_independent_layer_flag)733を含

み得る。vps_independent_layer_flag733は、レイヤ723および/または724などの、対応す

るレイヤがレイヤ間予測を使用するかどうかを指定する。したがって、vps_independent_

layer_flag[GeneralLayerIdx[nuh_layer_id]]は、インデックスGeneralLayerIdx[nuh_lay

er_id]を有する現在のレイヤがレイヤ間予測を使用するかどうかを指定する。しかしながら、vps_independent_layer_flag733を含むVPS711は、デコーダに送信されるレイヤ723が

サイマルキャストレイヤであるとき、デコーダに送信されない。それゆえに、参照はエラーを引き起こし得る。しかしながら、サイマルキャストレイヤは、レイヤ間予測を使用しない。それゆえに、サイマルキャストレイヤに対するvps_independent_layer_flag733は

、1に等しいと推論することができ、これは、対応するレイヤ723に対してレイヤ間予測が使用されないことを示す。したがって、vps_independent_layer_flag[GeneralLayerIdx[n

uh_layer_id]]は、sps_video_parameter_set_idが0に設定されるときに現在のレイヤに対

してレイヤ間予測が使用されないことを示すために1に設定され/1であると推論される。

この方法で、エラーは、レイヤ723などの、サイマルキャストレイヤの伝送前にビットス

10

20

30

40

50

トリームからVPSが除去されるときに回避される。その結果、エンコーダおよびデコーダの機能が向上する。さらに、コーディング効率は、サイマルキャストレイヤのみを含むビットストリームから不要なVPSを首尾よく除去することによって高められ、これにより、エンコーダとデコーダの両方においてプロセッサ、メモリ、および/またはネットワークシグナリングリソースの使用量を低減する。

【0105】

次に、前述の情報は、本明細書において以下でより詳細に説明される。レイヤード・ビデオ・コーディング(layered video coding)は、スケーラブルビデオコーディングまたはスケーラビリティを有するビデオコーディングとも称される。ビデオコーディングにおけるスケーラビリティは、マルチレイヤコーディング技術を使用することによってサポートされ得る。マルチレイヤビットストリームは、ベースレイヤ(BL)および1つまたは複数のエンハンスメントレイヤ(EL)を含む。スケーラビリティの例は、空間的スケーラビリティ、品質/信号対雑音比(SNR)スケーラビリティ、マルチビュースケーラビリティ、フレームレートスケーラビリティなどを含む。マルチレイヤコーディング技術が使用されるとき、ピクチャまたはその一部は、参照ピクチャを使用することなくコーディングされ(イントラ予測)、同じレイヤ内にある参照ピクチャを参照することによってコーディングされ(インター予測)、および/または他のレイヤ内にある参照ピクチャを参照することによってコーディングされ(レイヤ間予測)得る。現在のピクチャのレイヤ間予測に使用される参照ピクチャは、レイヤ間参照ピクチャ(ILRP)と称される。図6は、異なるレイヤ内のピクチャが異なる解像度を有する空間的スケーラビリティに対するマルチレイヤコーディングの一例を例示している。

【0106】

いくつかのビデオコーディングファミリーは、単一レイヤコーディングのためのプロファイルから分離されたプロファイルにおいてスケーラビリティに対するサポートを提供する。スケーラブルビデオコーディング(SVC)は、空間的、時間的、および品質的なスケーラビリティに対するサポートを提供する高度ビデオコーディング(AVC)のスケーラブルな拡張である。SVCについては、フラグがELピクチャ内の各マクロブロック(MB)においてシグナリングされ、それによりEL MBが下位レイヤからの同一の場所のブロックを使用して予測されるかどうかを示す。同一の場所のブロックからの予測は、テクスチャ、動きベクトル、および/またはコーディングモードを含み得る。SVCの実装形態では、その設計において未修正のAVC実装形態を直接再利用し得ない。SVC ELマクロブロックシンタックスおよびデコーディングプロセスは、AVCシンタックスおよびデコーディングプロセスと異なる。

【0107】

スケーラブルHEVC(SHVC)は、空間的および品質的スケーラビリティに対するサポートを提供するHEVCの拡張である。マルチビューHEVC(MV-HEVC)は、マルチビュースケーラビリティ

に対するサポートを提供するHEVCの拡張である。3D HEVC(3D-HEVC)は、MV-HEVCに比

べて高度であり効率的である3Dビデオコーディングに対するサポートを提供するHEVCの拡張

である。時間的スケーラビリティは、単一レイヤHEVCコーデックの不可欠な部分として含まれ得る。HEVCのマルチレイヤ拡張では、レイヤ間予測に使用されるデコード済みピクチャが、同じAUのみから来て、長期参照ピクチャ(LTRP)として扱われる。そのようなピクチャは、現在のレイヤ内の他の時間的参照ピクチャとともに参照ピクチャリスト内の参照インデックスを割り当てられる。レイヤ間予測(ILP)は、予測ユニット(PU)レベルにおいて、参照ピクチャリスト内のレイヤ間参照ピクチャを参照するように参照インデックスの値を設定することによって達成される。空間的スケーラビリティは、ILRPがエンコードされるかまたはデコードされている現在のピクチャと異なる空間解像度を有するときに参照

10

20

30

40

50

ピクチャまたはその一部を再サンプルする。参照ピクチャ再サンプリングは、ピクチャレベルまたはコーディングブロックレベルのいずれかで実現され得る。

【0108】

VVCは、レイヤード・ビデオ・コーディングもサポートし得る。VVCビットストリームは

、複数のレイヤを含むことができる。レイヤは、すべて互いから独立しているものとしてよい。たとえば、各レイヤは、レイヤ間予測を使用せずにコーディングされ得る。この場合、レイヤは、サイマルキャストレイヤとも呼ばれる。いくつかの場合において、レイヤのいくつかはILPを用いてコーディングされる。VPS内のフラグは、レイヤがサイマルキャストレイヤであるかどうか、一部のレイヤがILPを使用しているかどうかを示すことができる。いくつかのレイヤがILPを使用するとき、レイヤ間のレイヤ依存関係もVPSでシグナリングされる。SHVCおよびMV-HEVCと異なり、VVCはOLSを指定し得ない。OLSは、レイヤの

指定されたセットを含み、レイヤのセット内の1つまたは複数のレイヤは、出力レイヤとなるように指定される。出力レイヤは、出力されるOLSのレイヤである。VVCのいくつかの

実装形態において、レイヤがサイマルキャストレイヤであるとき、ただ1つのレイヤが、デコードおよび出力について選択され得る。VVCのいくつかの実装形態において、任意のレイヤがILPを使用するときすべてのレイヤを含むビットストリーム全体がデコードされるように指定される。さらに、レイヤのうちの特定のレイヤが出力レイヤとして指定される。出力レイヤは、最上位レイヤのみ、すべてのレイヤ、または最上位レイヤ+指示された下位レイヤのセットであると示されてよい。

【0109】

前述の態様は、いくつかのスケラビリティ関係問題を含んでいる。そのようなシステムにおけるスケラビリティ設計は、レイヤ特有のプロファイル、ティア、およびレベル(PTL)、さらにはレイヤ特有のコーディング済みピクチャバッファ(CPB)動作を含む。PTLシグナリング効率は、改善されるべきである。サブレイヤに対するシーケンスレベルHRDパラメータのシグナリング効率は改善されるべきである。DPBパラメータシグナリングは改善されるべきである。いくつかの設計は、単一レイヤビットストリームがVPSを参照することを引き起こす。そのような設計におけるnum_ref_entries[][]の値範囲は、不正であり、デコードに対して予期せぬエラーを引き起こす。このような設計におけるデコーディングプロセスは、サブビットストリーム抽出を伴い、これはデコード実装形態の負担を増す。そのような設計に対する一般デコーディングプロセスは、レイヤ間予測を有する複数のレイヤを含むスケラブルビットストリームに対しては機能し得ない。そのような設計における変数NoOutputOfPriorPicsFlagの値の導出は、ピクチャベースであり、そのよ

うな設計ではAUベースではあり得ない。そのような設計におけるスケラブルネスティングSEIメッセージは、nesting_ols_flagが1に等しいとき、OLSのレイヤではなく、OLSに直

接的に適用するように簡素化されるべきである。非スケラブルネスティングSEIメッセージは、payloadTypeが0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しいとき、0番目のOLSにのみ適用されるように指定され得る。

【0110】

一般に、本開示は、ビデオコーディングにおけるスケラビリティのための様々なアプローチについて説明する。技術の説明は、VVCに基づく。しかしながら、これらの技術は、他のビデオコーデック仕様に基づくレイヤード・ビデオ・コーディングにも適用される。上述の問題の1つまたは複数は、次のように解決され得る。具体的には、この開示は、ビデオコーディングにおける改善されたスケラビリティサポートのための方法を含む。

【0111】

次は、様々な定義例である。OPIは、OLSインデックスおよびTemporalIdの最高値によ

10

20

30

40

50

っ

て識別される、OLSの時間的サブセットであってよい。出力レイヤは、出力されるOLSのレイヤであり得る。OLSは、レイヤのセットであってよく、レイヤのセット内の1つまたは複数のレイヤは、出力レイヤとなるように指定される。OLSレイヤインデックスは、OLSにおけるレイヤのリストへの、OLS内のレイヤの、インデックスであってよい。サブビットストリーム抽出プロセスは、ターゲットOLSインデックスおよびターゲット最高TemporalIdによって決定される、ターゲットセットに属さないビットストリーム内のNALユニットをビットストリームから除去する指定されたプロセスであってよく、出力サブビットストリームは、ターゲットセットに属するビットストリーム内のNALユニットを含む。

【 0 1 1 2 】

例示的なビデオパラメータセットRBSPシンタックスは次のとおりである。

【 0 1 1 3 】

10

20

30

40

50

【表 1 A】

video_parameter_set_rbsp() {	記述子	
vps_video_parameter_set_id	u(4)	
vps_max_layers_minus1	u(6)	
vps_max_sub_layers_minus1	u(3)	
if(vps_max_layers_minus1 > 0 && vps_max_sub_layers_minus1 > 0)		
vps_all_layers_same_num_sub_layers_flag	u(1)	10
if(vps_max_layers_minus1 > 0)		
vps_all_independent_layers_flag	u(1)	
...		
vps_num_ptls	u(8)	
for(i = 0; i < vps_num_ptls; i++) {		
if(i > 0)		
pt_present_flag[i]	u(1)	
if(vps_max_sub_layers_minus1 > 0		20
&& !vps_all_layers_same_num_sub_layers_flag)		
ptl_max_temporal_id[i]	u(3)	
}		
while(!byte_aligned())		
vps_ptl_byte_alignment_zero_bit /* equal to 0 */	u(1)	
for(i = 0; i < vps_num_ptls; i++)		
profile_tier_level(pt_present_flag[i], ptl_max_temporal_id[i])		
for(i = 0; i < TotalNumOls; i++)		
if(NumLayersInOls[i] > 1 && vps_num_ptls > 1)		30
ols_ptl_idx[i]	u(8)	
if(!vps_all_independent_layers_flag)		
vps_num_dpb_params	ue(v)	
if(vps_num_dpb_params > 0) {		
same_dpb_size_output_or_nonoutput_flag	u(1)	
if(vps_max_sub_layers_minus1 > 0)		

10

20

30

40

【 0 1 1 4 】

50

【表 1 B】

vps_sub_layer_dpb_params_present_flag	u(1)	
}		
for(i = 0; i < vps_num_dpb_params; i++) {		
dpb_size_only_flag[i]	u(1)	
if(vps_max_sub_layers_minus1 > 0 && !vps_all_layers_same_num_sub_layers_flag)		
dpb_max_temporal_id[i]	u(3)	10
dpb_parameters(dpb_size_only_flag[i], dpb_max_temporal_id[i], vps_sub_layer_dpb_params_present_flag)		
}		
for(i = 0; i < vps_max_layers_minus1 && vps_num_dpb_params > 1; i++) {		
if(!vps_independent_layer_flag[i])		
layer_output_dpb_params_idx[i]	ue(v)	
if(LayerUsedAsRefLayerFlag[i] && !same_dpb_size_output_or_nonoutput_flag)		20
layer_nonoutput_dpb_params_idx[i]	ue(v)	
}		
general_hrd_params_present_flag	u(1)	
if(general_hrd_params_present_flag) {		
num_units_in_tick	u(32)	
time_scale	u(32)	
general_hrd_parameters()		30
}		
vps_extension_flag	u(1)	
if(vps_extension_flag)		
while(more_rbsp_data())		
vps_extension_data_flag	u(1)	
rbsp_trailing_bits()		
}		

10

20

30

40

【 0 1 1 5 】

例示的なシーケンスパラメータセットRBSPシンタックスは次のとおりである。

【 0 1 1 6 】

50

【表 2 A】

seq_parameter_set_rbsp() {	記述子	
sps_decoding_parameter_set_id	u(4)	
sps_video_parameter_set_id	u(4)	
sps_max_sub_layers_minus1	u(3)	
sps_reserved_zero_4bits	u(4)	10
sps_ptl_dpb_present_flag	u(1)	
if(sps_ptl_dpb_present_flag)		
profile_tier_level(1, sps_max_sub_layers_minus1)		
gdr_enabled_flag	u(1)	
sps_seq_parameter_set_id	ue(v)	
chroma_format_idc	ue(v)	
...		20
log2_max_pic_order_cnt_lsb_minus4	ue(v)	
poc_msb_in_rap_pics_flag	u(1)	
if(poc_msb_in_rap_pics_flag > 0)		
poc_msb_len_minus1	ue(v)	
if(sps_max_sub_layers_minus1 > 0)		
sps_sub_layer_dpb_params_flag	u(1)	
if(sps_ptl_dpb_present_flag)		30
dpb_parameters(0, sps_max_sub_layers_minus1, sps_sub_layer_dpb_params_flag)		
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {		
sps_max_dec_pic_buffering_minus1[i]	ue(v)	

【 0 1 1 7 】

40

50

【表 2 B】

sps_max_num_reorder_pics[i]	ue(v)	
sps_max_latency_increase_plus1[i]	ue(v)	
}		
long_term_ref_pics_flag	u(1)	
...		
sps_scaling_list_enabled_flag	u(1)	10
general_hrd_parameters_present_flag	u(1)	
if(general_hrd_parameters_present_flag) {		
num_units_in_tick	u(32)	
time_scale	u(32)	
sub_layer_cpb_parameters_present_flag	u(1)	
if(sub_layer_cpb_parameters_present_flag)		
general_hrd_parameters(0, sps_max_sub_layers_minus1)		20
else		
general_hrd_parameters(sps_max_sub_layers_minus1, sps_max_sub_layers_minus1)		
}		
vui_parameters_present_flag	u(1)	
if(vui_parameters_present_flag)		
vui_parameters()		30
sps_extension_flag	u(1)	
if(sps_extension_flag)		
while(more_rbsp_data())		
sps_extension_data_flag	u(1)	
rbsp_trailing_bits()		
}		

40

【 0 1 1 8 】

例示的なDPBパラメータシンタックスは次のとおりである。

【 0 1 1 9 】

50

【表 3】

dpb_parameters(dpbSizeOnlyFlag, maxSubLayersMinus1, subLayerInfoFlag)	記述子
{	
for(i = (subLayerInfoFlag ? 0 : maxSubLayersMinus1);	
i <= maxSubLayersMinus1; i++) {	
max_dec_pic_buffering_minus1[i]	ue(v)
if(!dpbSizeOnlyFlag) {	
max_num_reorder_pics[i]	ue(v)
max_latency_increase_plus1[i]	ue(v)
}	
}	
}	

10

【 0 1 2 0 】

20

例示的な一般HRDパラメータシンタックスは次のとおりである。

【 0 1 2 1 】

30

40

50

【表 4】

記述子	
general_hrd_parameters() {	
general_nal_hrd_params_present_flag	u(1)
general_vcl_hrd_params_present_flag	u(1)
if(general_nal_hrd_params_present_flag general_vcl_hrd_params_present_flag) {	
decoding_unit_hrd_params_present_flag	u(1)
if(decoding_unit_hrd_params_present_flag) {	
tick_divisor_minus2	u(8)
decoding_unit_cpb_params_in_pic_timing_sei_flag	u(1)
}	
bit_rate_scale	u(4)
cpb_size_scale	u(4)
if(decoding_unit_hrd_params_present_flag)	
cpb_size_du_scale	u(4)
}	
if(vps_max_sub_layers_minus1 > 0)	
sub_layer_cpb_params_present_flag	u(1)
if(TotalNumOls > 1)	
num_ols_hrd_params_minus1	ue(v)
hrd_cpb_cnt_minus1	ue(v)
for(i = 0; i <= num_ols_hrd_params_minus1; i++) {	
if(vps_max_sub_layers_minus1 > 0 && !vps_all_layers_same_num_sub_layers_flag)	
hrd_max_temporal_id[i]	u(3)
ols_hrd_parameters(hrd_max_temporal_id[i])	
}	
if(num_ols_hrd_params_minus1 > 0)	
for(i = 1; i < TotalNumOls; i++)	
ols_hrd_idx[i]	ue(v)
}	

10

20

30

40

【 0 1 2 2 】

例示的なOLD HRDパラメータシンタックスは次のとおりである。

【 0 1 2 3 】

50

【表 5】

	記述子
ols_hrd_parameters(hrdMaxTid) {	
firstSubLayer = sub_layer_cpb_params_present_flag ? 0: hrdMaxTid	
for(i = firstSubLayer; i <= hrdMaxTid; i++) {	
fixed_pic_rate_general_flag[i]	u(1)
if(!fixed_pic_rate_general_flag[i])	
fixed_pic_rate_within_cvs_flag[i]	u(1)
if(fixed_pic_rate_within_cvs_flag[i])	
elemental_duration_in_tc_minus1[i]	ue(v)
else if(hrd_cpb_cnt_minus1 == 0)	
low_delay_hrd_flag[i]	u(1)
if(general_nal_hrd_params_present_flag)	
sub_layer_hrd_parameters(i)	
if(general_vcl_hrd_params_present_flag)	
sub_layer_hrd_parameters(i)	
}	
}	

10

20

【 0 1 2 4 】

例示的なサブレイヤHRDパラメータシンタックスは次のとおりである。

【 0 1 2 5 】

【表 6】

	記述子
sub_layer_hrd_parameters(subLayerId) {	
for(j = 0; j <= hrd_cpb_cnt_minus1; j++) {	
bit_rate_value_minus1[subLayerId][j]	ue(v)
cpb_size_value_minus1[subLayerId][j]	ue(v)
if(decoding_unit_hrd_params_present_flag) {	
cpb_size_du_value_minus1[subLayerId][j]	ue(v)
bit_rate_du_value_minus1[subLayerId][j]	ue(v)
}	
cbr_flag[subLayerId][j]	u(1)
}	
}	

30

40

【 0 1 2 6 】

例示的なビデオパラメータセットRBSPセマンティクスは次のとおりである。vps_max_layers_minus1+1は、VPSを参照する各CVS内のレイヤの最大許容数を指定する。vps_max_sub_layers_minus1+1は、VPSを参照する各CVS内に存在している可能性のある時間的サブ

50

レイ

ヤの最大数を指定する。vps_max_sub_layers_minus1の値は、0から6までの範囲内にあり

得る。1に等しいvps_all_layers_same_num_sub_layers_flagは、時間的サブレイヤの数が

VPSを参照する各CVS内のすべてのレイヤについて同じであることを指定する。0に等しいv

ps_all_layers_same_num_sub_layers_flagは、VPSを参照する各CVSにおけるレイヤが同じ

数の時間的サブレイヤを有し得るか、または有し得ないことを指定する。存在していないとき、vps_all_layers_same_num_sub_layers_flagの値は、1に等しいと推論され得る。1

に等しいvps_all_independent_layers_flagは、CVS内のすべてのレイヤは、レイヤ間予測

を使用することなく独立してコーディングされることを指定する。0に等しいvps_all_independent_layers_flagは、CVS内のレイヤの1つまたは複数

がレイヤ間予測を使用していないことを指定する。存在していないとき、vps_all_independent_layers_flagの値は、1に等

しいと推論され得る。vps_all_independent_layers_flagが1に等しいとき、vps_independent_layer_flag[i]の値は、1に等しいと推論される。vps_all_independent_layers_flag

が0に等しいとき、vps_independent_layer_flag[0]の値は、1に等しいと推論される。

【 0 1 2 7 】

0に等しいvps_direct_dependency_flag[i][j]は、インデックスjを有するレイヤはインデックスiを有するレイヤに対する直接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を有するレイヤはインデックスiを有するレイヤに対する直接参照レイヤであることを指定する。vps_direct_dependency_flag

[i][j]が0からvps_max_layers_minus1までの範囲内のiおよびjについて存在していないとき、フラグは0に等しいと推論される。i番目のレイヤのj番目の直接従属レイヤを指定する、変数DirectDependentLayerIdx[i][j]、およびレイヤインデックスjを有するレイヤが任意の他のレイヤによって参照レイヤとして使用されるかどうかを指定する変数LayerUsedAsRefLayerFlag[j]は、次のように導出され得る。

```
for(i=0; i <=vps_max_layers_minus1; i++)
  LayerUsedAsRefLayerFlag[j]=0
```

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

【 0 1 2 8 】

nuh_layer_idがvps_layer_id[i]に等しいレイヤのレイヤインデックスを指定する、変数GeneralLayerIdx[i]は、次のように導出され得る。

```
for(i=0; i <=vps_max_layers_minus1; i++)
```

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

【 0 1 2 8 】

nuh_layer_idがvps_layer_id[i]に等しいレイヤのレイヤインデックスを指定する、変数GeneralLayerIdx[i]は、次のように導出され得る。

```
for(i=0; i <=vps_max_layers_minus1; i++)
```

10

20

30

40

50

```
GeneralLayerIdx[vps_layer_id[i]]=i
```

【 0 1 2 9 】

1に等しいeach_layer_is_an_ols_flagは、各出力レイヤセットがただ1つのレイヤを含み、ビットストリーム内の各レイヤそれ自体は、単一の含まれるレイヤが唯一の出力レイヤである出力レイヤセットであることを指定する。0に等しいeach_layer_is_an_ols_flagは、出力レイヤセットが複数のレイヤを含み得ることを指定する。vps_max_layers_minus1が0に等しい場合、each_layer_is_an_ols_flagの値は、1に等しいと推論される。そうでない場合、vps_all_independent_layers_flagが0に等しいとき、each_layer_is_an_ols_flagの値は、0に等しいと推論される。

【 0 1 3 0 】

0に等しいols_mode_idcは、VPSによって指定されたOLSの総数がvps_max_layers_minus1+1に等しく、i番目のOLSは0からiまでのレイヤインデックスを有するレイヤを含み、各OLSについてOLS内の最上位レイヤのみが出力されることを指定する。1に等しいols_mode_idcは、VPSによって指定されたOLSの総数がvps_max_layers_minus1+1に等しく、i番目のOLSは0からiまでのレイヤインデックスを有するレイヤを含み、各OLSについてOLS内のすべてのレイヤが出力されることを指定する。2に等しいols_mode_idcは、VPSによって指定されたOLSの総数が明示的にシグナリングされ、各OLSについてOLS内の最上位レイヤおよび下位レイヤの明示的にシグナリングされたセットが出力されることを指定する。ols_mode_idcの値は、0から2までの範囲内にあり得る。vps_all_independent_layers_flagが1に等しく、each_layer_is_an_ols_flagが0に等しいとき、ols_mode_idcの値は、2に等しいと推論される。num_output_layer_sets_minus1+1は、ols_mode_idcが2に等しいときにVPSによって指定されたOLSの総数を指定する。

【 0 1 3 1 】

VPSによって指定されたOLSの総数を指定する、変数TotalNumOlssは、次のように導出され得る。

```
if(vps_max_layers_minus1==0)
```

```
    TotalNumOlss=1
```

```
else if(each_layer_is_an_ols_flag || ols_mode_idc==0 || ols_mode_idc==1)
```

```
    TotalNumOlss=vps_max_layers_minus1+1
```

```
else if(ols_mode_idc==2)
```

```
    TotalNumOlss=num_output_layer_sets_minus1+1
```

【 0 1 3 2 】

layer_included_flag[i][j]は、j番目のレイヤ(たとえば、nuh_layer_idがvps_layer_id[j]に等しいレイヤ)が、ols_mode_idcが2に等しいときにi番目のOLS内に含まれることを指定する。1に等しいlayer_included_flag[i][j]は、j番目のレイヤがi番目のOLSに含まれることを指定する。0に等しいlayer_included_flag[i][j]は、j番目のレイヤがi番目の

10

20

30

40

50

OLSに含まれないことを指定する。i番目のOLS内のレイヤの数を指定する、変数NumLayers

InOls[i]およびi番目のOLS内のj番目のレイヤのnuh_layer_id値を指定する変数LayerIdIn

Ols[i][j]は、次のように導出され得る。

NumLayersInOls[0]=1

LayerIdInOls[0][0]=vps_layer_id[0]

```
for(i=1, i TotalNumOls; i++) {
```

```
  if(each_layer_is_an_ols_flag) {
```

```
    NumLayersInOls[i]=1
```

```
    LayerIdInOls[i][0]=vps_layer_id[i]
```

```
  } else if(ols_mode_idc==0 | | ols_mode_idc==1) {
```

```
    NumLayersInOls[i]=i+1
```

```
    for(j=0; j NumLayersInOls[i]; j++)
```

```
      LayerIdInOls[i][j]=vps_layer_id[j]
```

```
  } else if(ols_mode_idc==2) {
```

```
    for(k=0, j=0; k =vps_max_layers_minus1; k++)
```

```
      if(layer_included_flag[i][k])
```

```
        LayerIdInOls[i][j++]=vps_layer_id[k]
```

```
    NumLayersInOls[i]=j
```

```
  }
```

```
}
```

【 0 1 3 3 】

nuh_layer_idがLayerIdInOls[i][j]に等しいレイヤのOLSレイヤインデックスを指定する、変数OlsLayerIdx[i][j]は、次のように導出され得る。

```
for(i=0, i TotalNumOls; i++)
```

```
  for j=0; j NumLayersInOls[i]; j++)
```

```
    OlsLayerIdx[i][LayerIdInOls[i][j]]=j
```

【 0 1 3 4 】

各OLS内の最下レイヤは、独立レイヤであるものとする。言い換えると、0からTotalNum

Ols-1までの範囲内の各iについて、vps_independent_layer_flag[GeneralLayerIdx[Layer

IdInOls[i][0]]の値は1に等しいものとする。各レイヤは、VPSによって指定される少なくとも1つのOLS内に含まれるものとする。言い換えると、0からvps_max_layers_minus1の

範囲内のkについてvps_layer_id[k]の1つに等しいnuh_layer_id nuhLayerIdの特定の値を

有する各レイヤについて、iおよびjの値の少なくとも1つの対があるものとし、iは0からTotalNumOls-1の範囲内にあり、jはNumLayersInOls[i]-1の範囲内にあり、LayerIdIn

Ols[i][j]の値はnuhLayerIdに等しい。OLS内の任意のレイヤは、OLSの出力レイヤまたはOLSの

出力レイヤの(直接的または間接的)参照レイヤであるものとする。

【 0 1 3 5 】

vps_output_layer_flag[i][j]は、ols_mode_idcが2に等しいときにi番目のOLS内のj番

目のレイヤが出力されるかどうかを指定する。1に等しいvps_output_layer_flag[i]は、i番目のOLS内のj番目のレイヤが出力されることを指定する。0に等しいvps_output_laye

r_

10

20

30

40

50

flag[i]は、i番目のOLS内のj番目のレイヤが出力されないことを指定する。vps_all_independent_layers_flagが1に等しく、each_layer_is_an_ols_flagが0に等しいとき、vps_output_layer_flag[i]の値は、1に等しいと推論される。値1がi番目のOLS内のj番目のレイヤが出力されることを指定し、値0がi番目のOLS内のj番目のレイヤが出力されないことを指定する変数OutputLayerFlag[i][j]は、次のように導出され得る。

```
for(i=0; i TotalNumOls; i++) {
  OutputLayerFlag[i][NumLayersInOls[i]-1]=1
  for(j=0; j NumLayersInOls[i]-1; j++)
    if(ols_mode_idc[i]==0)
      OutputLayerFlag[i][j]=0
    else if(ols_mode_idc[i]==1)
      OutputLayerFlag[i][j]=1
    else if(ols_mode_idc[i]==2)
      OutputLayerFlag[i][j]=vps_output_layer_flag[i][j]
}
```

```
for(i=0; i TotalNumOls; i++) {
  OutputLayerFlag[i][NumLayersInOls[i]-1]=1
  for(j=0; j NumLayersInOls[i]-1; j++)
    if(ols_mode_idc[i]==0)
      OutputLayerFlag[i][j]=0
    else if(ols_mode_idc[i]==1)
      OutputLayerFlag[i][j]=1
    else if(ols_mode_idc[i]==2)
      OutputLayerFlag[i][j]=vps_output_layer_flag[i][j]
}
```

10

【0136】

0番目のOLSは、最下位レイヤ(たとえば、nuh_layer_idがvps_layer_id[0]に等しいレイ

ヤ)のみを含み、0番目のOLSについて、含まれているレイヤのみが出力されることに留意されたい。vps_num_ptlsは、VPSのprofile_tier_level()シンタックス構造の数を指定する。1に等しいpt_present_flag[i]は、プロファイル、ティア、および一般制約条件情報が、VPS内のi番目のprofile_tier_level()シンタックス構造内に存在することを指定する。0に等しいpt_present_flag[i]は、プロファイル、ティア、および一般制約条件情報が、VPS内のi番目のprofile_tier_level()シンタックス構造内に存在していないことを指定する。pt_present_flag[0]の値は、0に等しいと推論される。pt_present_flag[i]が0に等

20

しいとき、VPS内のi番目のprofile_tier_level()シンタックス構造に対するプロファイル、ティア、および一般制約条件情報は、VPS内の(i-1)番目のprofile_tier_level()シンタックス構造のものと同一であると推論される。

30

【0137】

ptl_max_temporal_id[i]は、レベル情報がVPS内のi番目のprofile_tier_level()シンタ

ックス構造に存在する最上位サブレイヤ表現のTemporalIdを指定する。ptl_max_temporal

_id[i]の値は、0からvps_max_sub_layers_minus1までの範囲内であるものとする。vps_max

x_sub_layers_minus1が0に等しいとき、ptl_max_temporal_id[i]の値は、0に等しいと推

論される。vps_max_sub_layers_minus1が0より大きく、vps_all_layers_same_num_sub_la

40

yers_flagが1に等しいとき、ptl_max_temporal_id[i]の値は、vps_max_sub_layers_minus

1に等しいと推論される。vps_ptl_byte_alignment_zero_bitは、0に等しいものとする。

【0138】

ols_ptl_idx[i]は、i番目のOLSに適用されるprofile_tier_level()シンタックス構造のVPS内のprofile_tier_level()シンタックス構造のリストへのインデックスを指定する。存在するときには、ols_ptl_idx[i]の値は、0からvps_num_ptls-1までの範囲内にあるものとする。NumLayersInOls[i]が1に等しいとき、i番目のOLSに適用されるprofile_tier

50

_l

evel()シンタックス構造は、i番目のOLS内のレイヤによって参照されるSPS内に存在する。vps_num_dpb_paramsは、VPS内のdpb_parameters()シンタックス構造の数を指定する。v

ps_num_dpb_paramsの値は、0から16までの範囲内にあるものとする。存在していないとき

、vps_num_dpb_paramsの値は、0に等しいと推論され得る。1に等しいsame_dpb_size_output

ut_or_nonoutput_flagは、layer_nonoutput_dpb_params_idx[i]シンタックス要素がVPS内

に存在しないことを指定する。0に等しいsame_dpb_size_output_or_nonoutput_flagは、l

ayer_nonoutput_dpb_params_idx[i]シンタックス要素がVPS内に存在し得るか、または存

在し得ないことを指定する。vps_sub_layer_dpb_params_present_flagは、VPS内のdpb_pa

rameters()シンタックス構造におけるmax_dec_pic_buffering_minus1[]、max_num_reord

er_pics[]、およびmax_latency_increase_plus1[]シンタックス要素の存在を制御するために使用される。存在しないときには、vps_sub_dpb_params_info_present_flagは

、0

に等しいと推論される。

【 0 1 3 9 】

1に等しいdpb_size_only_flag[i]は、max_num_reorder_pics[]およびmax_latency_inc

rease_plus1[]シンタックス要素がi番目のdpb_parameters()シンタックス構造のそのVPS

に存在していないことを指定する。1に等しいdpb_size_only_flag[i]は、max_num_reorde

r_pics[]およびmax_latency_increase_plus1[]シンタックス要素がi番目のdpb_paramet

ers()シンタックス構造のそのVPSに存在し得ることを指定する。dpb_max_temporal_id[i]

は、DPBパラメータがVPS内のi番目のdpb_parameters()シンタックス構造に存在し得る最

上位サブレイヤ表現のTemporalIdを指定する。dpb_max_temporal_id[i]の値は、0からvps

_max_sub_layers_minus1までの範囲内であるものとする。vps_max_sub_layers_minus1が0

に等しいとき、dpb_max_temporal_id[i]の値は、0に等しいと推論され得る。vps_max_sub

_layers_minus1が0より大きく、vps_all_layers_same_num_sub_layers_flagが1に等しい

とき、dpb_max_temporal_id[i]の値は、vps_max_sub_layers_minus1に等しいと推論され

る。layer_output_dpb_params_idx[i]は、OLS内の出力レイヤであるときにi番目のレイヤ

に適用されるdpb_parameters()シンタックス構造の、VPS内のdpb_parameters()シンタッ

クス構造のリストへのインデックスを指定する。存在するときには、layer_output_dpb

10

20

30

40

50

_p

arams_idx[i]の値は、0からvps_num_dpb_params-1までの範囲内にあるものとする。

【0 1 4 0】

vps_independent_layer_flag[i]が1に等しい場合、出力レイヤであるときにi番目のレイヤに適用されるdpb_parameters()シンタックス構造は、レイヤによって参照されるSPS内に存在するdpb_parameters()シンタックス構造である。そうでない場合(vps_indepe

ndependent_layer_flag[i]が1に等しい場合)、次が適用される。vps_num_dpb_paramsが1に等しい

とき、layer_output_dpb_params_idx[i]の値は、0に等しいと推論される。ビットスト

10

リーム適合性の要件は、layer_output_dpb_params_idx[i]の値が、dpb_size_only_flag[lay

er_output_dpb_params_idx[i]]が0に等しくなるような値であるものとしてよい。

【0 1 4 1】

layer_nonoutput_dpb_params_idx[i]は、i番目のレイヤがOLS内の非出力レイヤである

ときにi番目のレイヤに適用されるdpb_parameters()シンタックス構造の、VPS内のdpb

_pa
rameters()シンタックス構造のリストへのインデックスを指定する。存在するときには、

20

layer_nonoutput_dpb_params_idx[i]の値は、0からvps_num_dpb_params-1までの範囲内に

あるものとする。same_dpb_size_output_or_nonoutput_flagが1に等しい場合、次が適用

される。vps_independent_layer_flag[i]が1に等しい場合、i番目のレイヤが非出力レイ

30

ヤであるときにi番目のレイヤに適用されるdpb_parameters()シンタックス構造は、レイ

ヤによって参照されるSPS内に存在するdpb_parameters()シンタックス構造である。そ

うでない場合(vps_independent_layer_flag[i]が1に等しい場合)、layer_nonoutput_dp

b_pa
rams_idx[i]の値は、layer_output_dpb_params_idx[i]に等しいと推論される。そう

でない場合(same_dpb_size_output_or_nonoutput_flagが0に等しい場合)、vps_num_dp

40

b_param
sが1に等しいとき、layer_output_dpb_params_idx[i]の値は、0に等しいと推論される

。

【0 1 4 2】

1に等しいgeneral_hrd_params_present_flagは、シンタックス要素num_units_in

_tick
およびtime_scaleならびにシンタックス構造general_hrd_parameters()がSPS RBSP

シンタ
ックス構造内に存在することを指定する。0に等しいgeneral_hrd_params_present fla

50

gは、シンタックス要素num_units_in_tickおよびtime_scaleならびにシンタックス構造ge

ner
al_hrd_parameters()がSPS RBSPシンタックス構造内に存在しないことを指定する。nu

m_u
nits_in_tickは、クロックティックカウンタの1増分(クロックティックと呼ばれる)に対

応する周波数time_scalehertz(Hz)で動作するクロックの時間単位の数である。num_un

its
_in_tickは、0より大きいものとする。クロックティックは、秒単位であり、num_units

_i

n_tickをtime_scaleで除算した商に等しい。たとえば、ビデオ信号のピクチャレートが25 Hzであるとき、time_scaleは27,000,000に等しいものとしてよく、num_units_in_tickは1

,080,000に等しいものとしてよく、その結果、クロックティックは0.04秒に等しいものと

してよい。time_scaleは、1秒間に通過する時間単位の数である。たとえば、27MHzのクロ

ックを使用して時間を計測する時間座標系は、27,000,000のtime_scaleを有する。time_s

caleの値は、0より大きいものとする。

【0143】

0に等しいvps_extension_flagは、VPS RBSPシンタックス構造にvps_extension_data_fl

agシンタックス要素が存在していないことを指定する。1に等しいvps_extension_flagは、VPS RBSPシンタックス構造に存在するvps_extension_data_flagシンタックス要素があ

ることを指定する。vps_extension_data_flagは任意の値を有し得る。vps_extension_dat

a_flagの存在および値は、プロファイルへのデコーダ適合性に影響を及ぼし得ない。適合するデコーダは、すべてのvps_extension_data_flagシンタックス要素を無視し得る。

【0144】

例示的なシーケンスパラメータセットRBSPセマンティクスは次のとおりである。SPS RBSPは、TemporalIdが0に等しい少なくとも1つのアクセスユニットに含まれるか、または外

部手段を通じて提供される、参照される前のデコーディングプロセスに利用可能であるものとし、SPS RBSPを含むSPS NALユニットは、SPS NALユニットを参照するPPS NAL ユニッ

トの最低のnuh_layer_id値に等しいnuh_layer_idを有するものとする。CVS内のsps_s eq_p

arameter_set_idの特定の値を有するすべてのSPS NALユニットは、同じ内容を有するもの

とする。sps_decoding_parameter_set_idは、0よりも大きいとき、SPSによって参照され

るDPSに対するdps_decoding_parameter_set_idの値を指定する。sps_decoding_pa rameter

_set_idが0に等しいとき、SPSは、DPSを参照せず、SPSを参照する各CLVSをデコードす

るときにDPSは参照されない。sps_decoding_parameter_set_idの値は、ビットストリー ム内

のコーディング済みピクチャによって参照されるすべてのSPSにおいて同じであるものとする。

【0145】

sps_video_parameter_set_idは、0よりも大きいとき、SPSによって参照されるVPS に対

するvps_video_parameter_set_idの値を指定する。sps_video_parameter_set_idが0に等

しいとき、SPSは、VPSを参照し得ず、SPSを参照する各CLVSをデコードするときにVPS は参

照されず、GeneralLayerIdx[nuh_layer_id]の値は、0に等しいと推論されるものとし

10

20

30

40

50

、vp

s_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]の値は、1に等しいと推論され得る。vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいとき

、特定のnuh_layer_id値nuhLayerIdを有するCLVSによって参照されるSPSは、nuhLayerIdに等しいnuh_layer_idを有するものとする。

【0146】

sps_max_sub_layers_minus1+1は、SPSを参照する各CLVS内に存在している可能性のある

10

時間的サブレイヤの最大数を指定する。sps_max_sub_layers_minus1の値は、0からvps_max_sub_layers_minus1

までの範囲内にあるものとする。sps_reserved_zero_4bitsは、適合

するビットストリーム内で0に等しいものとする。sps_reserved_zero_4bitsに対する他の値は、予約され得る。

【0147】

1に等しいsps_ptl_dpb_present_flagは、profile_tier_level()シンタックス構造およびdpb_parameters()シンタックス構造がSPS内に存在していることを指定する。0に等しい

20

sps_ptl_dpb_present_flagは、profile_tier_level()シンタックス構造も、dpb_parameters()シンタックス構造も、SPS内に存在していないことを指定する。sps_ptl_dpb_present

_flagの値は、vps_independent_layer_flag[nuh_layer_id]に等しいものとする。vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しい場合、変数MaxDecPic

BuffMinus1は、SPS内のdpb_parameters()シンタックス構造においてmax_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]に等しくなるように設定される。そうでない場合

30

、MaxDecPicBuffMinus1は、VPS内のlayer_nonoutput_dpb_params_idx[GeneralLayerIdx[nuh_layer_id]]番目のdpb_parameters()シンタックス構造のmax_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]に等しくなるように設定される。1に等しいgdr_enabled_flagは、SPSを参照するCLVSにGDRピクチャが存在し得ることを指定する。0に等しいgdr_enabled_flagは、SPSを参照するCLVSにGDRピクチャが存在しないことを指定する。

40

【0148】

sps_sub_layer_dpb_params_flagは、SPS内のdpb_parameters()シンタックスにおけるmax_dec_pic_buffering_minus1[i]、max_num_reorder_pics[i]、およびmax_latency_increase_plus1[i]シンタックス要素の存在を制御するために使用される。存在しないときには、sps_sub_dpb_params_info_present_flagは、0に等しいと推論される。0に等しい

50

ong_t
erm_ref_pics_flagは、CLVS内の任意のコーディング済みピクチャのインター予測に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、CLVS内の1つ
または複数のコーディング済みピクチャのインター予測にLTRPが使用され得ることを指定する。

【0149】

例示的な一般的なプロファイル、ティア、およびレベルのセマンティクスは次のとおり
である。profile_tier_level()シンタックス構造は、レベル情報、および任意選択で、プロ
ファイル、ティア、サブプロファイル、および一般制約条件情報(PT情報として示され
る)を提供する。profile_tier_level()シンタックス構造がDPSに含まれるとき、OlsInSco
peは、DPSを参照するビットストリーム全体におけるすべてのレイヤを含むOLSである。
profile_tier_level()シンタックス構造が、VPSに含まれるとき、OlsInScopeは、VPSによ
って指定される1つまたは複数のOLSである。profile_tier_level()シンタックス構造が、SP
Sに含まれるとき、OlsInScopeは、独立レイヤであるものとする、SPSを参照するレイヤ
のうち最下位レイヤであるレイヤのみを含むOLSである。

【0150】

general_profile_idcは、OlsInScopeが適合するプロファイルを示す。general_tier
_fl
agは、general_level_idcの解釈に対するティアコンテキストを指定する。num_sub_pr
ofi
lesは、general_sub_profile_idc[i]シンタックス要素の数を指定する。general_sub_p
ro
file_idc[i]は、登録されているi番目の相互運用性メタデータを示している。general_le
vel_idcは、OlsInScopeが適合するレベルを示す。general_level_idcは値が大きいほど
レベルが高いことを示すことに留意されたい。OlsInScopeに対するDPSでシグナリングされ
る最大レベルは、OlsInScope内に含まれるCVSに対するSPSにおいてシグナリングされる
レベルより高いものとしてよい。OlsInScopeが複数のプロファイルに適合するとき、gener
a
l_profile_idcは、エンコーダによって決定されるように、好ましいデコード済み結果ま
たは好ましいビットストリーム識別を提供するプロファイルを示すべきであることにも留
意されたい。profile_tier_level()シンタックス構造がDPSに含まれ、OlsInScopeのCVS
が
異なるプロファイルに適合するとき、general_profile_idcおよびlevel_idcはOlsInSco
pe
をデコードすることができるデコーダに対するプロファイルおよびレベルを示すべきであ
ることにも留意されたい。

【0151】

1に等しいsub_layer_level_present_flag[i]は、レベル情報がiに等しいTemporalId
を
有するサブレイヤ表現に対するprofile_tier_level()シンタックス構造内に存在すること
を指定する。0に等しいsub_layer_level_present_flag[i]は、レベル情報がiに等しいTem
p
oralIdを有するサブレイヤ表現に対するprofile_tier_level()シンタックス構造内に存

10

20

30

40

50

在しないことを指定する。ptl_alignment_zero_bitsは0に等しいものとする。シンタックス

要素sub_layer_level_idc[i]のセマンティクスは、存在していない値の推論の指定とは別にして、シンタックス要素general_level_idcと同じであるが、iに等しいTemporalIdを

有するサブレイヤ表現に適用される。

【 0 1 5 2 】

例示的なDPBパラメータセマンティクスは次のとおりである。dpb_parameters(maxSubLayersMinus1, subLayerInfoFlag)シンタックス構造は、DPBサイズ、最大ピクチャ順序

変更

数、およびCVSの各CLVSに対する最大待ち時間の情報を提供する。dpb_parameters()シン

タックス構造が、VPSに含まれるとき、dpb_parameters()シンタックス構造が適用される0

LSは、VPSによって指定される。dpb_parameters()シンタックス構造が、SPSに含まれると

き、dpb_parameters()シンタックス構造は、独立レイヤであるものとする、SPSを参照するレイヤのうち最下位レイヤであるレイヤのみを含むOLSに適用される。

【 0 1 5 3 】

max_dec_pic_buffering_minus1[i]+1は、CVSの各CLVSについて、Htidがiに等しいとき

にピクチャストレージバッファのユニットでのデコード済みピクチャバッファの最大必要サイズを指定する。max_dec_pic_buffering_minus1[i]の値は、0からMaxDpbSize-1までの

範囲内にあるものとする。iが0より大きいとき、max_dec_pic_buffering_minus1[i]は、m

ax_dec_pic_buffering_minus1[i-1]以上であるものとする。subLayerInfoFlagが0に等し

いことに起因して、max_dec_pic_buffering_minus1[i]が、0からmaxSubLayersMinus1-1ま

での範囲内のiについて存在していないとき、max_dec_pic_buffering_minus1[i]は、max_

dec_pic_buffering_minus1[maxSubLayersMinus1]に等しいと推論される。

【 0 1 5 4 】

max_num_reorder_pics[i]は、CVSの各CLVSについて、Htidがiに等しいときにCLVSの任

意のピクチャにデコード順序で先行し出力順序でそのピクチャが後に続くことができるCLVSのピクチャの最大許容数を指定する。max_num_reorder_pics[i]の値は、0からmax_

dec_

pic_buffering_minus1[i]までの範囲内にあるものとする。iが0より大きいとき、max_

num

_reorder_pics[i]は、max_num_reorder_pics[i-1]以上であるものとする。subLayerI

nfoF

lagが0に等しいことに起因して、max_num_reorder_pics[i]が、0からmaxSubLayers

Minus1

-1までの範囲内のiについて存在していないとき、max_num_reorder_pics[i]は、max_

num_

reorder_pics[maxSubLayersMinus1]に等しいと推論される。

【 0 1 5 5 】

10

20

30

40

50

0に等しくないmax_latency_increase_plus1[i]は、MaxLatencyPictures[i]の値を計算するために使用され、これは、CVSの各CLVSについて、Htidがiに等しいときにCLVSの任意のピクチャに出力順序で先行しデコード順序でそのピクチャが後に続くことができるCLVS内のピクチャの最大数を指定する。max_latency_increase_plus1[i]が0に等しくないとき

、MaxLatencyPictures[i]の値は、次のように指定され得る。

MaxLatencyPictures[i]=max_num_reorder_pics[i]+max_latency_increase_plus1[i]-1

max_latency_increase_plus1[i]が0に等しいとき、対応する限度は表現されない。

【0156】

max_latency_increase_plus1[i]の値は、0から232-2までの範囲内であるべきである。s

ubLayerInfoFlagが0に等しいことに起因して、max_latency_increase_plus1[i]が、0から

maxSubLayersMinus1-1までの範囲内のiについて存在していないとき、max_latency_incre

ase_plus1[i]は、max_latency_increase_plus1[maxSubLayersMinus1]に等しいと推論され

る。

【0157】

例示的な一般HDRパラメータセマンティクスは、次のとおりである。general_hrd_param

eters()シンタックス構造は、HRD演算で使用されるHRDパラメータを規定する。num_ols_h

rd_params_minus1+1は、general_hrd_parameters()シンタックス構造に存在するols_hrd_

parameters()シンタックス構造の数を指定する。num_ols_hrd_params_minus1の値は、0か

ら63までの範囲内にあるものとする。TotalNumOlsが1より大きいとき、num_ols_hrd_par

ams_minus1の値は、0に等しいと推論される。hrd_cpb_cnt_minus1+1は、CVSのビットスト

リーム内の代替的CPB指定の数を指定する。hrd_cpb_cnt_minus1の値は、0から31までの範

囲内にあるものとする。hrd_max_temporal_id[i]は、HRDパラメータがi番目のlayer_l

l_hrd_parameters()シンタックス構造に含まれる最上位サブレイヤ表現のTemporalIdを指

定する。hrd_max_temporal_id[i]の値は、0からvps_max_sub_layers_minus1までの範囲内

であるものとする。vps_max_sub_layers_minus1が0に等しいとき、hrd_max_temporal_id[

i]の値は、0に等しいと推論される。ols_hrd_idx[i]は、i番目のOLSに適用されるols_hrd_

parameters()シンタックス構造のインデックスを指定する。ols_hrd_idx[[i]の値は、0からnum_ols_hrd_params_minus1までの範囲内であるものとする。存在していないとき、o

ls_hrd_idx[[i]の値は、0に等しいと推論される。

【0158】

10

20

30

40

50

例示的な参照ピクチャリスト構造セマンティクスは、次のとおりである。ref_pic_list_struct(listIdx, rplIdx)シンタックス構造は、SPS内、またはスライスヘッダ内に存在しているものとしてよい。シンタックス構造がスライスヘッダまたはSPSに含まれるかどうかに応じて、次が適用される。スライスヘッダ内に存在する場合、ref_pic_list_struct(listIdx, rplIdx)シンタックス構造は、現在のピクチャ(スライスを含むピクチャ)の参照ピクチャリストlistIdxを指定する。そうでない場合(SPSに存在する場合)、ref_pic_list_struct(listIdx, rplIdx)シンタックス構造は、参照ピクチャリストlistIdxに対する候補を指定し、この節の残りの部分で指定されるセマンティクスにおける「現在のピクチャ」という用語は、SPSに含まれるref_pic_list_struct(listIdx, rplIdx)シンタックス構造のリストへのインデックスに等しいref_pic_list_idx[listIdx]を含む1つまたは複数のスライスを含み、SPSを参照するCVS内にある各ピクチャを指す。num_ref_entries[listIdx][rplIdx]は、ref_pic_list_struct(listIdx, rplIdx)シンタックス構造内のエントリの数を指定する。num_ref_entries[listIdx][rplIdx]の値は、0からMaxDecPicBufferMius1+14までの範囲内であるものとする。

【0159】

例示的な一般的なデコーディングプロセスは、次のとおりである。このプロセスへの入力、ビットストリームBitstreamToDecodeである。このプロセスの出力は、デコード済みピクチャのリストである。デコーディングプロセスは、指定されたプロファイルおよびレベルに適合するすべてのデコーダが、そのプロファイルおよびレベルに適合するビットストリームに対してそのプロファイルに関連付けられているデコーディングプロセスを呼び出すときに数値的に同一のクロップされたデコード済み出力ピクチャを形成するように指定される。本明細書において説明されているプロセスによって形成されたものと同じのクロップされたデコード済み出力ピクチャを形成する任意のデコーディングプロセスは、(指定されているように、正しい出力順序または出力タイミングで)デコーディングプロセス要件に適合する。

【0160】

ビットストリーム内の各IRAP AUについて、次が適用される。AUがビットストリーム内のデコード順で最初のAUであるか、各ピクチャが瞬時デコーディングリフレッシュ(IDR)ピクチャであるか、または各ピクチャがデコード順でシーケンスNALユニットの終端の後に続くレイヤの最初のピクチャである場合に、変数NoIncorrectPicOutputFlagは1に等しくなるように設定される。そうでなければ、変数HandleCraAsCvsStartFlagがAUに対する値に設定されている場合、HandleCraAsCvsStartFlagは外部メカニズムによって提供される値に等しくなるように設定され、NoIncorrectPicOutputFlagはHandleCraAsCvsStartFlagに等しくなるように設定される。そうでなければ、HandleCraAsCvsStartFlagおよびNoIncorrectPicOutputFlagは両方とも0に等しくなるように設定される。

【0161】

ビットストリーム内の各漸進的デコーディングリフレッシュ(GDR)AUについて、次が適用される。AUがビットストリーム内のデコード順で最初のAUであるか、または各ピクチャがデコード順でシーケンスNALユニットの終端の後に続くレイヤの最初のピクチャである場合に、変数NoIncorrectPicOutputFlagは1に等しくなるように設定される。そうでなければ、何らかの外部メカニズムが、変数HandleGdrAsCvsStartFlagをAUに対する値に設定

10

20

30

40

50

するために利用可能である場合、HandleGdrAsCvsStartFlagは、外部メカニズムによつて

提供される値に等しくなるように設定され、NoIncorrectPicOutputFlagは、HandleGdrAsC

vsStartFlagに等しくなるように設定される。そうでなければ、HandleGdrAsCvsStartF

lagおよびNoIncorrectPicOutputFlagは両方とも0に等しくなるように設定される。IRAPピク

チャとGDRピクチャの両方について、上記の操作は、ビットストリーム中のCVSを識別す

るために使用される。デコーディングは、BitstreamToDecodeの各コーディング済みピクチャに対してデコード順に繰り返し呼び出される。

【0162】

参照ピクチャリスト構築のための例示的なデコーディングプロセスは、次のとおりである。このプロセスは、非IDRピクチャの各スライスに対してデコーディングプロセスの始めに呼び出される。参照ピクチャは、参照インデックスを通じてアドレス指定される。参照インデックスは、参照ピクチャリストへのインデックスである。Iスライスをデコードするとき、参照ピクチャリストは、スライスデータをデコードする際に使用されない。Pスライスをデコードするとき、参照ピクチャリスト0のみ(たとえば、RefPicList[0])が、スライスデータをデコードする際に使用される。Bスライスをデコードするとき、参照ピクチャリスト0と参照ピクチャリスト1(たとえば、RefPicList[1])の両方が、スライスデータをデコードする際に使用される。

【0163】

次の制約条件が、ビットストリーム適合性に対して適用される。0または1に等しい各*i*について、num_ref_entries[*i*][RplIdx[*i*]]は、NumRefIdxActive[*i*]より小さくないもの

とする。RefPicList[0]またはRefPicList[1]の各アクティブエントリによって参照されるピクチャは、DPB内に存在しているものとし、現在のピクチャのTemporalId以下のTemporal

Idを有するものとする。RefPicList[0]またはRefPicList[1]の各エントリによって参照されるピクチャは、現在のピクチャではないものとし、0に等しいnon_reference_picture

_flagを有するものとする。ピクチャのスライスのRefPicList[0]もしくはRefPicList[1]

内の短期参照ピクチャ(STRP)エントリおよび同じスライスまたは同じピクチャの異なるスライスのRefPicList[0]もしくはRefPicList[1]の長期参照ピクチャ(LTRP)エントリは、同じピクチャを参照すべきではない。RefPicList[0]またはRefPicList[1]に、現在のピクチャのPicOrderCntValとそのエントリによって参照されるピクチャのPicOrderCntValとの差

が224以上となるLTRPエントリはあるべきでない。

【0164】

setOfRefPicsを、現在のピクチャと同じnuh_layer_idを有するRefPicList[0]内のすべてのエントリおよび現在のピクチャと同じnuh_layer_idを有するRefPicList[1]内のすべてのエントリによって参照される固有のピクチャのセットであるとする。setOfRefPicsのピクチャの数は、MaxDecPicBuffMinus1以下であるものとし、setOfRefPicsは、ピクチャ

のすべてのスライスについて同じであるものとする。現在のピクチャが段階的時間的サブレイヤアクセス(STSA)ピクチャであるとき、RefPicList[0]またはRefPicList[1]に、現在のピクチャのTemporalIdに等しいTemporalIdを有するアクティブエントリがないものとする

る。現在のピクチャが、デコード順で、現在のピクチャとのTemporalIdに等しいTempo

10

20

30

40

50

ral

Idを有するSTSAピクチャの後に続くピクチャであるとき、デコード順でSTSAピクチャに先

行するRefPicList[0]またはRefPicList[1]内のアクティブエントリとして含まれる現在ピクチャのTemporalIdに等しいTemporalIdを有するピクチャはないものとする。

【0165】

現在のピクチャのスライスのRefPicList[0]またはRefPicList[1]内の各レイヤ間参照ピクチャ(ILRP)エントリによって参照されるピクチャは、現在のピクチャと同じアクセスユニットにあるものとする。現在のピクチャのスライスのRefPicList[0]またはRefPicList[1]内の各ILRPエントリによって参照されるピクチャは、DPB内に存在するものとし、現在のピクチャのnuh_layer_idより小さいnuh_layer_idを有するものとする。スライスのRefP

10

icList[0]またはRefPicList[1]内の各ILRPエントリは、アクティブエントリであるべきである。

【0166】

例示的なHRD指定は、次のとおりである。HRDは、ビットストリームおよびデコーダ適合

性をチェックするために使用される。entireBitstreamと表記される、ビットストリーム全体と称される、ビットストリームの適合性をチェックするためにビットストリーム適合性テストのセットが使用される。ビットストリーム適合性テストのセットは、VPSによって指定された各OLSの各OPの適合性をテストするためのものである。

20

【0167】

各テストについて、次の順序のステップがリストにされた順で適用され、その後、この節におけるこれらのステップの後に記述されるプロセスが続く。targetOpと表記される、テスト対象のオペレーションポイントが、OLSインデックスopOlsIdxおよび最高のTemporal

Id値opTidを有するターゲットOLSを選択することによって選択される。opOlsIdxの値は、0からTotalNumOls-1までの範囲内にある。opTidの値は、0からvps_max_sublayers_mi

nus1までの範囲内にある。opOlsIdxおよびopTidの選択された値の各対は、entireBitstream

30

am、opOlsIdx、およびopTidを入力としてサブビットストリーム抽出プロセスを呼び出すことによって出力されるサブビットストリームが次の条件を満たすような対であるものとする。BitstreamToDecodeにおけるLayerIdInOls[opOlsIdx]のnuh_layer_idに等しいnuh_l

ayer_id値を有するVCL NALユニットが少なくとも1つある。BitstreamToDecodeにおいてTe

mporalIdがopTidに等しいVCL NALユニットが少なくとも1つある。

【0168】

targetOpにおけるレイヤが、entireBitstream内のすべてのレイヤを含み、opTidがentire

40

Bitstream内のすべてのNALユニットの中で最高のTemporalId値と等しい大きいである場

合、BitstreamToDecodeはentireBitstreamと同一になるように設定される。そうでない場

合、BitstreamToDecodeは、entireBitstream、opOlsIdx、およびopTidを入力としてサブ

ビットストリーム抽出プロセスを呼び出すことによって出力されるように設定される。TargetOlsIdxおよびHtidの値は、それぞれtargetOpのopOlsIdxおよびopTidに等しくなるよ

50

うに設定される。Scldxの値が選択される。選択されたScldxは、0からhrd_cpb_cnt_minus

1までの範囲内にあるものとする。TargetOlsIdxに適用可能なバッファリング期間SEIメッセージ(TargetLayerBitstream内に存在するか、または外部メカニズムを通じて利用可能)

に関連付けられているBitstreamToDecodeにおけるアクセスユニットは、HRD初期化点とし

て選択され、ターゲットOLSの各レイヤに対してアクセスユニット0と参照される。

【0169】

その後のステップは、ターゲットOLS内のOLSレイヤインデックスTargetOlsLayerIdxを

有する各レイヤに適用される。BitstreamToDecodeに適用可能なols_hrd_parameters()

シンタックス構造およびsub_layer_hrd_parameters()シンタックス構造は次のように選択さ

れる。VPS内の(または外部メカニズムを通じて提供される)ols_hrd_idx[TargetOlsIdx]番

目のols_hrd_parameters()シンタックス構造が選択される。選択されたols_hrd_paramete

rs()シンタックス構造内で、BitstreamToDecodeがタイプIビットストリームである場合

、条件「if(general_vcl_hrd_params_present_flag)」の直後に来るsub_layer_hrd_paramet

ers(Htid)シンタックス構造が選択され、変数NalHrdModeFlagが0に等しくなるように設定

される。そうでない場合(BitstreamToDecodeがタイプIIビットストリームである場合)、条件「if(general_vcl_hrd_params_present_flag)」(この場合、変数NalHrdModeFlagは0

に等しくなるように設定される)または条件「if(general_nal_hrd_params_present_flag)

」(この場合、変数NalHrdModeFlagは1に等しくなるように設定される)のいずれかの直後

に来るsub_layer_hrd_parameters(Htid)シンタックス構造が選択される。BitstreamToDec

odeがタイプIIビットストリームであり、NalHrdModeFlagが0に等しいとき、フィルター

タNALユニットを除くすべての非VCL NALユニット、ならびにNALユニットストリームから

バイトストリームを形成するすべてのleading_zero_8bits、zero_byte、start_code_p

refix_one_3bytes、およびtrailing_zero_8bitsシンタックス要素は、存在するとき、Bits

trstreamToDecodeから破棄され、残りのビットストリームは、BitstreamToDecodeに割り

当てら

れる。

【0170】

decoding_unit_hrd_params_present_flagが1に等しいとき、CPBは、アクセスユニ

ットレベル(この場合、変数DecodingUnitHrdFlagは0に等しくなるように設定される)またはデ

10

20

30

40

50

コーディングユニットレベル(この場合、変数DecodingUnitHrdFlagは1に等しくなるように設定される)のいずれかで動作するようにスケジュールされる。そうでない場合、DecodingUnitHrdFlagは0に等しくなるように設定され、CPBはアクセスユニットレベルで動作するようにスケジュールされる。

【0171】

アクセスユニット0から始まるBitstreamToDecode内の各アクセスユニットについて、ア

クセスユニットに関連付けられ、TargetOlsIdxに適用されるバッファリング期間SEIメッセージ(BitstreamToDecode内に存在しているか、または外部メカニズムを通じて利用可能

10

である)が選択され、アクセスユニットに関連付けられ、TargetOlsIdxに適用されるピクチャタイミング期間SEIメッセージ(BitstreamToDecode内に存在しているか、または外部メカニズムを通じて利用可能である)が選択され、DecodingUnitHrdFlagが1に等しく、dec

oding_unit_cpb_params_in_pic_timing_sei_flagが0に等しいとき、アクセスユニット内

のデコーディングユニットに関連付けられ、TargetOlsIdxに適用されるデコーディングユニット情報SEIメッセージ(BitstreamToDecode内に存在しているか、または外部メカニズムを通じて利用可能である)が選択される。

20

【0172】

各適合性テストは、上記ステップの各々における1つのオプションの組合せを含む。ステップに対して複数のオプションがあるとき、任意の特定の適合性テストに対して、1つのオプションのみが選択される。すべてのステップの可能なすべての組合せが、適合性テストのセット全体を成す。テスト対象の各オペレーションポイントについて、実行されるべきビットストリーム適合性テストの数は、 $n0 \cdot n1 \cdot n2 \cdot n3$ に等しく、 $n0$ 、 $n1$ 、 $n2$ 、および n

3の値は、次のように指定される。 $n1$ は $hrd_cpb_cnt_minus1+1$ に等しい。 $n1$ は、バッファ

30

リング期間SEIメッセージに関連付けられているBitstreamToDecodeにおけるアクセスユニットの数である。 $n2$ は、次のように導出される。BitstreamToDecodeがタイプIビット

ストリームである場合、 $n0$ は1に等しい。そうでない場合(BitstreamToDecodeがタイプIIビット

ストリームである場合)、 $n0$ は2に等しい。 $n3$ は、次のように導出される。decoding_u

ni

t_hrd_params_present_flagが0に等しい場合、 $n3$ は1に等しい。そうでない場合、 $n3$ は2に

40

等しい。

【0173】

HRDは、ビットストリーム抽出器(任意選択で存在する)、コーディング済みピクチャバッファ(CPB)、瞬時デコーディングプロセス、各レイヤに対するサブDPBを概念的に含むデコード済みピクチャバッファ(DPB)、および出カクローピングを含んでいる。各ビットストリーム適合性テストについて、CPBサイズ(ビット数)は、CpbSize[Htid][ScIdx]であり、各レイヤに対するDPBパラメータmax_dec_pic_buffering_minus1[Htid]、max_number_reord

er_pics[Htid]、およびMaxLatencyPictures[Htid]は、レイヤが独立レイヤであるかどうか

50

か、およびレイヤがターゲットOLSの出力レイヤであるかどうかに応じてそのレイヤに適用されるdpb_parameters()シンタックス構造内に見つかるか、またはそれから導出される。

【0174】

HRDは次のように動作し得る。HDRは、デコーディングユニット0で初期化され、CPBおよびDPBの各サブDPBの両方は空に設定される(各サブDPBに対するサブDPBの充足量が0に等し

くなるように設定されている)。初期化後、HRDは、その後のバッファリング期間SEIメッセージによって再び初期化されることはあり得ない。指定された到着スケジュールに従って各CPBに流入するデコーディングユニットに関連付けられているデータは、仮想ストリームスケジューラ(HSS)によって配信される。各デコーディングユニットに関連付けられているデータが除去され、デコーディングユニットのCPB除去時間に瞬時デコーディングプロセスによって瞬時にデコードされる。各デコード済みピクチャは、DPB内に置かれる。デコード済みピクチャは、デコード済みピクチャがインター予測参照にもはや必要なくなり、また出力にもはや必要なくなったときにDPBから除去される。

10

【0175】

デコード済みピクチャバッファの例示的な動作は次のとおりである。これらの指定は、選択されたデコード済みピクチャバッファ(DPB)パラメータの各セットに独立して適用され得る。デコード済みピクチャバッファは、概念的にはサブDPBを含み、各サブDPBは、1つのレイヤのデコード済みピクチャを記憶するためのピクチャストレージバッファを含む。ピクチャストレージバッファの各々は、参照用使用とマークされるか、または後の出力のために保持されるデコード済みピクチャを含み得る。本明細書において説明されているプロセスは、順次適用され、OLS内のレイヤのnuh_layer_id値の増加順に、OLS内の最下位

20

レイヤから始まって各レイヤについて独立して適用される。特定のレイヤに対してこれらのプロセスが適用される時、特定のレイヤに対するサブDPBのみが影響を受ける。これらのプロセスの説明では、DPBは、特定のレイヤに対するサブDPBを指し、その特定のレイ

ヤは、現在のレイヤと称される。

30

【0176】

出力タイミングDPBの動作において、同一アクセスユニット内のPicOutputFlagが1に等

しいデコード済みピクチャは、デコード済みピクチャのnuh_layer_id値の昇順に連続的に出力される。ピクチャnおよび現在のピクチャを、nuh_layer_idの特定の値に対するアクセスユニットnのコーディング済みピクチャまたはデコード済みピクチャとし、nを非負の整数値とする。現在のピクチャのデコード前のDPBからのピクチャの除去は次のように生じる。現在のピクチャのデコード前のDPBからのピクチャの除去(ただし、現在のピクチャの最初のスライスのスライスヘッダの解析後)は、アクセスユニットnの最初のデコーディングユニットのCPB除去時間において実質的に瞬時に起こり(現在のピクチャを含む)、次のように進む。

40

【0177】

参照ピクチャリスト構築に対するデコーディングプロセスが呼び出され、参照ピクチャマーキングに対するデコーディングプロセスが呼び出される。現在のAUが、AU0でないコーディング済みビデオシーケンス開始(CVSS)AUであるとき、次の順序付けられたステップが適用される。変数NoOutputOfPriorPicsFlagは、次のようにテスト対象のデコーダにつ

いて導出される。現在のAU内の任意のピクチャについて導出されたpic_width_max_in_luma

a_samples、pic_height_max_in_luma_samples、chroma_format_idc、separate_

50

colour_plane_flag、bit_depth_luma_minus8、bit_depth_chroma_minus8またはmax_dec_pic_buffering_minus1[Htid]の値が、同じCLVS内の先行するピクチャについてそれぞれ導出されたpic_width_in_luma_samples、pic_height_in_luma_samples、chroma_format_idc、separate_colour_plane_flag、bit_depth_luma_minus8、bit_depth_chroma_minus8、またはmax_dec_pic_buffering_minus1[Htid]の値と異なる場合、NoOutputOfPriorPicsFlagは、no_output_of_prior_pics_flagの値に関係なく、テスト対象のデコーダによって1に設定されることがある。NoOutputOfPriorPicsFlagをno_output_of_prior_pics_flagに等しくなるように設定することはこれらの条件の下で好ましい場合があるが、テスト対象のデコーダは、この場合にNoOutputOfPriorPicsFlagを1に設定することを許される。そうでない場合、NoOutputOfPriorPicsFlagは、no_output_of_prior_pics_flagに等しくなるように設定される。

10

20

【0178】

テスト対象のデコーダについて導出されたNoOutputOfPriorPicsFlagの値はHRDに対して適用され、その結果、NoOutputOfPriorPicsFlagの値が1に等しいとき、DPB内のすべてのピクチャストレージバッファは、それらが含むピクチャの出力なしで空にされ、DPB充足量は0に等しくなるように設定される。DPB内の任意のピクチャkについて次の条件の両方が真であるとき、DPB内のそのようなすべてのピクチャkはDPBから除去される。ピクチャkは、参照用未使用としてマークされ得るか、またはピクチャkは、0に等しいPictureOutputFlagを有することができるか、もしくはDPB出力時間は、現在のピクチャnの最初のデコーディングユニット(デコーディングユニットmと表す)のCPB除去時間以下であり、DpbOutputTime[k]はDuCpbRemovalTime[m]以下である。DPBから除去される各ピクチャについて、DPB充足量は、1だけ減分される。

30

【0179】

出力順DPBの動作は次のとおりであるものとしてよい。これらのプロセスは、選択されたデコード済みピクチャバッファ(DPB)パラメータの各セットに独立して適用され得る。デコード済みピクチャバッファは、概念的にはサブDPBを含み、各サブDPBは、1つのレイヤのデコード済みピクチャを記憶するためのピクチャストレージバッファを含む。ピクチャストレージバッファの各々は、参照用使用とマークされるか、または将来の出力のために保持されるデコード済みピクチャを含む。現在のピクチャのデコード前にDPBからピクチャを出力し除去するためのプロセスが呼び出され、それに続いて現在のデコード済みピクチャのマーキングおよび記憶のためのプロセスが呼び出され、それに続いて最後に追加バンピングのプロセスが呼び出される。これらのプロセスは、OLS内の最下位レイヤから始まって、OLS内のレイヤのnuh_layer_id値の増加順に、各レイヤについて独立して適用される。特定のレイヤに対してこれらのプロセスが適用されるとき、特定のレイヤに対するサブDPBのみが影響を受ける。

40

50

【 0 1 8 0 】

出力順序DPBの動作において、出力タイミングDPBの動作と同じように、同じアクセスユニット内でPicOutputFlagが1に等しいデコード済みピクチャも、デコード済みピクチャのnuh_layer_id値の昇順に連続的に出力される。ピクチャnおよび現在のピクチャを、nuh_layer_idの特定の値に対するアクセスユニットnのコーディング済みピクチャまたはデコード済みピクチャとし、nを非負の整数値とする。DPBからのピクチャの出力および除去について次のように説明される。

【 0 1 8 1 】

現在のピクチャのデコード前(ただし、現在のピクチャの最初のスライスのスライスヘッダを解析した後)のDPBからのピクチャの出力および除去は、現在のピクチャを含むアクセスユニットの最初のデコーディングユニットがCPBから除去されたときに実質的に瞬時に生じ、次のように進む。参照ピクチャリスト構築に対するデコーディングプロセスが呼び出され、参照ピクチャマーキングに対するデコーディングプロセスが呼び出される。現在のAUが、AU0でないCVSS AUである場合に、次の順序付けられたステップが適用される。

変数NoOutputOfPriorPicsFlagは、次のようにテスト対象のデコーダについて導出される

。現在のAUの任意のピクチャについて導出されたpic_width_max_in_luma_samples、pic_height_max_in_luma_samples、chroma_format_idc、separate_colour_plane_flag、bit_depth_luma_minus8、bit_depth_chroma_minus8またはmax_dec_pic_buffering_minus1[Htid]の値が、同じCLVS内の先行するピクチャについてそれぞれ導出されたpic_width_in_luma_samples、pic_height_in_luma_samples、chroma_format_idc、separate_colour_plane_flag、bit_depth_luma_minus8、bit_depth_chroma_minus8、またはmax_dec_pic_buffering_minus1[Htid]の値と異なる場合、NoOutputOfPriorPicsFlagは、no_output_of_prior_pics_flagの値に関係なく、テスト対象のデコーダによって1に設定されることがある。

【 0 1 8 2 】

NoOutputOfPriorPicsFlagをno_output_of_prior_pics_flagに等しくなるように設定する

ことはこれらの条件の下で好ましいことがあるが、テスト対象のデコーダは、この場合にNoOutputOfPriorPicsFlagを1に設定することを許される。そうでない場合、NoOutputOfPriorPicsFlagは、no_output_of_prior_pics_flagに等しくなるように設定される。テスト対象のデコーダについて導出される変数NoOutputOfPriorPicsFlagの値は、次のようにRDについて適用される。NoOutputOfPriorPicsFlagが1に等しい場合、DPB内のすべてのピクチャストレージバッファは、それらが含むピクチャの出力なしに空にされ、DPB充足量は、0に等しくなるように設定される。そうでない場合(NoOutputOfPriorPicsFlagが0に等しい場合)、出力に必要なしおよび参照用未使用とマークされているピクチャを含むすべて

のピクチャは、出力に必要なしおよび参照用未使用とマークされているピクチャを含むすべて

でのピクチャストレージバッファは(出力なしで)空にされ、DPB内の空でないすべてのピクチャストレージバッファは、バンピングを繰り返し呼び出すことによって空にされ、DPB充足量は、0に等しくなるように設定される。

【0183】

そうでない場合(現在のピクチャがCLVSSピクチャではない場合)、出力に必要なしおよび参照用未使用とマークされているピクチャを含むすべてのピクチャストレージバッファが空にされる(出力なしで)。空にされた各ピクチャストレージバッファについて、DPB充足量が1だけ減分される。次の条件のうちの1つまたは複数が真であるとき、次の条件のどれも真でなくなるまで、空にされる各追加のピクチャストレージバッファについてDPB充足量をさらに1だけ減分しながらバンピングプロセスが繰り返し呼び出される。出力に必要とマークされているDPB内のピクチャの数は、max_num_reorder_pics[Htid]よりも大きい。

max_latency_increase_plus1[Htid]は0に等しくなく、関連付けられている変数PicLatencyCountがMaxLatencyPictures[Htid]以上である出力に必要とマークされている少なくとも1つのピクチャがDPB内にある。DPB内のピクチャの数は、max_dec_pic_buffering_min

us1[Htid]+1以上である。

【0184】

一例において、追加のバンピングは、次のように生じ得る。指定されているプロセスは、現在のピクチャを含むアクセスユニットnの最後のデコーディングユニットがCPBから除去されたときに実質的に瞬時に生じ得る。現在のピクチャが1に等しいPictureOutputFlag

を有するとき、出力に必要とマークされ、出力順序で現在のピクチャに続くDPB内の各ピクチャについて、関連付けられている変数PicLatencyCountは、PicLatencyCount+1に等しくなるように設定される。次も適用される。現在のデコード済みピクチャが1に等しいPictureOutputFlagを有する場合、現在のデコード済みピクチャは、出力に必要とマークされ、関連付けられている変数PicLatencyCountは、0に等しくなるように設定される。そうで

ない場合(現在のデコード済みピクチャは0に等しいPictureOutputFlagを有する場合)、現在のデコード済みピクチャは出力に必要なしとマークされる。

【0185】

次の条件のうちの1つまたは複数が真であるとき、バンピングプロセスは、次の条件のうちのどれも真でなくなるまで繰り返し呼び出される。出力に必要とマークされているDPB内のピクチャの数は、max_num_reorder_pics[Htid]よりも大きい。max_latency_increas

e_plus1[Htid]は0に等しくなく、関連付けられている変数PicLatencyCountがMaxLate

ncyPictures[Htid]以上である出力に必要とマークされている少なくとも1つのピクチャがDPB内にある。

【0186】

バンピングプロセスは、次の順序付けられたステップを含む。出力に関して最初である1つまたは複数のピクチャは、出力に必要とマークされたDPB内のすべてのピクチャのPicOrderCntValの最小値を有するものとして選択される。これらのピクチャの各々は、昇nuh

layer_id順で、ピクチャに対する適合性クロッピングウィンドウを使用して、クロップされ、クロップ済みピクチャが出力され、ピクチャは出力に必要なしとマークされる。参照

10

20

30

40

50

用未使用とマークされたピクチャを含み、クロップされ出力されたピクチャのうちの1つであった各ピクチャストレージバッファは空にされ、関連付けられているサブDPBの充足量が1だけ減分される。同じCVSに属し、バンピングプロセスによって出力される任意の2つのピクチャpicAおよびpicBについて、picAがpicBより早く出力されるとき、picAのPicOrderCntValの値はpicBのPicOrderCntValの値より小さい。

【0187】

例示的なサブビットストリーム抽出プロセスは次のとおりである。このプロセスへの入力、ビットストリームinBitstream、ターゲットOLSインデックスtargetOlsIdx、および

ターゲット最高TemporalId値tIdTargetである。このプロセスの出力は、サブビットストリームoutBitstreamである。ビットストリーム適合性の要件は、任意の入力ビットストリームについて、ビットストリーム、VPSによって指定されるOLSのリストへのインデックスに等しいtargetOlsIdx、および0から6までの範囲内の任意の値に等しいtIdTargetを入力とするこのプロセスからの出力であり、以下の条件を満たす出力サブビットストリームが適合するビットストリームであるものとするのであり、出力サブビットストリームは、LayerIdInOls[targetOlsIdx]のnuh_layer_id値の各々に等しいnuh_layer_idを有す

る少なくとも1つのVCL NALユニットを含む。出力サブビットストリームは、tIdTargetに等しいTemporalIdを有する少なくとも1つのVCL NALユニットを含む。適合するビット

ストリームは、0に等しいTemporalIdを有する1つまたは複数のコーディング済みスライスNAL

ユニットを含むが、0に等しいnuh_layer_idを有するコーディング済みスライスNALユニットを有しなくてもよい。

【0188】

出力サブビットストリームOutBitstreamは、次のように導出される。ビットストリームoutBitstreamは、ビットストリームinBitstreamと同一になるように設定される。tIdTarget

より大きいTemporalIdを有するすべてのNALユニットはoutBitstreamから除去される。

リストLayerIdInOls[targetOlsIdx]に含まれないnuh_layer_idを有するすべてのNAL

ユニットはoutBitstreamから除去される。1に等しいnesting_ols_flagを有するスケーラブルネ스팅SEIメッセージを含み、0からnesting_num_olss_minus1までの範囲内にi

の値がなく、NestingOlsIdx[i]はtargetOlsIdxに等しい、すべてのSEI NALユニットはoutB

itsreamから除去される。targetOlsIdxが0より大きいとき、0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しいpayloadTypeを有する非スケーラブルネ스팅SEIメッセージを含むすべてのSEI NALユニットはoutBitstream

から除去される。

【0189】

例示的なスケーラブルネ스팅SEIメッセージシンタックスは次のとおりである。

【0190】

10

20

30

40

50

【表 7】

scalable_nesting(payloadSize) {	記述子	
nesting_ols_flag	u(1)	
if(nesting_ols_flag) {		
nesting_num_olss_minus1	ue(v)	
for(i = 0; i <= nesting_num_olss_minus1; i++)		
nesting_ols_idx_delta_minus1[i]	ue(v)	
} else {		10
nesting_all_layers_flag	u(1)	
if(!nesting_all_layers_flag) {		
nesting_num_layers_minus1	ue(v)	
for(i = 1; i <= nesting_num_layers_minus1; i++)		
nesting_layer_id[i]	u(6)	
}		
}		
nesting_num_seis_minus1	ue(v)	20
while(!byte_aligned())		
nesting_zero_bit /* equal to 0 */	u(1)	
for(i = 0; i <= nesting_num_seis_minus1; i++)		
sei_message()		
}		

【 0 1 9 1 】

例示的な一般SEIペイロードセマンティクスは次のとおりである。非スケーラブルネストSEIメッセージの適用可能なレイヤまたはOLS上で、次が適用される。非スケーラブルネストSEIメッセージについて、payloadTypeが0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しいとき、非スケーラブルネストSEIメッセージは、0番目のOLSにのみ適用される。非スケーラブルネストSEIメッセージについて、payloadTypeがVclAssociatedSeiListのうちの任意の値に等しいとき、非スケーラブルネストSEIメッセージは、VCL NALユニットがSEIメッセージを含むようになるまでSEI

NALユニットのthe_nuh_layer_idに等しいnuh_layer_idを有するレイヤにのみ適用される

。

【 0 1 9 2 】

ビットストリーム適合性の要件は、SEI NALユニットのnuh_layer_idの値に次の制限が適用されることであってよい。非スケーラブルネストSEIメッセージが、0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しいpayloadTypeを有するとき、非スケーラブルネストSEIメッセージを含むSEI NALユニットは、

vps_layer_id[0]に等しいnuh_layer_idを有するものとする。非スケーラブルネストSEIメ

10

20

30

40

50

ッセージが、VclAssociatedSeiListのうちの任意の値に等しいpayloadTypeを有するとき

、非スケーラブルネストSEIメッセージを含むSEI NALユニットは、SEI NALユニットに関連付けられているVCL NALユニットのnuh_layer_idの値に等しいnuh_layer_idを有するも

のとする。スケーラブルネスティングSEIメッセージを含むSEI NALユニットは、スケーラブルネストSEIメッセージが適用されるすべてのレイヤのnuh_layer_idの最低値(スケーラブルネスティングSEIメッセージのnesting_ols_flagが0に等しいとき)またはスケーラブルネストSEIメッセージが適用されるOLS内のすべてのレイヤのnuh_layer_idの最低値(スケーラブルネスティングSEIメッセージのnesting_ols_flagが1に等しいとき)に等しいnuh

10

_layer_idを有するべきである。

【0193】

例示的なスケーラブルネスティングSEIメッセージセマンティクスは次のとおりである。スケーラブルネスティングSEIメッセージは、SEIメッセージを特定のOLSと、または特定のレイヤと関連付けるメカニズムを提供する。スケーラブルネスティングSEIメッセージは、1つまたは複数のSEIメッセージを含む。スケーラブルネスティングSEIメッセージに含まれるSEIメッセージは、スケーラブルネストSEIメッセージとも称される。ビットストリーム適合性の要件は、スケーラブルネスティングSEIメッセージ内にSEIメッセージを含めることに次の制限が適用されることであってよい。

20

【0194】

132(デコード済みピクチャハッシュ)または133(スケーラブルネスティング)に等しいpayloadTypeを有するSEIメッセージは、スケーラブルネスティングSEIメッセージに含まれないものとしてよい。スケーラブルネスティングSEIメッセージがバッファリング期間、ピクチャタイミング、またはデコーディングユニット情報SEIメッセージを含むとき、スケーラブルネスティングSEIメッセージは、0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しくないpayloadTypeを有する任意の他のSEIメッセージも含まないものとする。

【0195】

ビットストリーム適合性の要件は、スケーラブルネスティングSEIメッセージを含むSEI NALユニットのnal_unit_typeの値に次の制限が適用されることであるものとしてよい。スケーラブルネスティングSEIメッセージが、0(バッファリング期間)、1(ピクチャタイミング)、130(デコーディングユニット情報)、145(従属RAP指示)、または168(フレームフ

30

ィールド情報)に等しいpayloadTypeを有するSEIメッセージを含むとき、スケーラブルネスティングSEIメッセージを含むSEI NALユニットは、PREFIX_SEI_NUTに等しいnal_unit_type

eを有するべきである。

【0196】

1に等しいnesting_ols_flagは、スケーラブルネストSEIメッセージが特定のOLSに適用されることを指定する。0に等しいnesting_ols_flagは、スケーラブルネストSEIメッセージが特定のレイヤに適用されることを指定する。ビットストリーム適合性の要件は、nesting_ols_flagの値に次の制限が適用されることであってよい。スケーラブルネスティングSEIメッセージが、0(バッファリング期間)、1(ピクチャタイミング)、または130(デコーディングユニット情報)に等しいpayloadTypeを有するSEIメッセージを含むとき、nesting_ols_flag

40

_ols_flagの値は1に等しいものとする。スケーラブルネスティングSEIメッセージが、VclAssociatedSeiList内の値に等しいpayloadTypeを有するSEIメッセージを含むとき、nesting_ols_flag

_ols_flagの値は0に等しいものとする。nesting_num_olss_minus1+1は、スケーラブル

50

ルネ

ストSEIメッセージが適用されるOLSの数を指定する。nesting_num_olss_minus1の値は、0

からTotalNumOlss-1までの範囲内にあるものとする。

【0197】

nesting_ols_idx_delta_minus1[i]は、nesting_ols_flagが1に等しいときにスケラブル

ルネストSEIメッセージが適用されるi番目のOLSのOLSインデックスを指定する変数NestingOlsIdx

gOlsIdx[i]を導出するために使用される。nesting_ols_idx_delta_minus1[i]の値は、0

10

からTotalNumOlss-2までの範囲内にあるべきである。変数NestingOlsIdx[i]は、次のように

導出され得る。

if(i==0)

NestingOlsIdx[i]=nesting_ols_idx_delta_minus1[i]

else

NestingOlsIdx[i]=NestingOlsIdx[i-1]+nesting_ols_idx_delta_minus1[i]+1

【0198】

1に等しいnesting_all_layers_flagは、スケラブルネストSEIメッセージが現在のSEI NAL

20

ユニットのnuh_layer_id以上のnuh_layer_idを有するすべてのレイヤに適用されるこ

とを指定する。0に等しいnesting_all_layers_flagは、スケラブルネストSEIメッセージが現在のSEI NAL

ユニットのnuh_layer_id以上のnuh_layer_idを有するすべてのレイヤに適用され得るか、または適用され得ないことを指定する。nesting_num_layers_minus1+

30

1は、スケラブルネストSEIメッセージが適用されるレイヤの数を指定する。nesting_num_layers_minus1の値は、0からvps_max_layers_minus1-GeneralLayerIdx[nuh_layer_id]

までの範囲内にあるものとし、nuh_layer_idは、現在のSEI NALユニットのnuh_layer_idである。nesting_layer_id[i]は、nesting_all_layers_flagが0に等しいときにスケラブルネストSEIメッセージが適用されるi番目のレイヤのnuh_layer_id値を指定する。nesting_layer_id[i]の値は、nuh_layer_idより大きいものとし、nuh_layer_idは現在のSEI NAL

ALユニットのnuh_layer_idである。

【0199】

nesting_ols_flagが0に等しいとき、スケラブルネストSEIメッセージが適用されるレイヤの数を指定する変数NestingNumLayers、およびスケラブルネストSEIメッセージ

40

が適用されるレイヤnuh_layer_id値のリストを指定する、0からNestingNumLayers-1の範囲

内のiに対するリストNestingLayerId[i]は、次のように導出され得、nuh_layer_idは現在の

のSEI NALユニットのnuh_layer_idである。

if(nesting_all_layers_flag) {

NestingNumLayers=vps_max_layers_minus1+1-GeneralLayerIdx[nuh_layer_id]

for(i=0; i NestingNumLayers; i++)

50

```

    NestingLayerId[i]=vps_layer_id[GeneralLayerIdx[nuh_layer_id]+i]
} else {
    NestingNumLayers=nesting_num_layers_minus1+1
    for(i=0; i < NestingNumLayers; i++)
        NestingLayerId[i]=(i==0) ? nuh_layer_id: nesting_layer_id[i]
}

```

【0200】

nesting_num_seis_minus1+1は、スケーラブルネストSEIメッセージの数を指定する。nesting_num_seis_minus1の値は、0から63までの範囲内にあるものとする。nesting_zero_b
itは、0に等しいものとする。

【0201】

図8は、例示的なビデオコーディングデバイス800の概略図である。ビデオコーディングデバイス800は、本明細書において説明されているような開示された例/実施形態を実施するのに適している。ビデオコーディングデバイス800は、ネットワークの上流および/または下流でデータ通信を行うための送信器および/または受信器を含む、下流ポート820、上流ポート850、および/またはトランシーバユニット(Tx/Rx)810を備える。ビデオコーディ

ングデバイス800は、データを処理するための論理ユニットおよび/または中央演算処理装置(CPU)を含むプロセッサ830と、データを記憶するためのメモリ832とを備える。ビデオコーディングデバイス800は、電気、光、またはワイヤレス通信ネットワークを介してデータ通信を行うために、上流ポート850および/または下流ポート820に結合されている電気、光-電気(OE)コンポーネント、電気-光(EO)コンポーネント、および/またはワイヤレス通信コンポーネントも備え得る。ビデオコーディングデバイス800は、ユーザにデータを伝達し、ユーザからデータを受け取るための入力および/または出力(I/O)デバイス860も含み得る。I/Oデバイス860は、ビデオデータを表示するためのディスプレイ、音声データを出力するためのスピーカなどの出力デバイスを含み得る。I/Oデバイス860は、キーボード、マウス、トラックボールなどの入力デバイス、および/またはそのような出力デバイスをインタラクティブに操作するための対応するインターフェースも含み得る。

【0202】

プロセッサ830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830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)、フィールドプログラマブルゲートアレイ(FPGA)、特定用途向け集積回路(ASIC)、およびデジタルシグナルプロセッサ(DSP)として実装されてもよい。プロセッサ830は、下流ポート820、Tx/Rx810、上流ポート850、およびメモリ832と通信する。プロセッサ830は、コーディ

ングモジュール814を備える。コーディングモジュール814は、方法100、900、および1000

などの、本明細書において説明されている開示された実施形態を実装するものであり、マルチレイヤビデオシーケンス600、ビットストリーム700、および/またはサブビットストリーム701を採用し得る。コーディングモジュール814は、本明細書において説明されてい

る任意の他の方法/メカニズムも実装し得る。さらに、コーディングモジュール814は、コーデックシステム200、エンコーダ300、デコーダ400、および/またはHRD500を実装し得る

。たとえば、コーディングモジュール814は、サイマルキャストレイヤおよびVPSなしを含むビットストリームをエンコードし、抽出し、および/またはデコードするために採用され得る。さらに、コーディングモジュール814は、サブビットストリーム抽出の一部として抽出されるVPSへの参照に基づきエラーを回避するために様々なシンタックス要素およ

10

20

30

40

50

び/または変数を設定し、および/または推論するために採用され得る。それに加えて、コーディングモジュール814は、CLVSを含むレイヤがサイマルキャストレイヤであるときに

n
uh_layer_idがCLVSのnuh_layer_idと同じになるように制約するために採用され、それ

に
よりSPSの誤った抽出を防ぐことができる。したがって、コーディングモジュール814は、上で説明されている問題のうちの1つまたは複数に対処するためのメカニズムを遂行するように構成され得る。したがって、コーディングモジュール814は、ビデオデータをコーディングするときに追加の機能性および/またはコーディング効率を提供することをビデオコーディングデバイス800に行わせる。それゆえに、コーディングモジュール814は、

10

ビデオコーディングデバイス800の機能性を改善し、さらにはビデオコーディング技術に特有の問題に対処する。さらに、コーディングモジュール814は、ビデオコーディングデバイス800の異なる状態への変換を行う。代替的に、コーディングモジュール814は、メモリ

832に記憶され、プロセッサ830によって実行される命令として(たとえば、非一時的媒体に記憶されるコンピュータプログラム製品として)実装され得る。

【0203】

メモリ832は、ディスク、テープドライブ、ソリッドステートドライブ、リードオンリーメモリ(ROM)、ランダムアクセスメモリ(RAM)、フラッシュメモリ、三値連想メモリ(T

20

CA
M)、スタティックランダムアクセスメモリ(SRAM)などの1つまたは複数のメモリタイプを含む。メモリ832はプログラムを、そのようなプログラムが実行のために選択されたとき、記憶し、プログラムの実行中に読み出される命令およびデータを記憶するために、オーバーフローデータ記憶装置デバイスとして使用され得る。

【0204】

図9は、サイマルキャストレイヤに対するサブビットストリーム抽出プロセス729においてSPS713を保持することをサポートするために、ビットストリーム700などのビットストリームにマルチレイヤビデオシーケンスをエンコードする例示的な方法900のフローチャートである。方法900は、方法100を実行するとき、コーデックシステム200、エンコーダ3

30

00、および/またはビデオコーディングデバイス800などのエンコーダによって採用され得る。さらに、方法900は、HRD500上で動作してもよく、したがって、マルチレイヤビデオ

シーケンス600および/またはその抽出済みレイヤに対して適合性テストを実行してもよい。

【0205】

方法900は、エンコーダがビデオシーケンスを受信し、たとえばユーザ入力に基づき、そのビデオシーケンスをマルチレイヤビットストリームにエンコードすることを決定するときに開始し得る。ステップ901で、エンコーダは、コーディング済みピクチャのCLVSをビットストリーム内のレイヤにエンコードする。コーディング済みピクチャは、VCL NALユニットのセット内に収容される。たとえば、エンコーダは、ビデオシーケンス内のピクチャを1つまたは複数のCLVSとして1つまたは複数のレイヤにエンコードし、レイヤ/CLV

40

S
をマルチレイヤビットストリームにエンコードすることができる。したがって、ビットストリームは、1つまたは複数のレイヤと1つまたは複数のCLVSとを含む。レイヤは、同じレイヤIdを有するVCL NALユニットおよび関連付けられている非VCL NALユニットのセットを

含み得る。さらに、CLVSは、同じnuh_layer_id値を有するコーディング済みピクチャのシ

50

ーケンスである。具体例として、VCL NALユニットは、nuh_layer_idによって識別されるレイヤに関連付けられ得る。具体的には、VCL NALユニットのセットは、VCL NALユニット

のセットがすべてnuh_layer_idの特定の値を有するときにレイヤの一部である。レイヤは、エンコード済みピクチャのビデオデータさらにはそのようなピクチャをコーディングするために使用される任意のパラメータセットを収容するVCL NALユニットのセットを含み得る。そのようなパラメータは、VPS、SPS、PPS、ピクチャヘッダ、スライスヘッダ、または他のパラメータセットもしくはシンタックス構造に含まれ得る。レイヤの1つまたは複数は、出力レイヤであってもよく、したがって、レイヤに含まれるCLVSの1つまたは複数は、出力され得る。出力レイヤでないレイヤは参照レイヤと称され、出力レイヤを再構成することをサポートするようにエンコードされるが、そのようなサポートするレイヤ/CLVSは、デコーダでの出力を意図されていない。この方法で、エンコーダは、要求があったときにデコーダに伝送するためにレイヤ/CLVの様々な組合せをエンコードすることができる。レイヤ/CLVSは、ネットワーク条件、ハードウェア能力、および/またはユーザ設定に応じて、デコーダがビデオシーケンスの異なる表現を取得することを可能にする望み通りに伝送され得る。本発明の例では、レイヤの少なくとも1つは、レイヤ間予測を使用しないサイマルキャストレイヤである。したがって、少なくとも1つのCLVSは、サイマルキャストレイヤに含まれる。

【0206】

ステップ903で、エンコーダは、SPSをビットストリームにエンコードすることができる。SPSは、1つまたは複数のCLVSによって参照される。具体的には、SPSは、CLVSを含むレイヤがレイヤ間予測を使用しないときCLVSのnuh_layer_id値に等しいnuh_layer_id値を有する。さらに、SPSは、CLVSを含むレイヤがレイヤ間予測を使用するときCLVSのnuh_layer_id値以下であるnuh_layer_id値を有する。この方法で、SPSは、CLVSを参照する任意のサイマルキャストレイヤと同じnuh_layer_idを有する。さらに、SPSは、SPSを参照する任意の非サイマルキャストレイヤのnuh_layer_id以下であるnuh_layer_idを有する。この方法で、SPSは、複数のCLVS/レイヤから参照され得る。しかしながら、SPSは、サブビットストリーム抽出がサイマルキャストレイヤであるCLVS/レイヤに適用されるときにサブビットストリーム抽出によって除去されない。

【0207】

SPSは、vps_independent_layer_flagなどの、SPSによって参照されるVPSに対するIDの値を指定するsps_video_parameter_set_idを含むようにエンコードされ得る。それゆえに、sps_video_parameter_set_idは、sps_video_parameter_set_idが0より大きいときにSPSによって参照されるVPSのvps_video_parameter_set_idの値を指定する。さらに、SPSはVP Sを参照せず、sps_video_parameter_set_idが0に等しいときにSPSを参照する各コーディング済みレイヤビデオシーケンスをデコードする際にVPSはいっさい参照されない。したがって、sps_video_parameter_set_idは、sps_video_parameter_set_idがサイマルキャストレイヤに含まれるコーディング済みレイヤビデオシーケンスによって参照されるSPSが

10

20

30

40

50

ら取得されるときに0に設定され、および/または0であると推論される。

【0208】

HRDは、適合性テストを実行するために採用され、したがってVVCなどの、規格への適合性についてビットストリームをチェックするために採用され得る。HRDは、sps_video_parameter_set_idが0に等しいとき、GeneralLayerIdx[nuh_layer_id]を0に等しくなるように

設定し、および/または0に等しいと推論することができる。GeneralLayerIdx[nuh_layer_id]

は、対応するレイヤに対する現在のレイヤインデックスに等しく、したがって、現在のレイヤインデックスを示す。それゆえに、サイマルキャストレイヤに対する現在のレイヤインデックスは、0になるように設定され/0と推論される。さらに、HRDは、sps_video_

parameter_set_idが0に等しいときにvps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]の値が1に等しいと推論し得る。具体的には、vps_independent_layer_flag

は、対応するレイヤがレイヤ間予測を使用するかどうかを指定する。vps_independent_layer_flag[i]

は、VPSに含まれ、i番目のレイヤがレイヤ間予測を使用することを示すように0に設定され、またはi番目のレイヤがレイヤ間予測を使用しないことを示すように1に設定され得る。それゆえに、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]

は、インデックスGeneralLayerIdx[nuh_layer_id]を有する現在のレイヤがレイヤ間予測を使用するかどうかを指定する。それゆえに、レイヤは、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しいときにレイヤ間予測を使用しない。現在のレイヤがサイマルレイヤであるときには、VPSは省略され、レイヤ間予測は採用されない。

したがって、sps_video_parameter_set_idが0に等しいときの1の値の推論は、サイマルキャストレイヤが、サイマルキャストレイヤに対するビットストリーム抽出中に抽出される、VPSへの参照を回避しながら、HRDにおいて、またデコード中に、適切に動作することを

確実にする。それゆえに、この推論は、VPSがサイマルキャストレイヤについて除去されたときにそうしなければ発生することになるであろうサブビットストリーム抽出エラーを回避する。次いで、HRDは、SPS、sps_video_parameter_set_id、GeneralLayerIdx[nuh_layer_id]、および/またはvps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]に基づきサイマルキャストレイヤ/CLVS内のVCL NALユニットからのコーディング済みピクチャをデコードしてデコード済みピクチャを生成し得る。それゆえに、HRDは、マルチレイヤビットストリームのサイマルキャストレイヤが、サイマルキャストレイヤに対するVPSの省略によって引き起こされる予期せぬエラーなしにビットストリームに適合しているかどうかを検証することができる。

【0209】

ステップ905で、エンコーダは、要求があったとき、デコーダに向けた通信のためにビットストリームを記憶しておくことができる。エンコーダは、サブビットストリーム抽出

を実行して、サイマルキャストレイヤを取得し、望ましいときにビットストリーム/サブビットストリームをエンコーダに向けて伝送することもできる。

【0210】

図10は、ビットストリーム700などの、マルチレイヤビットストリームから抽出されたサイマルキャストレイヤを含む、サブビットストリーム701などの、ビットストリームからのビデオシーケンスをデコードする例示的な方法1000のフローチャートであり、SPS713

は、サブビットストリーム抽出プロセス729中に保持されている。方法1000は、方法100を

実行するとき、コーデックシステム200、デコーダ400、および/またはビデオコーディングデバイス800などのデコーダによって採用され得る。さらに、方法1000は、HRD500など

の、HRDによって適合性についてチェックされた、マルチレイヤビデオシーケンス600、また

たはその抽出済みレイヤ上で採用され得る。

【0211】

方法1000は、デコーダが、たとえば方法900の結果として、マルチレイヤビットストリームから抽出されたサイマルキャストレイヤのコーディング済みビデオシーケンスを含むビットストリームを受信し始めるときに開始し得る。ステップ1001で、デコーダは、レイヤ内のCLVSを含むビットストリームを受信する。レイヤは、エンコーダまたは他の中間コンテンツサーバによってマルチレイヤビットストリームから抽出されたサイマルキャストレイヤであってよい。サイマルキャストレイヤは、コーディング済みピクチャのセットを含むCLVSを含む。たとえば、ビットストリームは、各コーディング済みピクチャが、nuh

layer_idによって識別されるようなサイマルキャストレイヤに関連付けられている1つまたは複数のVCL NALユニットのセットに含まれるコーディング済みピクチャを含む。レイヤは、同じレイヤIdを有するVCL NALユニットおよび関連付けられている非VCL NALユニット

のセットを含み得る。たとえば、レイヤは、エンコード済みピクチャのビデオデータさらにはそのようなピクチャをコーディングするために使用される任意のパラメータセットを収容するVCL NALユニットのセットを含み得る。それゆえに、VCL NALユニットのセット

は、VCL NALユニットのセットがすべてnuh_layer_idの特定の値を有するときにレイヤの一部である。さらに、CLVSは、同じnuh_layer_id値を有するコーディング済みピクチャの

シーケンスである。サイマルキャストレイヤは、出力レイヤでもあり、レイヤ間予測を採用しない。したがって、CLVSは、また、レイヤ間予測を採用しない。

【0212】

ビットストリームは、CLVSによって参照されるSPSも含む。SPSは、CLVSを含むレイヤが

レイヤ間予測を使用しないときCLVSのnuh_layer_id値に等しいnuh_layer_id値を有する。

さらに、SPSは、CLVSを含むレイヤがレイヤ間予測を使用するときCLVSのnuh_layer_id

値以下であるnuh_layer_id値を有する。この方法で、SPSは、CLVSを参照する任意のサイ

マルキャストレイヤと同じnuh_layer_idを有する。さらに、SPSは、SPSを参照する任意の

非サイマルキャストレイヤのnuh_layer_id以下であるnuh_layer_idを有する。この方法で

10

20

30

40

50

、SPSは、複数のCLVS/レイヤから参照され得る。しかしながら、SPSは、サブビットストリーム抽出がサイマルキャストレイヤであるCLVS/レイヤに適用されるときにエンコーダ/コンテンツサーバにおいてサブビットストリーム抽出によって除去されない。

【0213】

SPSは、`vps_independent_layer_flag`などの、SPSによって参照されるVPSに対するIDの

値を指定する`sps_video_parameter_set_id`を含む。それゆえに、`sps_video_parameter_set`

`id`は、`sps_video_parameter_set_id`が0より大きいときにSPSによって参照されるVPSの

`vps_video_parameter_set_id`の値を指定する。さらに、SPSはVPSを参照せず、`sps_video_p`

`arameter_set_id`が0に等しいときにSPSを参照する各コーディング済みレイヤビデオシーケンスをデコードする際にVPSはいっさい参照されない。したがって、`sps_video_parameter`

`set_id`は、`sps_video_parameter_set_id`がサイマルキャストレイヤに含まれるコーディ

ング済みレイヤビデオシーケンスによって参照されるSPSから取得されるときに0に設定され、および/または0であると推論される。さらに、ビットストリームは、ビットストリームがレイヤ間予測を採用しないCLVSを含むサイマルキャストレイヤのみを含むときにVPSを含まない。

【0214】

ステップ1003で、デコーダは、SPSに基づきCLVSからのピクチャをデコードして、デ

コード済みピクチャを生成することができる。たとえば、デコーダは、`sps_video_parameter`

`set_id`が0に等しいとき、`GeneralLayerIdx[nuh_layer_id]`を0に等しくなるように設定

し、および/または0に等しいと推論することができる。`GeneralLayerIdx[nuh_layer_id]`

は、対応するレイヤに対する現在のレイヤインデックスに等しく、したがって、現在のレイヤインデックスを示す。それゆえに、サイマルキャストレイヤに対する現在のレイヤインデックスは、0になるように設定され/0と推論される。さらに、デコーダは、`sps_video_parameter_set_id`が0に等しいときに`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id][nuh_layer_id]]`の値が1に等しいと推論し得る。具体的には、`vps_independent_layer_flag`

は

、対応するレイヤがレイヤ間予測を使用するかどうかを指定する。`vps_independent_layer`

`flag[i]`は、VPSに含まれ、*i*番目のレイヤがレイヤ間予測を使用することを示すように0

に設定され、または*i*番目のレイヤがレイヤ間予測を使用しないことを示すように1に設定され得る。それゆえに、`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]`

は、インデックス`GeneralLayerIdx[nuh_layer_id]`を有する現在のレイヤがレイヤ間予測を使用するかどうかを指定する。それゆえに、レイヤは、`vps_independent_layer_flag`

`[GeneralLayerIdx[nuh_layer_id]]`が1に等しいときにレイヤ間予測を使用しない。現在のレ

イヤがサイマルレイヤであるとき、VPSは受信済みビットストリームに含まれず、レイヤ間予測は採用されない。したがって、`sps_video_parameter_set_id`が0に等しいときの

10

20

30

40

50

1の

値の推論は、サイマルキャストレイヤが、サイマルキャストレイヤに対するビットストリーム抽出中に抽出される、VPSへの参照を回避しながら、デコード中に、適切に動作することを確実にする。それゆえに、この推論は、VPSがサイマルキャストレイヤについて除去されたときにそうしなければ発生することになるであろうサブビットストリーム抽出エラーを回避する。次いで、デコーダは、SPS、sps_video_parameter_set_id、General Layer

Idx[nuh_layer_id]、および/またはvps_independent_layer_flag[GeneralLayerIdx[nuh_

layer_id]]に基づきサイマルキャストレイヤ/CLVS内のVCL NALユニットからのコーディン

10

グ済みピクチャをデコードしてデコード済みピクチャを生成し得る。次いで、デコーダは、ステップ1005で、デコード済みビデオシーケンスの一部として表示するためにデコード済みピクチャを転送することができる。

【0215】

図11は、サイマルキャストレイヤに対するサブビットストリーム抽出プロセス729においてSPS713を保持することをサポートするためにマルチレイヤビデオシーケンスをビットストリーム700内にコーディングするための例示的なシステム1100の概略図である。システム1100は、コーデックシステム200、エンコーダ300、デコーダ400、および/またはビデ

20

オコーディングデバイス800などのエンコーダおよびデコーダによって実装され得る。さらに、システム1100は、HRD500を採用して、マルチレイヤビデオシーケンス600、ビット

ストリーム700、および/またはサブビットストリーム701に対して適合性テストを実行するものとしてよい。それに加えて、システム1100は、方法100、900、および/または1000

を実装するとき採用され得る。

【0216】

システム1100は、ビデオエンコーダ1102を備える。ビデオエンコーダ1102は、CLVSをビ

30

ットストリーム内のレイヤにエンコードするためのエンコーディングモジュール1105を含んでいる。エンコーディングモジュール1105は、さらにビットストリーム内にCLVSによっ

て参照されるSPSをエンコードするためのものであって、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer_id値に等しいnuh_layer_id値を有するよう制約される。

ビデオエンコーダ1102は、デコーダに向けて通信するためにビットストリームを記憶する記憶モジュール1106をさらに備える。ビデオエンコーダ1102は、ビデオデコーダ1110に向

けてビットストリームを伝送するための伝送モジュール1107をさらに備える。ビデオエンコーダ1102は、方法900のステップのいずれかを実行するようにさらに構成され得る。

40

【0217】

システム1100は、ビデオデコーダ1110も備える。ビデオデコーダ1110は、レイヤ内のCL

VSおよびCLVSによって参照されるSPSを含むビットストリームを受信するための受信モジュール1111を備え、SPSは、レイヤがレイヤ間予測を使用しないときCLVSのnuh_layer_id

値に等しいnuh_layer_id値を有する。ビデオデコーダ1110は、SPSに基づきCLVSからのコ

ーディング済みピクチャをデコードして、デコード済みピクチャを生成するためのデコー

50

ディングモジュール1113をさらに備える。ビデオデコーダ1110は、デコード済みビデオシ

ーケンスの一部として表示するためにデコード済みピクチャを転送する転送モジュール1115をさらに備える。ビデオデコーダ1110は、方法1000のステップのいずれかを実行するよ

うにさらに構成され得る。

【0218】

第1のコンポーネントは、第1のコンポーネントと第2のコンポーネントとの間の線、トレース、または別の媒体を除き、介在するコンポーネントがないとき、第2のコンポーネントに直接的に結合されている。第1のコンポーネントは、第1のコンポーネントと第2のコンポーネントとの間の線、トレース、または別の媒体以外の介在するコンポーネントがあるとき、第2のコンポーネントに間接的に結合されている。「結合されている」という言い回しおよびその変形は、直接的に結合されることと間接的に結合されることの両方を含む。また、「約」という言い回しは、特に断らない限り、後続の数値の $\pm 10\%$ を含む範囲を意味する。

10

【0219】

本明細書において述べられている例示的な方法のステップは、必ずしも説明されている順序で実行される必要はないとも理解されるべきであり、そのような方法のステップの順序は、単に例示的であると理解されるべきである。同様に、本開示の様々な実施形態と一致する方法において、追加のステップがそのような方法に含まれてもよく、特定のステップが省略されるか、または組み合わされてもよい。

20

【0220】

本開示においていくつかの実施形態が提供されているが、開示されているシステムおよび方法は、本開示の精神または範囲から逸脱することなく多くの他の特定の形態で具現化されることも可能であることは理解され得る。本例は、例示的であり、制限的でない、と考えられるべきであり、本発明は、明細書に示されている詳細に限定されるべきでない。たとえば、様々な要素またはコンポーネントは、別のシステム内に組み合わされ得るか、もしくは一体化され得るか、またはいくつかの特徴が省略され得るか、もしくは実施され得ない。

【0221】

それに加えて、離散または分離していると様々な実施形態において説明され例示されている技術、システム、サブシステム、および方法は、本開示の範囲から逸脱することなく他のシステム、コンポーネント、技術、または方法と組み合わされるか、もしくは一体化され得る。変更、代用、および改変の他の例は、当業者によって確かめることができ、本明細書で開示されている精神および範囲から逸脱することなくなされ得る。

30

【符号の説明】

【0222】

- 100 動作方法
- 200 コーディングおよびデコーディング(コーデック)システム
- 201 区分化済みビデオ信号
- 211 一般コーダ制御コンポーネント
- 213 変換スケーリングおよび量子化コンポーネント
- 215 ピクチャ内推定コンポーネント
- 217 ピクチャ内予測コンポーネント
- 219 動き補償コンポーネント
- 221 動き推定コンポーネント
- 223 デコード済みピクチャバッファコンポーネント
- 225 ループ内フィルタコンポーネント
- 227 フィルタ制御分析コンポーネント
- 229 スケーリングおよび逆変換コンポーネント

40

50

231	ヘッダフォーマッティングおよびコンテキスト適応2値算術コーディング(CABAC)	
)コンポーネント	
300	ビデオエンコーダ	
301	区分化済みビデオ信号	
313	変換および量子化コンポーネント	
317	ピクチャ内予測コンポーネント	
321	動き補償コンポーネント	
323	デコード済みピクチャバッファコンポーネント	
325	ループ内フィルタコンポーネント	
329	逆変換および量子化コンポーネント	10
331	エントロピーコーディングコンポーネント	
400	ビデオデコーダ	
417	ピクチャ内予測コンポーネント	
421	動き補償コンポーネント	
423	デコード済みピクチャバッファコンポーネント	
425	ループ内フィルタコンポーネント	
429	逆変換および量子化コンポーネント	
433	エントロピーデコーディングコンポーネント	
500	HRD	
541	仮想ストリームスケジューラ(HSS)	20
543	CPB	
545	デコーディングプロセスコンポーネント	
547	DPB	
549	出力クロッピングコンポーネント	
551	ビットストリーム	
553	デコーディングユニット(DU)	
555	デコード済みDU	
556	参照ピクチャ	
557	ピクチャ	
559	出力クロッピング済みピクチャ	30
600	マルチレイヤビデオシーケンス	
611、612、613、614	ピクチャ	
615、616、617、618	ピクチャ	
621	レイヤ間予測	
623	インター予測	
631	レイヤN	
632	レイヤN+1	
700	ビットストリーム	
701	サブビットストリーム	
711	VPS	40
713	SPS	
715	ピクチャパラメータセット(PPS)	
717	スライスヘッダ	
720	画像データ	
723、724	レイヤ	
725、726	ピクチャ	
726	ピクチャのセット	
727、728	スライス	
729	サブビットストリーム抽出プロセス	
731	sps_video_parameter_set_id	50

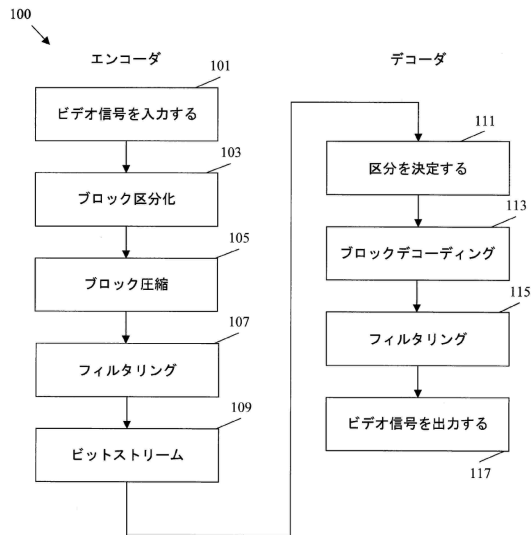
- 732 nuh_layer_id
- 733 VPS独立レイヤフラグ(vps_independent_layer_flag)
- 735 vps_video_parameter_set_id
- 741 VCL NALユニット
- 742 非VCL NALユニット
- 743 コーディング済みレイヤビデオシーケンス(CLVS)
- 744 CLVS
- 800 ビデオコーディングデバイス
- 810 トランシーバユニット(Tx/Rx)
- 814 コーディングモジュール
- 820 下流ポート
- 830 プロセッサ
- 832 メモリ
- 850 上流ポート
- 860 入力および/または出力(I/O)デバイス
- 900 方法
- 1000 方法
- 1100 システム
- 1102 ビデオエンコーダ
- 1105 エンコーディングモジュール
- 1106 記憶モジュール
- 1107 伝送モジュール
- 1110 ビデオデコーダ
- 1111 受信モジュール
- 1113 デコーディングモジュール

10

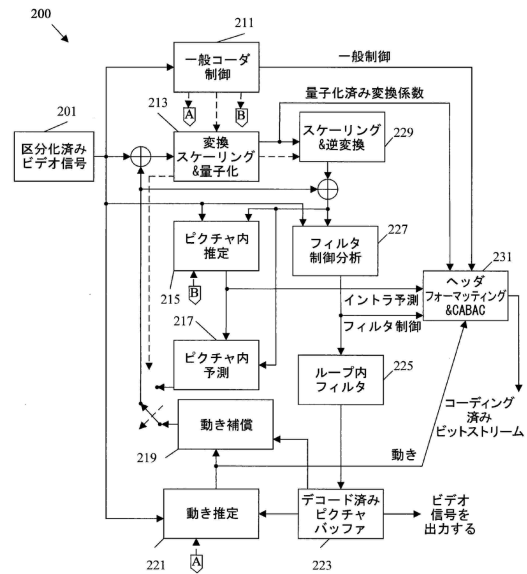
20

【 図面 】

【 図 1 】



【 図 2 】

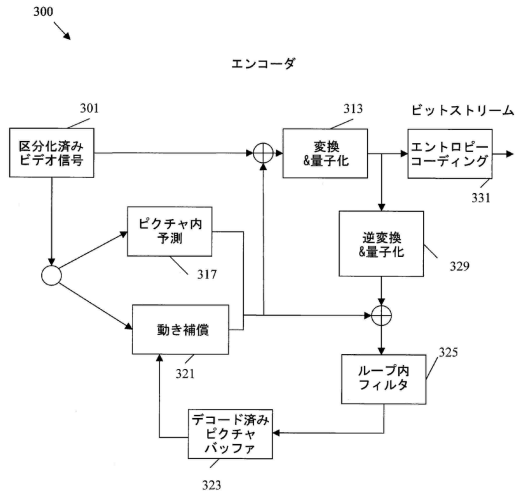


30

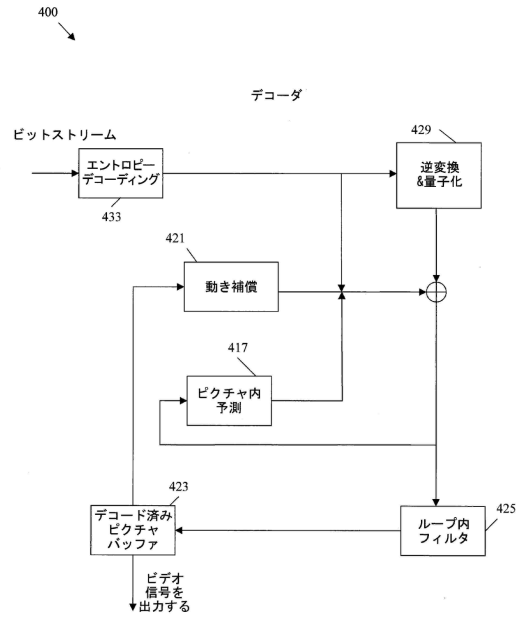
40

50

【図3】



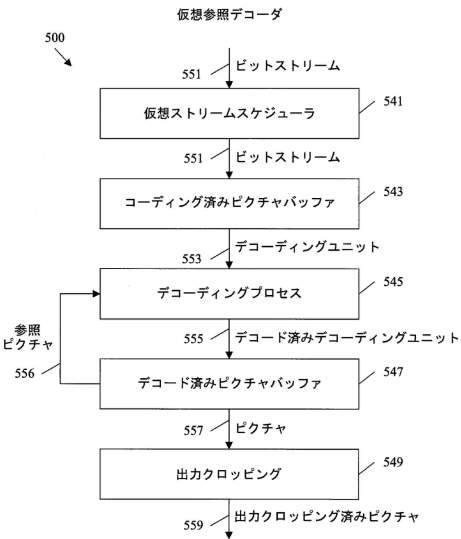
【図4】



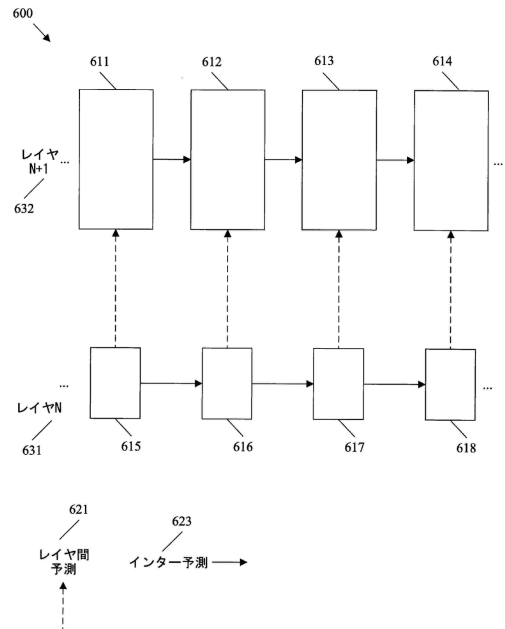
10

20

【図5】



【図6】

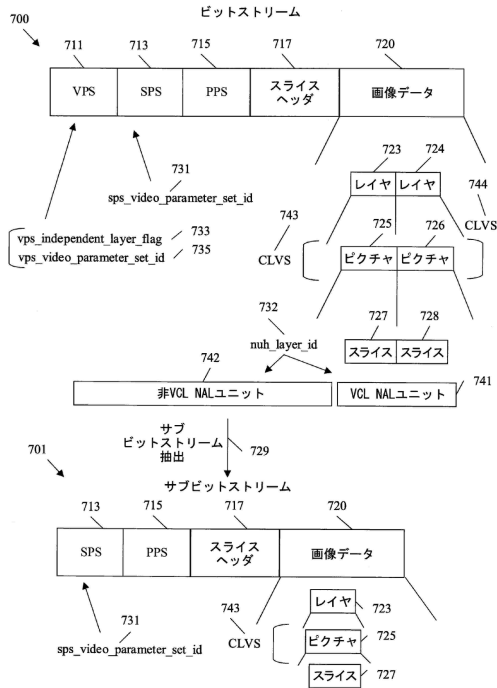


30

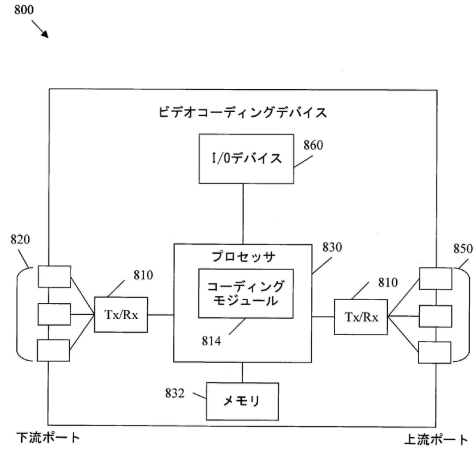
40

50

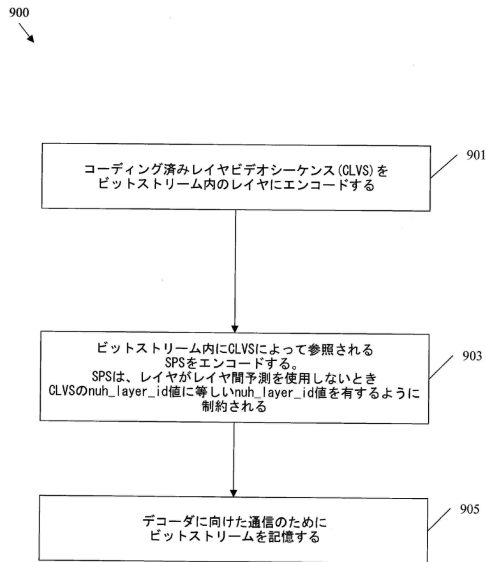
【図7】



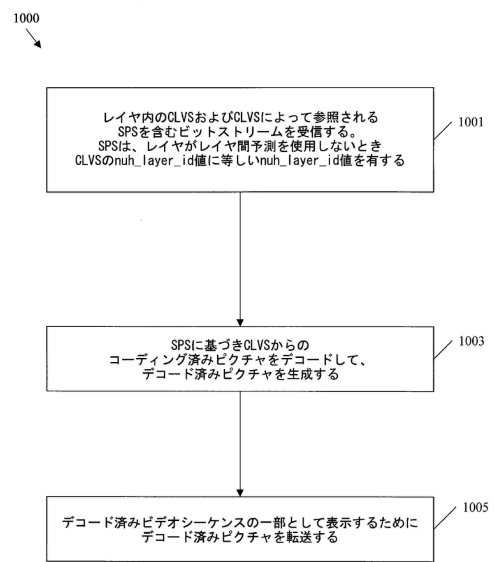
【図8】



【図9】



【図10】



10

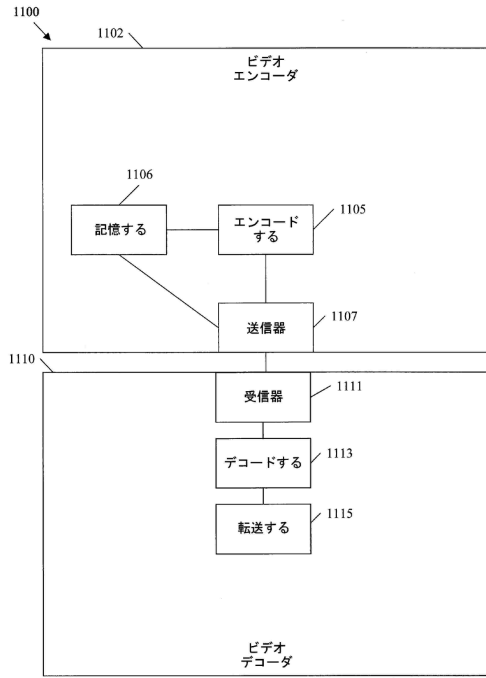
20

30

40

50

【図 11】



10

20

30

40

50

フロントページの続き

クレスト・ウェイ・6264

審査官 白川 瑞樹

(56)参考文献 B. Choi et al. , AHG8: On signaling interlayer reference picture list , JVET-P0221(version 3) , ITU , 2019年10月04日 , pp.1-3 , [online], [retrieved on 2024-10-29], Retrieved from the Internet: URL:https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0221-v3.zip , JVET-P0221-v2.docx

M. M. Hannuksela et al. , AHG8/AHG17: Removing dependencies on VPS from the decoding process of a non-scalable bitstream , JVET-P0097(version 1) , ITU , 2019年09月24日 , pp .1-5 , [online], [retrieved on 2024-10-29], Retrieved from the Internet: URL:https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0097-v1.zip , JVET-P0097.docx

(58)調査した分野 (Int.Cl. , D B 名)

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