

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7475455号
(P7475455)

(45)発行日 令和6年4月26日(2024.4.26)

(24)登録日 令和6年4月18日(2024.4.18)

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

請求項の数 19 (全76頁)

(21)出願番号	特願2022-540581(P2022-540581)	(73)特許権者	000001007 キャノン株式会社 東京都大田区下丸子3丁目30番2号
(86)(22)出願日	令和3年3月5日(2021.3.5)	(74)代理人	110003281 弁理士法人大塚国際特許事務所
(65)公表番号	特表2023-516244(P2023-516244 A)	(72)発明者	ラロシュ, ギローム フランス国 レンヌ - アタラント, セデ ックス セッソン - セヴィニエ 3 5 5 1 7, リュ ドゥ ラ トゥッシュランペ ル キャノン リサーチ センター フラン ス エス . エー . エス 内
(43)公表日	令和5年4月19日(2023.4.19)	(72)発明者	ウエドラオゴ, ナエル フランス国 レンヌ - アタラント, セデ ックス セッソン - セヴィニエ 3 5 5 1 7, リュ ドゥ ラ トゥッシュランペ ル
(86)国際出願番号	PCT/EP2021/055661		
(87)国際公開番号	WO2021/180595		
(87)国際公開日	令和3年9月16日(2021.9.16)		
審査請求日	令和4年9月12日(2022.9.12)		
(31)優先権主張番号	2003562.2		
(32)優先日	令和2年3月11日(2020.3.11)		
(33)優先権主張国・地域又は機関	英国(GB)		
(31)優先権主張番号	2005741.0		
(32)優先日	令和2年4月20日(2020.4.20)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ビデオの符号化および復号のためのハイレベルシンタックス

(57)【特許請求の範囲】

【請求項1】

ビットストリームからビデオデータを復号する方法であって、

前記ビットストリームは、1つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、を含み、

前記方法は、

複数のシンタックス要素を復号することと、前記復号した複数のシンタックス要素を用いて前記ビットストリームから前記ビデオデータを復号することと、

を含み、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされる場合、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つフラグが復号されるよう制約され、前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ(ALF)のためのアダプテーションパラメータセットidであり、前記アダプテーションパラメータセットidにより示されるアダプテーションパラメータセット(APS)はルマに対してフィルターの係数のための1つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含む

ことを特徴とする方法。

【請求項 2】

前記方法は、前記情報が前記ピクチャヘッダ内でシグナリングされるか否かを示す第 1 のシンタックス要素をパースすることと、前記第 1 のシンタックス要素に基づいて、前記スライスヘッダまたは前記ピクチャヘッダの何れか一方のみにおいて、スライスヘッダ内およびピクチャヘッダ内でシグナリングされ得る前記情報の前記パースを許可することと、をさらに含む

請求項 1 に記載の方法。

【請求項 3】

前記第 1 のシンタックス要素は、ピクチャパラメータセットフラグ内の情報である

請求項 2 に記載の方法。

10

【請求項 4】

前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第 1 のシンタックス要素が示す場合、前記スライスヘッダ内の前記情報のパースは許可されない

請求項 2 または 3 に記載の方法。

【請求項 5】

前記情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプルアダプテーションオフセット (SAO) 情報、重み付け予測情報、アダプテーションループフィルタ (ALF) 情報のうちの 1 つ以上を含む

請求項 1 乃至 4 の何れか 1 項に記載の方法。

【請求項 6】

前記情報は、ピクチャヘッダ内およびスライスヘッダ内でシグナリングされ得るすべての情報を含む

請求項 1 乃至 5 の何れか 1 項に記載の方法。

20

【請求項 7】

前記参照ピクチャリスト情報は、`slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx` のうちの 1 つ以上を含む

請求項 5 に記載の方法。

【請求項 8】

ビットストリームからビデオデータを復号する方法であって、

前記ビットストリームは、1 つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、1 つのスライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、を含み、

前記方法は、

複数のシンタックス要素を復号することと、

前記復号した複数のシンタックス要素を用いて前記ビットストリームから前記ビデオデータを復号することと、

を含み、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされることを示す値を有するシンタックス要素を前記ビットストリームが含む場合、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つフラグが復号されるよう制約され、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ (ALF) のためのアダプテーションパラメータセット `id` であり、前記アダプテーションパラメータセット `id` により示されるアダプテーションパラメータセット (APS) はルマに対してフィルターの係数のための 1 つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含む

ことを特徴とする方法。

【請求項 9】

30

40

50

ビデオデータをビットストリームに符号化する方法であって、

前記ビットストリームは、1つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、前記ピクチャヘッダが前記スライスヘッダ内にあるか否かを示すフラグと、を含み、

前記方法は、複数のシンタックス要素を用いて前記ビデオデータを符号化することを含み、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされる場合、前記フラグは、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つよう制約され、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ (ALF) のためのアダプテーションパラメータセット `id` であり、前記アダプテーションパラメータセット `id` により示されるアダプテーションパラメータセット (APS) はルマに対してフィルタの係数のための1つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含むことを特徴とする方法。

【請求項 10】

前記符号化は、前記情報が前記ピクチャヘッダ内でシグナリングされるか否かを示す第1のシンタックス要素を符号化することと、前記第1のシンタックス要素に基づいて、前記スライスヘッダまたは前記ピクチャヘッダの何れか一方のみにおいて、スライスヘッダ内およびピクチャヘッダ内でシグナリングされ得る前記情報の符号化を許可することと、をさらに含む

請求項 9 に記載の方法。

【請求項 11】

前記第1のシンタックス要素は、ピクチャパラメータセットフラグ内の情報である請求項 10 に記載の方法。

【請求項 12】

前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第1のシンタックス要素が示す場合、前記スライスヘッダ内の前記情報の符号化は許可されない

請求項 10 または 11 に記載の方法。

【請求項 13】

前記情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプルアダプテーションオフセット (SAO) 情報、重み付け予測情報、アダプテーションループフィルタ (ALF) 情報のうちの1つ以上を含む

請求項 9 乃至 12 の何れか1項に記載の方法。

【請求項 14】

前記情報は、ピクチャヘッダ内およびスライスヘッダ内でシグナリングされ得るすべての情報を含む

請求項 9 乃至 13 の何れか1項に記載の方法。

【請求項 15】

前記参照ピクチャリスト情報は、`slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx` のうちの1つ以上を含む

請求項 13 に記載の方法。

【請求項 16】

ビットストリームからビデオデータを復号する復号装置であって、

前記ビットストリームは、1つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、を含み、

10

20

30

40

50

前記復号装置は、
複数のシンタックス要素を復号する手段と、
前記復号した複数のシンタックス要素を用いて前記ビットストリームから前記ビデオデータを復号する手段と、
 を有し、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされる場合、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つフラグが復号されるよう制約され、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ (ALF) のためのアダプテーションパラメータセット i d であり、前記アダプテーションパラメータセット i d により示されるアダプテーションパラメータセット (APS) はルマに対してフィルタの係数のための 1 つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含む

ことを特徴とする復号装置。

【請求項 17】

ビットストリームからビデオデータを復号する復号装置であって、

前記ビットストリームは、1 つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、1 つのスライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、を含み、

前記復号装置は、

複数のシンタックス要素を復号する手段と、

前記復号した複数のシンタックス要素を用いて前記ビットストリームから前記ビデオデータを復号する手段と、

を有し、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされることを示す値を有するシンタックス要素を前記ビットストリームが含む場合、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つフラグが復号されるよう制約され、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ (ALF) のためのアダプテーションパラメータセット i d であり、前記アダプテーションパラメータセット i d により示されるアダプテーションパラメータセット (APS) はルマに対してフィルタの係数のための 1 つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含む

ことを特徴とする復号装置。

【請求項 18】

ビデオデータをビットストリームに符号化する符号化装置であって、

前記ビットストリームは、1 つ以上のスライスを復号するときを使用される複数のシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用される複数のシンタックス要素を含むスライスヘッダと、前記ピクチャヘッダが前記スライスヘッダ内にあるか否かを示すフラグと、を含み、

前記符号化装置は、複数のシンタックス要素を用いて前記ビデオデータを符号化する手段を有し、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る情報が前記ピクチャヘッダにおいてシグナリングされる場合、前記フラグは、前記ピクチャヘッダが前記スライスヘッダには無いことを示す値を持つよう制約され、

前記ピクチャヘッダまたは前記スライスヘッダにおいてシグナリングされ得る前記情報は、アダプティブループフィルタ (ALF) のためのアダプテーションパラメータセット i d であり、前記アダプテーションパラメータセット i d により示されるアダプテーションパラメータセット (APS) はルマに対してフィルタの係数のための 1 つ以上のクリッピングインデックスが復号されるかどうかを示すフラグを含む

10

20

30

40

50

ことを特徴とする符号化装置。

【請求項 19】

コンピュータに、請求項 1 乃至 15 の何れか 1 項に記載の方法を実行させるコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオの符号化および復号に関し、特にビットストリームで使用されるハイレベルシンタックスに関するものである。

【背景技術】

【0002】

最近、MPEGとITU-Tスタディグループ16のVCEGが結成した共同チームであるジョイントビデオエキスパートチーム(JVET)は、多用途ビデオ符号化(VVC)と呼ばれる新しいビデオ符号化規格の策定作業を開始した。VVCの目標は、既存のHEVC規格と比較して圧縮性能を大幅に向上させる(典型的に従前の2倍)ことであり、2020年に完成する予定である。主なターゲットとなるアプリケーションやサービスには、360度動画やハイダイナミックレンジ(HDR)ビデオが含まれるが、これらに限定されるものではない。JVETは、独立したテストラボによる正式な主観テストにより、32の組織からの回答を評価した。いくつかの提案では、HEVCを使用した場合と比較して、典型的に40%以上の圧縮効率の向上が示された。特に、超高精細(UHD)ビデオのテスト素材に有効であることが示された。したがって、最終的な規格の目標である50%をはるかに超える圧縮効率の向上が期待される。

【0003】

JVET探査モデル(JEM)は、すべてのHEVCツールを使用し、多くの新しいツールを導入している。これらの変更により、ビットストリームの構造、特にビットストリームの全体的なビットレートに影響を与える可能性のあるハイレベルシンタックスの変更が必要となった。

【発明の概要】

【0004】

本発明は、符号化性能を低下させることなく複雑さを低減させるハイレベルシンタックス構造の改良に関するものである。

【0005】

本発明の第1の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、前記ビットストリームは1つ以上のスライスに対応するビデオデータを含み、前記ビットストリームは、1つ以上のスライスを復号するとき使用されることになる複数のシンタックス要素を含むピクチャヘッダと、1つのスライスを復号するとき使用されることになる複数のシンタックス要素を含むスライスヘッダと、を含み、前記復号は、前記ピクチャヘッダ内または前記スライスヘッダ内でシグナリングされ得る情報が前記ピクチャヘッダ内でシグナリングされる場合、前記ピクチャヘッダが前記スライスヘッダ内がないことを課す(例えば、制約をかける)ことと、前記複数のシンタックス要素を用いて前記ビットストリームを復号することと、を含む。

【0006】

オプションで、前記復号は、前記情報が前記ピクチャヘッダ内でシグナリングされるか否かを示す第1のシンタックス要素をパースすることと、前記第1のシンタックス要素に基づいて、前記スライスヘッダまたは前記ピクチャヘッダの何れか一方のみにおいて、スライスヘッダ内およびピクチャヘッダ内でシグナリングされ得る前記情報の前記パースを許可することと、をさらに含む。

【0007】

オプションで、前記第1のシンタックス要素は、ピクチャパラメータセットフラグ内の情報またはピクチャヘッダフラグ内の情報である。

10

20

30

40

50

【0008】

オプションで、前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第1のシンタックス要素が示す場合、前記スライスヘッダ内の前記情報のパースは許可されない。

【0009】

オプションで、前記方法は、前記ピクチャヘッダが前記スライスヘッダ内にあるか否かを示す第2のシンタックス要素をパースすることをさらに含み、前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第1のシンタックス要素が示す場合、前記ピクチャヘッダが前記スライスヘッダ内でないことを前記第2のシンタックス要素が示すことがビットストリーム適合性の要件である。

10

【0010】

オプションで、前記情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプルアダプテーションオフセット(SAO)情報、重み付け予測情報、アダプテーションループフィルタ(ALF)情報のうちの1つ以上を含む。

【0011】

オプションで、前記情報は、ピクチャヘッダ内およびスライスヘッダ内でシグナリングされ得るすべての情報を含む。

【0012】

オプションで、前記参照ピクチャリスト情報は、`slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx`のうちの1つ以上を含む。

20

【0013】

本発明の第2の態様によれば、ビデオデータをビットストリームに符号化する方法が提供され、前記ビデオデータは1つ以上のスライスに対応し、前記ビットストリームは、1つ以上のスライスを復号するときを使用されることになる複数のシンタックス要素を含むピクチャヘッダと、1つのスライスを復号するときを使用されることになる複数のシンタックス要素を含むスライスヘッダと、を含み、前記符号化は、前記ピクチャヘッダ内または前記スライスヘッダ内でシグナリングされ得る情報が前記ピクチャヘッダ内でシグナリングされる場合、前記ピクチャヘッダが前記スライスヘッダ内でないことをシグナリングすることと、前記複数のシンタックス要素を用いて前記ビデオデータを符号化することと、を含む。

30

【0014】

オプションで、前記符号化は、前記情報が前記ピクチャヘッダ内でシグナリングされるか否かを示す第1のシンタックス要素を符号化することと、前記第1のシンタックス要素に基づいて、前記スライスヘッダまたは前記ピクチャヘッダの何れか一方のみにおいて、スライスヘッダ内およびピクチャヘッダ内でシグナリングされ得る前記情報の符号化を許可することと、をさらに含む。

【0015】

オプションで、前記第1のシンタックス要素は、ピクチャパラメータセットフラグ内の情報またはピクチャヘッダフラグ内の情報である。

40

【0016】

オプションで、前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第1のシンタックス要素が示す場合、前記スライスヘッダ内の前記情報の符号化は許可されない。

【0017】

オプションで、前記方法は、前記ピクチャヘッダが前記スライスヘッダ内にあるか否かを示す第2のシンタックス要素をパースすることをさらに含み、前記情報が前記ピクチャヘッダ内でシグナリングされることを前記第1のシンタックス要素が示す場合、前記ピクチャヘッダが前記スライスヘッダ内でないことを前記第2のシンタックス要素が示すこと

50

がビットストリーム適合性の要件である。

【0018】

オプションで、前記情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプルアダプテーションオフセット(SAO)情報、重み付け予測情報、アダプテーションループフィルタ(ALF)情報のうちの1つ以上を含む。

【0019】

オプションで、前記情報は、ピクチャヘッダ内およびスライスヘッダ内でシグナリングされ得るすべての情報を含む。

【0020】

オプションで、前記参照ピクチャリスト情報は、`slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx`のうちの1つ以上を含む。

10

【0021】

本発明の代替的な態様では、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときに使用するシンタックス要素を含むピクチャヘッダと、スライスを復号するときに使用するシンタックス要素を含むスライスヘッダと、を含み、復号は、ピクチャヘッダまたはスライスヘッダにシグナリングできる情報がスライスヘッダにおいてシグナリングされている場合、スライスヘッダにはシグナリングされていないことに課すことと、シンタックス要素を用いてビットストリームを復号することと、を含む。

20

【0022】

本発明の他の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときに使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するときに使用されるシンタックス要素を含むスライスヘッダと、を含み、復号は、(a) ツール情報がスライスヘッダではなくピクチャヘッダにシグナリングされることを示すシンタックス要素と(b) スライスヘッダにシグナリングされることを示すシンタックス要素とを組み合わせるとして適用不可として扱うことと、シンタックス要素を用いてビットストリームを復号することと、を含む。復号は、組み合わせで適用不可として扱われるシンタックス要素(a)及び(b)が存在する場合には行われない。

30

【0023】

本発明の関連する態様によれば、ビットストリームは、1つ以上のスライスに対応するビデオデータと、1つ以上のスライスを復号するときに使用されるシンタックス要素を含むピクチャヘッダと、スライスを復号するときに使用されるシンタックス要素を含むスライスヘッダと、を含む。ビットストリームは、(a) ツール情報がスライスヘッダではなくピクチャヘッダでシグナリングされることを示すシンタックス要素と、(b) ピクチャヘッダがスライスヘッダでシグナリングされることを示すシンタックス要素と、を組み合わせるとしてそこに存在してはならない、という制約を有している。ツール情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報及び適応ループフィルタ(ALF)情報のうちのいずれか1つであってよい。関連する態様では、ビットストリームを復号する方法が提供される。別の関連する態様では、ビットストリームを復号するように構成された復号器が提供される。ビットストリームは、ビデオ符号化規格に適合するように制約されてもよい。一実施形態において、ビデオ符号化規格は、多用途ビデオ符号化規格である。制約は、ビットストリーム全体に系統的に適用されてもよい。例えば、実施形態では、制約は、ビットストリーム内のシーケンス、ピクチャ及びスライスのいずれか又は全てに対して適用される。

40

50

【0024】

本発明の他の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときを使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用されるシンタックス要素を含むスライスヘッダと、を含み、復号は、(a) ツール情報がピクチャヘッダではなくスライスヘッダにシグナリングされていることを示すシンタックス要素と(b) ピクチャヘッダがスライスヘッダにシグナリングされていることを示すシンタックス要素とを組み合わせ適用不可として処理することと、シンタックス要素を用いてビットストリームを復号することと、を含む。復号は、組み合わせで適用不可として扱われるシンタックス要素(a)及び(b)が存在する場合には実行されない。

10

【0025】

本発明の他の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときを使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用されるシンタックス要素を含むスライスヘッダと、を含み、ピクチャヘッダは、スライスヘッダにおいてシグナリングされることになっており、スライスヘッダまたはピクチャヘッダのうち的一方のみにおいてシグナリングされ得る情報のパースが制約され、復号は、シンタックス要素(`xxx__info__in__ph__flag`)がピクチャヘッダにツール情報があることを示す場合(例えば、`xxx__info__in__ph__flag = 1`の場合)には、スライスヘッダにシグナリングしない(例えば、`picture__header__in__slice__header__flag`を0に強制)ことを含む。

20

【0026】

本発明の関連する態様によれば、ビットストリームは、1つ以上のスライスに対応するビデオデータと、1つ以上のスライスを復号するときを使用されるシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用されるシンタックス要素を含むスライスヘッダと、を含む。ビットストリームは、(a) ツール情報がピクチャヘッダではなくスライスヘッダでシグナリングされることを示すシンタックス要素と、(b) ピクチャヘッダがスライスヘッダでシグナリングされることを示すシンタックス要素と、が組み合わせられてそこに存在してはならないという制約を有している。

30

【0027】

本発明の他の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときを使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用されるシンタックス要素を含むスライスヘッダと、を含み、ピクチャヘッダは、スライスヘッダにおいてシグナリングされることになっており、スライスヘッダまたはピクチャヘッダのうち的一方のみにおいてシグナリングされ得る情報のパースが制約され、復号は、シンタックス要素(`xxx__info__in__ph__flag`)がピクチャヘッダにツール情報がないことを示す場合(例えば、`xxx__info__in__ph__flag = 0`の場合)には、スライスヘッダにシグナリングしない(例えば、`picture__header__in__slice__header__flag`を0に強制)ことを含む。

40

【0028】

本発明の第1の更なる態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するときを使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するときを使用されるシンタックス要素を含むスライスヘッダと、を含み、復号は、ピクチャヘッダがスライスヘッダにシグナリングされる場合、スライスヘッダまたはピクチャヘッダの一方のみで、スライスヘッダお

50

よびピクチャヘッダにシグナリングされる可能性がある情報をパースすることを許可することと、シンタックス要素を用いてビットストリームを復号することと、を含む。

【0029】

ピクチャヘッダがスライスヘッダ内にある場合、これは現在のピクチャに対して1つのスライスだけが存在することを意味する。したがって、スライスとピクチャの両方について情報を送信または送信可能にしても、パラメータが同じになるため、符号化器または復号器の柔軟性が向上しない。つまり、ピクチャヘッダに情報がある場合、スライスヘッダの対応する情報は冗長となる。同様に、スライスヘッダに情報がある場合、ピクチャヘッダ内の対応する情報は冗長となる。ピクチャヘッダがスライスヘッダにある場合には、ピクチャヘッダまたはスライスヘッダにある情報のみを許可することにより、シグナリング

10

【0030】

復号は、スライスヘッダにおいてピクチャヘッダをシグナリングするか否かを示す第1のシンタックス要素をパースすることと、第1のシンタックス要素に基づいてスライスヘッダまたはピクチャヘッダの一方のみにシグナリングされ得る情報のパースを許可することと、をさらに含んでもよい。第1のシンタックス要素は、スライスヘッダ中のピクチャヘッダフラグであってもよい。

【0031】

オプションで、スライスヘッダ内の情報のパースは、第1のシンタックス要素が、スライスヘッダ内でピクチャヘッダがシグナリングされることを示す場合には、許可されない。情報がピクチャヘッダにあるか否かを示す第2のシンタックス要素が解析されてもよく、第1のシンタックス要素がピクチャヘッダがスライスヘッダにシグナリングされていることを示す場合に、第2のシンタックス要素が情報がピクチャヘッダにシグナリングされていることを示すことがビットストリーム適合性の要件とされる。

20

【0032】

代替的に、第1のシンタックス要素が、ピクチャヘッダがスライスヘッダ内でシグナリングされることを示す場合には、ピクチャヘッダ内の情報のシグナリングは許可されない。この方法は、情報がピクチャヘッダ内にあるか否かを示す第2のシンタックス要素をパースすることをさらに含んでもよく、第1のシンタックス要素がピクチャヘッダがスライスヘッダ内でシグナリングされることを示す場合に、第2のシンタックス要素が情報がピクチャヘッダ内にあることを示すことがビットストリーム適合性の要件とされる。第2のシンタックス要素は、ピクチャパラメータ設定フラグであってもよく、フラグが設定された場合に情報がピクチャヘッダにあり、設定されていない場合に情報がスライスヘッダにあるか、存在しないかである。

30

【0033】

実施形態によれば、情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報及びアダプティブループフィルタ(ALF)情報のうちの1つ以上を含むことができる。例えば、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報、及びアダプティブループフィルタ(ALF)情報の全てを含むことができる。

40

【0034】

オプションで、この情報は、ピクチャヘッダ内とスライスヘッダ内でシグナリングされる可能性のあるすべての情報を含む。

【0035】

参照ピクチャリスト情報は、以下のシンタックス要素 `slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx` を1つ以上含むことができる。

50

【0036】

スライスヘッダ内でピクチャヘッダをシグナリングする場合、パース可能な重み付け予測の重みの数を制限することができる。

【0037】

本発明の第2の更なる態様によれば、ビデオデータをビットストリームに符号化する方法が提供され、ビデオデータが1つ以上のスライスに対応し、ビットストリームが、1つ以上のスライスを復号するときに使用されるシンタックス要素を含むピクチャヘッダと、スライスを復号するときに使用されるシンタックス要素を含むスライスヘッダと、を含み、符号化は、スライスヘッダ内でピクチャヘッダがシグナリングされる場合において、スライスヘッダおよびピクチャヘッダにおいてシグナリングされ得る情報を、スライスヘッダまたはピクチャヘッダのいずれか一方においてのみ符号化を許可することと、シンタックス要素を用いてビデオデータの符号化することと、を含む。

10

【0038】

ピクチャヘッダがスライスヘッダ内にある場合、これは現在のピクチャに対して1つのスライスだけが存在することを意味する。したがって、スライスとピクチャの両方について情報を送信または送信可能にしても、パラメータが同じになるため、符号化器または復号器の柔軟性が向上しない。つまり、ピクチャヘッダに情報がある場合、スライスヘッダの対応する情報は冗長となる。同様に、スライスヘッダに情報がある場合、ピクチャヘッダ内の対応する情報は冗長となる。ピクチャヘッダがスライスヘッダにある場合に、ピクチャヘッダまたはスライスヘッダにある情報のみを許可することにより、シグナリングにおける冗長性を制限して、復号器の実装を簡素化することができる。したがって、符号化効率を損なうことなく符号化を簡略化し、シグナリングコストを削減することができる（関連情報がビットストリームに一度だけ含まれるため）。

20

【0039】

符号化は、スライスヘッダ内でピクチャヘッダをシグナリングするか否かを示す第1のシンタックス要素を符号化し、第1のシンタックス要素に基づいてスライスヘッダまたはピクチャヘッダの一方のみにシグナリングされ得る情報の符号化を許可することを更に含んでもよい。

【0040】

第1のシンタックス要素は、スライスヘッダフラグのピクチャヘッダであってもよい。

30

【0041】

第1のシンタックス要素が、ピクチャヘッダがスライスヘッダ内でシグナリングされることを示す場合、スライスヘッダでの情報の符号化は許可されない場合がある。

【0042】

情報がピクチャヘッダにあるか否かを示す第2のシンタックス要素を符号化してもよく、第1のシンタックス要素がスライスヘッダでピクチャヘッダをシグナリングすることを示す場合に、第2のシンタックス要素が情報がピクチャヘッダにあることを示すことがビットストリーム適合性の要件とされる。

【0043】

代替的に、第1のシンタックス要素がスライスヘッダ内でピクチャヘッダがシグナリングされることを示す場合には、ピクチャヘッダの情報のシグナリングは許可されない。

40

【0044】

情報がピクチャヘッダにあるか否かを示す第2のシンタックス要素を符号化してもよく、第1のシンタックス要素がスライスヘッダ内でピクチャヘッダをシグナリングすることを示す場合に、第2のシンタックス要素が情報がピクチャヘッダ内にあることを示すことがビットストリームパフォーマンスの要件とされる。

【0045】

第2のシンタックス要素は、ピクチャパラメータセットフラグまたはピクチャヘッダフラグの情報であってもよく、フラグがセットされているときはピクチャヘッダ内で情報がシグナリングされ、セットされていないときはスライスヘッダ内で情報がシグナリングされ

50

るか、または存在しないかである。

【0046】

情報は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報、アダプティブループフィルタ(ALF)情報のうちの1つ以上を含んでもよい。任意選択で、情報は、ピクチャヘッダおよびスライスヘッダにおいてシグナリングされ得るすべての情報を含む。例えば、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報、及びアダプティブループフィルタ(ALF)情報の全てである。

【0047】

参照画像リスト情報は、`slice_collocated_from_l0_flag`、`slice_collocated_ref_idx`、`ph_collocated_from_l0_flag`、`ph_collocated_ref_idx`のうちの1つ以上を含んでいる。

【0048】

オプションで、ピクチャヘッダがスライスヘッダでシグナリングされる場合、重み付け予測のための重みの数は制限されてもよい。

【0049】

本発明の第3の更なる態様によれば、1つ以上のスライスに対応するビデオデータを含むビットストリームを復号する方法が提供され、ビットストリームは、1つ以上のスライスを復号するとき使用されるべきシンタックス要素を含むピクチャヘッダと、スライスを復号するとき使用されるべきシンタックス要素を含むスライスヘッダと、を含み、方法は、スライスヘッダにおいて、ピクチャヘッダがスライスヘッダ内でシグナリングされているかどうかを示すシンタックス要素をパースすることであって、ALFのAPS_ID関連シンタックス要素は、ピクチャヘッダがスライスヘッダにおいてシグナリングされているかどうかを示すシンタックス要素よりも前にパースされる。ALFのAPS_ID関連情報は、スライスヘッダの近傍または冒頭でパースされてもよい。

【0050】

本発明の第4の更なる態様によれば、1つ以上のスライスを含むビデオデータをビットストリームに符号化する方法が提供され、ビットストリームは、1つ以上のスライスを復号するとき使用するシンタックス要素を含むピクチャヘッダと、スライスを復号するとき使用するシンタックス要素を含むスライスヘッダと、を含み、方法は、スライスヘッダにおいて、ピクチャヘッダがスライスヘッダにシグナリングされているかどうかを示すシンタックス要素をパースすることを含み、ここで、ALFのAPS_ID関連シンタックス要素は、ピクチャヘッダがスライスヘッダ内でシグナリングされているかどうかを示すシンタックス要素より先に符号化される。ALFのAPS_ID関連情報は、スライスヘッダの近傍または冒頭に符号化されてもよい。

【0051】

本発明の第5の更なる態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するとき使用するシンタックス要素を含むピクチャヘッダと、スライスを復号するとき使用するシンタックス要素を含むスライスヘッダと、を含み、復号は、スライスヘッダにシグナリングされるべき場合、重み予測モードについてシグナリングするウェイト数を制限することと、シンタックス要素を用いてビットストリームを復号することと、を含む。スライスヘッダ内でピクチャヘッダをシグナリングする場合、スライスヘッダ内とピクチャヘッダ内でシグナリングされ得る情報のパースは、スライスヘッダまたはピクチャヘッダのいずれか一方でのみ許可され得る。

【0052】

本発明の第6の更なる態様によれば、ビットストリームからビデオデータを符号化する

10

20

30

40

50

方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するとき使用するシンタックス要素を含むピクチャヘッダと、スライスを復号するとき使用するシンタックス要素を含むスライスヘッダと、を含み、符号化は、スライスヘッダにシグナリングする場合、重み付き予測モードに対して符号化する重みの数を制限することと、シンタックス要素を用いてビットストリームを符号化することと、を含む、スライスヘッダ内でピクチャヘッダをシグナリングする場合、スライスヘッダ内とピクチャヘッダ内でシグナリングされ得る情報の符号化を、スライスヘッダまたはピクチャヘッダのいずれか一方でのみ許可するようにしてもよい。

【0053】

本発明の第7の更なる態様によれば、ビットストリームからビデオデータを復号するための復号器が提供され、復号器は、第1、第3または第5のいずれかの更なる態様の方法を実行するように構成される。

【0054】

本発明の第8の更なる態様によれば、ビデオデータをビットストリームに符号化するための符号化器が提供され、符号化器は、第2、第4または第6のいずれかの更なる態様の方法を実行するように構成される。

【0055】

本発明の第9の更なる態様によれば、実行により、第1から第6の更なる態様のいずれかに記載の方法を実行させるコンピュータプログラムが提供される。プログラムは、それ自体で提供されてもよいし、キャリア媒体上で、キャリア媒体によって、またはキャリア媒体中で搬送され得る。キャリア媒体は、非一時的のもの、例えば記憶媒体、特にコンピュータ読み取り可能な記憶媒体であってもよい。搬送媒体はまた、例えば信号または他の伝送媒体のような一時的のものであってもよい。信号は、インターネットを含む任意の適切なネットワークを介して送信されてもよい。本発明の更なる特徴は、独立請求項及び従属請求項によって特徴付けられる。

【0056】

本発明の第1の更なる他の態様によれば、ビットストリームからビデオデータを復号する方法が提供され、ビットストリームは、1つ以上のスライスに対応するビデオデータを含み、ビットストリームは、1つ以上のスライスを復号するとき使用するシンタックス要素を含むピクチャヘッダと、スライスを復号するとき使用するシンタックス要素を含むスライスヘッダと、を含む。ビットストリームが、ピクチャヘッダまたはスライスヘッダにおいてシグナリングできる情報がピクチャヘッダにおいてシグナリングされることを示す値を有する第1のシンタックス要素を含む場合に、ビットストリームが、スライスヘッダにないことを示す値を有する第2のシンタックス要素も含むように制約し、方法は、シンタックス要素を用いてビットストリームを復号化することを含む。ビットストリームは、ビデオ符号化規格に適合するように制約されてもよい。実施形態において、ビデオコーディング規格は、多用途ビデオ符号化規格である。第2のシンタックス要素は、スライスヘッダシンタックス要素におけるピクチャヘッダであってもよい。第1のシンタックス要素は、量子化パラメータ値情報、参照ピクチャリスト情報、デブロッキングフィルタ情報、サンプル適応オフセット(SAO)情報、重み付け予測情報及びアダプティブループフィルタ(ALF)情報のうちの1つ以上のピクチャヘッダ内の符号化を示すフラグであってよい。ビットストリーム制約は、系統的に適用されてもよい。例えば、実施形態では、制約は、ビットストリーム内のシーケンス、ピクチャ及びスライスのいずれか又は全てに対して適用される。

【0057】

本発明の第2の更なる他の態様によれば、ビデオデータをビットストリームにまたはビットストリームから符号化または復号するための方法が提供され、方法は、ピクチャヘッダまたはスライスヘッダにおいてシグナリングされ得る情報がピクチャヘッダにおいてシグナリングされるかどうかに基づいて、スライスヘッダにおいてピクチャヘッダが許可さ

10

20

30

40

50

れているかどうかに関連する制約を適用することを含む。本発明の第3の更なる他の態様によれば、第2の更なる他の態様の方法を実行するように構成された装置が提供される。本発明の第4の更なる他の態様によれば、実行されることにより第2の更なる他の態様の方法を実行させる命令を含むコンピュータプログラムが提供される。

【0058】

本発明の一態様における任意の特徴は、任意の適切な組み合わせで、本発明の他の態様に適用され得る。特に、方法の態様は、装置の態様に適用されてもよく、その逆もまた然りである。

【0059】

さらに、ハードウェアで実施された特徴は、ソフトウェアで実施されてもよく、その逆もまた然りである。本明細書では、ソフトウェアおよびハードウェアの特徴についての言及は、それに応じて解釈される。

10

【0060】

本明細書に記載されるような任意の装置特徴はまた、方法特徴として提供されてもよく、その逆もまた同様である。本明細書で使用されるように、手段プラス機能の特徴は、適切にプログラムされたプロセッサ及び関連するメモリのようなそれらの対応する構造の観点から代替的に表現され得る。

【0061】

また、本発明の任意の態様で説明および定義される様々な特徴の特定の組み合わせは、独立して実施および/または供給および/または使用できることを理解されたい。

20

【図面の簡単な説明】

【0062】

次に、例として、添付の図面を参照する。

【0063】

【図1】HEVCおよびVVCで使用される符号化構造の説明に用いる図である。

【図2】本発明の1つ以上の実施形態が実施され得るデータ通信システムを模式的に示すブロック図である。

【図3】本発明の1つ以上の実施形態が実施され得る処理装置の構成要素を示すブロック図である。

【図4】本発明の実施形態に係る符号化方法のステップを示すフローチャートである。

30

【図5】本発明の実施形態に係る復号方法のステップを示すフローチャートである。

【図6】例示的な符号化方式VVCにおけるビットストリームの構造を示す図である。

【図7】例示的な符号化方式VVCにおけるビットストリームの別の構造を示す図である。

【図8】ルマモデリングクロマスケーリング(LMCS)を説明する図である。

【図9】LMCSのサブツールを示す図である。

【図10】本発明の実施形態に係る符号化器又は復号器と通信ネットワークとを含むシステムを示す図である。

【図11】本発明の1つ以上の実施形態の実施のためのコンピューティングデバイスの概略ブロック図である。

【図12】ネットワークカメラシステムを説明するための図である。

40

【図13】スマートフォンを説明するための図である。

【発明を実施するための形態】

【0064】

図1は、高効率ビデオ符号化(HEVC)ビデオ規格で使用される符号化構造に関するものである。ビデオシーケンス1は、連続するデジタル画像 i で構成され、そのような各デジタル画像は、1つ以上の行列によって表される。行列の係数は、画素を表す。

【0065】

シーケンスの画像2は、複数のスライス3に分割され得る。スライスは、場合によっては、画像全体を構成することがある。これらのスライスは、オーバーラップしない複数の符号化ツリーユニット(CTU)に分割される。符号化ツリーユニット(CTU)は、高効

50

率ビデオ符号化 (H E V C) ビデオ規格の基本処理単位であり、概念的には、いくつかの以前のビデオ規格で使用されていたマクロブロックユニットに構造上対応する。C T U は最大符号化ユニット (L C U) と呼ばれることがある。C T U は、ルマとクロマのコンポーネント部分を持ち、コンポーネント部分のそれぞれは符号化ツリーブロック (C T B) と呼ばれる。これらの異なる色コンポーネントは、図 1 では示されていない。

【 0 0 6 6 】

C T U は一般に 6 4 画素 × 6 4 画素の大きさである。各 C T U は、四分木 (クワッドツリー) 分解を使用して、より小さな可変サイズの複数の符号化ユニット (C U) に反復的に分割され得る。

【 0 0 6 7 】

符号化ユニットは基本的な符号化要素であり、予測ユニット (P U) と変換ユニット (T U) と呼ばれる 2 種類のサブユニットで構成される。P U や T U の最大サイズは C U のサイズと等しい。予測ユニットは、画素値を予測するための C U の分割に相当する。C U を P U に分割する場合、6 0 6 で示したように、4 つの正方形 P U に分割する方法や、2 つの長方形 P U に分割する方法など、さまざまな分割が可能である。変換ユニットは、D C T を使用して空間変換を受ける基本ユニットである。C U は、四分木表現 6 0 7 に基づいて複数の T U に分割され得る。

【 0 0 6 8 】

各スライスは 1 つのネットワーク抽象化レイヤ (N A L) ユニットに埋め込まれている。さらに、ビデオシーケンスの符号化パラメータは、パラメータセットと呼ばれる専用の N A L ユニットに格納される。H E V C と H . 2 6 4 / A V C では、2 種類のパラメータセット N A L ユニットが採用される：まず、シーケンスパラメータセット (S P S) N A L ユニットは、ビデオシーケンス全体で変更されないすべてのパラメータを収集するものである。典型的には、符号化プロファイル、複数のビデオフレームのサイズ、他の複数のパラメータを扱う。次に、ピクチャパラメータセット (P P S) N A L ユニットには、シーケンスのある画像 (またはフレーム) から別の画像へと変化する可能性がある複数のパラメータが含まれる。H E V C はまた、ビットストリームの全体的な構造を記述する複数のパラメータを含むビデオパラメータセット (V P S) N A L ユニットを含む。V P S は、H E V C で定義された新しいタイプのパラメータセットであり、ビットストリームのすべてのレイヤに適用される。1 つのレイヤには複数のサブレイヤが含まれる場合があり、バージョン 1 のビットストリームはすべて 1 つのレイヤに制限されている。H E V C はスケーラビリティとマルチビューのために特定のレイヤ拡張を持ち、これらは下位互換性のあるバージョン 1 の基本レイヤで、複数のレイヤを可能にする。

【 0 0 6 9 】

図 2 は、本発明の 1 つ以上の実施形態が実施され得るデータ通信システムを示す図である。データ通信システムは、データ通信ネットワーク 2 0 0 を介して、データストリームのデータパケットを受信装置 (この場合はクライアント端末 2 0 2) に送信するように動作可能な送信装置 (この場合はサーバ 2 0 1) から構成される。データ通信ネットワーク 2 0 0 は、ワイドエリアネットワーク (W A N) またはローカルエリアネットワーク (L A N) であってよい。このようなネットワークは、例えば、無線ネットワーク (W i f i / 8 0 2 . 1 1 a または b または g)、イーサネットネットワーク、インターネットネットワーク、または複数の異なるネットワークを含む混合ネットワークであってよい。本発明の特定の実施形態では、データ通信システムは、サーバ 2 0 1 が複数のクライアントに同じデータコンテンツを送信するデジタルテレビ放送システムであってよい。

【 0 0 7 0 】

サーバ 2 0 1 から提供されるデータストリーム 2 0 4 は、ビデオデータおよびオーディオデータを表すマルチメディアデータで構成されてもよい。オーディオ及びビデオデータストリームは、本発明のいくつかの実施形態において、マイクロフォン及びカメラをそれぞれ使用してサーバ 2 0 1 によって取得されてもよい。いくつかの実施形態では、データストリームは、サーバ 2 0 1 に格納されてもよいし、サーバ 2 0 1 が他のデータプロバイ

10

20

30

40

50

ダから受信してもよいし、サーバ201で生成してもよい。サーバ201は、特にビデオ及びオーディオストリームを符号化するための符号化器を備え、符号化器への入力として提示されたデータのよりコンパクトな表現である送信用の圧縮されたビットストリームを提供するためのものである。

【0071】

送信データの量に対する送信データの質のより良い比率を得るために、ビデオデータの圧縮は、例えば、HEVC形式またはH.264/AVC形式に従って行うことができる。

【0072】

クライアント202は、送信されたビットストリームを受信し、再構成されたビットストリームを復号して、ディスプレイ装置でビデオ画像を再生し、ラウドスピーカーでオーディオデータを再生する。

10

【0073】

図2の例ではストリーミングシナリオが考慮されているが、本発明のいくつかの実施形態では、符号化器と復号器の間のデータ通信は、例えば光ディスクなどのメディア記憶装置を用いて実行されてもよいことが理解されよう。

【0074】

本発明の1つ以上の実施形態では、ビデオ画像は、最終画像内のフィルタリングされた複数の画素を提供するために画像の再構成された複数の画素に適用する補償オフセットの代表的なデータとともに送信される。

【0075】

20

図3は、本発明の少なくとも1つの実施形態を実施するように構成された処理装置300を模式的に示している。処理装置300は、マイクロコンピュータ、ワークステーション、または軽量の携帯機器などの装置であってよい。装置300は、

- マイクロプロセッサなどの中央処理装置311(CPUと表記)；

- 本発明を実施するためのコンピュータプログラムを格納するための読み出し専用メモリ306(ROMと表記)；

- 本発明の実施形態の方法の実行可能コード、ならびに本発明の実施形態によるデジタル画像のシーケンスを符号化する方法および/またはビットストリームを復号する方法を実施するために必要な変数およびパラメータを記録するために適合されたレジスタを記憶するためのランダムアクセスメモリ312(RAMと表記)；および

30

- 処理されるデジタルデータが送受信される通信ネットワーク303に接続される通信インターフェース302

に接続された通信バス313を含む。

【0076】

オプションで、装置300は、以下の構成要素を含み得る：

- 本発明の1つ以上の実施形態の方法を実施するためのコンピュータプログラム、および本発明の1つ以上の実施形態の実施中に使用または生成されるデータを格納するための、データ格納手段304(ハードディスク)；

- ディスク306のためのディスクドライブ305であって、ディスクドライブは、ディスク306からデータを読み取るか、またはディスクにデータを書き込むように適合されている；

40

- キーボード310または他の任意のポインティング手段によって、データを表示するため、および/またはユーザとのグラフィカルインターフェースとして機能するためのスクリーン309。

【0077】

装置300は、例えばデジタルカメラ320やマイクロフォン308などの様々な周辺機器に接続することができ、それぞれは装置300にマルチメディアデータを供給するように入出力カード(図示せず)に接続される。

【0078】

通信バスは、装置300に含まれる又はそれに接続される様々な要素間の通信及び相互

50

運用性を提供する。バスの表現は限定的ではなく、特に中央処理装置は、装置 300 の任意の要素に直接又は装置 300 の別の要素によって命令を伝達するように動作可能である。

【0079】

ディスク 306 は、例えば、書き換え可能か否かを問わずコンパクトディスク (CD-ROM)、ZIP ディスク、メモリカードなどの任意の情報媒体で置き換えることができ、一般的には、マイクロコンピュータまたはマイクロプロセッサによって読み取り可能な情報記憶手段で、装置に組み込まれているか否かを問わず、取り外し可能で、その実行により本発明によるデジタル画像のシーケンスを符号化する方法および/またはビットストリームを復号する方法を実施することができる 1 以上のプログラムの格納に適合されている可能性があるものであれば、どのような媒体でもよい。

10

【0080】

実行可能コードは、前述したように、読み取り専用メモリ 306、ハードディスク 304 上、または例えばディスク 306 のような取り外し可能なデジタル媒体のいずれかに格納することができる。変形例によれば、プログラムの実行可能コードは、実行される前に装置 300 の記憶手段の 1 つ、例えばハードディスク 304 に記憶されるように、インタフェース 302 を介して、通信ネットワーク 303 の手段によって受信されることが可能である。

【0081】

中央処理装置 311 は、本発明によるプログラムまたはプログラムのソフトウェアコードの命令または部分、前述の記憶手段の 1 つに格納されている命令の実行を制御および指示するように適合されている。電源投入時に、例えばハードディスク 304 上または読み取り専用メモリ 306 内の不揮発性メモリに格納されているプログラムまたは複数のプログラムは、ランダムアクセスメモリ 312 に転送され、その後、プログラムまたはプログラムの実行可能コード、および本発明を実施するために必要な変数およびパラメータを格納するためのレジスタが格納される。

20

【0082】

この実施形態では、装置は、本発明を実施するためにソフトウェアを使用するプログラマブルな装置である。しかしながら、代替的に、本発明は、ハードウェア (例えば、特定用途向け集積回路すなわち ASIC の形態) において実施され得る。

【0083】

図 4 は、本発明の少なくとも 1 つの実施形態による符号化器のブロック図である。符号化器は、接続された複数のモジュールによって表され、各モジュールは、例えば、装置 300 の CPU 311 によって実行されるプログラミング命令の形態で、本発明の 1 つ以上の実施形態による画像のシーケンスの画像を符号化する少なくとも 1 つの実施形態を実施する方法の少なくとも 1 つの対応するステップを実施するように適合される。

30

【0084】

複数のデジタル画像 ($i_0 \sim i_n$) 401 のオリジナルシーケンスは、符号化器 400 によって入力として受け取られる。各デジタル画像は、画素として知られるサンプルの集合によって表される。

【0085】

符号化処理の実施後、符号化器 400 によってビットストリーム 410 が出力される。ビットストリーム 410 は、複数の符号化ユニットまたはスライスを含み、各スライスは、スライスの符号化に使用される符号化パラメータの符号化値を送信するためのスライスヘッダと、符号化されたビデオデータを含むスライスボディと、を含む。

40

【0086】

入力されたデジタル画像 ($i_0 \sim i_n$) 401 は、モジュール 402 によって画素のブロックに分割される。ブロックは画像部分に対応し、可変サイズ (例えば、 4×4 、 8×8 、 16×16 、 32×32 、 64×64 、 128×128 画素およびいくつかの矩形ブロックサイズも考慮できる) であってもよい。符号化モードは、各入力ブロックごとに選択される。空間予測に基づく符号化モード (イントラ予測) と時間予測に基づく符号化 (

50

インター符号化、マージ、SKIP)の2つのファミリの符号化モードがある。可能な符号化モードがテストされる。

【0087】

モジュール403は、イントラ予測プロセスを実行し、符号化される所定のブロックは、符号化されるブロックの近傍の複数の画素から計算される予測器によって予測される。選択されたイントラ予測器の表示と与えられたブロックとその予測器との間の差は、イントラ符号化が選択された場合、残差を提供するために符号化される。

【0088】

時間予測は、動き推定モジュール404及び動き補償モジュール405によって実施される。まず、一組の参照画像416の中から1つの参照画像が選択され、参照画像の一部(参照領域または画像部分とも呼ばれ、符号化されるべき所定のブロックに最も近い領域である)が、動き推定モジュール404により選択される。そして、動き補償モジュール405は、選択された領域を用いて、符号化されるブロックを予測する。選択された参照領域と与えられたブロックとの間の差は残差ブロックとも呼ばれ、動き補償モジュール405によって計算される。選択された参照領域は、動きベクトルによって示される。

【0089】

このように、両方(場合空間予測と時間予測)の場合において、オリジナルのブロックから予測値を差し引くことで残差が計算される。

【0090】

モジュール403によって実施されるイントラ予測では、予測方向が符号化される。時間予測では、少なくとも1つの動きベクトルが符号化される。モジュール404、405、416、418、417によって実施されるインター予測では、少なくとも1つの動きベクトルまたはそのような動きベクトルを識別するためのデータが、時間予測に対して符号化される。

【0091】

インター予測を選択した場合は、動きベクトルと残差ブロックの相対的な情報が符号化される。さらにビットレートを減らすために、動きが均質であると仮定して、動きベクトルは、動きベクトル予測器に関する差分によって符号化される。動きベクトル予測および符号化モジュール417によって、動き情報予測器のセットの動きベクトル予測器が動きベクトルフィールド418から取得される。

【0092】

符号化器400は、さらに、レート-歪み基準などの符号化コスト基準を適用して符号化モードを選択するための選択モジュール406を備える。さらに冗長性を減らすために、変換モジュール407によって変換(DCTなど)が残差ブロックに適用され、得られた変換データは、次に量子化モジュール408によって量子化され、エントロピー符号化モジュール409によってエントロピー符号化される。最後に、符号化されている現在のブロックの符号化された残差ブロックがビットストリーム410に挿入される。

【0093】

また、符号化器400は、後続の画像の動き推定のための参照画像を生成するために、符号化された画像の復号を実行する。これにより、ビットストリームを受信する符号化器と復号器は、同じ参照フレームを持つことができる。逆量子化モジュール411は、量子化されたデータの逆量子化を行い、その後、逆変換モジュール412による逆変換を行う。逆イントラ予測モジュール413は、予測情報を用いて、所定のブロックに使用する予測器を決定し、逆動き補償モジュール414は、モジュール412によって得られた残差を、参照画像セット416から得られた参照領域に実際に加算する。

【0094】

その後、モジュール415によってポストフィルタリングが適用され、再構成された画素のフレームをフィルタリングする。本発明の実施形態では、SAOループフィルタが使用され、補償オフセットが、再構成された画像の再構成された画素の画素値に加算される

【0095】

10

20

30

40

50

図5は、本発明の実施形態による符号化器からデータを受信するために使用され得る復号器60のブロック図である。復号器は、接続された複数のモジュールによって表され、各モジュールは、例えば、装置300のCPU311によって実行されるプログラミング命令の形態で、復号器60によって実施される方法の対応するステップを実施するように適合されている。

【0096】

復号器60は、それぞれが、符号化パラメータの情報を含むヘッダと、符号化されたビデオデータを含むボディと、を含む複数の符号化ユニットを含むビットストリーム61を受信する。VVCにおけるビットストリームの構造は、図6を参照して以下に詳細に説明する。図4に関して説明したように、符号化ビデオデータはエンтроピー符号化されており、動きベクトル予測器のインデックスは、所定のブロックについて、所定のビット数で符号化されている。受信された符号化ビデオデータは、モジュール62によってエンтроピー復号される。そして、モジュール63によって残留データが逆量子化され、その後、モジュール64によって逆変換が施され、画素値が取得される。

10

【0097】

また、符号化モードを示すモードデータをエンтроピー復号し、そのモードに基づいて、符号化された画像データブロックに対してイントラタイプ復号またはインタータイプ復号を行う。

【0098】

イントラモードの場合、ビットストリームで指定されたイントラ予測モードに基づいて、イントラ逆予測モジュール65によりイントラ予測器が決定される。

20

【0099】

インターモードの場合、符号化器が使用する参照領域を見つけるために、ビットストリームから動き予測情報を抽出する。動き予測情報は、参照フレームインデックスと動きベクトル残差から構成される。動きベクトル予測情報は、動きベクトル復号モジュール70によって動きベクトルを得るために、動きベクトル残差に加えられる。

【0100】

動きベクトル復号モジュール70は、動き予測によって符号化された各現在のブロックに対して、動きベクトル復号を適用する。現在のブロックに対する動きベクトル予測器のインデックスが得られると、現在のブロックに関連する動きベクトルの実際の値が復号され、モジュール66によって逆動き補償を適用するために使用されることが可能である。復号された動きベクトルによって示される参照画像部分は、逆動き補償66を適用するために、参照画像68から抽出される。動きベクトルフィールドデータ71は、後続の復号された動きベクトルの逆予測に使用するために、復号された動きベクトルで更新される。

30

【0101】

最後に、復号されたブロックが得られる。ポストフィルタリングは、ポストフィルタリングモジュール67によって適用される。復号器60によって、最終的に復号されたビデオ信号69が提供される。

【0102】

図6は、JVET-Q2001-vDに記載された例示的な符号化システムVVCにおけるビットストリームの構成を示す図である。

40

【0103】

VVC符号化方式によるビットストリーム61は、シンタックス要素と符号化データの順序シーケンスで構成される。シンタックス要素と符号化データは、ネットワーク抽象化レイヤ(NAL)ユニット601~608に配置される。NALユニットには様々なタイプがある。ネットワーク抽象化レイヤは、RTP/IP(リアルタイムプロトコル/インターネットプロトコル)、ISOベースメディアファイルフォーマットなどの異なるプロトコルにビットストリームをカプセル化する能力を提供する。また、ネットワーク抽象化レイヤは、パケットロスの回復のためのフレームワークを提供する。

【0104】

50

NALユニットは、ビデオ符号化レイヤ(VCL)NALユニットと非VCL_NALユニットに分けられる。VCL_NALユニットには、実際に符号化されたビデオデータが含まれる。非VCL_NALユニットには、付加的な情報が含まれる。この追加情報は、符号化されたビデオデータの復号に必要なパラメータ、または復号されたビデオデータの使い勝手を向上させる可能性のある補足的なデータであってもよい。NALユニット606は、スライスに対応し、ビットストリームのVCL_NALユニットを構成する。

【0105】

異なるNALユニット601~605は、異なるパラメータセットに対応し、これらのNALユニットは非VCL_NALユニットである。復号器パラメータセット(DPS)NALユニット301は、所定の復号プロセスに対して一定であるパラメータを含む。ビデオパラメータセット(VPS)NALユニット602は、ビデオ全体、ひいてはビットストリーム全体に対して定義されたパラメータを含む。DPS_NALユニットは、VPSのパラメータよりも静的なパラメータを定義してもよい。すなわち、DPSのパラメータは、VPSのパラメータよりも変更頻度が低い。

10

【0106】

シーケンスパラメータセット(PPS)NALユニット603は、ビデオシーケンスに対して定義されたパラメータを含む。特に、PPS_NALユニットは、ビデオシーケンスのサブピクチャレイアウト及び関連するパラメータを定義することができる。各サブピクチャに関連するパラメータは、サブピクチャに適用される符号化制約を指定する。特に、サブピクチャ間の時間予測が、同じサブピクチャから来るデータに制限されることを示すフラグを含む。別のフラグは、サブピクチャの境界を越えてループフィルタを有効または無効にすることができる。

20

【0107】

ピクチャパラメータセット(PPS)NALユニット604、PPSは、ピクチャまたはピクチャのグループに対して定義されたパラメータを含む。アダプテーションパラメータセット(APS)NALユニット605は、典型的にはアダプテーションループフィルタ(ALF)またはリシェーパモデル(またはクロマスケーリング付きルママッピング(LMCS)モデル)またはスライスレベルで使用されるスケーリングマトリックスに対するパラメータを含んでいる。

【0108】

VVCの現行バージョンで提案されているPPSのシンタックスは、ルマサンプルでピクチャのサイズを指定し、さらに各ピクチャのタイルとスライスに分割するシンタックス要素で構成されている。

30

【0109】

PPSは、フレーム内のスライス位置を決定するためのシンタックス要素を含む。サブピクチャはフレーム内で矩形領域を形成するため、パラメータセットNALユニットでサブピクチャに属するスライスの集合、タイルの部分、タイルを決定することが可能である。PPSはAPSと同様にID機構を持ち、同じPPSの送信量を制限している。

【0110】

PPSとピクチャヘッダの主な違いはその伝送で、PPSは一般的にピクチャのグループに対して伝送されるのに対し、PHは各ピクチャに対してシステムティックに伝送される。従って、PPSはPHに比べ、複数のピクチャで一定となるパラメータを含んでいる。

40

【0111】

また、ビットストリームには、補足拡張情報(SEI)NALユニット(図6には表されていない)が含まれる場合がある。これらのパラメータセットのビットストリームにおける出現の周期は可変である。ビットストリーム全体に対して定義されるVPSは、ビットストリーム中に一度だけ出現する可能性がある。逆に、スライスに対して定義されるAPSは、各ピクチャの各スライスに対して一度だけ発生する可能性がある。実際には、異なるスライスが同じAPSに依存することがあり、したがって、一般に、各ピクチャにおけるスライスよりも少ないAPSが存在する。特に、APSはピクチャヘッダで定義され

50

る。しかし、ALFのAPSは、スライスヘッダで洗練されることができる。

【0112】

アクセスユニットデリミタ(AUD)NALユニット607は、2つのアクセスユニットを分離する。アクセスユニットは、同じ復号タイムスタンプを持つ1つ以上の符号化ピクチャを構成することができるNALユニットのセットである。このオプションのNALユニットは、現在のVVC仕様において1つのシンタックス要素: pic_typeのみを含み、このシンタックス要素は、AU内の符号化ピクチャの全てのスライスに対する slice_typeの値を示す。pic_typeが0である場合、AUはイントラスライスのみを含む。1である場合、PとIのスライスが含まれる。2の場合、B, P, イントラのいずれかのスライスを含む。このNALユニットは、pic_typeという1つのシンタックス要素のみを含む。

10

【0113】

【表1】

表1 シンタックスAUD

access_unit_delimiter_rbsp() {	Descriptor
pic_type	u(3)
rbsp_trailing_bits()	
}	

20

【0114】

JVET-Q2001-vDでは、pic_typeは次のように定義されている:
「pic_typeは、AUデリミタNALユニットを含むAU内の符号化ピクチャの全スライスの slice_type 値が、表2のpic_typeの値で示される集合のメンバであることを示す。pic_typeの値は、本規格の本バージョンに準拠したビットストリームにおいて、0、1、または2でなければならない。pic_typeの他の値は、ITU-TやISO/IECが将来使用するために予約されている。本仕様に準拠する復号器は、pic_typeの予約値を無視するものとする。」

30

【0115】

rbsp_trailing_bits()は、バイトの末尾に整列させるためにビットを追加する関数である。したがって、この関数を実行すると、パースされるビットストリームの量は整数バイトになる。

【0116】

【表2】

表2 pic_typeの解釈

pic_type	AU内に存在し得るslice_typeの値
0	I
1	P, I
2	B, P, I

40

【0117】

PH_NALユニット608は、1つの符号化ピクチャのスライス集合に共通するパラメータをグループ化するピクチャヘッダNALユニットである。ピクチャは、ピクチャのスライスによって使用されるAFLパラメータ、リシェーパモデル及びスケーリングマトリクスを示すために、1つ以上のAPSを参照することができる。

50

【 0 1 1 8 】

VCL_NALユニット606の各々は、スライスを含む。スライスは、ピクチャ全体またはサブピクチャ、単一のタイルまたは複数のタイルまたはタイルの一部に対応してもよい。例えば、図3のスライスは、複数のタイル620を含む。スライスは、スライスヘッダ610と、符号化ブロック640として符号化された符号化画素データを含むローバイトシーケンスペイロード(RBSP)611と、で構成される。

【 0 1 1 9 】

VVCの現行バージョンで提案されているPPSのシンタックスは、ルマサンプルでピクチャのサイズを指定し、さらに各ピクチャのタイルとスライスに分割するシンタックス要素で構成されている。

10

【 0 1 2 0 】

PPSは、フレーム内のスライス位置を決定するためのシンタックス要素を含む。サブピクチャはフレーム内の矩形領域を形成するため、パラメータセットNALユニットでスライスの集合、タイルの部分、またはサブピクチャに属するタイルを決定することが可能である。

【 0 1 2 1 】

NALユニットスライス

NALユニットスライス層は、表3に示すように、スライスヘッダとスライスデータを含む。

【 0 1 2 2 】

20

【表3】

表3 スライスレイヤのシンタックス

slice_layer_rbsp() {	Descriptor
slice_header()	
slice_data()	
rbpslice_trailing_bits()	
}	

30

【 0 1 2 3 】

APS

アダプテーションパラメータセット(APS)NALユニット605は、シンタックス要素を示す表4で定義されている。

【 0 1 2 4 】

表4に示すように、aps__params__typeシンタックス要素で与えられるAPSの種類は3つある：

- ALF__AP : ALFパラメータ用
- LMCS__APS : LMCSパラメータ用
- SCALING__APS : スケーリングリスト相対パラメータ用

40

【 0 1 2 5 】

50

【表 4】

表4 アダプテーションパラメータセットのシンタックス

adaptation_parameter_set_rbsp() {	Descriptor
adaptation_parameter_set_id	u(5)
aps_params_type	u(3)
if(aps_params_type == ALF_APS)	
alf_data()	
else if(aps_params_type == LMCS_APS)	
lmcs_data()	
else if(aps_params_type == SCALING_APS)	
scaling_list_data()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

10

20

【0126】

これら3種類のAPSパラメータについて、以下に順を追って説明する。

【0127】

ALFのAPS

ALFのパラメータは、アダプティブループフィルタのデータシンタックス要素に記述される(表5)。まず、4つのフラグにより、ルマおよびクロマのALFフィルタの有無、CbおよびCr成分のCC-ALF(クロスコンポーネント-アダプティブループフィルタ)の有無が指定される。ルマフィルタフラグが有効な場合、クリップ値がシグナリングされているかどうかを知るために、別のフラグが復号される(alf_luma_clip_flag)。次に、alf_luma_num_filters_signalled_minus1シンタックス要素を使用して、シグナリングされたフィルタの数が復号される。必要であれば、ALF係数デルタを表すシンタックス要素"alf_luma_coeff_delta_idx"は、各有効なフィルタに対して復号される。その後、各フィルタの各係数の絶対値と符号が復号される。

30

【0128】

alf_luma_clip_flagが有効な場合、有効な各フィルタの各係数のクリップインデックスが復号される。

40

【0129】

同様に、ALFのクロマ係数は必要に応じて復号される。

【0130】

CC-ALFがCrまたはCbに対して有効な場合、フィルタの数は復号され(alf_cc_cb_filters_signalled_minus1またはalf_cc_cr_filters_signalled_minus1)、関連する係数が復号される(alf_cc_cb_mapped_coeff_absおよびalf_cc_cb_coeff_signまたはそれぞれalf_cc_cr_mapped_coeff

50

f__abs および alf_cc_cr_coeff_sign)。

【 0 1 3 1 】

【 表 5 】

表5 アダプテーションループフィルタデータのシンタックス

alf_data() {	Descriptor
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
alf_cc_cb_filter_signal_flag	u(1)
alf_cc_cr_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	ue(v)
if(alf_luma_num_filters_signalled_minus1 > 0)	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	u(v)
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++)	
for(j = 0; j < 12; j++) {	
alf_luma_coeff_abs[sIdx][j]	ue(v)
if(alf_luma_coeff_abs[sIdx][j])	
alf_luma_coeff_sign[sIdx][j]	u(1)
}	
if(alf_luma_clip_flag)	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++)	
for(j = 0; j < 12; j++)	
alf_luma_clip_idx[sIdx][j]	u(2)
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_clip_flag	u(1)
alf_chroma_num_alt_filters_minus1	ue(v)
for(altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++) {	
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[altIdx][j]	ue(v)
if(alf_chroma_coeff_abs[altIdx][j] > 0)	
alf_chroma_coeff_sign[altIdx][j]	u(1)
}	
if(alf_chroma_clip_flag)	
for(j = 0; j < 6; j++)	
alf_chroma_clip_idx[altIdx][j]	u(2)
}	
}	

10

20

30

40

50

if(alf_cc_cb_filter_signal_flag) {	
alf_cc_cb_filters_signalled_minus1	ue(v)
for(k = 0; k < alf_cc_cb_filters_signalled_minus1 + 1; k++) {	
for(j = 0; j < 7; j++) {	
alf_cc_cb_mapped_coeff_abs[k][j]	u(3)
if(alf_cc_cb_mapped_coeff_abs[k][j])	
alf_cc_cb_coeff_sign[k][j]	u(1)
}	
}	
}	
if(alf_cc_cr_filter_signal_flag) {	
alf_cc_cr_filters_signalled_minus1	ue(v)
for(k = 0; k < alf_cc_cr_filters_signalled_minus1 + 1; k++) {	
for(j = 0; j < 7; j++) {	
alf_cc_cr_mapped_coeff_abs[k][j]	u(3)
if(alf_cc_cr_mapped_coeff_abs[k][j])	
alf_cc_cr_coeff_sign[k][j]	u(1)
}	
}	
}	
}	

10

20

【 0 1 3 2 】

ルママッピングとクロマスケールリング両方のLMCSシンタックス要素

以下の表6は、aps_params_typeパラメータが1に設定されるとき(LMCS_APS)、アダプテーションパラメータセット(APS)シンタックス構造で符号化されるすべてのLMCSシンタックス要素を与える。最大4つのLMCS_APSは、符号化されたビデオシーケンスにおいて使用することができるが、与えられたピクチャに対して単一のLMCS_APSのみを使用することができる。

30

【 0 1 3 3 】

これらのパラメータは、ルマの順方向および逆方向マッピング関数、クロマのスケールリング関数を構築するために使用される。

【 0 1 3 4 】

40

50

【表 6】

表6 クロマスケーリングを有するルママッピングデータのシンタックス

	Descriptor
lmcs_data() {	
lmcs_min_bin_idx	ue(v)
lmcs_delta_max_bin_idx	ue(v)
lmcs_delta_cw_prec_minus1	ue(v)
for(i = lmcs_min_bin_idx; i <= LmcsMaxBinIdx; i++) {	
lmcs_delta_abs_cw[i]	u(v)
if(lmcs_delta_abs_cw[i] > 0)	
lmcs_delta_sign_cw_flag[i]	u(1)
}	
lmcs_delta_abs_crs	u(3)
if(lmcs_delta_abs_crs > 0)	
lmcs_delta_sign_crs_flag	u(1)
}	

10

【 0 1 3 5 】

スケーリングリスト A P S

スケーリングリストは、定量化に使用される量子化マトリックスを更新する可能性を提供する。VVCでは、このスケーリングマトリックスは、スケーリングリストデータシンタックス要素（表7のスケーリングリストデータシンタックス）で説明されているように、APSでシグナリングされる。最初のシンタックス要素は、フラグ `scaling_matrix_for_lfnst_disabled_flag` に基づいて、LFNST（低周波非分離変換）ツールにスケーリングマトリックスが使用されるかどうかを指定する。2つ目は、クロマコンポーネントのためにスケーリングリストが使用される場合に指定される（`scaling_list_chroma_present_flag`）。それから、スケーリングマトリックスを構築するために必要なシンタックス要素が復号される（`scaling_list_copy_mode_flag`, `scaling_list_pred_mode_flag`, `scaling_list_pred_id_delta`, `scaling_list_dc_coef`, `scaling_list_delta_coef`）。

20

30

【 0 1 3 6 】

40

50

【表 7】

表7 スケーリングリストデータのシンタックス

scaling_list_data() {	Descriptor
scaling_matrix_for_lfnst_disabled_flag	u(1)
scaling_list_chroma_present_flag	u(1)
for(id = 0; id < 28; id ++)	
matrixSize = (id < 2) ? 2 : ((id < 8) ? 4 : 8)	
if(scaling_list_chroma_present_flag (id % 3 == 2) (id == 27)) {	
scaling_list_copy_mode_flag[id]	u(1)
if(!scaling_list_copy_mode_flag[id])	
scaling_list_pred_mode_flag[id]	u(1)
if((scaling_list_copy_mode_flag[id] scaling_list_pred_mode_flag[id]) && id != 0 && id != 2 && id != 8)	
scaling_list_pred_id_delta[id]	ue(v)
if(!scaling_list_copy_mode_flag[id]) {	
nextCoef = 0	
if(id > 13) {	
scaling_list_dc_coef[id - 14]	se(v)
nextCoef += scaling_list_dc_coef[id - 14]	
}	
for(i = 0; i < matrixSize * matrixSize; i++) {	
x = DiagScanOrder[3][3][i][0]	
y = DiagScanOrder[3][3][i][1]	
if(!(id > 25 && x >= 4 && y >= 4)) {	
scaling_list_delta_coef[id][i]	se(v)
nextCoef += scaling_list_delta_coef[id][i]	
}	
ScalingList[id][i] = nextCoef	
}	
}	
}	
}	
}	

10

20

30

【 0 1 3 7 】

ピクチャヘッダ

ピクチャヘッダは各ピクチャの先頭で他のスライスデータより先に伝送される。これは、以前の標準草案におけるヘッダと比較して非常に大きなものである。これらすべてのパラメータの完全な説明は、J V E T - Q 2 0 0 1 - v Dに記載されている。表 9 は、現在のピクチャヘッダ復号シンタックスにおけるこれらのパラメータを示す。

40

【 0 1 3 8 】

復号可能な関連するシンタックス要素は以下の通りである：

- ・このピクチャの使い方、参照フレームか否か
- ・ピクチャのタイプ
- ・出力フレーム
- ・ピクチャの番号
- ・サブピクチャ利用（必要な場合）
- ・参照画像リスト（必要な場合）

50

- ・カラープレーン（必要な場合）
- ・上書きフラグが有効な場合のパーティション更新
- ・デルタQPパラメータ（必要な場合）
- ・動き情報パラメータ（必要な場合）
- ・ALFパラメータ（必要な場合）
- ・SAOパラメータ（必要な場合）
- ・定量化パラメータ（必要な場合）
- ・LMCSパラメータ（必要な場合）
- ・スケーリングリストパラメータ（必要な場合）
- ・ピクチャヘッダ拡張（必要な場合）
- ・などなど...

10

【0139】

ピクチャ”タイプ”

最初のフラグは `gdr_or_irap_pic_flag` で、現在のピクチャが再同期ピクチャ（IRAPまたはGDR）であるかどうかを示すものである。このフラグが真の場合、`gdr_pic_flag` は、現在のピクチャがIRAPまたはGDRピクチャであるかどうかを知るために復号される。

【0140】

その後、`ph_inter_slice_allowed_flag` が復号され、インタースライスが許可されていることを識別する。

20

【0141】

許可された場合、フラグ `ph_intra_slice_allowed_flag` が復号され、現在のピクチャでイントラスライスが許可されているかどうかを知ることができる。

【0142】

次に、`non_reference_picture_flag`、PPS_IDを示す `ph_pic_parameter_set_id`、ピクチャオーダーカウンント `ph_pic_order_cnt_lsb` が復号される。ピクチャオーダーカウンントは、現在のピクチャの番号を示す。

【0143】

ピクチャがGDRまたはIRAPピクチャの場合、フラグ `no_output_of_prior_pics_flag` を復号する。また、ピクチャがGDRの場合、`recover_poc_cnt` が復号される。そして、必要に応じて `ph_poc_msb_present_flag` と `poc_msb_val` が復号される。

30

【0144】

ALF

現在のピクチャに関する重要な情報を記述するこれらのパラメータの後、ALFがSPSレベルで有効である場合、およびALFがピクチャヘッダレベルで有効である場合、ALFのAPS_IDシンタックス要素の集合が復号される。ALFは、`sps_alf_enabled_flag` フラグにより、SPSレベルで有効である。また、ALFシグナリングは、`alf_info_in_ph_flag` が1であるためピクチャヘッダレベルで有効であり、それ以外の場合（`alf_info_in_ph_flag` が0）はALFシグナリングはスライスレベルでシグナリングされる。

40

【0145】

`alf_info_in_ph_flag` は以下のように定義される：

「1である `alf_info_in_ph_flag` は、ALF情報がPHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッダには存在しないことを指定する。0である `alf_info_in_ph_flag` は、ALF情報がPHシンタックス構造には存在せず、PHシンタックス構造を含まないPPSを参照するスライスヘッダに存在し得ることを指定する。」

50

【0146】

まず、`ph_half_enabled_flag`が復号されるべきかどうか判断するために、`ph_half_enabled_present_flag`が復号される。`ph_half_enabled_flag`が有効な場合、現在のピクチャのすべてのスライスに対してALFが有効になる。

【0147】

ALFが有効な場合、luma用のALFのAPS__idの量は`pic_num_half_aps_ids_luma`シンタックス要素を使用して復号される。各APS__IDについて、lumaのためのAPS__ID値は、"`ph_half_aps_id_luma`"と復号される。

10

【0148】

クロマについては、シンタックス要素`ph_half_chroma_idc`を復号して、ALFが、クロマに対して有効か、Crのみ有効か、Cbのみ有効かを判断する。有効である場合、クロマ用のAPS__IDの値は`ph_half_aps_id_chroma`シンタックス要素を使用して復号される。CC-ALF方式のAPS__IDは、Cbおよび/またはCrコンポーネントに必要な場合、復号される。

【0149】

LMCS

LMCSがSPSレベルで有効であった場合、LMCSのAPS__IDシンタックス要素のセットが次に復号される。まず、`ph_lmcs_enabled_flag`が復号され、現在のピクチャに対してLMCSが有効であるか否かが判断される。LMCSが有効な場合、ID値`ph_lmcs_aps_id`が復号される。クロマのみに対して、`ph_chroma_residual_scale_flag`が復号され、クロマに対するメソッドを有効または無効にする。

20

【0150】

スケーリングリスト

スケーリングリストがSPSレベルで有効である場合、スケーリングリストのAPS__IDのセットが次に復号される。`ph_scaling_list_present_flag`は、現在のピクチャに対してスケーリングマトリックスが有効であるか否かを決定するために復号される。そして、APS__IDの値である`ph_scaling_list_aps_id`が復号される。

30

【0151】

サブピクチャ

サブピクチャパラメータは、SPSで有効になっておりサブピクチャIDシグナリングが無効になっている場合に有効である。また、仮想境界に関するいくつかの情報を含む。サブピクチャパラメータについては、8つのシンタックス要素が定義されている。

- ・`ph_virtual_boundaries_present_flag`
- ・`ph_num_ver_virtual_boundaries`
- ・`ph_virtual_boundaries_pos_x[i]`
- ・`ph_num_hor_virtual_boundaries`
- ・`ph_virtual_boundaries_pos_y[i]`

40

【0152】

出力フラグ

これらのサブピクチャパラメータの後には、存在すれば`pic_output_flag`が続く。

【0153】

参照ピクチャリスト

参照ピクチャリストがピクチャヘッダで通知される場合(`rpl_info_in_ph_flag`が1)、参照ピクチャリストのパラメータ`ref_pic_lists()`は復号され、以下のシンタックス要素を含んでいる：

50

- ・ r p l _ s p s _ f l a g []
- ・ r p l _ i d x []
- ・ p o c _ l s b _ l t [] []
- ・ d e l t a _ p o c _ m s b _ p r e s e n t _ f l a g [] []
- ・ d e l t a _ p o c _ m s b _ c y c l e _ l t [] []

【0154】

パーティション

パーティションパラメータのセットは、必要に応じて復号され、以下のシンタックス要素を含む：

- ・ p a r t i t i o n _ c o n s t r a i n t s _ o v e r r i d e _ f l a g 10
- ・ p h _ l o g 2 _ d i f f _ m i n _ q t _ m i n _ c b _ i n t r a _ s l i c e _ l u m a
- ・ p h _ m a x _ m t t _ h i e r a r c h y _ d e p t h _ i n t r a _ s l i c e _ l u m a
- ・ p h _ l o g 2 _ d i f f _ m a x _ b t _ m i n _ q t _ i n t r a _ s l i c e _ l u m a
- ・ p h _ l o g 2 _ d i f f _ m a x _ t t _ m i n _ q t _ i n t r a _ s l i c e _ l u m a
- ・ p h _ l o g 2 _ d i f f _ m i n _ q t _ m i n _ c b _ i n t r a _ s l i c e _ c h r o m a 20
- ・ p h _ m a x _ m t t _ h i e r a r c h y _ d e p t h _ i n t r a _ s l i c e _ c h r o m a
- ・ p h _ l o g 2 _ d i f f _ m a x _ b t _ m i n _ q t _ i n t r a _ s l i c e _ c h r o m a
- ・ p h _ l o g 2 _ d i f f _ m a x _ t t _ m i n _ q t _ i n t r a _ s l i c e _ c h r o m a
- ・ p h _ l o g 2 _ d i f f _ m i n _ q t _ m i n _ c b _ i n t e r _ s l i c e
- ・ p h _ m a x _ m t t _ h i e r a r c h y _ d e p t h _ i n t e r _ s l i c e
- ・ p h _ l o g 2 _ d i f f _ m a x _ b t _ m i n _ q t _ i n t e r _ s l i c e
- ・ p h _ l o g 2 _ d i f f _ m a x _ t t _ m i n _ q t _ i n t e r _ s l i c e 30

【0155】

重み付き予測

重み付き予測パラメータ `pred_weight_table()` は、PPSレベルで重み付き予測方法が有効であり、かつ重み付き予測パラメータがピクチャヘッダでシグナリングされている場合 (`wp_info_in_ph_flag` が 1)、復号される。

【0156】

`pred_weight_table()` は、リスト L0 のための重み付け予測パラメータと、双方向予測重み付け予測が有効である場合のリスト L1 のための重み付け予測パラメータを含む。重み付き予測パラメータがピクチャヘッダで伝送されるとき、各リストの重みの数は、`pred_weight_table()` シンタックステーブル (表 8) に描かれるように、明示的に伝送される。

【0157】

【表 8】

表8 重み付け予測パラメータのシンタックス

	Descriptor	
pred_weight_table() {		
luma_log2_weight_denom	ue(v)	
if(ChromaArrayType != 0)		
delta_chroma_log2_weight_denom	se(v)	
if(wp_info_in_ph_flag)		
num_l0_weights	ue(v)	
for(i = 0; i < NumWeightsL0; i++)		
luma_weight_l0_flag[i]	u(1)	10
if(ChromaArrayType != 0)		
for(i = 0; i < NumWeightsL0; i++)		
chroma_weight_l0_flag[i]	u(1)	
for(i = 0; i < NumWeightsL0; i++) {		
if(luma_weight_l0_flag[i]) {		
delta_luma_weight_l0[i]	se(v)	
luma_offset_l0[i]	se(v)	
}		
if(chroma_weight_l0_flag[i])		
for(j = 0; j < 2; j++) {		
delta_chroma_weight_l0[i][j]	se(v)	20
delta_chroma_offset_l0[i][j]	se(v)	
}		
}		
if(pps_weighted_bipred_flag && wp_info_in_ph_flag)		
num_l1_weights	ue(v)	
for(i = 0; i < NumWeightsL1; i++)		
luma_weight_l1_flag[i]	u(1)	
if(ChromaArrayType != 0)		
for(i = 0; i < NumWeightsL1; i++)		
chroma_weight_l1_flag[i]	u(1)	30
for(i = 0; i < NumWeightsL1; i++) {		
if(luma_weight_l1_flag[i]) {		
delta_luma_weight_l1[i]	se(v)	
luma_offset_l1[i]	se(v)	
}		
if(chroma_weight_l1_flag[i])		
for(j = 0; j < 2; j++) {		
delta_chroma_weight_l1[i][j]	se(v)	
delta_chroma_offset_l1[i][j]	se(v)	
}		
}		
}		
}		
}		
}		40

【 0 1 5 8 】

デルタ Q P

ピクチャがイントラの場合、必要に応じて `ph_cu_qp_delta_subdiv_intra_slice` と `ph_cu_chroma_qp_offset_subdiv_intra_slice` が復号される。また、インタースライスが許可されている場合、`ph_cu_qp_delta_subdiv_inter_slice` と `ph_cu_chroma_qp_offset_subdiv_inter_slice` は、必要に応じて復号される。最後に、必要に応じてピクチャヘッダ拡張シンタックス要素が復号される。

【 0 1 5 9 】

alf_info_in_ph_flag, rpl_info_in_ph_flag, qp_delta_info_in_ph_flag, sao_info_in_ph_flag, dbf_info_in_ph_flag, wp_info_in_ph_flag のすべてのパラメータは、PPS でシグナリングされる。

【 0 1 6 0 】

【表 9】

表9 ピクチャヘッダの構造

picture_header_structure() {	Descriptor	
gdr_or_irap_pic_flag	u(1)	10
if(gdr_or_irap_pic_flag)		
gdr_pic_flag	u(1)	
ph_inter_slice_allowed_flag	u(1)	
if(ph_inter_slice_allowed_flag)		
ph_intra_slice_allowed_flag	u(1)	
non_reference_picture_flag	u(1)	
ph_pic_parameter_set_id	ue(v)	
ph_pic_order_cnt_lsb	u(v)	
if(gdr_or_irap_pic_flag)		
no_output_of_prior_pics_flag	u(1)	20
if(gdr_pic_flag)		
recovery_poc_cnt	ue(v)	
for(i = 0; i < NumExtraPhBits; i++)		
ph_extra_bit[i]	u(1)	
if(sps_poc_msb_flag) {		
ph_poc_msb_present_flag	u(1)	
if(ph_poc_msb_present_flag)		
poc_msb_val	u(v)	
}		
if(sps_alf_enabled_flag && alf_info_in_ph_flag) {		30
ph_alf_enabled_flag	u(1)	
if(ph_alf_enabled_flag) {		
ph_num_alf_aps_ids_luma	u(3)	
for(i = 0; i < ph_num_alf_aps_ids_luma; i++)		
ph_alf_aps_id_luma[i]	u(3)	
if(ChromaArrayType != 0)		
ph_alf_chroma_idc	u(2)	
if(ph_alf_chroma_idc > 0)		
ph_alf_aps_id_chroma	u(3)	
if(sps_ccalf_enabled_flag) {		40

10

20

30

40

50

ph_cc_alf_cb_enabled_flag	u(1)	
if(ph_cc_alf_cb_enabled_flag)		
ph_cc_alf_cb_aps_id	u(3)	
ph_cc_alf_cr_enabled_flag	u(1)	
if(ph_cc_alf_cr_enabled_flag)		
ph_cc_alf_cr_aps_id	u(3)	
}		
}		
}		
if(sps_lmcs_enabled_flag) {		10
ph_lmcs_enabled_flag	u(1)	
if(ph_lmcs_enabled_flag) {		
ph_lmcs_aps_id	u(2)	
if(ChromaArrayType != 0)		
ph_chroma_residual_scale_flag	u(1)	
}		
}		
if(sps_scaling_list_enabled_flag) {		
ph_scaling_list_present_flag	u(1)	
if(ph_scaling_list_present_flag)		
ph_scaling_list_aps_id	u(3)	
}		20
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {		
ph_virtual_boundaries_present_flag	u(1)	
if(ph_virtual_boundaries_present_flag) {		
ph_num_ver_virtual_boundaries	u(2)	
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)		
ph_virtual_boundaries_pos_x[i]	u(13)	
ph_num_hor_virtual_boundaries	u(2)	
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)		
ph_virtual_boundaries_pos_y[i]	u(13)	
}		
}		
if(output_flag_present_flag)		30
pic_output_flag	u(1)	
if(rpl_info_in_ph_flag)		
ref_pic_lists()		
if(partition_constraints_override_enabled_flag)		
partition_constraints_override_flag	u(1)	
if(ph_intra_slice_allowed_flag) {		
if(partition_constraints_override_flag) {		
ph_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)	
ph_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)	
if(ph_max_mtt_hierarchy_depth_intra_slice_luma != 0) {		
ph_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)	
ph_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)	40

10

20

30

40

50

}	
if(qbtt_dual_tree_intra_flag) {	
ph_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
ph_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
ph_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
ph_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(ph_inter_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph_log2_diff_min_qt_min_cb_inter_slice	ue(v)
ph_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(ph_max_mtt_hierarchy_depth_inter_slice != 0) {	
ph_log2_diff_max_bt_min_qt_inter_slice	ue(v)
ph_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag) {	
ph_temporal_mvp_enabled_flag	u(1)
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplsIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplsIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
}	
mvd_l1_zero_flag	u(1)
if(sps_fpel_mmvd_enabled_flag)	
ph_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
ph_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
ph_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	

10

20

30

40

50

ph_disable_prof_flag	u(1)	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag)		
pred_weight_table()		
}		
if(qp_delta_info_in_ph_flag)		
ph_qp_delta	se(v)	
if(sps_joint_cbr_enabled_flag)		
ph_joint_cbr_sign_flag	u(1)	
if(sps_sao_enabled_flag && sao_info_in_ph_flag) {		10
ph_sao_luma_enabled_flag	u(1)	
if(ChromaArrayType != 0)		
ph_sao_chroma_enabled_flag	u(1)	
}		
if(sps_dep_quant_enabled_flag)		
ph_dep_quant_enabled_flag	u(1)	
if(sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag)		
pic_sign_data_hiding_enabled_flag	u(1)	
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {		20
ph_deblocking_filter_override_flag	u(1)	
if(ph_deblocking_filter_override_flag) {		
ph_deblocking_filter_disabled_flag	u(1)	
if(!ph_deblocking_filter_disabled_flag) {		
ph_beta_offset_div2	se(v)	
ph_tc_offset_div2	se(v)	
ph_cb_beta_offset_div2	se(v)	
ph_cb_tc_offset_div2	se(v)	
ph_cr_beta_offset_div2	se(v)	
ph_cr_tc_offset_div2	se(v)	
}		30
}		
}		
if(picture_header_extension_present_flag) {		
ph_extension_length	ue(v)	
for(i = 0; i < ph_extension_length; i++)		
ph_extension_data_byte[i]	u(8)	
}		
}		

【 0 1 6 1 】

スライスヘッダ

スライスヘッダは、各スライスの先頭で送信される。スライスヘッダは、約 65 個のシンタックス要素を含む。これは、以前のビデオ符号化規格のスライスヘッダと比較して非常に大きい。すべてのスライスヘッダパラメータの完全な説明は、JVET-Q2001-vD に記載されている。表 10 は、現在のスライスヘッダ復号シンタックスにおけるこれらのパラメータを示す。

【 0 1 6 2 】

10

20

30

40

50

【表 1 0】

表10 部分スライスヘッダ

slice_header() {	Descriptor	
picture_header_in_slice_header_flag	u(1)	
if(picture_header_in_slice_header_flag)		
picture_header_structure()		
if(subpic_info_present_flag)		
slice_subpic_id	u(v)	
if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1))		
slice_address	u(v)	10
for(i = 0; i < NumExtraShBits; i++)		
sh_extra_bit[i]	u(1)	
if(!rect_slice_flag && NumTilesInPic > 1)		
num_tiles_in_slice_minus1	ue(v)	
if(ph_inter_slice_allowed_flag)		
slice_type	ue(v)	
if(sps_alf_enabled_flag && !alf_info_in_ph_flag) {		
slice_alf_enabled_flag	u(1)	
if(slice_alf_enabled_flag) {		
slice_num_alf_aps_ids_luma	u(3)	
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)		
slice_alf_aps_id_luma[i]	u(3)	20
if(ChromaArrayType != 0)		
slice_alf_chroma_idc	u(2)	
if(slice_alf_chroma_idc)		
slice_alf_aps_id_chroma	u(3)	
if(sps_ccalf_enabled_flag) {		
slice_cc_alf_cb_enabled_flag	u(1)	
if(slice_cc_alf_cb_enabled_flag)		
slice_cc_alf_cb_aps_id	u(3)	
slice_cc_alf_cr_enabled_flag	u(1)	
if(slice_cc_alf_cr_enabled_flag)		
slice_cc_alf_cr_aps_id	u(3)	
}		30
}		
}		
if(separate_colour_plane_flag == 1)		
colour_plane_id	u(2)	
if(!rpl_info_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))		
ref_pic_lists()		
if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && ((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {		
num_ref_idx_active_override_flag	u(1)	40
if(num_ref_idx_active_override_flag)		

10

20

30

40

50

for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag) {	
if(slice_type == B)	
slice_collocated_from_l0_flag	u(1)
if((slice_collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!slice_collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
slice_collocated_ref_idx	ue(v)
}	
if(!wp_info_in_ph_flag && ((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)))	
pred_weight_table()	
}	
if(!qp_delta_info_in_ph_flag)	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !sao_info_in_ph_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
slice_cb_beta_offset_div2	se(v)
slice_cb_tc_offset_div2	se(v)
slice_cr_beta_offset_div2	se(v)
slice_cr_tc_offset_div2	se(v)
}	
}	

10

20

30

40

50

slice_ts_residual_coding_disabled_flag	u(1)
if(ph_lmcs_enabled_flag)	
slice_lmcs_enabled_flag	u(1)
if(ph_scaling_list_present_flag)	
slice_scaling_list_present_flag	u(1)
if(NumEntryPoints > 0) {	
offset_len_minus1	uc(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	uc(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

【 0 1 6 3 】

20

まず、`picture_header_in_slice_header_flag`を復号して、スライスヘッダに`picture_header_structure()`が存在するかどうかを知る。

【 0 1 6 4 】

次に、必要であれば、`slice_subpic_id`が復号され、現在のスライスのサブピクチャIDが決定される。次に、`slice_address`は、現在のスライスのアドレスを決定するために復号される。次に、現在のピクチャのタイル数が1より大きい場合、`num_tiles_in_slice_minus1`が復号される。

【 0 1 6 5 】

その後、`slice_type`が復号される。

30

【 0 1 6 6 】

ALFがSPSレベルで有効であり(`sps_alf_enabled_flag`)、ALFがスライスヘッダでシグナリングされる場合(`alf_info_in_ph_flag`が0)、ALF情報は復号される。これには、現在のスライスに対してALFが有効であることを示すフラグ(`slice_alf_enabled_flag`)が含まれる。それが有効である場合、ルマのためのAPSのALF_IDの数(`slice_num_alf_aps_ids_luma`)が復号され、次にAPS_IDが復号される(`slice_alf_aps_id_luma[i]`)。次に、`slice_alf_chroma_idc`が復号され、クロマ成分に対してALFが有効であるかどうか、またどのクロマ成分に対して有効であるかを知る。次に、クロマ用のAPS_IDslice_alf_aps_id_chromaが必要に応じて復号される。同様に、CC_ALF方式が有効であるかどうかを知るために、必要に応じて、`slice_cc_alf_cb_enabled_flag`が復号される。もしCC_ALFがCr及び/又はCbに対して有効である場合、Cr及び/又はCbの関連するAPS_IDが復号される。

40

【 0 1 6 7 】

複数のカラープレーンが独立して伝送される場合(`separate_colour_plane_flag`が1)、`color_plane_id`が復号される。ピクチャヘッダで参照ピクチャリストを伝送せず(`rpl_info_in_ph_flag`が0)、かつ、NALユニットがIDRでないまたはIDRピクチャに対して参照ピクチャリストを伝送する場合(`sps_idr_rpl_present_flag`が1)、参照ピ

50

クチャリストパラメータが復号される；これらはピクチャヘッダのパラメータと同様のパラメータである。

【0168】

参照ピクチャリストがピクチャヘッダで伝送される場合 (`rpl__info__in__ph__flag`が1)、または、NALユニットがIDRでないまたは参照ピクチャリストがIDRピクチャで伝送される場合 (`sps__idr__rpl__present__flag`が1)、少なくとも1つのリストに対する参照数が1より大きい場合、オーバーライドフラグ `num__ref__idx__active__override__flag` が復号される。このフラグが有効な場合、各リストの参照インデックスが復号される。

【0169】

スライスタイプがイントラではない場合、および必要に応じて、`cabac__init__flag` が復号される。スライスヘッダで参照ピクチャリストが伝送されが来ている場合、`slice__collocated__from__l0__flag` および `slice__collocated__ref__idx` が復号される。これらのデータは、CABAC符号化および動きベクトル配置に関連する。同様に、スライスタイプがイントラでない場合は、重み付き予測 `pred__weight__table()` のパラメータが復号される。

【0170】

`slice__qp__delta` は、デルタQP情報がスライスヘッダで伝送される場合 (`qp__delta__info__in__ph__flag`が0) に復号される。必要であれば、`slice__cb__qp__offset`、`slice__cr__qp__offset`、`slice__joint__cbcr__qp__offset`、`cu__chroma__qp__offset__enabled__flag` のシンタックス要素も復号される。

【0171】

SAO情報がスライスヘッダで伝送され (`sao__info__in__ph__flag`が0)、SPSレベルで有効になっている場合 (`sps__sao__enabled__flag`) は、ルマおよびクロマの両方でSAOに対する有効フラグが復号される：`slice__sao__luma__flag`、`slice__sao__chroma__flag`。次に、デブロッキングフィルタパラメータは、それらがスライスヘッダでシグナリングされる場合 (`dbf__info__in__ph__flag`が0) に、復号される。

【0172】

フラグ `slice__ts__residual__coding__disabled__flag` は、現在のスライスに対して変換スキップ残留符号化法が有効であるかどうかを知るために系統的に復号される。

【0173】

LMCSがピクチャヘッダで有効であった場合 (`ph__lmcs__enabled__flag`が1)、フラグ `slice__lmcs__enabled__flag` が復号される。

【0174】

同様に、スケーリングリストがピクチャヘッダで有効になっていた場合 (`phpic__scaling__list__present__enabled__flag`が1) は、フラグ `slice__scaling__list__present__flag` が復号される。

【0175】

その後、必要に応じて他のパラメータが復号される。

【0176】

スライスヘッダ内のピクチャヘッダ

特定のシグナリング方法では、図7に描かれているように、ピクチャヘッダ708は、スライスヘッダ710の内部でシグナリングされ得る。その場合、ピクチャヘッダ608のみを含むNALユニットは存在しない。ユニット701、702、703、704、705、706、707、720、740は、図6の601、602、603、604、605、606、607、620、640に対応しており、したがって、先の説明から理解することが可能である。これは、フラグ `picture__header__in__slice`

10

20

30

40

50

e_header_flagのおかげで、スライスヘッダにおいて有効にすることができる。さらに、ピクチャヘッダがスライスヘッダ内にシグナリングされる場合、ピクチャは1つのスライスのみを含むものとする。従って、1枚のピクチャには常に1つのピクチャヘッダしか存在しない。さらに、フラグpicture_header_in_slice_header_flagは、CLVS（符号化済レイヤビデオシーケンス）のすべてのピクチャで同じ値を持つものとする。これは、第1のIRAPを含む2つのIRAP間のすべてのピクチャは、1つのピクチャにつき1つのスライスのみであることを意味する。

【0177】

フラグpicture_header_in_slice_header_flagは次のように定義されている：

10

「picture_header_in_slice_header_flagが1の場合、スライスヘッダにPHシンタックス構造が存在することを示す。picture_header_in_slice_header_flagが0の場合、スライスヘッダにPHシンタックス構造が存在しないことを示す。

CLVS内のすべての符号化スライスにおいてpicture_header_in_slice_header_flagの値が同じであることは、ビットストリーム適合性の要件である。符号化スライスでpicture_header_in_slice_header_flagが1のときは、CLVS内にnal_unit_typeがPH_NUTであるVCL_NALユニットが存在しないことが適合性の要件となる。

picture_header_in_slice_header_flagが0のとき、現在のピクチャのすべての符号化スライスはpicture_header_in_slice_header_flagが0となり、現在のPUはPH_NALユニットを持つ。

20

picture_header_structure()は、スタッフビットrbsp_trailing_bits()を除くpicture_rbsp()のシンタックス要素を含む。」

【0178】

スライスヘッダにおけるピクチャヘッダとピクチャヘッダおよびスライスヘッダにおけるツールシグナリングの相互作用

QPデルタ情報、参照ピクチャリストパラメータ、デブロッキングフィルタパラメータ、サンプルアダプテーションオフセットパラメータ、重み付き予測パラメータ及びALFパラメータは、それぞれのフラグにより、ピクチャヘッダ又はスライスヘッダにおいて伝送することができる：

30

- qp_delta_info_in_ph_flag
- rpl_info_in_ph_flag
- dbf_info_in_ph_flag
- sa_o_info_in_ph_flag
- wp_info_in_ph_flag
- alf_info_in_ph_flag

これらのフラグは、PPS内で送信される。

40

【0179】

表11（フラグpicture_header_in_slice_header_flagおよびxxx_info_in_ph_flagに従ったXXXシグナリングの概要）に示されるように、言及したツールの1つに対する「xxx」を考慮すれば、現在のシンタックスでxxx_info_in_ph_flagが0に設定されるとxxxはスライスヘッダ内でシグナリングすることが可能である。すなわち、以降の説明では、ピクチャヘッダとスライスヘッダの両方でシグナリングされ得る情報タイプ（その例は上に示した）を参照するために、XXXまたはxxxという略記を使用する。

【0180】

50

【表 1 1】

表11 フラグpicture_header_in_slice_header_flagおよびxxx_info_in_ph_flagに従ったXXXシグナリングの概要

	xxx_info_in_ph_flag = 1	xxx_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	ピクチャヘッダ内にXXX	スライスヘッダ内にXXX、または存在しない
picture_header_in_slice_header_flag = 0	ピクチャヘッダ内にXXX	スライスヘッダ内にXXX、または存在しない

10

【0 1 8 1】

ピクチャヘッダがスライスヘッダ内にある場合、これは現在のピクチャに対して1つのスライスのみが存在することを意味する。したがって、スライスとピクチャの両方について情報を送信または送信可能にしても、パラメータが同じになるため、符号化器または復号器の柔軟性が向上しない。つまり、ピクチャヘッダに情報がある場合、スライスヘッダの対応する情報は冗長となる。同様に、情報がスライスヘッダ内にある場合、ピクチャヘッダ内の対応する情報は冗長となる。本明細書で説明する実施形態は、同じ符号化のシグナリングにおける冗長性を制限することによって、復号器の実施を簡素化する。特に、実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされるとき、情報はピクチャヘッダ内にあることは許されるが、スライスヘッダ内にあることは許されない。

20

【0 1 8 2】

ストリーミングアプリケーション

ストリーミングアプリケーションによっては、ビットストリームの特定の部分のみを抽出するものもある。これらの抽出は、空間的（サブピクチャとして）または時間的（ビデオシーケンスのサブパート）であることができる。そして、これらの抽出された部分は、他のビットストリームとマージされ得る。また、一部のフレームのみを抽出することでフレームレートを低下させるものもある。一般に、これらのストリーミングアプリケーションの主な目的は、許容される帯域幅の最大値を使用して、エンドユーザーに最大限の品質を提供することである。

30

【0 1 8 3】

VVCでは、フレームレート低減のため、あるフレームの新しいAPS_ID番号を時間的階層の上位のフレームに使用できないように、APS_IDのナンバリングを制限している。しかし、ビットストリームの一部を抽出するストリーミングアプリケーションでは、フレーム（IRAPとして）がAPS_IDのナンバリングをリセットしないため、ビットストリームの一部に対してどのAPSを保持すべきかを追跡する必要がある。

【0 1 8 4】

L MCS（クロマスケーリング付きルママッピング）

クロマスケーリング付きルママッピング（L MCS）技術は、VVCなどのビデオ復号器でループフィルタを適用する前にブロックに適用するサンプル値の変換方法である。

40

【0 1 8 5】

L MCSは2つのサブツールに分けることができる。後述のように、1番目のツールはルマブロックに適用され、2番目のツールはクロマブロックに適用される：

1) 1番目のサブツールは、適応的ピースワイズリニアモデルに基づくルマ成分のインループマッピングである。ルマ成分のインループマッピングは、圧縮効率を向上させるために、ダイナミックレンジ全体にコードワードを再分配することによって、入力信号のダイナミックレンジを調整する。ルママッピングでは、「マッピング領域」への順方向のマッピング関数と、「入力領域」に戻るための対応する逆方向のマッピング関数が使用される。

50

2) 2番目のサブツールはクロマ成分に関するもので、ルマに依存したクロマ残差スケールリングが適用される。クロマ残差スケールリングは、ルマ信号とその対応するクロマ信号の間の相互作用を補償するように設計されている。クロマ残差スケールリングは、現在のブロックの再構成された近隣（上および/または左）のルマサンプルの平均値に依存する。

【0186】

LMCSは、VVCなどのビデオ符号化の他のツールと同様に、SPSフラグを使用してシーケンスレベルで有効/無効を設定することができる。クロマ残差スケールリングが有効かどうかは、スライスレベルでも通知される。ルママッピングが有効な場合、ルマ依存のクロマ残差スケールリングが有効か否かを示すフラグが追加で通知される。ルママッピングを使用しない場合、ルマ依存のクロマ残差スケールリングは完全に無効である。また、クロマブロックのサイズが4以下の場合、クロマ依存のクロマ残差スケールリングは常に無効となる。

10

【0187】

図8は、上記で説明したLMCSの原理をルママッピングサブツールについて示したものである。図8のハッチングを施したブロックが、ルマ信号の順方向および逆方向のマッピングを含む新しいLMCSの機能ブロックである。LMCSを使用する場合、一部の復号オペレーションは「マッピングされた領域」で適用されることに留意する必要がある。これらの演算は、この図8において破線のブロックによって表される。これらは通常、逆量子化、逆変換、ルマイントラ予測、ルマ予測とルマ残差の加算を含む再構成ステップに対応する。逆に、図8の実線のブロックは、オリジナル（すなわち、マップされていない）領域で復号処理が適用される場所を示し、これには、デブロッキング、ALF、SAOなどのループフィルタ処理、動き補償予測、および参照ピクチャ（DPB）としての復号画像の保存が含まれる。

20

【0188】

図9は、図8と同様の図であるが、今回はLMCSツールのクロマスケールリングサブツールの図である。図9のハッチングを施したブロックは、ルマ依存のクロマスケールリング処理を含む新しいLMCSの機能ブロックである。ただし、クロマでは、ルマの場合と比較して、いくつかの重要な相違点がある。ここでは、クロマサンプルに対して、逆量子化と破線のブロックで表される逆変換のみが「マッピングされた領域」で実行される。クロマ内予測、動き補償、ループフィルタリングの他のすべてのステップは、オリジナルの領域で実行される。図9に示すように、スケールリング処理のみ存在し、ルママッピングのような順変換と逆変換の処理はない。

30

【0189】

ピースワイズリニアモデルによるルママッピング。

ルママッピングサブツールでは、ピースワイズリニアモデルを使用している。つまり、ピースワイズリニアモデルは入力信号のダイナミックレンジを16等分のサブレンジに分離し、各サブレンジに対して、そのリニアマッピングのパラメータを、そのレンジに割り当てられたコードワードの数で表現している。

【0190】

ルママッピングのセマンティクス

40

`lmc_s_min_bin_idx`は、クロマスケールリング付きルママッピング（LMCS）構築処理で使用する最小のピンインデックスを指定する。`lmc_s_min_bin_idx`の値は、0から15の範囲である。

【0191】

`lmc_s_delta_max_bin_idx`は、クロマスケールリング付きルママッピング構築処理で使用される最大ピンインデックス`Lmc_s_Max_Bin_idx`と15との差分値を指定する。`lmc_s_delta_max_bin_idx`の値は、0から15までの範囲でなければならない。`Lmc_s_Max_Bin_idx`の値は、`15 - lmc_s_delta_max_bin_idx`に設定される。`Lmc_s_Max_Bin_idx`の値は`lmc_s_min_bin_idx`以上でなければならない。

50

【0192】

シンタックス要素 `lmcs_delta_cw_minus1` プラス1は、シンタックス `lmcs_delta_abs_cw[i]` の表現に使用するビット数を指定する。

【0193】

シンタックス要素 `lmcs_delta_abs_cw[i]` は、*i* 番目のビンの絶対デルタコードワード値を指定する。

【0194】

シンタックス要素 `lmcs_delta_sign_cw_flag[i]` は、変数 `lmcsDeltaCW[i]` の符号を指定する。 `lmcs_delta_sign_cw_flag[i]` が存在しない場合、0であると推論される。

10

【0195】

ルママッピングのためのLMCS中間変数計算

順方向および逆方向のルママッピング処理を適用するために、いくつかの中間変数とデータ配列が必要である。

【0196】

まず、変数 `OrgCW` は以下のように導出される：

$$\text{OrgCW} = (1 \ll \text{BitDepth}) / 16$$

【0197】

次に、変数 `lmcsDeltaCW[i]` ($i = \text{lmcs_min_bin_idx} \dots \text{lmcs_max_bin_idx}$) は、次のように計算される：

20

$$\text{lmcsDeltaCW}[i] = (1 - 2 * \text{lmcs_delta_sign_cw_flag}[i]) * \text{lmcs_delta_abs_cw}[i]$$

【0198】

新しい変数 `lmcsCW[i]` は次のように導出される：

$i = 0 \dots \text{lmcs_min_bin_idx} - 1$ の場合、`lmcsCW[i]` は0に設定される。

$i = \text{lmcs_min_bin_idx} \dots \text{lmcs_max_bin_idx}$ の場合、以下が適用される：

$$\text{lmcsCW}[i] = \text{OrgCW} + \text{lmcsDeltaCW}[i]$$

30

`lmcsCW[i]` の値は、 $(\text{OrgCW} \gg 3)$ から $(\text{OrgCW} \ll 3 - 1)$ の範囲でなければならない。

$i = \text{lmcs_max_bin_idx} + 1 \dots 15$ に対して、`lmcsCW[i]` は0に設定される。

【0199】

変数 `InputPivot[i]` ($i = 0 \dots 16$) は、次のように導出される：

$$\text{InputPivot}[i] = i * \text{OrgCW}$$

【0200】

変数 `LmcsPivot[i]` ($i = 0 \dots 16$)、変数 `ScaleCoeff[i]` および `InvScaleCoeff[i]` ($i = 0 \dots 15$) は、次のように計算される：

40

```

LmcsPivot[0] = 0;
for (i = 0; i <= 15; i++) {
    LmcsPivot[i+1] = LmcsPivot[i] + lmcsCW[i]
    ScaleCoeff[i] = (lmcsCW[i] * (1 << 11) + (1 <
< (Log2(OrgCW) - 1))) >> (Log2(OrgCW))
    if (lmcsCW[i] == 0)
        InvScaleCoeff[i] = 0
    else
        InvScaleCoeff[i] = OrgCW * (1 << 11) / lmcs

```

`CW[i]`

50

【0201】

順ルママッピング

図8に示すように、LMCSをルマに適用した場合、予測サンプル $predSamples[i][j]$ から、 $predMapSamples[i][j]$ というLumaリマップサンプルが得られる。

【0202】

$predMapSamples[i][j]$ は以下のように計算される：

まず、予測サンプル $predSamples[i][j]$ から、位置 (i, j) のインデックス $idxY$ が計算される。

$$idxY = predSamples[i][j] \gg \text{Log2}(OrgCW) \quad 10$$

そして、第0セクションの中間変数 $idxY$ 、 $LmcsPivot[idxY]$ 、 $InputPivot[idxY]$ を用いて、 $predMapSamples[i][j]$ は以下のように導出される：

$$predMapSamples[i][j] = LmcsPivot[idxY] + (ScaleCoeff[idxY] * (predSamples[i][j] - InputPivot[idxY]) + (1 \ll 10)) \gg 11$$

【0203】

ルマ再構成サンプル

予測されたルマサンプル $predMapSample[i][j]$ と残留ルマサンプル $resiSamples[i][j]$ から再構成処理を求める。

【0204】

再構成されたルマ画像サンプル $recSamples[i][j]$ は、以下のように $resiSamples[i][j]$ に $predMapSample[i][j]$ を加算するのみで得られる：

$$recSamples[i][j] = Clip1(predMapSamples[i][j] + resiSamples[i][j])$$

【0205】

上述の関係において、 $Clip1$ 関数は、再構成されたサンプルが 0 と $1 \ll BitDepth - 1$ の間にあることを確認するためのクリッピング関数である。

【0206】

逆ルママッピング

図8に従って逆ルママッピングを適用する場合、処理中の現在のブロックの各サンプル $recSample[i][j]$ に対して、以下の演算が適用される：

【0207】

まず、位置 (i, j) の再構成サンプル $recSamples[i][j]$ から、インデックス $idxY$ を計算する。

$$idxY = recSamples[i][j] \gg \text{Log2}(OrgCW)$$

逆マップされたルマサンプル $invLumaSample[i][j]$ を以下のように導出する：

$$invLumaSample[i][j] = InputPivot[idxYInv] + (InvScaleCoeff[idxYInv] * (recSample[i][j] - LmcsPivot[idxYInv]) + (1 \ll 10)) \gg 11 \quad 40$$

その後、クリッピング操作を行い、最終的なサンプルを得る：

$$finalSample[i][j] = Clip1(invLumaSample[i][j])$$

【0208】

クロマスケーリング

クロマスケーリングのためのLMCSセマンティクス

表6のシンタックス要素 $lmcs_delta_abs_crs$ は、変数 $lmcsDeltaCrs$ のコードワード絶対値を指定する。 $lmcs_delta_abs_crs$

の値は、0 から 7 の範囲である。存在しない場合、`lmcs_delta_abs_crs` は 0 であると推論される。

【0209】

シンタックス要素 `lmcs_delta_sign_crs_flag` は、変数 `lmcsDeltaCrs` の符号を指定する。存在しない場合、`lmcs_delta_sign_crs_flag` は 0 であると推論される。

【0210】

クロマスケーリングのための LMCS 中間変数計算

クロマスケーリング処理を適用するためには、いくつかの中間変数が必要である。

変数 `lmcsDeltaCrs` は以下のように導出される：

```
lmcsDeltaCrs = (1 - 2 * lmcs_delta_sign_crs_flag) * lmcs_delta_abs_crs
```

【0211】

変数 `ChromaScaleCoeff[i]` ($i = 0 \dots 15$) は、次のように導出される：

```
if (lmcsCW[i] == 0)
    ChromaScaleCoeff[i] = (1 << 11)
else
    ChromaScaleCoeff[i] = OrgCW * (1 << 11) / (lmcsCW[i] + lmcsDeltaCrs)
```

【0212】

クロマスケーリング処理

最初のステップでは、現在の対応するクロマブロックの周囲の再構成されたルマサンプルの平均ルマ値を計算するために、変数 `invAvgLuma` が導出される。平均ルマは、対応するクロマブロックを囲む左と上のルマブロックから計算される。サンプルがない場合、変数 `invAvgLuma` は以下のように設定される：

```
invAvgLuma = 1 << (BitDepth - 1)
```

【0213】

そして、第 0 セクションの中間配列 `lmcsPivot[]` に基づいて、変数 `idxYInv` を以下のように導出する：

```
For (idxYInv = lmcs_min_bin_idx; idxYInv <= lmcsMaxBinIdx; idxYInv++) {
    if (invAvgLuma < lmcsPivot[idxYInv + 1])
        break
}
IdxYInv = Min (idxYInv, 15)
```

【0214】

変数 `varScale` は以下のように導出される：

```
varScale = ChromaScaleCoeff[idxYInv]
```

【0215】

現在のクロマブロックに変換を施すと、再構成されたクロマ画像サンプル配列 `recSamples` が以下のように導出される：

```
recSamples[i][j] = Clip1 (predSamples[i][j] + Sign (resSamples[i][j]) * (Abs (resSamples[i][j]) * varScale + (1 << 10)) >> 11)
```

現在のブロックに変換が適用されていない場合は、次のようになる：

```
recSamples[i][j] = Clip1 (predSamples[i][j])
```

【0216】

符号化器の検討

10

20

30

40

50

LMCS符号化器の基本原理は、まず、ダイナミックレンジのセグメントが平均分散よりも低いコードワードを持つレンジに、より多くのコードワードを割り当てるというものである。これの別形式として、LMCSの主なターゲットは、平均分散よりも高いコードワードを持つダイナミックレンジセグメントに、より少ないコードワードを割り当てることである。この方法では、画像の滑らかな領域は、平均よりも多くのコードワードで符号化され、その逆も同様である。

【0217】

APSに格納されるLMCSツールのすべてのパラメータ(表6参照)は、符号化器側で決定される。LMCS符号化器アルゴリズムは、局所的なルマ分散の評価に基づいており、上記の基本原理に従ってLMCSパラメータの決定を最適化している。そして、与えられたブロックの最終的な再構成サンプルについて、最良のPSNRメトリックを得るために最適化が行われる。

10

【0218】

スライスヘッダではなくピクチャヘッダで情報を通知する実施形態

実施形態において、ピクチャヘッダ内またはスライスヘッダ内でシグナリングすることができる情報のシグナリングは、ピクチャヘッダがスライスヘッダ内でシグナリングされるとき、ピクチャヘッダ内でシグナリングされ、スライスヘッダ内ではシグナリングされない。また、ピクチャヘッダ内またはスライスヘッダ内でシグナリング可能な情報のシグナリングをスライスヘッダ内でシグナリングする場合、ピクチャヘッダはスライスヘッダ内でシグナリングされないという等価な方法がある。別の等価な方法では、ピクチャヘッダ中またはスライス中にシグナリングすることができる情報のシグナリングがピクチャヘッダ内でシグナリングされるとき、ピクチャヘッダは、スライスヘッダ内でシグナリングされる。表12は、ツール名をXXXに置き換えた実施形態を示す表である。この表では、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合、スライスヘッダ内においてパラメータのシグナリングが許可されない。

20

【0219】

【表12】

表12 シグナリングの概要

	xxx_info_in_ph_flag = 1	xxx_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	ピクチャヘッダ内にXXX	適用無し
picture_header_in_slice_header_flag = 0	ピクチャヘッダ内にXXX	スライスヘッダ内にXXX、または存在しない

30

【0220】

一実施形態では、xxx_info_in_ph_flagのセマンティクスにおいて、以下の条件を追加する：

「PPSを参照するスライスヘッダがPHシンタックス構造を含む場合、xxx_info_in_ph_flagが1であることがビットストリーム適合性の要件である。」

40

および/または

「xxx_info_in_ph_flagが0であるとき、picture_header_in_slice_header_flagは0である。」

【0221】

ピクチャヘッダがスライスヘッダ内にある場合、これは現在のピクチャに対して1つのスライスのみが存在することを意味する。したがって、スライスとピクチャの両方について情報を送信または送信可能にしても、パラメータが同じになるため、符号化器または復号器の柔軟性が向上しない。つまり、ピクチャヘッダに情報がある場合、スライスヘッダの対応する情報は冗長となる。同様に、スライスヘッダに情報がある場合、ピクチャヘッ

50

ダ内の対応する情報は冗長となる。このように、ピクチャヘッダがスライスヘッダにある場合には、その情報はピクチャヘッダにあり、スライスヘッダにないように条件を強制することにより、シグナリングの冗長性を制限して、復号器の実施を簡略化することができる。

【0222】

QP (量子化パラメータ) デルタ

実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされる時、QPデルタシグナリングは、スライスヘッダ内で回避される。等価な方法では、QPデルタシグナリングが、スライスヘッダ内でシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされない。別の等価な方法では、ピクチャヘッダにおいてQPデルタがシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされる。つまり、上記のツールXXXは、QPデルタである。

10

【0223】

表13は、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合、QPデルタ情報のシグナリングが認可されない(すなわち、許可されない)ように、これが実施される方法を示す。

【0224】

【表13】

表13 QPデルタシグナリング

	<i>qp_delta_info_in_ph_flag</i> = 1	<i>qp_delta_info_in_ph_flag</i> = 0
<i>picture_header_in_slice_header_flag</i> = 1	ピクチャヘッダ内にQPデルタ	適用無し
<i>picture_header_in_slice_header_flag</i> = 0	ピクチャヘッダ内にQPデルタ	スライスヘッダ内にQPデルタ、または存在しない

20

【0225】

ピクチャヘッダがスライスヘッダ内にある場合(現在のピクチャに対して1つのスライスのみが存在することを意味する)、パラメータが同じであるため、スライスおよびピクチャヘッダにおいて伝達可能な情報は、符号化器または復号器の柔軟性を増大させない。したがって、復号器の実施の複雑さを低減するために、同じ符号化の可能性(この場合、QPデルタパラメータ(*qp_delta_info*))を符号化するために、スライスまたはピクチャヘッダのうちの1つの可能なシグナリングQPデルタ情報のみを有することがより好ましい。

30

【0226】

第1の実施形態の実施において、*qp_delta_info_in_ph_flag* のセマンティクスに以下の条件を追加してもよい:

「PPSを参照するスライスヘッダがPHシンタックス構造を含む場合、*qp_delta_info_in_ph_flag* が1であることがビットストリーム適合性の要件である。」

40

および/または

「*qp_delta_info_in_ph_flag* が0であるとき、*picture_header_in_slice_header_flag* は0である。」

【0227】

別の実施形態では、スライスヘッダ内のQPデルタパラメータの復号は、表14に描かれているように、フラグ*picture_header_in_slice_header_flag* の値が0に設定される場合にのみ許可される。シンタックスの変更は、下線で示されている。この表において、スライスヘッダの*slice_qp_delta* 情報は、QPデルタ情報がスライスレベルでシグナリングされ(*qp_delta_info*

50

`_in_ph_flag`が0)、ピクチャヘッダがスライスヘッダ内で伝送されない場合 (`picture_header_in_slice_header_flag`が0)に復号され得る。

【0228】

【表14】

表14 QPデルタの修正を示す部分スライスヘッダ

	Descriptor
<code>slice_header() {</code>	
<code>picture_header_in_slice_header_flag</code>	<code>u(1)</code>
<code>if(picture_header_in_slice_header_flag)</code>	
<code>picture_header_structure()</code>	
...	
<code>if(!qp_delta_info_in_ph_flag && !picture_header_in_slice_header_flag)</code>	
<code>slice_qp_delta</code>	<code>se(v)</code>
...	
<code>}</code>	

10

【0229】

実施形態において、ピクチャヘッダのQPデルタパラメータの復号は、表15に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、系統的に許可される。この表によれば、スライスヘッダのQPデルタ情報は、QPデルタ情報がピクチャヘッダでシグナリングされる場合 (`qp_delta_info_in_ph_flag`が1)、またはピクチャヘッダがスライスヘッダ内で送信される場合 (`picture_header_in_slice_header_flag`が1)のみ、復号することが可能である。

20

【0230】

【表15】

表15 QPデルタの修正を示す部分ピクチャヘッダ

	Descriptor
<code>picture_header_structure() {</code>	
...	
<code>if(qp_delta_info_in_ph_flag picture_header_in_slice_header_flag)</code>	
<code>ph_qp_delta</code>	<code>se(v)</code>
...	
<code>}</code>	

30

【0231】

参照ピクチャリスト (RPL)

一実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされるとき、参照ピクチャリストのシグナリングは、スライスヘッダ内で回避される。等価な方法では、ピクチャヘッダは、参照ピクチャリストのシグナリングが、スライスヘッダにおいてシグナリングされるとき、スライスヘッダにおいてシグナリングされない。別の等価な方法では、ピクチャヘッダにおいて参照ピクチャリストのシグナリングが行われるとき、ピクチャヘッダはスライスヘッダ内においてシグナリングされる。言い換えれば、上記のツールXXXはRPLである。

40

【0232】

表16は、スライスヘッダ内でピクチャヘッダをシグナリングする場合にRPLシグナリングを許可しない本実施形態の実施例を示している。

【0233】

50

【表 1 6】

表16 RPLシグナリングの概要

	rpl_info_in_ph_flag = 1	rpl_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	ピクチャヘッド内にRPL	適用無し
picture_header_in_slice_header_flag = 0	ピクチャヘッド内にRPL	スライスヘッド内にRPL、または存在しない

10

【0 2 3 4】

ピクチャヘッドがスライスヘッド内にある場合（現在のピクチャに対して1つのスライスのみが存在することを意味する）、スライスヘッドおよびピクチャヘッドにおいて情報が伝達可能であることは、パラメータが同じになるため、符号化器または復号器に対する柔軟性を増加させない。この実施形態によれば、同じ符号化の可能性（この場合は参照ピクチャリスト（RPL）情報（rpl_info））を符号化するために、スライスまたはピクチャヘッドのうちの1つで、情報の可能なシグナリングのみを有する方が良いので、復号器の実施の複雑さは低減される。

【0 2 3 5】

実施形態では、rpl_info_in_ph_flagのセマンティクスにおいて、以下の条件を追加する：

20

「PPSを参照するスライスヘッドがPHシンタックス構造を含む場合、rpl_info_in_ph_flagが1であることがビットストリーム適合性の要件である。」
および/または

「rpl_info_in_ph_flagが0であるとき、picture_header_in_slice_header_flagは0である。」

【0 2 3 6】

実施形態において、スライスヘッド内の参照ピクチャリストパラメータの復号は、表17に描かれているように、フラグpicture_header_in_slice_header_flagの値が0に設定される場合にのみ、許可される。この表において、参照ピクチャリスト情報がスライスレベルでシグナリングされ（rpl_info_in_ph_flagが0）、ピクチャヘッドがスライスヘッド内で伝送されない（picture_header_in_slice_header_flagが0）場合、スライスヘッドのref_pic_lists（）情報が復号され得る。

30

【0 2 3 7】

実施形態では、表17に描かれているように、ピクチャヘッドがスライスヘッド内にあるとき、時間パラメータslice_collocated_from_l0_flag及びslice_collocated_ref_idxは復号され得ない。

【0 2 3 8】

40

50

【表 1 7】

表17 RPLの修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(!rpl_info_in_ph_flag && !picture_header_in_slice_header_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))	
ref_pic_lists()	
if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && (slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
if(slice_type == B)	
slice_collocated_from_10_flag	u(1)
if((slice_collocated_from_10_flag && NumRefIdxActive[0] > 1) (!slice_collocated_from_10_flag && NumRefIdxActive[1] > 1))	
slice_collocated_ref_idx	ue(v)
}	
}	
}	
...	
}	

10

20

30

【0 2 3 9】

実施形態において、ピクチャヘッダのRPLパラメータの復号は、表18に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定される場合に、系統的に許可される。この表において、ピクチャヘッダのRPL情報は、RPL情報がピクチャレベルでシグナリングされる場合 (`rpl_info_in_ph_flag` が1)、またはピクチャヘッダがスライスヘッダで伝送される場合 (`picture_header_in_slice_header_flag` が1) のみ、復号され得る。

40

【0 2 4 0】

実施形態では、表18に描かれているように、ピクチャヘッダがスライスヘッダ内にあるとき、時間パラメータ `ph_collocated_from_10_flag` 及び `ph_collocated_ref_idx` は復号され得る。

【0 2 4 1】

50

【表 1 8】

表18 RPLの修正を示す部分ピクチャヘッダ

	Descriptor
picture_header_structure() {	
...	
if(rpl_info_in_ph_flag picture_header_in_slice_header_flag)	
ref_pic_lists()	
...	
if(ph_temporal_mvp_enabled_flag && (rpl_info_in_ph_flag picture_header_in_slice_header_flag)) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
...	
}	

10

【 0 2 4 2】

20

デブロッキングフィルタ (D B F)

実施形態において、デブロッキングフィルタパラメータのシグナリングは、ピクチャヘッダがスライスヘッダ内でシグナリングされるとき、スライスヘッダにおいて回避される。等価な方法では、デブロッキングフィルタパラメータのシグナリングが、スライスヘッダ内でシグナリングされるとき、ピクチャヘッダは、スライスヘッダ内でシグナリングされない。別の等価な方法では、ピクチャヘッダにおいてデブロッキングフィルタパラメータのシグナリングが行われるとき、ピクチャヘッダはスライスヘッダ内でシグナリングされる。すなわち、上述したツール X X X は、 D B F である。

【 0 2 4 3】

30

表 1 9 は、スライスヘッダ内でピクチャヘッダをシグナリングする場合に、 D B F シグナリングを許可しない本実施形態による実施を示す図である。

【 0 2 4 4】

【表 1 9】

表19 DBFシグナリングの概要

	dbf_info_in_ph_flag = 1	dbf_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	ピクチャヘッダ内にDBF	適用無し
picture_header_in_slice_header_flag = 0	ピクチャヘッダ内にDBF	ピクチャヘッダ内にDBF

40

【 0 2 4 5】

ピクチャヘッダがスライスヘッダ内にある場合、現在のピクチャに対して1つのスライスしか存在しないため、スライスまたはピクチャで伝送される情報は、パラメータが同じであるため、符号化器または復号器の柔軟性を増加させない。そこで、復号器の実施の複雑さを軽減するために、同じ符号化の可能性を符号化するために、 D B F 情報の可能な符号化は 1 つのみである方が良い。

【 0 2 4 6】

実施形態では、 d b f _ i n f o _ i n _ p h _ f l a g のセマンティクスに以下の条

50

件を追加する：

「 P P Sを参照するスライスヘッダが P Hシンタックス構造を含む場合、 d b f _ i n f o _ i n _ p h _ f l a g が 1 であることがビットストリーム適合性の要件である。」
および/または

「 d b f _ i n f o _ i n _ p h _ f l a g が 0 であるとき、 p i c t u r e _ h e a d e r _ i n _ s l i c e _ h e a d e r _ f l a g は 0 である。」

【 0 2 4 7 】

実施形態では、表 2 0 に描かれているように、フラグ p i c t u r e _ h e a d e r _ i n _ s l i c e _ h e a d e r _ f l a g の値が 0 に設定される場合にのみ、スライスヘッダの D B F パラメータが認可される。この表において、D B F 情報がスライスレベルでシグナリングされ (d b f _ i n f o _ i n _ p h _ f l a g が 0)、ピクチャヘッダがスライスヘッダで伝送されない (p i c t u r e _ h e a d e r _ i n _ s l i c e _ h e a d e r _ f l a g が 0) 場合、スライスヘッダの s l i c e _ d e b l o c k i n g _ f i l t e r _ o v e r r i d e _ f l a g フラグは復号され得る。

10

【 0 2 4 8 】

【 表 2 0 】

表20 DBFの修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag && !picture_header_in_slice_header_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
...	
}	
}	
}	
...	
}	

20

30

【 0 2 4 9 】

実施形態では、表 2 1 に描かれているように、フラグ p i c t u r e _ h e a d e r _ i n _ s l i c e _ h e a d e r _ f l a g の値が 1 に設定される場合、ピクチャヘッダの D B F パラメータは、系統的に認可される。この表では、D B F 情報がピクチャヘッダでシグナリングされる場合 (d b f _ i n f o _ i n _ p h _ f l a g が 1)、またはピクチャヘッダがスライスヘッダで伝送される場合 (p i c t u r e _ h e a d e r _ i n _ s l i c e _ h e a d e r _ f l a g が 1) のみスライスヘッダ内の D B F 情報が復号されることが可能である。

40

【 0 2 5 0 】

【表 2 1】

表21 DBFの修正を示す部分ピクチャヘッダ

picture_header_structure() {	Descriptor
...	
if(deblocking_filter_override_enabled_flag && (dbf_info_in_ph_flag picture_header_in_slice_header_flag)) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
...	
}	
}	
}	

10

【0 2 5 1】

S A O (サンプルアダプティブオフセット)

一実施形態において、S A Oシグナリングは、ピクチャヘッダがスライスヘッダ内でシグナリングされる時、スライスヘッダにおいて回避される。等価な方法では、S A Oシグナリングが、スライスヘッダ内でシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされない。別の等価な方法では、S A Oシグナリングがピクチャヘッダ内でシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされる。すなわち、上記のツールXXXは、S A Oである。

20

【0 2 5 2】

表 2 2 は、スライスヘッダ内でピクチャヘッダをシグナリングする場合に、S A Oシグナリングを許可しない本実施形態を示すものである。

【0 2 5 3】

【表 2 2】

表22 SAOシグナリングの概要

	sao_info_in_ph_flag = 1	sao_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	ピクチャヘッダ内にSAO	適用無し
picture_header_in_slice_header_flag = 0	ピクチャヘッダ内にSAO	スライスヘッダ内にSAO、または存在しない

30

【0 2 5 4】

ピクチャヘッダがスライスヘッダ内にある場合、現在のピクチャに対して1つのスライスしか存在しないので、スライスまたはピクチャで伝送される情報は、パラメータが同じであるため、符号化器または復号器に対する柔軟性を増大させない。そこで、復号器の実施の複雑さを軽減するために、同じ符号化の可能性を符号化するために、S A O情報の可能な符号化を1つのみ持つことが良い。

40

【0 2 5 5】

実施形態では、s a o _ i n f o _ i n _ p h _ f l a g のセマンティクスにおいて、以下の条件を追加する：

「 P P S を参照するスライスヘッダが P H シンタックス構造を含む場合、s a o _ i n f o _ i n _ p h _ f l a g が 1 であることがビットストリーム適合性の要件である。」

および/または

50

「sao_info_in_ph_flagが0であるとき、picture_header_in_slice_header_flagは0である。」

【0256】

実施形態において、スライスヘッダのSAOパラメータの復号は、以下に図示するように、フラグpicture_header_in_slice_header_flagの値が0に設定される場合にのみ、許可される。

【0257】

表23。この表において、スライスレベルでSAO情報がシグナリングされ(sao_info_in_ph_flagが0)、かつスライスヘッダ内でピクチャヘッダが伝送されない場合(picture_header_in_slice_header_flagが0)、スライスヘッダのslice_sao_luma_flagが復号され得る。

10

【0258】

【表23】

表23 SAOの修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(sps_sao_enabled_flag && !sao_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
...	
}	

20

30

【0259】

実施形態において、ピクチャヘッダのSAOパラメータの復号は、表24に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定される場合に、系統的に認可される。この表では、SAO情報がピクチャヘッダでシグナリングされる場合(sao_info_in_ph_flagが1)、またはピクチャヘッダがスライスヘッダで伝送される場合(picture_header_in_slice_header_flagが1)のみ、スライスヘッダのSAO情報が復号され得る。

【0260】

40

50

【表 2 4】

表24 SAOの修正を示す部分ピクチャヘッダ

picture_header_structure() {	Descriptor
...	
if(sps_sao_enabled_flag && (sao_info_in_ph_flag picture_header_in_slice_header_flag)) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
....	
}	

10

【0 2 6 1】

加重平均予測 (W P)

一実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされる時、重み付き予測のシグナリングは、スライスヘッダで回避される。等価な方法では、重み付けされた予測のシグナリングが、スライスヘッダ内でシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされない。別の等価な方法では、重み付き予測のシグナリングがピクチャヘッダ内でシグナリングされる時、ピクチャヘッダは、スライスヘッダ内でシグナリングされる。すなわち、上述したツール X X X は、 W P である。

20

【0 2 6 2】

表 2 5 に示すように、 W P パラメータは、現在のシンタックスで `wp_info_in_ph_flag` を 0 に設定すると、スライスヘッダでシグナリングすることができる。

【0 2 6 3】

表 2 5 は、スライスヘッダ内でピクチャヘッダをシグナリングする場合に W P シグナリングを許可しない本実施形態の実施例を示している。

【0 2 6 4】

【表 2 5】

表25 WPシグナリングの概要

	<code>wp_info_in_ph_flag = 1</code>	<code>wp_info_in_ph_flag = 0</code>
<code>picture_header_in_slice_header_flag = 1</code>	ピクチャヘッダ内にWP	適用無し
<code>picture_header_in_slice_header_flag = 0</code>	ピクチャヘッダ内にWP	スライスヘッダ内にWP、または存在しない

30

40

【0 2 6 5】

ピクチャヘッダがスライスヘッダ内にある場合、現在のピクチャに対して1つのスライスしか存在しないので、スライスまたはピクチャで伝送される情報は、パラメータが同じであるため、符号化器または復号器の柔軟性を増大させない。そこで、復号器の実施の複雑さを軽減するために、同じ符号化の可能性を符号化するために、 W P 情報の可能な符号化を1つのみ持つことが良い。

【0 2 6 6】

実施形態では、 `wp_info_in_ph_flag` のセマンティクスにおいて、以下の条件を追加する：

「 P P S を参照するスライスヘッダが P H シンタックス構造を含む場合、 `wp_info`

50

o_in_ph_flagが1であることがビットストリーム適合性の要件である。」
および/または

「wp_info_in_ph_flagが0である場合、picture_header_in_slice_header_flagは0である。」

【0267】

実施形態において、スライスヘッダのWPパラメータの復号は、表26に描かれているように、フラグpicture_header_in_slice_header_flagの値が0に設定される場合にのみ、許可される。この表では、WP情報がスライスレベルでシグナリングされ(wp_info_in_ph_flagが0)、ピクチャヘッダがスライスヘッダ内で伝送されない(picture_header_in_slice_header_flagが0)場合、重み付け予測パラメータを含むpred_weight_table()関数が復号され得る。

10

【0268】

【表26】

表26 WPの修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(!wp_info_in_ph_flag && !picture_header_in_slice_header_flag && ((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)))	
pred_weight_table()	
...	
}	

20

【0269】

実施形態において、ピクチャヘッダのWPパラメータの複合は、表27に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定される場合に、系統的に許可される。この表において、WP情報は、WP情報がピクチャヘッダ内でシグナリングされる場合(wp_info_in_ph_flagが1)、またはピクチャヘッダがスライスヘッダ内で伝送される場合(picture_header_in_slice_header_flagが1)のみ、復号され得る。

30

【0270】

【表27】

表27 WPの修正を示す部分ピクチャヘッダ

picture_header_structure() {	Descriptor
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && (wp_info_in_ph_flag picture_header_in_slice_header_flag))	
pred_weight_table()	u(1)
....	
}	

40

【0271】

A L F

50

一実施形態において、ALFのシグナリングは、ピクチャヘッダがスライスヘッダ内でシグナリングされるとき、スライスヘッダにおいて回避される。等価な方法では、ALFのシグナリングが、スライスヘッダ内でシグナリングされるとき、ピクチャヘッダは、スライスヘッダ内でシグナリングされない。別の等価な方法では、ALFのシグナリングがピクチャヘッダ内でシグナリングされるとき、ピクチャヘッダは、スライスヘッダ内でシグナリングされる。すなわち、上記のツールXXXは、ALFである。

【0272】

実際、現在、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合 (`picture_header_in_slice_header_flag` が 1)、およびALFがスライスヘッダ内でシグナリングされる場合 (`alf_info_in_ph_flag` が 0)、ALFのAPS_IDを得る前にピクチャヘッダのすべてのパラメータがパース(解析)されるべきである。その結果、PPS、SPS、ピクチャヘッダなどのすべての変数がALFのAPS_IDをパースするためにメモリ内に保持されなければならないので、いくつかのストリーミングアプリケーションのパースの複雑さを増加させる。

10

【0273】

さらに、ピクチャヘッダがスライスヘッダ内にある場合、現在のピクチャに対して1つのスライスしか存在しないため、スライスまたはピクチャで伝送される情報はパラメータが同じであるため、符号化器または復号器の柔軟性を増加させない。そこで、復号器の実施の複雑さを軽減するために、同じ符号化の可能性のあるシグナリングは1つのみにしておく方がよい。表28は、スライスヘッダでピクチャヘッダをシグナリングする場合に、ALFシグナリングを許可しない本実施形態を示すものである。

20

【0274】

【表28】

表28 ALFシグナリングの概要

	<code>alf_info_in_ph_flag = 1</code>	<code>alf_info_in_ph_flag = 0</code>
<code>picture_header_in_slice_header_flag = 1</code>	ピクチャヘッダ内にALF	適用無し
<code>picture_header_in_slice_header_flag = 0</code>	ピクチャヘッダ内にALF	スライスヘッダ内にALF、または存在しない

30

【0275】

実施形態では、`alf_info_in_ph_flag` のセマンティクスにおいて、以下の条件を追加する：

「PPSを参照するスライスヘッダがPHシンタックス構造を含む場合、`alf_info_in_ph_flag` が1であることがビットストリーム適合性の要件である。」

および/または

「`alf_info_in_ph_flag` が0であるとき、`picture_header_in_slice_header_flag` は0である。」

【0276】

実施形態において、スライスヘッダのALFパラメータの復号は、表29に描かれているように、フラグ`picture_header_in_slice_header_flag`の値が0に設定される場合にのみ、許可される。この表では、ALFがSPSレベルで有効であり(`sps_alf_enabled_flag`が1)、ALF情報がスライスレベルでシグナリングされ(`alf_info_in_ph_flag`が0)、ピクチャヘッダがスライスヘッダで送信されない(`picture_header_in_slice_header_flag`が0)場合のみ、スライスヘッダのALF情報が復号され得る。

40

【0277】

50

【表 2 9】

表29 ALFの修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(sps_alf_enabled_flag && !alf_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
slice_cc_alf_cb_enabled_flag	u(1)
if(slice_cc_alf_cb_enabled_flag)	
slice_cc_alf_cb_aps_id	u(3)
slice_cc_alf_cr_enabled_flag	u(1)
if(slice_cc_alf_cr_enabled_flag)	
slice_cc_alf_cr_aps_id	u(3)
}	
}	
}	
}	
}	
...	
}	

10

20

30

【0 2 7 8】

実施形態において、ピクチャヘッダのALFパラメータの復号は、表30に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定される場合に、系統的に許可される。この表において、スライスヘッダのALF情報は、ALFがSPSレベルで有効である場合 (`sps_alf_enabled_flag` が1)、及びALF情報がピクチャレベルでシグナリングされる場合 (`alf_info_in_ph_flag` が1)、又はスライスヘッダ内でピクチャヘッダが伝送される場合 (`picture_header_in_slice_header_flag` が1) のみ、復号され得る。

40

【0 2 7 9】

50

【表 3 0】

表30 ALFの修正を示す部分ピクチャヘッダ

picture_header_structure() {	Descriptor
...	
if(sps_alf_enabled_flag && (alf_info_in_ph_flag picture_header_in_slice_header_flag)) {	
ph_alf_enabled_flag	u(1)
if(ph_alf_enabled_flag) {	
ph_num_alf_aps_ids_luma	u(3)
for(i = 0; i < ph_num_alf_aps_ids_luma; i++)	
ph_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
ph_alf_chroma_idc	u(2)
if(ph_alf_chroma_idc > 0)	
ph_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
ph_cc_alf_cb_enabled_flag	u(1)
if(ph_cc_alf_cb_enabled_flag)	
ph_cc_alf_cb_aps_id	u(3)
ph_cc_alf_cr_enabled_flag	u(1)
if(ph_cc_alf_cr_enabled_flag)	
ph_cc_alf_cr_aps_id	u(3)
}	
}	
}	

10

20

【 0 2 8 0】

すべてのツール/パラメータ

一実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合、ピクチャヘッダ内またはスライスヘッダ内でシグナリングされ得るすべてのツール（および/またはパラメータ）は、ピクチャヘッダ内でシグナリングされるように制限される。上記の説明で言及したように、実施形態では、関連するツールは以下の通りである。QPデルタ情報、参照ピクチャリスト、デブロッキングフィルタ、SAO重み付け予測、ALFである。しかしながら、他のツールも可能であり、それらは、スライス及びピクチャヘッダの両方においてシグナリングされ得る。

30

【 0 2 8 1】

これは、以下の制約を加えることで表現できる：

「フラグ rpl_info_in_ph_flag, dbf_info_in_ph_flag, sao_info_in_ph_flag, alf_info_in_ph_flag, wp_info_in_ph_flag, qp_delta_info_in_ph_flag の少なくとも1つが0に設定されると、picture_header_in_slice_header_flag の値は0とする。」

40

および/または、以下を加える：

「picture_header_in_slice_header_flag が1であるとき、フラグ rpl_info_in_ph_flag, dbf_info_in_ph_flag, sao_info_in_ph_flag, wp_info_in_ph_flag, qp_delta_info_in_ph_flag は1とする。」

【 0 2 8 2】

また、各 XXX_info_in_ph_flag に以下の制約を追加する：

「PPSを参照するスライスヘッダがPHシンタックス構造を含む場合、XXX_in

50

f o _ i n _ p h _ f l a g が 1 であることがビットストリーム適合性の要件である。」
【 0 2 8 3 】

これらのパラメータがすべて同じヘッダでシグナリングされる場合、同じ符号化の可能性を符号化するためのシグナリングが 1 つしかない方が良いので、復号器の実施の複雑さを軽減することができる。

【 0 2 8 4 】

シグナリング順序の実施形態

A L F に関連するこれまでのものの 1 つの代替実施形態では、表 3 1 に描かれているように、スライスヘッダの A L F の A P S _ I D に関連する情報は、ピクチャヘッダの構造より前に設定される。この実施形態では、スライスヘッダ内で A L F がシグナリングされたとき、およびスライスヘッダで内ピクチャヘッダがシグナリングされたときに、ピクチャヘッダのすべてのパラメータをパースすることなく A P S _ I D を迅速に取得することができる。

10

【 0 2 8 5 】

【表 3 1】

表31 複数の修正を示す部分スライスヘッダ

slice_header() {	Descriptor
if(sps_alf_enabled_flag && !alf_info_in_ph_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
slice_cc_alf_cb_enabled_flag	u(1)
if(slice_cc_alf_cb_enabled_flag)	
slice_cc_alf_cb_aps_id	u(3)
slice_cc_alf_cr_enabled_flag	u(1)
if(slice_cc_alf_cr_enabled_flag)	
slice_cc_alf_cr_aps_id	u(3)
}	
}	
}	
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
....	
}	

20

30

40

【 0 2 8 6 】

重み数の実施形態

ピクチャヘッダがスライスヘッダ内にある場合の実施形態では、シンタックス要素の表 3 2 の部分に描かれているように、各リスト L 0 L 1 に対する重み付け予測の重みの数が復号される。従って、重み数のシグナリングは、ピクチャヘッダに限定され得る。

【 0 2 8 7 】

50

【表 3 2】

表32 重み付け予測パラメータの部分シンタックステーブル

pred_weight_table() {	Descriptor
...	
if(wp_info_in_ph_flag picture_header_in_slice_header_flag)	
num_10_weights	ue(v)
...	
if(pps_weighted_bipred_flag && (wp_info_in_ph_flag picture_header_in_slice_header_flag))	
num_11_weights	ue(v)
...	
}	

10

【0 2 8 8】

スライスヘッダ内で情報を通知し、ピクチャヘッダ内では通知しない実施形態の概要
 一実施形態では、ピクチャヘッダ内またはスライスヘッダ内にシグナリングすることができる情報のシグナリングは、ピクチャヘッダがスライスヘッダ内でシグナリングされる
 とき、スライスヘッダにシグナリングされ、ピクチャヘッダにシグナリングされない。また、
 ピクチャヘッダ内またはスライスヘッダ内にシグナリング可能な情報のシグナリングを、
 ピクチャヘッダ内にシグナリングする場合には、スライスヘッダ内にシグナリングし
 ないようにする。別の等価な方法では、ピクチャヘッダ内またはスライスヘッダ内にシグ
 ナリングすることができる情報のシグナリングがスライスヘッダ内にシグナリングされる
 とき、ピクチャヘッダは、スライスヘッダ内にシグナリングされる。表 3 3 は、ツール（
 またはパラメータ）名を X X X に置き換えた本実施の形態を示す。この表では、ピクチャ
 ヘッダがスライスヘッダ内でシグナリングされる場合、ピクチャヘッダでのパラメータの
 シグナリングは許可されていない。

20

【0 2 8 9】

【表 3 3】

表33 ツールシグナリングの概要

	xxx_info_in_ph_flag = 1	xxx_info_in_ph_flag = 0
picture_header_in_slice_header_flag = 1	適用無し	スライスヘッダ内にXXX、または存在しない
picture_header_in_slice_header_flag = 0	ピクチャヘッダ内にXXX	スライスヘッダ内にXXX、または存在しない

30

【0 2 9 0】

一実施形態では、xxx_info_in_ph_flag のセマンティクスにおいて、以下の条件を追加する：
 「PPSを参照するスライスヘッダがPHシンタックスを含む場合、XXX_info_in_ph_flag が0であることがビットストリーム適合性の要件である。」

40

および/または

「XXX_info_in_ph_flag が1であるとき、picture_header_in_slice_header_flag は0である。」

【0 2 9 1】

ピクチャヘッダがスライスヘッダ内にある場合、これは現在のピクチャに対して1つのスライスのみが存在することを意味する。したがって、スライスとピクチャの両方について情報を送信または送信可能にしても、パラメータが同じになるため、符号化器または復

50

号器の柔軟性が向上しない。つまり、ピクチャヘッダに情報がある場合、スライスヘッダの対応する情報は冗長となる。同様に、情報がスライスヘッダ内にある場合、ピクチャヘッダ内の対応する情報は冗長となる。本明細書で説明する実施形態は、同じ符号化のシグナリングにおける冗長性を制限することによって、復号器の実施を簡素化する。特に、実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合、情報がスライスヘッダ内にあることは許容されるが、ピクチャヘッダ内にあることは許容されない。

【0292】

QPデルタ

一実施形態では、ツールXXXは、QPデルタである。一実施形態では、表33に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、スライスヘッダのQPデルタパラメータの復号は許可される。また、表33に示すように、`picture_header_in_slice_header_flag` は、QPデルタパラメータがピクチャヘッダ内でシグナリングされる場合、0に等しい値に設定される。別の等価な方法では、表33に描かれているように、QPデルタパラメータがスライスヘッダ内でシグナリングされるとき、`picture_header_in_slice_header_flag` は1に設定される。この表において、QPデルタ情報がスライスレベルでシグナリングされる場合 (`qp_delta_info_in_ph_flag` が0)、またはピクチャヘッダがスライスヘッダにおいて伝送される場合 (`picture_header_in_slice_header_flag` が1)、スライスヘッダの `slice_qp_delta` 情報は、復号され得る。

【0293】

これは、セマンティクスに以下を追加することで得られる：

「PPSを参照するスライスヘッダがPHシンタックスを含む場合、`qp_delta_info_in_ph_flag` は0であるというのがビットストリーム適合性の要件である。」

および/または

「`qp_delta_info_in_ph_flag` が1であるとき、`picture_header_in_slice_header_flag` は0である。」

【0294】

実施形態では、表35に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、ピクチャヘッダのQPデルタパラメータの復号が系統的に回避される。この表において、ピクチャヘッダのQPデルタ情報は、QPデルタ情報がピクチャヘッダにおいてシグナリングされる場合 (`qp_delta_info_in_ph_flag` が1)、及び `picture_header_in_slice_header_flag` が0に設定される場合のみ復号され得る。

【0295】

RPL (参照ピクチャリスト)

一実施形態では、ツールXXXは、参照ピクチャリストである。一実施形態では、表33に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定される場合にのみ、スライスヘッダにおける参照ピクチャリストのパラメータの復号が認可される。また、`picture_header_in_slice_header_flag` は、表33に示すように、ピクチャヘッダ内で参照ピクチャのリストパラメータがシグナリングされるとき、0に設定される。また、`picture_header_in_slice_header_flag` は、表33に示すように、スライスヘッダ内で参照ピクチャのパラメータを通知する場合に1に設定される。この表において、参照ピクチャリスト情報がスライスレベルでシグナリングされ (`rpl_info_in_ph_flag` が0)、ピクチャヘッダがスライスヘッダで伝送される場合 (`picture_header_in_slice_header_f`

lagが1)、スライスヘッダのref_pic_lists()情報は復号され得る。

【0296】

実施形態では、表34に描かれているように、ピクチャヘッダがスライスヘッダ内にあるとき、時間パラメータslice_collocated_from_l0_flag及びslice_collocated_ref_idxは復号され得る。

【0297】

実施形態では、表35に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定されるとき、ピクチャヘッダのRPLパラメータの復号は系統的に回避される。この表では、RPL情報がピクチャレベルでシグナリングされ(rpl_info_in_ph_flagが1)、picture_header_in_slice_header_flagが0に設定される場合のみ、ピクチャヘッダのRPL情報を復号することが可能である。

10

【0298】

実施形態では、表35に描かれているように、ピクチャヘッダがスライスヘッダ内にある場合、時間パラメータph_collocated_from_l0_flag及びph_collocated_ref_idxは復号できない。

【0299】

デブロッキングフィルタ(DBF)

一実施形態では、ツールXXXは、デブロッキングフィルタ(DBF)である。代替または追加の一実施形態では、表33に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定されるとき、スライスヘッダのDBFパラメータが認可される。また、表33に示すように、picture_header_in_slice_header_flagを0に設定すると、DBFパラメータがピクチャヘッダ内でシグナリングされる。また、表33に示すように、DBFパラメータがスライスヘッダ内に格納されている場合、picture_header_in_slice_header_flagは1に設定される。この表では、DBF情報がスライスレベルでシグナリングされる場合(dbf_info_in_ph_flagが0)、またはピクチャヘッダがスライスヘッダ内で伝送される場合(picture_header_in_slice_header_flagが1)、スライスヘッダのslice_deblocking_filter_override_flagフラグが復号され得る。

20

30

【0300】

実施形態では、表35に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定されるとき、ピクチャヘッダのDBFパラメータは系統的に回避される。この表では、ピクチャヘッダのDBF情報は、DBF情報がピクチャヘッダ内でシグナリングされ(dbf_info_in_ph_flagが1)、picture_header_in_slice_header_flagが0である場合にのみ復号され得る。

【0301】

サンプルアダプティブオフセット(SAO)

一実施形態では、ツールXXXは、SAO(サンプルアダプティブオフセット)である。ある代替または追加の実施形態において、スライスヘッダのSAOパラメータは、表33に描かれているように、フラグpicture_header_in_slice_header_flagの値が1に設定される場合にのみ、認可される。また、表33に示すように、picture_header_in_slice_header_flagは、SAOパラメータがピクチャヘッダにシグナリングされるとき、0に設定される。別の等価な方法では、picture_header_in_slice_header_flagは、表33に描かれるようにSAOパラメータがスライスヘッダにシグナリングされるとき、1に設定される。この表において、SAO情報がスライスレベルでシグナリングされ(sa_o_info_in_ph_flagが0)、ピクチャヘッダがスライス

40

50

ヘッダにおいて伝送される場合 (`picture_header_in_slice_header_flag` が 1)、スライスヘッダの `slice_sao_luma_flag` は復号され得る。

【0302】

実施形態では、表35に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、ピクチャヘッダのSAOパラメータは系統的に回避される。この表において、ピクチャヘッダのSAO情報は、SAO情報がピクチャヘッダ内でシグナリングされ (`sao_info_in_ph_flag` が 1)、`picture_header_in_slice_header_flag` が 0 に設定される場合にのみ復号され得る。

10

【0303】

WP (重み付き予測)

一実施形態では、ツールXXXは、WP (重み付き予測) である。一実施形態において、スライスヘッダ内のWPパラメータは、表33に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定される場合にのみ、認可される。また、表33に示すように、`picture_header_in_slice_header_flag` は、ピクチャヘッダ内でWPパラメータがシグナリングされるとき、0に設定される。また、表33に示すように、スライスヘッダ内でWPパラメータを通知する場合、`picture_header_in_slice_header_flag` は1に設定される。この表では、WP情報がスライスレベルでシグナリングされる場合 (`wp_info_in_ph_flag` が 0)、またはピクチャヘッダがスライスヘッダで伝送される場合 (`picture_header_in_slice_header_flag` が 1)、重み付き予測パラメータを含む `pred_weight_table ()` 関数が復号され得る。

20

【0304】

追加の一実施形態では、表35に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、ピクチャヘッダ内のWPパラメータの復号は系統的に回避される。この表では、WP情報がピクチャヘッダ内でシグナリングされる場合 (`wp_info_in_ph_flag` が 1)、および `picture_header_in_slice_header_flag` が 0 に設定される場合にのみ、WP情報が復号され得る。

30

【0305】

ALF (アダプティブループフィルタ)

一実施形態では、ツールXXXは、ALF (アダプティブループフィルタ) である。一実施形態において、スライスヘッダのALFパラメータの復号は、表33に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定される場合にのみ、許可される。また、`picture_header_in_slice_header_flag` は、ALFパラメータがピクチャヘッダ内でシグナリングされる場合、0に設定される (表33)。別の等価な方法では、`picture_header_in_slice_header_flag` は1に設定される。この表では、ALFがSPSレベルで有効である場合 (`sps_alf_enabled_flag` が 1)、ALF情報がスライスレベルでシグナリングされる場合 (`alf_info_in_ph_flag` が 0)、またはスライスヘッダ内でピクチャヘッダが伝送される場合 (`picture_header_in_slice_header_flag` が 1) のみ、スライスヘッダのALF情報が解読され得る。

40

【0306】

実施形態では、表35に描かれているように、フラグ `picture_header_in_slice_header_flag` の値が1に設定されるとき、ピクチャヘッダのALFパラメータの復号は、系統的に回避される。この表において、スライスヘッダのALF情報は、ALFがSPSレベルで有効であり (`sps_alf_enabled_`

50

flag が 1)、ALF 情報がピクチャレベルでシグナリングされ (alf_info_in_ph_flag が 1)、picture_header_in_slice_header_flag は 0 に設定されていれば、復号され得るのみである。

【0307】

【表34】

表34 複数の修正を示す部分スライスヘッダ

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
...	
if(sps_alf_enabled_flag && (!alf_info_in_ph_flag picture_header_in_slice_header_flag)) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
....	
}	
}	
...	
if((!rpl_info_in_ph_flag picture_header_in_slice_header_flag) && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))	
ref_pic_lists()	

10

20

30

40

50

if((rpl_info_in_ph_flag && ! picture_header_in_slice_header_flag) ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) && (slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {		
num_ref_idx_active_override_flag	u(1)	
if(num_ref_idx_active_override_flag)		
for(i = 0; i < (slice_type == B ? 2 : 1); i++)		
if(num_ref_entries[i][RplIdx[i]] > 1)		
num_ref_idx_active_minus1[i]	ue(v)	10
}		
...		
if(ph_temporal_mvp_enabled_flag && (! rpl_info_in_ph_flag picture_header_in_slice_header_flag)) {		
if(slice_type == B)		
slice_collocated_from_l0_flag	u(1)	
if((slice_collocated_from_l0_flag && NumRefIdxActive[0] > 1) (! slice_collocated_from_l0_flag && NumRefIdxActive[1] > 1))		
slice_collocated_ref_idx	ue(v)	
}		
if((! wp_info_in_ph_flag picture_header_in_slice_header_flag) && ((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)))		20
pred_weight_table()		
}		
if((! qp_delta_info_in_ph_flag picture_header_in_slice_header_flag))		
slice_qp_delta	se(v)	
...		
if(sps_sao_enabled_flag && (! sao_info_in_ph_flag picture_header_in_slice_header_flag)) {		
slice_sao_luma_flag	u(1)	30
if(ChromaArrayType != 0)		
slice_sao_chroma_flag	u(1)	
}		
if(deblocking_filter_override_enabled_flag && (! dbf_info_in_ph_flag picture_header_in_slice_header_flag))		
slice_deblocking_filter_override_flag	u(1)	
...		
}		

【 0 3 0 8 】

40

50

【表 3 5】

表35 複数の修正を示す部分ピクチャヘッダ

picture_header_structure() {	Descriptor
...	
if(sps_alf_enabled_flag && (alf_info_in_ph_flag && !picture_header_in_slice_header_flag)) {	
ph_alf_enabled_flag	u(1)
if(ph_alf_enabled_flag) {	
...	
}	
}	
...	
if(rpl_info_in_ph_flag && !picture_header_in_slice_header_flag)	
ref_pic_lists()	
...	
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag && !picture_header_in_slice_header_flag)	
pred_weight_table()	
}	
if(qp_delta_info_in_ph_flag && !picture_header_in_slice_header_flag)	
ph_qp_delta	se(v)
...	
if(sps_sao_enabled_flag && sao_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
...	
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag && !picture_header_in_slice_header_flag) {	
ph_deblocking_filter_override_flag	u(1)
...	
}	
...	
}	

10

20

30

40

【 0 3 0 9】

すべてのツール / パラメータ

実施形態では、ピクチャヘッダがスライスヘッダ内でシグナリングされる場合、ピクチャヘッダ内またはスライスヘッダ内でシグナリングされ得るすべてのツール（および/またはパラメータ）が、スライスヘッダにシグナリングされる。実施形態における上記説明で言及したように、関連するツールは以下の通りである。QPデルタ情報、参照ピクチャリスト、デブロッキングフィルタ、SAO重み付け予測、及びALF。しかし、他のツールは、それらがスライス及びピクチャヘッダの両方でシグナリングされることが可能である場合、可能であり得る。

50

【0310】

これは、以下の制約を加えることで表現できる：

「フラグ `rpl_info_in_ph_flag`, `dbf_info_in_ph_flag`, `sao_info_in_ph_flag`, `alf_info_in_ph_flag`, `wp_info_in_ph_flag`, `qp_delta_info_in_ph_flag` の少なくとも1つが1となる時、`picture_header_in_slice_header_flag` は0となるものとする。」

および/または、以下を加える：

「`picture_header_in_slice_header_flag` が1である時、フラグ `rpl_info_in_ph_flag`, `dbf_info_in_ph_flag`, `sao_info_in_ph_flag`, `wp_info_in_ph_flag`, `qp_delta_info_in_ph_flag` は0とする。」

10

【0311】

また、各 `XXX_info_in_ph_flag` に以下の制約を追加する：

「PPSを参照するスライスヘッダがPHシンタックスを含む場合、`XXX_info_in_ph_flag` が0であることがビットストリーム適合性の要件である。」

【0312】

これらのパラメータがすべて同じヘッダ内でシグナリングされる場合、ツールまたはパラメータが必然的に同じ値または特性を持つ場合、シグナリングが1つのみ可能である方がよいので、復号器の実施の複雑さが軽減される。

20

【0313】

実施

図10は、本発明の実施形態による符号化器150または復号器100の少なくとも1つと通信ネットワーク199とを含むシステム191、195を示す図である。実施形態によれば、システム195は、例えば、復号器100を構成するユーザ端末または復号器100と通信可能なユーザ端末のユーザインタフェースを介して、復号器100にアクセスするユーザに対してコンテンツ（例えば、表示/出力またはストリーミング用のビデオおよび音声コンテンツ）を処理し提供するためのシステムである。このようなユーザ端末は、コンピュータ、携帯電話、タブレット、または、ユーザに（提供される/ストリーム配信される）コンテンツを提供/表示することができる他のタイプの装置であってもよい。システム195は、通信ネットワーク199を介してビットストリーム101を（連続ストリームまたは信号の形態で（例えば、先のビデオ/オーディオが表示/出力されている間））取得/受信する。実施形態によれば、システム191は、コンテンツを処理し、処理されたコンテンツ、例えば、後の時間に表示/出力/ストリーミングするために処理されたビデオ及びオーディオコンテンツを記憶するためのものである。システム191は、符号化器150によって受信され処理（本発明によるデブロッキングフィルタによるフィルタリングを含む）されたオリジナル画像シーケンス151を含むコンテンツを取得/受信し、符号化器150は、通信ネットワーク199を介して復号器100に伝達されることになるビットストリーム101を発生させる。そして、ビットストリーム101は、いくつかの方法で復号器100に通信されるが、例えば、符号化器150によって予め生成され、ユーザがストレージ装置からコンテンツ（すなわち、ビットストリームデータ）を要求するまで、通信ネットワーク199内のストレージ装置（例えば、サーバやクラウドストレージ）にデータとして格納され、この時点でストレージ装置から復号器100にデータが通信/ストリーム配信されてもよい。また、システム191は、ユーザに対して（例えば、ユーザ端末に表示されるユーザインタフェースのデータを通信することによって）、ストレージ装置に格納されたコンテンツのコンテンツ情報（例えば、コンテンツのタイトル、およびコンテンツを識別、選択および要求するための他のメタ/格納場所データ）を提供/ストリーム化し、要求されたコンテンツがストレージ装置からユーザ端末に配信/ストリーム化できるようにコンテンツのユーザ要求を受信して処理するコンテンツ提供装置を含み得る。あるいは、符号化器150は、ユーザがコンテンツを要求したとき

30

40

50

に、ビットストリーム101を生成し、復号器100に直接通信/ストリーミングする。その後、復号器100は、ビットストリーム101(または信号)を受信し、本発明によるデブロッキングフィルタによるフィルタリングを行い、ビデオ信号109および/またはオーディオ信号を得る/生成し、これをユーザ端末が使用して、要求されたコンテンツをユーザに提供する。

【0314】

本発明による方法/プロセスの任意のステップまたは本明細書に記載された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組み合わせで実施されてもよい。ソフトウェアで実施される場合、ステップ/機能は、1つ以上の命令またはコードまたはプログラム、またはコンピュータ可読媒体として、上に格納されるかまたは上に伝送されてもよく、PC(パーソナルコンピュータ)であってもよいプログラマブル計算機などの1つ以上のハードウェアベースの処理装置によって実行されてもよい。DSP(デジタル信号プロセッサ)、回路、プロセッサおよびメモリ、汎用マイクロプロセッサまたは中央処理装置、マイクロコントローラ、ASIC(特定用途向け集積回路)、フィールドプログラマブルロジックアレイ(FPGA)、または他の同等の集積またはディスクリートロジック回路であってもよい。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造のいずれか、または本明細書で説明する技術の実施に適した他の構造を指す場合がある。

【0315】

本発明の実施形態は、無線ハンドセット、集積回路(IC)またはICセット(例えばチップセット)を含む多種多様なデバイスまたは装置によって実現することも可能である。様々なコンポーネント、モジュール、またはユニットは、それらの実施形態を実行するように構成されたデバイス/装置の機能的側面を説明するために本明細書に記載されているが、必ずしも異なるハードウェアユニットによる実現が必要なわけではない。むしろ、様々なモジュール/ユニットをコーデックハードウェアユニットに組み合わせたり、適切なソフトウェア/ファームウェアと連携した1つ以上のプロセッサを含む相互運用可能なハードウェアユニットの集合体によって提供したりすることができる。

【0316】

本発明の実施形態は、記憶媒体に記録されたコンピュータ実行可能命令(例えば、1つ以上のプログラム)を読み出して実行し上述した実施形態の1つ以上のモジュール/ユニット/機能を実行する、および/または、上述した実施形態の1つ以上の機能を実行するための1つ以上の処理ユニット又は回路を含む、システムまたは装置のコンピュータによって実現することができ、例えば、記憶媒体からコンピュータ実行可能命令を読み出して実行し、上述の実施形態の1つ以上の機能を実行すること、および/または、上述の実施形態の1つ以上の機能を実行するために1つ以上の処理ユニット又は回路を制御するシステム又は装置のコンピュータによって実行される方法によって実現され得る。コンピュータは、コンピュータ実行可能命令を読み出して実行するための別個のコンピュータ又は別個の処理ユニットのネットワークを含んでもよい。コンピュータ実行可能命令は、例えば、ネットワークを介した通信媒体や有形記憶媒体などのコンピュータ読取可能媒体からコンピュータに提供されてもよい。通信媒体は、信号/ビットストリーム/搬送波であってもよい。有形記憶媒体は、「非一時的コンピュータ可読記憶媒体」であり、例えば、ハードディスク、ランダムアクセスメモリ(RAM)、リードオンリーメモリ(ROM)、分散コンピューティングシステムのストレージ、光ディスク(コンパクトディスク(CD)、デジタル多用途ディスク(DVD)、ブルーレイディスク(BD)(商標)など)、フラッシュメモリ装置、メモリカードなどの1または複数が含まれてもよい。また、ステップ/機能の少なくとも一部は、FPGA(フィールドプログラマブルゲートアレイ)やASIC(特定用途向け集積回路)などのマシンまたは専用部品によってハードウェアで実施されてもよい。

【0317】

図11は、本発明の1つ以上の実施形態の実施のためのコンピューティングデバイス2

10

20

30

40

50

000の概略ブロック図である。コンピューティングデバイス2000は、マイクロコンピュータ、ワークステーション、または軽量の携帯デバイスなどのデバイスであってよい。コンピューティングデバイス2000は、通信バスに接続された、マイクロプロセッサのような中央処理装置(CPU)2001；本発明の実施形態の方法の実行可能コードおよび本発明の実施形態による画像の少なくとも一部を符号化または復号する方法を実施するために必要な変数およびパラメータを記録するために適合したレジスタを記憶するためのランダムアクセスメモリ(RAM)2002(メモリ容量は例えば、拡張ポートに接続したオプションRAMにより拡張することが可能である)；本発明の実施形態を実施するためのコンピュータプログラムを記憶するためのリードオンリーメモリ(ROM)2003；通常、処理されるべきデジタルデータが送信または受信される通信ネットワークに接続されるネットワークインタフェース(NET)2004；で構成される。ネットワークインタフェース(NET)2004は、単一のネットワークインタフェースであってもよいし、異なるネットワークインタフェース(例えば、有線インタフェースと無線インタフェース、あるいは異なる種類の有線または無線インタフェース)の集合で構成されていてもよい。データパケットは、CPU2001で実行されるソフトウェアアプリケーションの制御下で、送信のためにネットワークインタフェースに書き込まれるか、受信のためにネットワークインタフェースから読み出され、ユーザインタフェース(UI)2005は、ユーザからの入力を受け取るため、またはユーザに対する情報を表示するために使用され得、ハードディスク(HD)2006は、大量記憶デバイスとして提供され得、入出力モジュール(IO)2007は、ビデオソースまたはディスプレイなどの外部デバイスからのノとのデータの受信/送信に使用され得る、などである。実行可能コードは、ROM2003、HD2006、またはディスクなどの取り外し可能なデジタル媒体のいずれかに格納することができる。変形例によれば、プログラムの実行可能コードは、実行される前に、HD2006などの通信装置2000の記憶手段の1つに格納されるために、NET2004を介して、通信ネットワーク手段によって受信されることが可能である。CPU2001は、本発明の実施形態によるプログラムまたはプログラムのソフトウェアコードの命令または部分の実行を制御および指示するように適合されており、この命令は前述の記憶手段の1つに格納される。電源投入後、CPU2001は、例えばプログラムROM2003またはHD2006からそれらの命令がロードされた後、ソフトウェアアプリケーションに関連するメインRAMメモリ2002からの命令を実行することが可能である。このようなソフトウェアアプリケーションは、CPU2001によって実行されると、本発明による方法のステップを実行させる。

【0318】

また、本発明の別の実施形態によれば、前述の実施形態による復号器は、コンピュータ、携帯電話(セルラーフォン)、テーブルなどのユーザ端末、またはユーザにコンテンツを提供/表示できる他のタイプの装置(例えば、表示装置)に設けられることが理解される。さらに別の実施形態によれば、前述の実施形態による符号化器は、カメラ、ビデオカメラまたはネットワークカメラ(例えば、閉回路テレビジョンまたはビデオ監視カメラ)をも含む撮像装置に設けられ、符号化器が符号化するためのコンテンツを捕らえて提供する。そのような2つの例を、図11および図12を参照して以下に提供する。

【0319】

・ネットワークカメラ

図11は、ネットワークカメラ2102とクライアント装置2104を含むネットワークカメラシステム2100を説明するための図である。

ネットワークカメラ2102は、撮像部2106、符号化部2108、通信部2110、及び制御部2112を含む。

ネットワークカメラ2102とクライアント装置2104は、ネットワーク200を介して相互に通信可能に接続されている。

撮像部2106は、レンズとイメージセンサ(例えば、電荷結合素子(CCD)や相補型金属酸化膜半導体(CMOS))を含み、対象物の画像を撮影し、その画像に基づく画

10

20

30

40

50

像データを生成する。この画像は、静止画であっても、動画であってもよい。

符号化部 2108 は、上述した符号化方法を用いて、画像データを符号化する

ネットワークカメラ 2102 の通信部 2110 は、符号化部 2108 により符号化された符号化画像データをクライアント装置 2104 に送信する。

さらに、通信部 2110 は、クライアント装置 2104 からコマンドを受信する。コマンドには、符号化部 2108 の符号化のためのパラメータを設定するコマンドが含まれる。

制御部 2112 は、通信部 2110 が受信したコマンドに従って、ネットワークカメラ 2102 内の他のユニットを制御する。

クライアント装置 2104 は、通信部 2114 と、復号部 2116 と、制御部 2118 とを含む。

クライアント装置 2104 の通信部 2114 は、ネットワークカメラ 2102 にコマンドを送信する。

さらに、クライアント装置 2104 の通信部 2114 は、ネットワークカメラ 2102 から符号化された画像データを受信する。

復号部 2116 は、上述した復号方法を用いて、符号化された画像データを復号する。

クライアント装置 2104 の制御部 2118 は、ユーザ操作や通信部 2114 が受信したコマンドに従って、クライアント装置 2104 内の他のユニットを制御する。

クライアント装置 2104 の制御部 2118 は、復号部 2116 で復号された画像を表示するように、表示装置 2120 を制御する。

また、クライアント装置 2104 の制御部 2118 は、ネットワークカメラ 2102 のパラメータの値を指定する GUI (グラフィカルユーザインタフェース) を表示するように表示装置 2120 を制御し、符号化部 2108 の符号化のためのパラメータを含む。

また、クライアント装置 2104 の制御部 2118 は、表示装置 2120 が表示する GUI に対するユーザ操作入力に応じて、クライアント装置 2104 内の他のユニットを制御する。

クライアント装置 2104 の制御部 2119 は、表示装置 2120 が表示する GUI に対するユーザの操作入力に応じて、ネットワークカメラ 2102 に対するパラメータの値を指定するコマンドをネットワークカメラ 2102 に送信するように、クライアント装置 2104 の通信部 2114 を制御する。

【0320】

・スマートフォン

図 12 は、スマートフォン 2200 を説明するための図である。

スマートフォン 2200 は、通信部 2202、復号部 2204、制御部 2206、表示部 2208、画像記録装置 2210、及びセンサ 2212 を含む。

通信部 2202 は、ネットワーク 200 を介して符号化された画像データを受信する。

復号部 2204 は、通信部 2202 が受信した符号化された画像データを復号する。

復号部 2204 は、上述した復号方法を用いて、符号化された画像データを復号する。

制御部 2206 は、通信部 2202 が受信したユーザ操作やコマンドに従って、スマートフォン 2200 内の他のユニットを制御する。

例えば、制御部 2206 は、復号部 2204 が復号した画像を表示するように、表示部 2208 を制御する。

【0321】

本発明を実施形態を参照して説明したが、本発明は開示された実施形態に限定されないことが理解されよう。添付の特許請求の範囲に定義された本発明の範囲から逸脱することなく、様々な変更および修正がなされ得ることは、当業者には理解されよう。本明細書 (添付の請求項、要旨及び図面を含む) に開示された全ての特徴、及び/又は、そのように開示された任意の方法又はプロセスの全てのステップは、かかる特徴及び/又はステップの少なくとも一部が相互に排他的である組み合わせを除き、任意の組み合わせで組み合わせることが可能である。本明細書 (添付の請求項、要約および図面を含む) に開示された各特徴は、明示的に別段の記載がない限り、同一、同等または類似の目的を果たす代替の

10

20

30

40

50

特徴に置き換えることができる。したがって、明示的に別段の記載がない限り、開示された各特徴は、同等または類似の特徴の一般的な一連の一例のみである。

【0322】

また、上記の比較、決定、評価、選択、実行 (execution)、実行 (performing)、または検討の任意の結果 (例えば、符号化またはフィルタリング処理中に行われた選択) は、ビットストリーム中のデータ (例えば、フラグまたは結果を示すデータ) において示されるかまたは決定 / 参照可能であり、示されるかまたは決定 / 参照された結果は、例えば、復号処理中に比較、決定、評価、選択、実行、実行、または検討を実際に行う代わりに処理で使用できることが、理解される。

【0323】

特許請求の範囲において、「含む (comprising)」という語は、他の要素又はステップを排除するものではなく、不定冠詞「a」又は「an」は、複数を排除するものでもない。異なる特徴が相互に異なる従属請求項に記載されているという事実のみでは、これらの特徴の組合せが有利に使用できないことを示すことはない。

【0324】

特許請求の範囲に記載された参照数字は、説明のためのものであり、特許請求の範囲を限定するものではない。

10

20

30

40

50

【図 5】

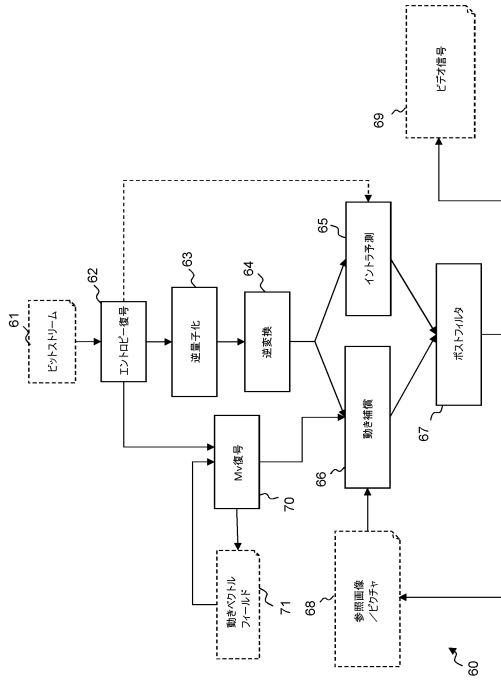


Figure 5

【図 6】

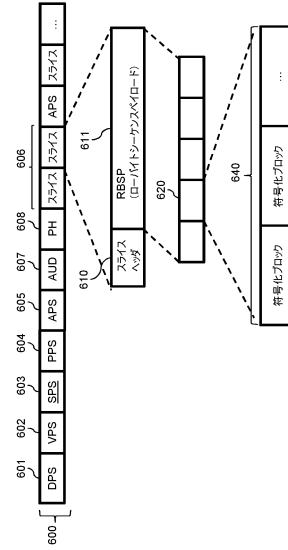


Figure 6

【図 7】

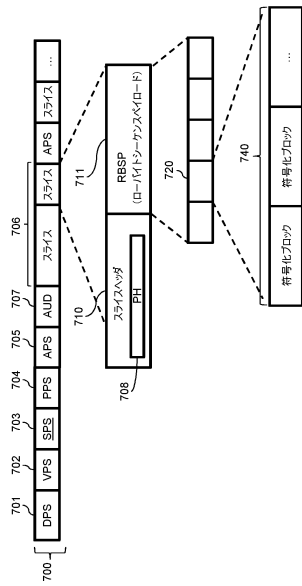


Figure 7

【図 8】

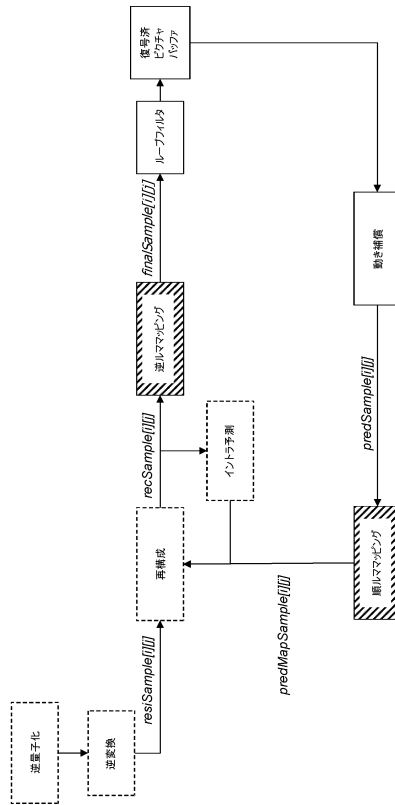


Figure 8

10

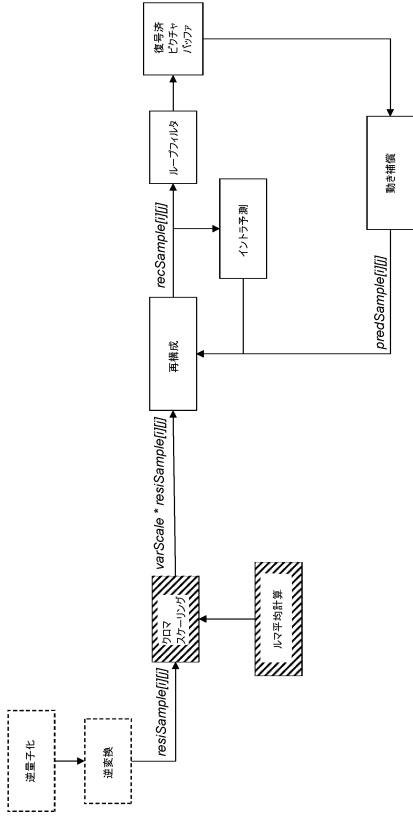
20

30

40

50

【図 9】



【図 11】

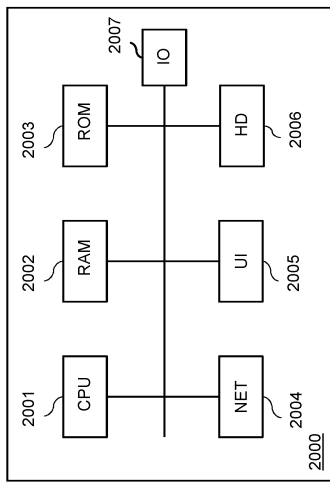


Figure 11

【図 10】

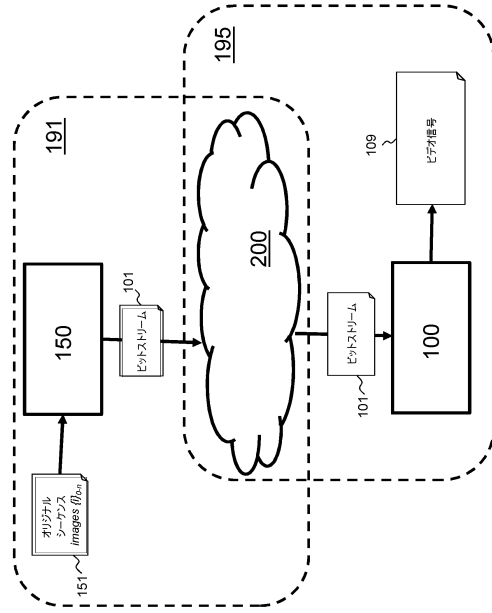


Figure 9

Figure 10

【図 12】

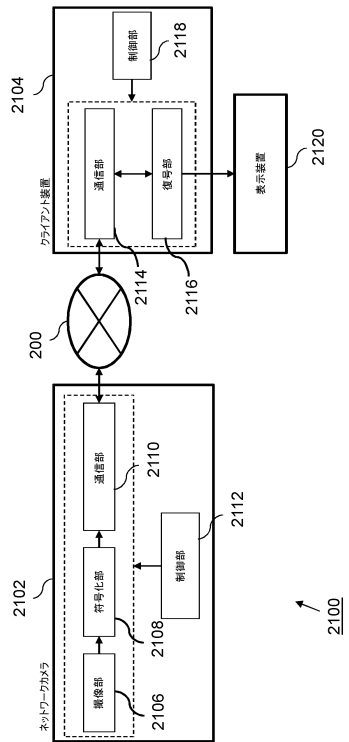


Figure 12

【 図 13 】

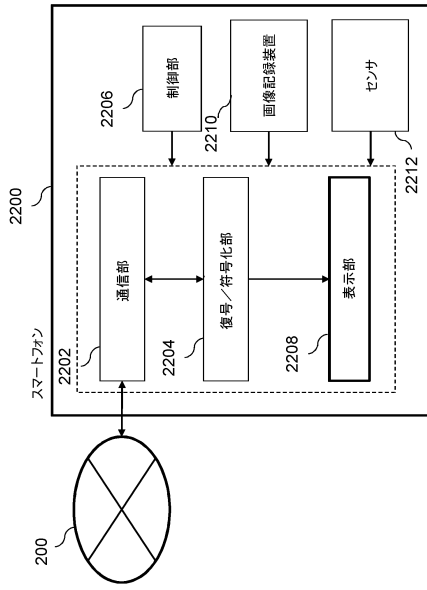


Figure 13

10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

英国(GB)

ル キヤノン リサーチ センター フランス エス . エー . エス 内

(72)発明者 オンノ , パトリス

フランス国 レンヌ - アタラント , セデックス セッソン - セヴィニエ 3 5 5 1 7 , リュ ドゥ
ラ トゥッシュランベール キヤノン リサーチ センター フランス エス . エー . エス 内

審査官 松元 伸次

(56)参考文献

Benjamin Bross et al. , Versatile Video Coding (Draft 8) , Joint Video Experts Team (JVET)
of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, 7-17 Ja
nuary 2020 , JVET-Q2001-v7 , 2020年01月20日 , pp.44-45,47,56-57,112,118,135,139,
165

(58)調査した分野 (Int.Cl. , D B 名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8