

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7375196号
(P7375196)

(45)発行日 令和5年11月7日(2023.11.7)

(24)登録日 令和5年10月27日(2023.10.27)

(51)国際特許分類 F I
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70
H 0 4 N 19/82 (2014.01) H 0 4 N 19/82

請求項の数 3 (全74頁)

(21)出願番号	特願2022-529064(P2022-529064)	(73)特許権者	502032105
(86)(22)出願日	令和2年11月17日(2020.11.17)		エルジー エレクトロニクス インコーポ レイティド
(65)公表番号	特表2023-503070(P2023-503070 A)		L G E L E C T R O N I C S I N C .
(43)公表日	令和5年1月26日(2023.1.26)		大韓民国, ソウル, ヨンドウンポーク, ヨイーデロ, 1 2 8
(86)国際出願番号	PCT/KR2020/016140		1 2 8, Yeoui-daero, Y eongdeungpo-gu, 07
(87)国際公開番号	WO2021/101203		3 3 6 Seoul, Republic of Korea
(87)国際公開日	令和3年5月27日(2021.5.27)		
審査請求日	令和4年6月13日(2022.6.13)	(74)代理人	100099759
(31)優先権主張番号	62/937,230		弁理士 青木 篤
(32)優先日	令和1年11月18日(2019.11.18)	(74)代理人	100123582
(33)優先権主張国・地域又は機関	米国(US)		弁理士 三橋 真二
		(74)代理人	100165191

最終頁に続く

(54)【発明の名称】 フィルタリングに基づく画像コーディング装置及び方法

(57)【特許請求の範囲】

【請求項1】

デコード装置により実行される画像デコード方法において、
ビットストリームを介してレジデュアル情報を含む画像情報を取得するステップと、
前記レジデュアル情報に基づいて現在ピクチャの復元サンプルを生成するステップと、
前記復元サンプルに対するインループフィルタリング手順に基づいて、修正された復元
サンプルを生成するステップと、を含み、

前記画像情報は、SPS(sequence parameter set)を含み、
前記SPSは、SPS仮想境界可用フラグを含み、

前記SPS仮想境界可用フラグの値に基づいて、前記SPSはSPS仮想境界存在フラグ
をさらに含み、

前記SPSの仮想境界可用フラグに基づいて、前記インループフィルタリング手順は仮想
境界を横切って実行されるか否かを決定する、画像デコード方法。

【請求項2】

エンコード装置により実行される画像エンコード方法において、
現在ブロックに対するレジデュアルサンプルを生成するステップと、
前記現在ブロックに対する前記レジデュアルサンプルに基づいてレジデュアル情報を生
成するステップと、

現在ピクチャの復元サンプルのためのインループフィルタリング関連情報を生成するス
テップと、

10

20

前記レジデュアル情報及び前記インループフィルタリング関連情報を含む画像情報をエンコードするステップと、を含み、

前記画像情報は、SPS (sequence parameter set) を含み、
前記SPSは、SPS仮想境界可用フラグを含み、

前記SPS仮想境界可用フラグの値に基づいて、前記SPSはSPS仮想境界存在フラグをさらに含み、

前記インループフィルタリング関連情報は、インループフィルタリング手順が仮想境界を横切って実行されるかどうかに関連したSPSの仮想境界可用フラグを含む、画像エンコード方法。

【請求項3】

画像に対するデータの送信方法であって、

前記画像に対するビットストリームを取得するステップであって、

前記ビットストリームは、現在ブロックに対するレジデュアルサンプルを生成し、前記現在ブロックに対するレジデュアルサンプルに基づいて、レジデュアル情報を生成し、現在ピクチャの回復したサンプルに対するインループフィルタリング関連情報を生成し、前記レジデュアル情報及び前記インループフィルタリング関連情報を含む映像情報をエンコーディングすることにより生成される、ステップと、

前記ビットストリームを含む前記データを送信するステップを含み、

前記映像情報は、シーケンスパラメータセット (SPS) を含み、

前記SPSは、SPS仮想境界可用フラグを含み、

前記SPS仮想境界可用フラグの値に基づいて、前記SPSはSPS仮想境界存在フラグをさらに含み、

前記インループフィルタリング関連情報は、インループフィルタリングプロセスが仮想境界を横切って実行されるかに関連する前記SPS仮想境界可用フラグを含む、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本文書は、フィルタリングに基づく画像コーディング装置及び方法に関する。

【背景技術】

【0002】

近年、4K又は8K以上のUHD (Ultra High Definition) 画像／ビデオのような高解像度、高品質の画像／ビデオに対する需要が様々な分野で増加している。画像／ビデオデータが高解像度、高品質になるほど、既存の画像／ビデオデータに比べて相対的に送信される情報量又はビット量が増加するので、既存の有無線広帯域回線のような媒体を利用して画像データを送信するか、既存の格納媒体を利用して画像／ビデオデータを格納する場合、送信費用と格納費用が増加される。

【0003】

また、近年、VR (Virtual Reality)、AR (Artificial Reality) コンテンツやホログラムなどの実感メディア (Immersive Media) に対する関心及び需要が増加しており、ゲーム画像のように、現実画像と異なる画像特性を有する画像／ビデオに対する放送が増加している。

【0004】

これにより、前記のような様々な特性を有する高解像度・高品質の画像／ビデオの情報を効果的に圧縮して送信するか、格納し、再生するために高効率の画像／ビデオ圧縮技術が求められる。

【0005】

具体的に、主観的／客観的ビジュアルの品質を高めるためにインループフィルタリング手順が行われ、仮想境界に基づくインループフィルタリングの実行のための情報のシグナリングの効率を高めるための案に対する議論がある。

【発明の概要】

【課題を解決するための手段】

【0006】

本文書の一実施形態によると、画像／ビデオコーディングの効率を高める方法及び装置を提供する。

【0007】

本文書の一実施形態によると、効率的なフィルタリング適用方法及び装置を提供する。

【0008】

本文書の一実施形態によると、デブロックキング、SAO (sample adaptive loop)、ALF (adaptive loop filtering) を効率的に適用する方法及び装置を提供する。

【0009】

本文書の一実施形態によると、仮想境界に基づいてインループフィルタリングが行われる。

【0010】

本文書の一実施形態によると、SPS (sequence parameter set) は仮想境界を横切ってインループフィルタリングが実行されるかどうかを示すSPSの仮想境界可用フラグを含むことができる。

【0011】

本文書の一実施形態によると、SPSの仮想境界可用フラグに基づいて、仮想境界を横切ってインループフィルタリングを実行することができる。

【0012】

本文書の一実施形態によると、ビデオ／画像エンコードを行うエンコード装置を提供する。

【0013】

本文書の一実施形態によると、本文書の実施形態の少なくとも1つに開示されたビデオ／画像エンコード方法によって生成されたエンコードされたビデオ／画像情報が格納されたコンピュータ読み取り可能なデジタル格納媒体を提供する。

【0014】

本文書の一実施形態によると、デコード装置により本文書の実施形態の少なくとも1つに開示されたビデオ／画像デコード方法を行うように引き起こすエンコードされた情報又はエンコードされたビデオ／画像情報が格納されたコンピュータ読み取り可能なデジタル格納媒体を提供する。

【発明の効果】

【0015】

本文書の一実施形態によると、全般的な画像／ビデオ圧縮効率を上げることができる。

【0016】

本文書の一実施形態によると、効率的なフィルタリングを介して主観的／客観的ビジュアル品質を高めることができる。

【0017】

本文書の一実施形態による仮想境界に基づくインループフィルタリング手順は、ハードウェアリソースを節約することができる。

【0018】

本文書の一実施形態によると、仮想境界に基づくインループフィルタリング手順を効率的に実行することができ、また、フィルタリング性能を改善することができる。

【0019】

本文書の一実施形態によると、仮想境界に基づくインループフィルタリングのための情報を効率的にシグナリングできる。

【図面の簡単な説明】

【0020】

【図1】本文書の実施形態に適用されることができるビデオ／画像コーディングシステム

10

20

30

40

50

の例を概略的に示す。

【図 2】本文書の実施形態に適用されることができるビデオ／画像エンコード装置の構成を概略的に説明する図面である。

【図 3】本文書の実施形態に適用されることができるビデオ／画像デコード装置の構成を概略的に説明する図面である。

【図 4】コーディングされた画像／ビデオに対する階層構造を例示的に示す。

【図 5】エンコード装置におけるフィルタリングに基づくエンコード方法を説明するためのフローチャートである。

【図 6】デコード装置におけるフィルタリングに基づくデコード方法を説明するためのフローチャートである。

【図 7】本文書の実施形態によるビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

【図 8】本文書の実施形態によるビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

【図 9】本文書の実施形態による画像／ビデオデコード方法及び関連コンポーネントの一例を概略的に示す。

【図 10】本文書の実施形態による画像／ビデオデコード方法及び関連コンポーネントの一例を概略的に示す。

【図 11】本文書で開示された実施形態が適用され得るコンテンツストリーミングシステムの例を示す。

【発明を実施するための形態】

【0021】

本文書は、様々な変更を加えることができ、種々の実施形態を有することができ、特定実施形態を図面に例示し、詳細に説明しようとする。しかしながら、これは、本文書を特定実施形態に限定しようとするものではない。本明細書で常用する用語は、単に特定の実施形態を説明するために使用されたものであって、本文書の技術的思想を限定しようとする意図で使用されるものではない。単数の表現は、文脈上明白に異なるように意味しない限り、複数の表現を含む。本明細書において「含む」または「有する」などの用語は、明細書上に記載された特徴、数字、ステップ、動作、構成要素、部品、またはこれらを組み合わせたものが存在することを指定しようとするものであり、1つまたはそれ以上の他の特徴や数字、ステップ、動作、構成要素、部品、またはこれらを組み合わせたものなどの存在または付加可能性を予め排除しないことと理解されるべきである。

【0022】

一方、本文書で説明される図面上の各構成は、互いに異なる特徴的な機能に関する説明の都合上、独立的に図示されたものであって、各構成が互いに別個のハードウェアや別個のソフトウェアで実現されるということを意味するものではない。例えば、各構成のうち、2つ以上の構成が結合されて1つの構成をなすこともでき、1つの構成を複数の構成に分けることもできる。各構成が統合及び／又は分離された実施形態も本文書の本質から外れない限り、本文書の権利範囲に含まれる。

【0023】

以下、添付の図面を参照して、本文書の好ましい実施形態をより詳細に説明する。以下、図面上の同一の構成要素に対しては同一の参照符号を使用し、同一の構成要素について重複した説明は省略する。

【0024】

この文書は、ビデオ／画像コーディングに関する。例えば、この文書において開示された方法／実施形態は、VVC (Versatile Video Coding) 標準 (ITU-T Rec. H. 266)、VVC 以後の次世代ビデオ／イメージコーディング標準、またはそれ以外のビデオコーディング関連標準 (例えば、HEVC (High Efficiency Video Coding) 標準 (ITU-T Rec. H. 265)、EVC (essential video coding) 標準、AVS2 標準など) に

10

20

30

40

50

関連する。

【0025】

この文書ではビデオ／画像コーディングに関する様々な実施形態を提示し、他の言及がない限り、前記実施形態は、互いに組み合わせて実行されることもできる。

【0026】

この文書においてビデオ (video) は、時間の流れによる一連の画像 (image) の集合を意味し得る。ピクチャ (picture) は、一般に特定の時間帯の1つの画像を示す単位を意味し、スライス (slice)／タイル (tile) はコーディングにおいてピクチャの一部を構成する単位である。スライス／タイルは、1つ以上のCTU (coding tree unit) を含んでもよい。1つのピクチャは1つ以上のスライス／タイルで構成されてもよい。1つのピクチャは1つ以上のタイルグループで構成されてもよい。1つのタイルグループは1つ以上のタイルを含んでもよい。

10

【0027】

ピクセル (pixel) 又はペル (pel) は、1つのピクチャ (又は、画像) を構成する最小の単位を意味し得る。また、ピクセルに対応する用語として「サンプル (sample)」が使用されてもよい。サンプルは、一般的にピクセル又はピクセルの値を示し、ルマ (luma) 成分のピクセル／ピクセル値のみを示すこともでき、クロマ (chroma) 成分のピクセル／ピクセル値のみを示すこともできる。または、サンプルは、空間ドメインでのピクセル値を意味することもでき、このようなピクセル値が周波数ドメインに変換されると、周波数ドメインでの変換係数を意味することもできる。

20

【0028】

ユニット (unit) は、画像処理の基本単位を示す。ユニットは、ピクチャの特定領域及び当該領域に関する情報のうち少なくとも1つを含む。1つのユニットは、1つのルマブロック及び2つのクロマ (例えば、cb、cr) ブロックを含む。ユニットは、場合によって、ブロック (block) または領域 (area) などの用語と混用して使用されてもよい。一般的な場合、M×Nブロックは、M個の列とN個の行からなるサンプル (または、サンプルアレイ) または変換係数 (transform coefficient) の集合 (又は、アレイ) を含む。

【0029】

本文書において、「／」と「、」は、「及び／又は」と解釈される。例えば、「A／B」は「A及び／又はB」と解釈され、「A、B」は「A及び／又はB」と解釈される。追加的に、「A／B／C」は「A、B及び／又はCの少なくとも1つ」を意味する。また、「A、B、C」も「A、B及び／又はCの少なくとも1つ」を意味する。

30

【0030】

追加的に、本文書において「又は」は「及び／又は」と解釈される。例えば、「A又はB」は、1)「A」のみを意味し、2)「B」のみを意味するか、3)「A及びB」を意味し得る。言い換えれば、本文書の「又は」は「追加的に又は代替的に (additionally or alternatively)」を意味し得る。

【0031】

本明細書において「少なくとも1つのA及びB (at least one of A and B)」は、「Aのみ」、「Bのみ」又は「AとBの両方」を意味し得る。また、本明細書において「少なくとも1つのA又はB (at least one of A or B)」や「少なくとも1つのA及び／又はB (at least one of A and/or B)」という表現は、「少なくとも1つのA及びB (at least one of A and B)」と同様に解釈され得る。

40

【0032】

また、本明細書において、「少なくとも1つのA、B及びC (at least one of A, B and C)」は、「Aのみ」、「Bのみ」、「Cのみ」又は「A、B及びCの任意の全ての組み合わせ (any combination of A, B and C)」を意味し得る。また、「少なくとも1つのA、B又はC (at least one

50

of A, B or C)」や「少なくとも1つのA、B及び／又はC (at least one of A, B and C)」は「少なくとも1つのA、B及びC (at least one of A, B and C)」を意味し得る。

【0033】

また、本明細書において用いられる括弧は「例えば (for example)」を意味し得る。具体的に、「予測 (イントラ予測)」と表示されている場合、「予測」の一例として「イントラ予測」が提案されたものであり得る。言い換えると、本明細書の「予測」は「イントラ予測」に制限 (limit) されることなく、「イントラ予測」が「予測」の一例として提案されるものであり得る。また、「予測 (すなわち、イントラ予測)」と表示されている場合にも、「予測」の一例として「イントラ予測」が提案されたものであり得る。

10

【0034】

本明細書において1つの図面内において個別に説明される技術的特徴は、個別に実現されてもよく、同時に実現されてもよい。

【0035】

図1は、本文書が適用されることができるビデオ／画像コーディングシステムの例を概略的に示す。

【0036】

図1に示すように、ビデオ／画像コーディングシステムはソースデバイス及び受信デバイスを備えることができる。ソースデバイスは、エンコードされたビデオ (video)／画像 (image) 情報又はデータをファイル又はストリーミング形態でデジタル格納媒体又はネットワークを介して受信デバイスに伝達することができる。

20

【0037】

前記ソースデバイスは、ビデオソース、エンコード装置、送信部を備えることができる。前記受信デバイスは、受信部、デコード装置、及びレンダラーを備えることができる。前記エンコード装置は、ビデオ／画像エンコード装置と呼ばれることができ、前記デコード装置は、ビデオ／画像デコード装置と呼ばれることができる。送信機は、エンコード装置に備えられることができる。受信機は、デコード装置に備えられることができる。レンダラーは、ディスプレイ部を備えることができ、ディスプレイ部は、別個のデバイス又は外部コンポーネントで構成されることもできる。

30

【0038】

ビデオソースは、ビデオ／画像のキャプチャ、合成、又は生成過程などを介してビデオ／画像を取得することができる。ビデオソースは、ビデオ／画像キャプチャデバイス及び／又はビデオ／画像生成デバイスを含むことができる。ビデオ／画像キャプチャデバイスは、例えば、1つ以上のカメラ、以前にキャプチャされたビデオ／画像を含むビデオ／画像アーカイブなどを含むことができる。ビデオ／画像生成デバイスは、例えば、コンピュータ、タブレット、及びスマートフォンなどを含むことができ、(電子的に) ビデオ／画像を生成することができる。例えば、コンピュータなどを介して仮想のビデオ／画像が生成され得るし、この場合、関連データが生成される過程にビデオ／画像キャプチャ過程が代替されることができる。

40

【0039】

エンコード装置は、入力ビデオ／画像をエンコードすることができる。エンコード装置は、圧縮及びコーディング効率のために、予測、変換、量子化など、一連の手順を行うことができる。エンコードされたデータ (エンコードされたビデオ／画像情報) は、ビットストリーム (bit stream) 形態で出力されることができる。

【0040】

送信部は、ビットストリーム形態で出力されたエンコードされたビデオ／画像情報又はデータをファイル又はストリーミング形態でデジタル格納媒体又はネットワークを介して受信デバイスの受信部に伝達することができる。デジタル格納媒体は、USB、SD、CD、DVD、ブルーレイ、HDD、SSDなど、様々な格納媒体を含むことができる。送

50

信部は、予め決められたファイルフォーマットを介してメディアファイルを生成するためのエレメントを含むことができ、放送／通信ネットワークを介しての送信のためのエレメントを含むことができる。受信部は、前記ビットストリームを受信／抽出してデコード装置に伝達することができる。

【0041】

デコード装置は、エンコード装置の動作に対応する逆量子化、逆変換、予測など、一連の手順を行ってビデオ／画像をデコードすることができる。

【0042】

レンダラーは、デコードされたビデオ／画像をレンダリングすることができる。レンダリングされたビデオ／画像は、ディスプレイ部を介してディスプレイされることができる。

10

【0043】

図2は、本文書が適用されることができるビデオ／画像エンコード装置の構成を概略的に説明する図面である。以下、ビデオエンコード装置というのは、画像エンコード装置を含むことができる。

【0044】

図2に示すように、エンコード装置200は、画像分割部 (image partitioner) 210、予測部 (predictor) 220、レジデュアル処理部 (residual processor) 230、エントロピーエンコード部 (entropy encoder) 240、加算部 (adder) 250、フィルタリング部 (filter) 260、及びメモリ (memory) 270を備えて構成されることができる。予測部220は、インター予測部221及びイントラ予測部222を備えることができる。レジデュアル処理部230は、変換部 (transformer) 232、量子化部 (quantizer) 233、逆量子化部 (dequantizer) 234、逆変換部 (inverse transformer) 235を備えることができる。レジデュアル処理部230は、減算部 (subtractor, 231) をさらに備えることができる。加算部250は、復元部 (reconstructor) 又は復元ブロック生成部 (reconstructed block generator) と呼ばれることができる。前述した画像分割部210、予測部220、レジデュアル処理部230、エントロピーエンコード部240、加算部250、及びフィルタリング部260は、実施形態によって1つ以上のハードウェアコンポーネント (例えば、エンコーダチップセット又はプロセッサ) によって構成されることができる。また、メモリ270は、DPB (decoded picture buffer) を備えることができ、デジタル格納媒体によって構成されることもできる。前記ハードウェアコンポーネントは、メモリ270を内／外部コンポーネントとしてさらに備えることもできる。

20

30

【0045】

画像分割部210は、エンコード装置200に入力された入力画像 (又は、ピクチャ、フレーム) を1つ以上の処理ユニット (processing unit) に分割することができる。一例として、前記処理ユニットは、コーディングユニット (coding unit, CU) と呼ばれることができる。この場合、コーディングユニットは、コーディングツリーユニット (coding tree unit, CTU) 又は最大コーディングユニット (largest coding unit, LCU) からQTBT (Quad-tree binary-tree ternary-tree) 構造によって再帰的に (recursively) 分割されることができる。例えば、1つのコーディングユニットは、クアドツリー構造、バイナリツリー構造、及び／又はターナリ構造に基づいて下位 (deeper) デプスの複数のコーディングユニットに分割されることができる。この場合、例えば、クアドツリー構造が先に適用され、バイナリツリー構造及び／又はターナリ構造がその後適用されることができる。または、バイナリツリー構造が先に適用されることもできる。それ以上分割されない最終コーディングユニットに基づいて本開示に係るコーディング手順が行われ得る。この場合、画像特性によるコーディング効率などに基づいて、最大コーディングユニットが最終コーディングユニットとして使用さ

40

50

れることができ、又は、必要に応じてコーディングユニットは、再帰的に (r e c u r s i v e l y) もっと下位デプスのコーディングユニットに分割されて最適のサイズのコーディングユニットが最終コーディングユニットとして使用されることができる。ここで、コーディング手順とは、後述する予測、変換、及び復元などの手順を含むことができる。他の例として、前記処理ユニットは、予測ユニット (P U : P r e d i c t i o n U n i t) 又は変換ユニット (T U : T r a n s f o r m U n i t) をさらに備えることができる。この場合、前記予測ユニット及び前記変換ユニットは、各々前述した最終コーディングユニットから分割又はパーティショニングされることができる。前記予測ユニットは、サンプル予測の単位であることができ、前記変換ユニットは、変換係数を導く単位及び／又は変換係数からレジデュアル信号 (r e s i d u a l s i g n a l) を導く単位であることができる。

10

【0046】

ユニットは、場合によって、ブロック (b l o c k) 又は領域 (a r e a) などの用語と混用して使用されることができる。一般的な場合、 $M \times N$ ブロックは、 M 個の列と N 個の行からなるサンプル又は変換係数 (t r a n s f o r m c o e f f i c i e n t) 等の集合を示すことができる。サンプルは、一般的にピクセル又はピクセルの値を示すことができ、輝度 (l u m a) 成分のピクセル／ピクセル値のみを示すことができ、彩度 (c h r o m a) 成分のピクセル／ピクセル値のみを示すこともできる。サンプルは、1つのピクチャ (又は、画像) をピクセル (p i x e l) 又はペル (p e l) に対応する用語として使用することができる。

20

【0047】

減算部231は、入力画像信号 (原本ブロック、原本サンプル又は原本サンプルアレイ) で予測部220から出力された予測信号 (予測されたブロック、予測サンプル又は予測サンプルアレイ) を減算してレジデュアル信号 (レジデュアルブロック、レジデュアルサンプル又はレジデュアルサンプルアレイ) を生成することができ、生成されたレジデュアル信号は、変換部232に送信される。予測部220は、処理対象ブロック (以下、現在ブロックという) に対する予測を行い、前記現在ブロックに対する予測サンプルを含む予測されたブロック (p r e d i c t e d b l o c k) を生成することができる。予測部220は、現在ブロック又はCU単位でイントラ予測が適用されるか、又はインター予測が適用されるか決定することができる。予測部は、各予測モードについての説明で後述するように、予測モード情報など、予測に関する様々な情報を生成してエントロピーエンコード部240に伝達することができる。予測に関する情報は、エントロピーエンコード部240でエンコードされてビットストリーム形態で出力されることができる。

30

【0048】

イントラ予測部222は、現在ピクチャ内のサンプルを参照して現在ブロックを予測できる。前記参照されるサンプルは、予測モードによって前記現在ブロックの隣接 (n e i g h b o r) に位置することができ、又は、離れて位置することもできる。イントラ予測で予測モードは、複数の非方向性モードと複数の方向性モードとを含むことができる。非方向性モードは、例えば、DCモード及びプラナーモード (P l a n a r モード) を含むことができる。方向性モードは、予測方向の細かい程度によって、例えば、33個の方向性予測モード又は65個の方向性予測モードを含むことができる。ただし、これは、例示であり、設定によってそれ以上又はそれ以下の個数の方向性予測モードが使用され得る。イントラ予測部222は、隣接ブロックに適用された予測モードを用いて、現在ブロックに適用される予測モードを決定することもできる。

40

【0049】

インター予測部221は、参照ピクチャ上で動きベクトルにより特定される参照ブロック (参照サンプルアレイ) に基づいて、現在ブロックに対する予測されたブロックを導くことができる。このとき、インター予測モードで送信される動き情報の量を減らすために、隣接ブロックと現在ブロックとの間の動き情報の相関性に基づいて、動き情報をブロック、サブブロック、又はサンプル単位で予測することができる。前記動き情報は、動きベ

50

クトル及び参照ピクチャインデックスを含むことができる。前記動き情報は、インター予測方向（L0予測、L1予測、Bi予測等）情報をさらに含むことができる。インター予測の場合に、隣接ブロックは、現在ピクチャ内に存在する空間的隣接ブロック（`spatial neighboring block`）と参照ピクチャに存在する時間的隣接ブロック（`temporal neighboring block`）とを含むことができる。前記参照ブロックを含む参照ピクチャと前記時間的隣接ブロックを含む参照ピクチャとは同じであることができ、異なることもできる。前記時間的隣接ブロックは、同一位置参照ブロック（`collocated reference block`）、同一位置CU（`col CU`）などの名前で呼ばれることができ、前記時間的隣接ブロックを含む参照ピクチャは、同一位置ピクチャ（`collocated picture`、`colPic`）と呼ばれることもできる。例えば、インター予測部221は、隣接ブロックに基づいて動き情報候補リストを構成し、前記現在ブロックの動きベクトル及び／又は参照ピクチャインデックスを導出するために、どの候補が使用されるかを指示する情報を生成することができる。様々な予測モードに基づいてインター予測が行われ得るし、例えば、スキップモードとマージモードとの場合に、インター予測部221は、隣接ブロックの動き情報を現在ブロックの動き情報として利用することができる。スキップモードの場合、マージモードとは異なり、レジデュアル信号が送信されないことがある。動き情報予測（`motion vector prediction`、MVP）モードの場合、隣接ブロックの動きベクトルを動きベクトル予測子（`motion vector predictor`）として用い、動きベクトル差分（`motion vector difference`）をシグナリングすることにより、現在ブロックの動きベクトルを指示することができる。

【0050】

予測部220は、後述する様々な予測方法に基づいて予測信号を生成することができる。例えば、予測部は、1つのブロックに対する予測のために、イントラ予測又はインター予測を適用することができるだけでなく、イントラ予測とインター予測とを同時に適用することができる。これは、`combined inter and intra prediction`（CIIP）と呼ばれることができる。また、予測部は、ブロックに対する予測のために、イントラブロックコピー（`intra block copy`、IBC）を実行することができる。前記イントラブロックコピーは、例えば、SCC（`screen content coding`）などのように、ゲームなどのコンテンツ画像／動画コーディングのために使用されることができる。IBCは、基本的に現在ピクチャ内で予測を行うが、現在ピクチャ内で参照ブロックを導出する点においてインター予測と同様に行われることができる。すなわち、IBCは、本文書において説明されるインター予測技法のうち、少なくとも1つを用いることができる。

【0051】

インター予測部221及び／又はイントラ予測部222を含むを介して生成された予測信号は、復元信号を生成するために用いられるか、レジデュアル信号を生成するために用いられることができる。変換部232は、レジデュアル信号に変換技法を適用して、変換係数（`transform coefficients`）を生成することができる。例えば、変換技法は、DCT（`Discrete Cosine Transform`）、DST（`Discrete Sine Transform`）、GBT（`Graph-Based Transform`）、又はCNT（`Conditionally Non-linear Transform`）などを含むことができる。ここで、GBTは、ピクセル間の関係情報をグラフで表現するという際に、このグラフから得られた変換を意味する。CNTは、以前に復元された全てのピクセル（`all previously reconstructed pixel`）を用いて予測信号を生成し、それに基づいて取得される変換を意味する。また、変換過程は正方形の同じサイズを有するピクセルブロックに適用されてもよく、正方形ではない可変サイズのブロックに適用されてもよい。

【0052】

量子化部233は、変換係数を量子化してエントロピーエンコード部240に送信され

10

20

30

40

50

、エントロピーエンコード部240は、量子化された信号（量子化された変換係数に関する情報）をエンコードしてビットストリームに出力することができる。前記量子化された変換係数に関する情報は、レジデュアル情報と呼ばれることができる。量子化部233は、係数スキャン順序（scan order）に基づいてブロック形態の量子化された変換係数を1次元ベクトル形態で再整列することができ、前記1次元ベクトル形態の量子化された変換係数に基づいて前記量子化された変換係数に関する情報を生成することもできる。エントロピーエンコード部240は、例えば、指数ゴロム（exponential Golomb）、CAVLC（context-adaptive variable length coding）、CABAC（context-adaptive binary arithmetic coding）などのような様々なエンコード方法を行うことができる。エントロピーエンコード部240は、量子化された変換係数の他に、ビデオ／イメージ復元に必要な情報（例えば、シンタックス要素（syntax elements）の値等）を共に又は別にエンコードすることもできる。エンコードされた情報（例えば、エンコードされたビデオ／画像情報）は、ビットストリーム形態でNAL（network abstraction layer）ユニット単位で送信又は格納されることができる。前記ビデオ／画像情報は、アダプテーションパラメータセット（APS）、ピクチャパラメータセット（PPS）、シーケンスパラメータセット（SPS）、又はビデオパラメータセット（VPS）等、様々なパラメータセットに関する情報をさらに含むことができる。また、前記ビデオ／画像情報は、一般制限情報（general constraint information）をさらに含むことができる。本文書において、後述されるシグナリング／送信される情報及び／又はシンタックス要素は、前述したエンコード手順を介してエンコードされて前記ビットストリームに含まれることができる。前記ビットストリームは、ネットワークを介して送信されることができ、又はデジタル格納媒体に格納されることができる。ここで、ネットワークは、放送網及び／又は通信網などを含むことができ、デジタル格納媒体は、USB、SD、CD、DVD、ブルーレイ、HDD、SSDなど、様々な格納媒体を含むことができる。エントロピーエンコード部240から出力された信号は、送信する送信部（図示せず）及び／又は格納する格納部（図示せず）がエンコード装置200の内／外部エレメントとして構成されることができ、又は送信部は、エントロピーエンコード部240に含まれることもできる。

【0053】

量子化部233から出力された量子化された変換係数は、予測信号を生成するために用いられることができる。例えば、量子化された変換係数に逆量子化部234及び逆変換部235を介して逆量子化及び逆変換を適用することにより、レジデュアル信号（レジデュアルブロック又はレジデュアルサンプル）を復元できる。加算部155は、復元されたレジデュアル信号を予測部220から出力された予測信号に加えることにより、復元（reconstructed）信号（復元ピクチャ、復元ブロック、復元サンプル又は復元サンプルアレイ）が生成され得る。スキップモードが適用された場合のように、処理対象ブロックに対するレジデュアルがない場合、予測されたブロックが復元ブロックとして使用されることができる。生成された復元信号は、現在ピクチャ内の次の処理対象ブロックのイントラ予測のために使用されることができ、後述するように、フィルタリングを経て次のピクチャのインター予測のために使用されることもできる。

【0054】

一方、ピクチャエンコード及び／又は復元過程でLMCS（luma mapping with chroma scaling）が適用されることもできる。

【0055】

フィルタリング部260は、復元信号にフィルタリングを適用して主観的／客観的画質を向上させることができる。例えば、フィルタリング部260は、復元ピクチャに様々なフィルタリング方法を適用して修正された（modified）復元ピクチャを生成することができ、前記修正された復元ピクチャをメモリ270、具体的に、メモリ270のDPBに格納することができる。前記様々なフィルタリング方法は、例えば、デブロッキング

10

20

30

40

50

グフィルタリング、サンプル適応的オフセット (sample adaptive offset、SAO)、適応的ループフィルタ (adaptive loop filter)、両方向フィルタ (bilateral filter) などを含むことができる。フィルタリング部 260 は、各フィルタリング方法についての説明で後述するように、フィルタリングに関する様々な情報を生成してエントロピーエンコード部 240 に伝達することができる。フィルタリングに関する情報は、エントロピーエンコード部 240 でエンコードされてビットストリーム形態で出力されることができる。

【0056】

メモリ 270 に送信された修正された復元ピクチャは、インター予測部 221 で参照ピクチャとして使用されることができる。エンコード装置は、これを介してインター予測が適用される場合、エンコード装置 100 とデコード装置における予測ミスマッチを避けることができ、符号化効率も向上させることができる。

【0057】

メモリ 270 の DPB は、修正された復元ピクチャをインター予測部 221 における参照ピクチャとして使用するために格納することができる。メモリ 270 は、現在ピクチャ内の動き情報が導出された (又は、エンコードされた) ブロックの動き情報及び／又は既に復元されたピクチャ内のブロックの動き情報を格納することができる。前記格納された動き情報は、空間的隣接ブロックの動き情報又は時間的隣接ブロックの動き情報として活用するために、インター予測部 221 に伝達することができる。メモリ 270 は、現在ピクチャ内の復元されたブロックの復元サンプルを格納することができ、イントラ予測部 222 に伝達することができる。

【0058】

図 3 は、本文書が適用されることができるビデオ／画像デコード装置の構成を概略的に説明する図面である。

【0059】

図 3 に示すように、デコード装置 300 は、エントロピーデコード部 (entropy decoder) 310、レジデュアル処理部 (residual processor) 320、予測部 (predictor) 330、加算部 (adder) 340、フィルタリング部 (filter) 350、及びメモリ (memoery) 360 を備えて構成されることができる。予測部 330 は、インター予測部 331 及びイントラ予測部 332 を備えることができる。レジデュアル処理部 320 は、逆量子化部 (dequantizer) 321 及び逆変換部 (inverse transformer) 321 を備えることができる。前述したエントロピーデコード部 310、レジデュアル処理部 320、予測部 330、加算部 340、及びフィルタリング部 350 は、実施形態によって 1 つのハードウェアコンポーネント (例えば、デコーダチップセット又はプロセッサ) により構成されることができる。また、メモリ 360 は、DPB (decoded picture buffer) を備えることができ、デジタル格納媒体により構成されることもできる。前記ハードウェアコンポーネントは、メモリ 360 を内／外部コンポーネントとしてさらに備えることもできる。

【0060】

ビデオ／画像情報を含むビットストリームが入力されれば、デコード装置 300 は、図 3 のエンコード装置でビデオ／画像情報が処理されたプロセスに対応して画像を復元できる。例えば、デコード装置 300 は、前記ビットストリームから取得したブロック分割関連情報に基づいてユニット／ブロックを導出できる。デコード装置 300 は、エンコード装置で適用された処理ユニットを用いてデコードを行うことができる。したがって、デコードの処理ユニットは、例えば、コーディングユニットであることができ、コーディングユニットは、コーディングツリーユニット又は最大コーディングユニットからクアドツリー構造、バイナリツリー構造、及び／又はターナリツリー構造にしたがって分割されることができる。コーディングユニットから 1 つ以上の変換ユニットが導出され得る。そして、デコード装置 300 を介してデコード及び出力された復元画像信号は、再生装置を介

10

20

30

40

50

して再生されることができる。

【0061】

デコード装置300は、図3のエンコード装置から出力された信号をビットストリーム形態で受信することができ、受信された信号は、エントロピーデコード部310を介してデコードされることができる。例えば、エントロピーデコード部310は、前記ビットストリームをパースングして画像復元（又は、ピクチャ復元）に必要な情報（例えば、ビデオ／画像情報）を導出できる。前記ビデオ／画像情報は、アダプテーションパラメータセット（APS）、ピクチャパラメータセット（PPS）、シーケンスパラメータセット（SPS）、又はビデオパラメータセット（VPS）など、様々なパラメータセットに関する情報をさらに含むことができる。また、前記ビデオ／画像情報は、一般制限情報（general constraint information）をさらに含むことができる。デコード装置は、前記パラメータセットに関する情報及び／又は前記一般制限情報に基づいてさらにピクチャをデコードすることができる。本文書において後述されるシグナリング／受信される情報及び／又はシンタックス要素は、前記デコード手順を介してデコードされて、前記ビットストリームから取得されることができる。例えば、エントロピーデコード部310は、指数ゴロム符号化、CAVLC、又はCABACなどのコーディング方法に基づいてビットストリーム内の情報をデコードし、画像復元に必要なシンタックス要素の値、レジデュアルに関する変換係数の量子化された値を出力できる。より具体的に、CABACエントロピーデコード方法は、ビットストリームで各シンタックス要素に該当するピンを受信し、デコード対象のシンタックス要素情報と隣接及びデコード対象ブロックのデコード情報又は以前ステップでデコードされたシンボル／ピンの情報を利用して文脈（context）モデルを決定し、決定された文脈モデルによってピン（bin）の発生確率を予測してピンの算術デコード（arithmetic decoding）を行い、各シンタックス要素の値に該当するシンボルを生成することができる。このとき、CABACエントロピーデコード方法は、文脈モデル決定後、次のシンボル／ピンの文脈モデルのためにデコードされたシンボル／ピンの情報を利用して文脈モデルをアップデートすることができる。エントロピーデコード部310でデコードされた情報のうち、予測に関する情報は、予測部330に提供され、エントロピーデコード部310でエントロピーデコードが行われたレジデュアル情報、すなわち、量子化された変換係数及び関連パラメータ情報は、逆量子化部321に入力されることができる。また、エントロピーデコード部310でデコードされた情報のうち、フィルタリングに関する情報は、フィルタリング部350に提供されることができる。一方、エンコード装置から出力された信号を受信する受信部（図示せず）がデコード装置300の内／外部エレメントとしてさらに構成されることができ、又は受信部は、エントロピーデコード部310の構成要素であることもできる。一方、本文書に係るデコード装置は、ビデオ／画像／ピクチャデコード装置と呼ばれることができ、前記デコード装置は、情報デコーダ（ビデオ／画像／ピクチャ情報デコーダ）及びサンプルデコーダ（ビデオ／画像／ピクチャサンプルデコーダ）に区分することもできる。前記情報デコーダは、前記エントロピーデコード部310を備えることができ、前記サンプルデコーダは、前記逆量子化部321、逆変換部322、予測部330、加算部340、フィルタリング部350、及びメモリ360のうち、少なくとも1つを備えることができる。

【0062】

逆量子化部321では、量子化された変換係数を逆量子化して変換係数を出力できる。逆量子化部321は、量子化された変換係数を2次元のブロック形態で再整列することができる。この場合、前記再整列は、エンコード装置で行われた係数スキャン順序に基づいて再整列を行うことができる。逆量子化部321は、量子化パラメータ（例えば、量子化ステップサイズ情報）を用いて量子化された変換係数に対する逆量子化を行い、変換係数（transform coefficient）を取得することができる。

【0063】

逆変換部322では、変換係数を逆変換してレジデュアル信号（レジデュアルブロック

、レジデュアルサンプルアレイ）を取得するようになる。

【0064】

予測部は、現在ブロックに対する予測を行い、前記現在ブロックに対する予測サンプルを含む予測されたブロック（predicted block）を生成することができる。予測部は、エントロピーデコード部310から出力された前記予測に関する情報に基づいて、前記現在ブロックにイントラ予測が適用されるか、又はインター予測が適用されるか決定することができ、具体的なイントラ／インター予測モードを決定することができる。

【0065】

予測部は、後述する様々な予測方法に基づいて予測信号を生成することができる。例えば、予測部は、1つのブロックに対する予測のために、イントラ予測又はインター予測を適用することができるだけでなく、イントラ予測とインター予測とを同時に適用することができる。これは、combined inter and intra prediction（CIIP）と呼ばれることができる。また、予測部は、ブロックに対する予測のために、イントラブロックコピー（intra block copy、IBC）を実行することができる。前記イントラブロックコピーは、例えば、SCC（screen content coding）などのように、ゲームなどのコンテンツ画像／動画コーディングのために使用されることができる。IBCは、基本的に現在ピクチャ内で予測を行うが、現在ピクチャ内で参照ブロックを導出する点においてインター予測と同様に行われることができる。すなわち、IBCは、本文書において説明されるインター予測技法のうち、少なくとも1つを利用することができる。パレットモードは、イントラコーディング又はイントラ予測の一例と見なすことができる。

【0066】

イントラ予測部331は、現在ピクチャ内のサンプルを参照して現在ブロックを予測できる。前記参照されるサンプルは、予測モードによって前記現在ブロックの隣接（neighbor）に位置することができ、又は離れて位置することができる。イントラ予測において予測モードは、複数の非方向性モードと複数の方向性モードとを含むことができる。イントラ予測部331は、隣接ブロックに適用された予測モードを用いて、現在ブロックに適用される予測モードを決定することもできる。

【0067】

インター予測部332は、参照ピクチャ上で動きベクトルにより特定される参照ブロック（参照サンプルアレイ）に基づいて、現在ブロックに対する予測されたブロックを導くことができる。このとき、インター予測モードで送信される動き情報の量を減らすために、隣接ブロックと現在ブロックとの間の動き情報の相関性に基づいて動き情報をブロック、サブブロック、又はサンプル単位で予測することができる。前記動き情報は、動きベクトル及び参照ピクチャインデックスを含むことができる。前記動き情報は、インター予測方向（L0予測、L1予測、Bi予測等）情報をさらに含むことができる。インター予測の場合に、隣接ブロックは、現在ピクチャ内に存在する空間的隣接ブロック（spatial neighboring block）と参照ピクチャに存在する時間的隣接ブロック（temporal neighboring block）とを備えることができる。例えば、インター予測部332は、隣接ブロックに基づいて動き情報候補リストを構成し、受信した候補選択情報に基づいて前記現在ブロックの動きベクトル及び／又は参照ピクチャインデックスを導出できる。様々な予測モードに基づいてインター予測が行われ得るし、前記予測に関する情報は、前記現在ブロックに対するインター予測のモードを指示する情報を含むことができる。

【0068】

加算部340は、取得されたレジデュアル信号を予測部から出力された予測信号（予測されたブロック、予測サンプルアレイ）に加えることにより、復元信号（復元ピクチャ、復元ブロック、復元サンプルアレイ）を生成することができる。スキップモードが適用された場合のように、処理対象ブロックに対するレジデュアルがない場合、予測されたブロックが復元ブロックとして使用されることができる。

【0069】

加算部340は、復元部又は復元ブロック生成部と呼ばれることができる。生成された復元信号は、現在ピクチャ内の次の処理対象ブロックのイントラ予測のために使用されることができ、後述するように、フィルタリングを経て出力されることができ、又は次のピクチャのインター予測のために使用されることもできる。

【0070】

一方、ピクチャデコード過程でLMCS (luma mapping with chroma scaling) が適用されることもできる。

【0071】

フィルタリング部350は、復元信号にフィルタリングを適用して主観的／客観的画質を向上させることができる。例えば、フィルタリング部350は、復元ピクチャに様々なフィルタリング方法を適用して修正された(modified)復元ピクチャを生成することができ、前記修正された復元ピクチャをメモリ360、具体的に、メモリ360のDPBに送信することができる。前記様々なフィルタリング方法は、例えば、デブロッキングフィルタリング、サンプル適応的オフセット(sample adaptive offset)、適応的ループフィルタ(adaptive loop filter)、両方向フィルタ(bilateral filter)などを含むことができる。

【0072】

メモリ360のDPBに格納された(修正された)復元ピクチャは、インター予測部332で参照ピクチャとして使用されることができる。メモリ360は、現在ピクチャ内の動き情報が導出された(又は、デコードされた)ブロックの動き情報及び／又は既に復元されたピクチャ内のブロックの動き情報を格納できる。前記格納された動き情報は、空間的隣接ブロックの動き情報又は時間的隣接ブロックの動き情報として活用するために、インター予測部332に伝達することができる。メモリ360は、現在ピクチャ内の復元されたブロックの復元サンプルを格納でき、イントラ予測部331に伝達することができる。

【0073】

本明細書において、デコード装置300の予測部330、逆量子化部321、逆変換部322、及びフィルタリング部350などで説明された実施形態は、各々、エンコード装置200の予測部220、逆量子化部234、逆変換部235、及びフィルタリング部260などにも同一又は対応されるように適用されることができる。

【0074】

前述したように、ビデオコーディングを実行するにあたって、圧縮効率を上げるために予測を実行する。これを介してコーディング対象ブロックである現在ブロックに対する予測サンプルを含む予測されたブロックを生成することができる。ここで、前記予測されたブロックは、空間ドメイン(又は、ピクセルドメイン)での予測サンプルを含む。前記予測されたブロックは、エンコード装置及びデコード装置で同じく導出され、前記エンコード装置は、原本ブロックの原本サンプル値自体でない前記原本ブロックと前記予測されたブロックとの間のレジデュアルに対する情報(レジデュアル情報)をデコード装置にシグナリングすることで画像コーディング効率を上げることができる。デコード装置は、前記レジデュアル情報に基づいてレジデュアルサンプルを含むレジデュアルブロックを導出し、前記レジデュアルブロックと前記予測されたブロックとを合わせて復元サンプルを含む復元ブロックを生成することができ、復元ブロックを含む復元ピクチャを生成することができる。

【0075】

前記レジデュアル情報は、変換及び量子化手順を介して生成されることができる。例えば、エンコード装置は、前記原本ブロックと前記予測されたブロックとの間のレジデュアルブロックを導出し、前記レジデュアルブロックに含まれているレジデュアルサンプル(レジデュアルサンプルアレイ)に変換手順を実行して変換係数を導出し、前記変換係数に量子化手順を実行して量子化された変換係数を導出することで、関連したレジデュアル情報を(ビットストリームを介して)デコード装置にシグナリングできる。ここで、前記レ

ジデュアル情報は、前記量子化された変換係数の値情報、位置情報、変換技法、変換カーネル、量子化パラメータなどの情報を含むことができる。デコード装置は、前記レジデュアル情報に基づいて逆量子化／逆変換の手順を実行してレジデュアルサンプル（又は、レジデュアルブロック）を導出することができる。デコード装置は、予測されたブロックと前記レジデュアルブロックに基づいて復元ピクチャを生成することができる。また、エンコード装置は、以後ピクチャのインター予測のための参照のために量子化された変換係数を逆量子化／逆変換してレジデュアルブロックを導出し、これに基づいて復元ピクチャを生成することができる。

【0076】

本文書において、量子化／逆量子化及び／又は変換／逆変換のうち少なくとも1つは、省略されることができる。前記量子化／逆量子化が省略される場合、前記量子化された変換係数は、変換係数と呼ばれることができる。前記変換／逆変換が省略される場合、前記変換係数は、係数又はレジデュアル係数と呼ばれることもでき、又は、表現の統一性のために、依然として変換係数と呼ばれることもできる。

【0077】

本文書において、量子化された変換係数及び変換係数は、各々、変換係数及びスケールリングされた（scaled）変換係数と呼ばれることができる。この場合、レジデュアル情報は、変換係数（ら）に関する情報を含むことができ、前記変換係数（ら）に関する情報は、レジデュアルコーディングシンタックスを介してシグナリングされることができる。前記レジデュアル情報（又は、前記変換係数（ら）に関する情報）に基づいて変換係数が導出されることができ、前記変換係数に対する逆変換（スケールリング）を介してスケールリングされた変換係数が導出されることができる。前記スケールリングされた変換係数に対する逆変換（変換）に基づいてレジデュアルサンプルが導出されることができる。これは本文書の他の部分でも同様に適用／表現されることができる。

【0078】

エンコード装置／デコード装置の予測部は、ブロック単位でインター予測を行って予測サンプルを導出することができる。インター予測は、現在ピクチャ以外のピクチャのデータ要素（ex. サンプル値、又は動き情報）に依存的な方法で導出される予測を示すことができる。現在ブロックにインター予測が適用される場合、参照ピクチャのインデックスの指す参照ピクチャ上で動きベクトルにより特定される参照ブロック（参照サンプルアレイ）に基づいて、現在ブロックに対する予測されたブロック（予測サンプルアレイ）を誘導することができる。このとき、インター予測モードで送信される動き情報の量を減らすために、隣接ブロックと現在ブロックとの間の動き情報の相関性に基づいて、現在ブロックの動き情報をブロック、サブブロック又はサンプル単位で予測することができる。前記動き情報は、動きベクトル及び参照ピクチャのインデックスを含むことができる。前記動き情報は、インター予測タイプ（L0予測、L1予測、Bi予測等）の情報を更に含むことができる。インター予測が適用される場合、隣接ブロックは、現在ピクチャ内に存在する空間的隣接ブロック（spatial neighboring block）と、参照ピクチャに存在する時間的隣接ブロック（temporal neighboring block）とを含むことができる。前記参照ブロックを含む参照ピクチャと、前記時間的隣接ブロックを含む参照ピクチャとは同一であってもよく、異なってもよい。前記時間的隣接ブロックは、同じ位置参照ブロック（collocated reference block）、同じ位置CU（colCU）等の名称で呼ばれ得、前記時間的隣接ブロックを含む参照ピクチャは、同じ位置ピクチャ（collocated picture、colPic）と呼ばれることもある。例えば、現在ブロックの隣接ブロックに基づいて動き情報の候補リストが構成されることができ、前記現在ブロックの動きベクトル及び／又は参照ピクチャのインデックスを導出するために、どの候補が選択（使用）されるかを指示するフラグ又はインデックス情報がシグナリングされることができる。様々な予測モードに基づいてインター予測が行われ、例えば、スキップモードとマージモードの場合、現在ブロックの動き情報は、選択された隣接ブロックの動き情報を同一であり得る。スキッ

10

20

30

40

50

ブモードの場合、マージモードと異なり、レジデュアル信号が送信されないことがある。動き情報予測 (motion vector prediction、MVP) モードの場合、選択された隣接ブロックの動きベクトルを動きベクトル予測子 (motion vector predictor) として利用し、動きベクトル差分 (motion vector difference) はシグナリングされることができる。この場合、前記動きベクトル予測子及び動きベクトル差分の和を用いて、前記現在ブロックの動きベクトルを導出することができる。

【0079】

前記動き情報は、インター予測タイプ (L0予測、L1予測、Bi予測等) に応じて、L0動き情報及び／又はL1動き情報を含むことができる。L0方向の動きベクトルは、L0動きベクトル又はMVL0と呼ばれ得、L1方向の動きベクトルは、L1動きベクトル又はMVL1と呼ばれ得る。L0動きベクトルに基づいた予測は、L0予測と呼ばれ得、L1動きベクトルに基づいた予測をL1予測と呼ばれ得、前記L0動きベクトル及び前記L1動きベクトルの両方に基づいた予測を双 (Bi) 予測と呼ばれ得る。ここで、L0動きベクトルは、参照ピクチャリストL0 (L0) に関連する動きベクトルを示すことができ、L1動きベクトルは、参照ピクチャリストL1 (L1) に関連する動きベクトルを示すことができる。参照ピクチャリストL0は、前記現在ピクチャよりも、出力順序上、以前のピクチャを参照ピクチャに含むことができ、参照ピクチャリストL1は、前記現在ピクチャよりも、出力順序上、以後のピクチャを含むことができる。前記以前のピクチャは、順方向 (参照) ピクチャと呼ばれ得、前記以後のピクチャは、逆方向 (参照) ピクチャと呼ばれ得る。前記参照ピクチャリストL0は、前記現在ピクチャよりも、出力順序上、以後のピクチャを参照ピクチャにさらに含むことができる。この場合、前記参照ピクチャリストL0内で前記以前のピクチャが先にインデキシングされ、前記以後のピクチャは、その後にインデキシングされ得る。前記参照ピクチャリストL1は、前記現在ピクチャよりも、出力順序上、以前のピクチャを参照ピクチャにさらに含むことができる。この場合、前記参照ピクチャリストL1内で前記以後のピクチャが先にインデキシングされ、前記以前のピクチャは、その後にインデキシングされ得る。ここで、出力順序は、POC (picture order count) 順序 (order) に対応し得る。

【0080】

図4は、コーディングされた画像／ビデオに対する階層構造を例示的に示す。

【0081】

図4に示すように、コーディングされた画像／ビデオは、画像／ビデオのデコード処理及びその自体を扱うVCL (video coding layer、ビデオコーディング階層)、符号化された情報を送信して格納する下位システム、及びVCLと下位システムとの間に存在してネットワーク適応機能を担当するNAL (network abstraction layer、ネットワーク抽象階層) に区分されている。

【0082】

VCLでは、圧縮された画像データ (スライスデータ) を含むVCLデータを生成し、又は、ピクチャパラメータセット (Picture Parameter Set: PPS)、シーケンスパラメータセット (Sequence Parameter Set: SPS)、ビデオパラメータセット (Video Parameter Set: VPS) などの情報を含むパラメータセット又は画像のデコード過程に付加的に必要なSEI (Supplemental Enhancement Information) メッセージを生成することができる。

【0083】

NALでは、VCLで生成されたRBSP (Raw Byte Sequence Payload) にヘッダ情報 (NALユニットヘッダ) を付加してNALユニットを生成することができる。このとき、RBSPは、VCLで生成されたスライスデータ、パラメータセット、SEIメッセージなどを意味する。NALユニットヘッダには該当NALユニットに含まれるRBSPデータによって特定されるNALユニットタイプ情報を含むこと

ができる。

【0084】

前記図面に示すように、NALユニットは、VCLで生成されたRBS Pによって、VCL NALユニットとNon-VCL NALユニットとに区分されることができる。VCL NALユニットは、画像に対する情報（スライスデータ）を含んでいるNALユニットを意味することができ、Non-VCL NALユニットは、画像をデコードするために必要な情報（パラメータセット又はSEIメッセージ）を含んでいるNALユニットを意味することができる。

【0085】

前述したVCL NALユニット、Non-VCL NALユニットは、下位システムのデータ規格によってヘッダ情報を付けてネットワークを介して送信されることができる。例えば、NALユニットは、H. 266/VVCファイルフォーマット、RTP (Real-time Transport Protocol)、TS (Transport Stream) などのような所定規格のデータ形態に変形されて多様なネットワークを介して送信されることができる。

【0086】

前述したように、NALユニットは、該当NALユニットに含まれるRBS Pデータ構造 (structure) によってNALユニットタイプが特定されることができ、このようなNALユニットタイプに対する情報は、NALユニットヘッダに格納されてシグナリングされることができる。

【0087】

例えば、NALユニットが画像に対する情報（スライスデータ）を含むかどうかによって、大別して、VCL NALユニットタイプとNon-VCL NALユニットタイプとに分類されることができる。VCL NALユニットタイプは、VCL NALユニットが含むピクチャの性質及び種類などによって分類されることができ、Non-VCL NALユニットタイプは、パラメータセットの種類などによって分類されることができる。

【0088】

下記は、Non-VCL NALユニットタイプが含むパラメータセットの種類などによって特定されたNALユニットタイプの一例においてある。

【0089】

－APS (Adaptation Parameter Set) NAL unit: APSを含むNALユニットに対するタイプ

【0090】

－DPS (Decoding Parameter Set) NAL unit: DPSを含むNALユニットに対するタイプ

【0091】

－VPS (Video Parameter Set) NAL unit: VPSを含むNALユニットに対するタイプ

【0092】

－SPS (Sequence Parameter Set) NAL unit: SPSを含むNALユニットに対するタイプ

【0093】

－PPS (Picture Parameter Set) NAL unit: PPSを含むNALユニットに対するタイプ

【0094】

－PH (Picture header) NAL unit: PHを含むNALユニットに対するタイプ

【0095】

前述したNALユニットタイプは、NALユニットタイプのためのシンタックス情報を有し、前記シンタックス情報は、NALユニットヘッダに格納されてシグナリングされる

10

20

30

40

50

ことができる。例えば、前記シンタックス情報は、`nal_unit_type`であり、NALユニットタイプは、`nal_unit_type`値に特定されることができる。

【0096】

一方、前述したように、1つのピクチャは、複数のスライスを含むことができ、1つのスライスは、スライスヘッダ及びスライスデータを含むことができる。この場合、1つのピクチャ内の複数のスライス（スライスヘッダ及びスライスデータ集合）に対して1つのピクチャヘッダがさらに付加されることができる。前記ピクチャヘッダ（ピクチャヘッダシンタックス）は、前記ピクチャに共通的に適用できる情報／パラメータを含むことができる。本文書において、スライスは、タイルグループに混用又は代替されることができる。また、本文書において、スライスヘッダは、タイプグループヘッダに混用又は代替されることができる。

10

【0097】

前記スライスヘッダ（スライスヘッダシンタックス、スライスヘッダ情報）は、前記スライスに共通的に適用できる情報／パラメータを含むことができる。前記APS（APSシンタックス）又はPPS（PPSシンタックス）は、1つ以上のスライス又はピクチャに共通的に適用できる情報／パラメータを含むことができる。前記SPS（SPSシンタックス）は、1つ以上のシーケンスに共通的に適用できる情報／パラメータを含むことができる。前記VPS（VPSシンタックス）は、多重レイヤに共通的に適用できる情報／パラメータを含むことができる。前記DPS（DPSシンタックス）は、ビデオ全般に共通的に適用できる情報／パラメータを含むことができる。前記DPSは、CVS（`coded video sequence`）の接合（`concatenation`）に関連する情報／パラメータを含むことができる。本文書において、上位レベルシンタックス（`High level syntax`、HLS）とは、前記APSシンタックス、PPSシンタックス、SPSシンタックス、VPSシンタックス、DPSシンタックス、ピクチャヘッダシンタックス、スライスヘッダシンタックスのうち少なくとも1つを含むことができる。

20

【0098】

本文書において、エンコード装置からデコード装置にエンコードされてビットストリーム形態でシグナリングされる画像／ビデオ情報は、ピクチャ内のパーティショニング関連情報、イントラ／インター予測情報、レジデュアル情報、インループフィルタリング情報などを含むだけでなく、前記スライスヘッダに含まれている情報、前記ピクチャヘッダに含まれている情報、前記APSに含まれている情報、前記PPSに含まれている情報、SPSに含まれている情報、VPSに含まれている情報、及び／又はDPSに含まれている情報を含むことができる。また、前記画像／ビデオ情報は、NALユニットヘッダの情報をさらに含むことができる。

30

【0099】

一方、量子化など、圧縮符号化過程で発生するエラーによる原本（`original`）画像と復元画像の差異を補償するために、前述したように、復元サンプル又は復元ピクチャにインループフィルタリング手順が実行されることができる。前述したように、インループフィルタリングは、エンコード装置のフィルタ部及びデコード装置のフィルタ部で実行されることができる。デブロッキングフィルタ、SAO及び／又は適応的ループフィルタ（ALF）が適用されることができる。例えば、ALF手順は、デブロッキングフィルタリング手順及び／又はSAO手順が完了した後に実行されることができる。ただし、この場合も、デブロッキングフィルタリング手順及び／又はSAO手順が省略されることもできる。

40

【0100】

以下、ピクチャ復元及びフィルタリングに対する具体的な説明が記述される。画像／ビデオコーディングにおいて、各ブロック単位でイントラ予測／インター予測に基づいて復元ブロックが生成されることができ、復元ブロックを含む復元ピクチャが生成されることができる。現在ピクチャ／スライスがIピクチャ／スライスである場合、前記現在ピクチャ／スライスに含まれるブロックは、イントラ予測のみに基づいて復元されることができ

50

る。一方、現在ピクチャ／スライスがP又はBピクチャ／スライスである場合、前記現在ピクチャ／スライスに含まれるブロックは、イントラ予測又はインター予測に基づいて復元されることができる。この場合、現在ピクチャ／スライス内の一部ブロックに対してはイントラ予測が適用され、残りのブロックに対してはインター予測が適用されることもできる。

【0101】

イントラ予測は、現在ブロックが属するピクチャ（以下、現在ピクチャという）内の参照サンプルに基づいて現在ブロックに対する予測サンプルを生成する予測を示すことができる。現在ブロックにイントラ予測が適用される場合、現在ブロックのイントラ予測に使用する隣接参照サンプルが導出されることができる。前記現在ブロックの隣接参照サンプルは、 $nW \times nH$ 大きさの現在ブロックの左側（left）境界に隣接したサンプル及び左下側（bottom-left）に隣接した総 $2 \times nH$ 個のサンプル、現在ブロックの上側（top）境界に隣接したサンプル及び右上側（top-right）に隣接した総 $2 \times nW$ 個のサンプル及び現在ブロックの左上側（top-left）に隣接した1個のサンプルを含むことができる。または、前記現在ブロックの隣接参照サンプルは、複数列の上側隣接サンプル及び複数行の左側隣接サンプルを含むこともできる。また、前記現在ブロックの隣接参照サンプルは、 $nW \times nH$ 大きさの現在ブロックの右側（right）境界に隣接した総 nH 個のサンプル、現在ブロックの下側（bottom）境界に隣接した総 nW 個のサンプル、及び現在ブロックの右下側（bottom-right）に隣接した1個のサンプルを含むこともできる。

【0102】

ただし、現在ブロックの隣接参照サンプルのうち一部は、まだデコードされない、又は利用可能でない場合がある。この場合、デコーダは、利用可能なサンプルとして利用可能でないサンプルを代替（substitution）して予測に使用する隣接参照サンプルを構成することができる。または、利用可能なサンプルの補間（interpolation）を介して予測に使用する隣接参照サンプルを構成することができる。

【0103】

隣接参照サンプルが導出された場合、（i）現在ブロックの隣接（neighboring）参照サンプルの平均（average）又は補間（interpolation）に基づいて予測サンプルを誘導することができ、（ii）現在ブロックの隣接参照サンプルのうち、予測サンプルに対して特定（予測）方向に存在する参照サンプルに基づいて前記予測サンプルを誘導することもできる。（i）の場合は、非方向性モード又は非角度モードと呼ばれ、（ii）の場合は、方向性（directional）モード又は角度（angular）モードと呼ばれることができる。また、前記隣接参照サンプルのうち前記現在ブロックの予測サンプルを基準にして、前記現在ブロックのイントラ予測モードの予測方向の反対方向に位置する前記第2の隣接サンプルと前記第1の隣接サンプルとの補間を介して前記予測サンプルが生成されることもできる。前述した場合は、線形補間イントラ予測（Linear interpolation intra prediction、LIP）と呼ばれることができる。また、線形モデル（linear model）を利用してルマサンプルに基づいてクロマ予測サンプルが生成されることもできる。この場合は、LMモードと呼ばれることができる。また、フィルタリングされた隣接参照サンプルに基づいて前記現在ブロックの臨時予測サンプルを導出し、前記既存の隣接参照サンプル、即ち、フィルタリングされない隣接参照サンプルのうち、前記イントラ予測モードにより導出された少なくとも1つの参照サンプルと前記臨時予測サンプルとを加重和（weighted sum）して前記現在ブロックの予測サンプルを導出することもできる。前述した場合は、PDPC（Position dependent intra prediction）と呼ばれることができる。また、現在ブロックの隣接多重参照サンプルラインの中から最も予測正確度が高い参照サンプルラインを選択して該当ラインで予測方向に位置する参照サンプルを利用して予測サンプルを導出し、この時に使用された参照サンプルラインをデコード装置に指示（シグナリング）する方法でイントラ予測符号化を

10

20

30

40

50

実行することができる。前述した場合は、`multi-reference line (MRL) intra prediction`又はMRLベースのイントラ予測と呼ばれることができる。また、現在ブロックを垂直又は水平のサブパーティションに分けて同じイントラ予測モードに基づいてイントラ予測を実行し、前記サブパーティション単位で隣接参照サンプルを導出して利用できる。即ち、この場合、現在ブロックに対するイントラ予測モードが前記サブパーティションに同じく適用され、前記サブパーティション単位で隣接参照サンプルを導出して利用することによって、場合によって、イントラ予測性能を高めることができる。このような予測方法は、`intra sub-partitions (ISP)`又はISPベースのイントラ予測と呼ばれることができる。前述したイントラ予測方法は、目次1及び2でのイントラ予測モードと区分してイントラ予測タイプと呼ばれることができる。前記イントラ予測タイプは、イントラ予測技法又は付加イントラ予測モードなど、多様な用語で呼ばれることができる。例えば、前記イントラ予測タイプ（又は、付加イントラ予測モードなど）は、前述したLIP、PDPC、MRL、ISPのうち少なくとも1つを含むことができる。前記LIP、PDPC、MRL、ISPなどの特定イントラ予測タイプを除外した一般イントラ予測方法は、ノーマルイントラ予測タイプと呼ばれることができる。ノーマルイントラ予測タイプは、前記のような特定イントラ予測タイプが適用されない場合、一般的に適用されることができ、前述したイントラ予測モードに基づいて予測が実行されることができ、必要によって、導出された予測サンプルに対する後処理フィルタリングが実行されることもできる。

【0104】

具体的に、イントラ予測手順は、イントラ予測モード／タイプ決定ステップ、隣接参照サンプル導出ステップ、イントラ予測モード／タイプベースの予測サンプル導出ステップを含むことができる。また、必要によって、導出された予測サンプルに対する後処理フィルタリング（`post-filtering`）ステップが実行されることもできる。

【0105】

インループフィルタリング手順により修正された（`modified`）復元ピクチャが生成され、デコード装置から前記修正された復元ピクチャがデコードされたピクチャとして出力され、またエンコード装置／デコード装置の復号ピクチャバッファ又はメモリに格納されて、以後にピクチャのエンコード／デコード時のインター予測手順において参照ピクチャとして使用されることができる。前記インループフィルタリング手順は、前述のようにデブロッキングフィルタリング手順、SAO（`sample adaptive offset`）手順及び／又はALF（`adaptive loop filter`）手順などを含む。この場合、前記デブロッキングフィルタリング手順、SAO（`sample adaptive offset`）手順、ALF（`adaptive loop filter`）手順及びバイラテラルフィルタ（`bi-lateral filter`）手順のうちの1つ又は一部が順次適用されてもよく、または全てが順次適用されてもよい。例えば、復元ピクチャに対してデブロッキングフィルタリング手順が適用された後、SAO手順が行われてもよい。または、例えば、復元ピクチャに対してデブロッキングフィルタリング手順が適用された後、ALF手順が行われてもよい。これはエンコード装置においても同様に行われる。

【0106】

デブロッキングフィルタリングは、復元されたピクチャでブロック間の境界に発生する歪曲を除去するフィルタリング技法である。デブロッキングフィルタリング手順は、例えば、復元ピクチャでターゲット境界を導出し、前記ターゲット境界に対するbS（`boundary strength`）を決定し、前記bSに基づいて前記ターゲット境界に対するデブロッキングフィルタリングを実行することができる。前記bSは、前記ターゲット境界を隣接する2つのブロックの予測モード、動きベクトル差、参照ピクチャ同一可否、0でない有効係数の存在可否などに基づいて決定されることができる。

【0107】

SAOは、サンプル単位で復元ピクチャと原本ピクチャとのオフセット差を補償する方

法であって、例えば、バンドオフセット (Band Offset)、エッジオフセット (Edge Offset) などのタイプに基づいて適用されることができる。SAOによると、各SAOタイプによってサンプルを互いに異なるカテゴリに分類し、カテゴリに基づいて各サンプルにオフセット値を足すことができる。SAOのためのフィルタリング情報は、SAO適用可否に関する情報、SAOタイプ情報、SAOオフセット値情報などを含むことができる。SAOは、前記デブロッキングフィルタリング適用後の復元ピクチャに対して適用されることもできる。

【0108】

ALF (Adaptive Loop Filter) は、復元ピクチャに対してフィルタ形状によるフィルタ係数に基づいてサンプル単位でフィルタリングする技法である。エンコード装置は、復元ピクチャと原本ピクチャの比較を介してALF適用可否、ALF形状及び／又はALFフィルタリング係数などを決定することができ、デコード装置にシグナリングすることができる。即ち、ALFのためのフィルタリング情報は、ALF適用可否に関する情報、ALFフィルタ形状 (shape) 情報、ALFフィルタリング係数情報などを含むことができる。ALFは、前記デブロッキングフィルタリング適用後の復元ピクチャに対して適用されることもできる。

【0109】

図5は、エンコード装置におけるフィルタリングに基づくエンコード方法を説明するためのフローチャートである。図5の方法は、S500乃至S530のステップを含むことができる。

【0110】

S500のステップにおいて、エンコード装置は復元ピクチャを生成することができる。S500のステップは、前述した復元ピクチャ (又は復元サンプル) の生成手順に基づいて実行されることができる。

【0111】

S510のステップにおいて、エンコード装置はインループフィルタリング関連情報に基づいて (仮想境界を横切って) インループフィルタリングが適用されるかどうかを決定することができる。ここで、インループフィルタリングは、前述したデブロッキングフィルタリング、SAO、又はALFの少なくとも1つを含み得る。

【0112】

S520のステップにおいて、エンコード装置はS510のステップの前記決定に基づいて修正された復元ピクチャ (修正された復元サンプル) を生成することができる。ここで、修正された復元ピクチャ (修正された復元サンプル) はフィルタリングされた復元ピクチャ (フィルタリングされた復元サンプル) であり得る。

【0113】

S530のステップにおいて、エンコード装置はインループフィルタリング手順に基づいてインループフィルタリング関連情報を含む画像／ビデオ情報をエンコードすることができる。

【0114】

図6は、デコード装置におけるフィルタリングに基づくデコード方法を説明するためのフローチャートである。図6の方法は、S600乃至S630のステップを含むことができる。

【0115】

S600のステップにおいて、デコード装置はビットストリームからインループフィルタリング関連情報を含む画像／ビデオ情報を取得することができる。ここで、ビットストリームはエンコード装置から送信されたエンコードされた画像／ビデオ情報に基づくことができる。

【0116】

S610のステップにおいて、デコード装置は復元ピクチャを生成することができる。S610のステップは、前述した復元ピクチャ (又は復元サンプル) の生成手順に基づい

10

20

30

40

50

て実行されることができる。

【0117】

S620のステップにおいて、デコード装置はインループフィルタリング関連情報に基づいて（仮想境界を横切って）インループフィルタリングが適用されるかどうかを決定することができる。ここで、インループフィルタリングは、前述したデブロッキングフィルタリング、SAO、又はALFの少なくとも1つを含み得る。

【0118】

S630のステップにおいて、デコード装置はS620のステップの前記決定に基づいて修正された復元ピクチャ（修正された復元サンプル）を生成することができる。ここで、修正された復元ピクチャ（修正された復元サンプル）は、フィルタリングされた復元ピクチャ（フィルタリングされた復元サンプル）であり得る。

【0119】

前述したように、復元ピクチャにインループフィルタリング手順が適用できる。この場合、復元ピクチャの主観的／客観的ビジュアルの品質をより高めるために仮想境界を定義し、前記仮想境界を横切って（*across*）前記インループフィルタリング手順を適用することもできる。前記仮想境界は、例えば、360度画像、VR画像、又はPIP（*picture in picture*）などの不連続エッジを含み得る。例えば、前記仮想境界は予め定められた約束した位置に存在し得、その存否及び／又は位置がシグナリングされ得る。一例として、前記仮想境界はCTU行における上方の4番目のサンプルライン（具体的に、例えば、前記CTU行における上方の4番目のサンプルラインの上側）に位置し得る。他の例として、前記仮想境界は、その存否及び／又は位置に関する情報がHLSを介してシグナリングされることもある。前記HLSは、前述したように、SPS、PPS、ピクチャヘッダ、スライスヘッダなどを含み得る。

【0120】

以下では、本文書の実施形態に関する上位レベルシンタックスのシグナリング及びセマンティクスについて説明する。

【0121】

本文書の一実施形態は、ループフィルタを制御する方法を含み得る。ループフィルタを制御する本方法は、復元ピクチャに対して適用され得る。インループフィルタ（ループフィルタ）は、エンコードされたビットストリームのデコードのために使用できる。ループフィルタは、前述したデブロッキング、SAO、ALFを含み得る。SPSは、デブロッキング、SAO、ALFの各々と関連したフラグを含み得る。前記フラグは、前記SPSを参照するCLVS（*coded layer video sequence*）、CVS（*coded video sequence*）のコーディングのために、各ツールが可用であるかどうかを示すことができる。

【0122】

前記ループフィルタがCVSに対して可用である場合、前記ループフィルタの適用は、特定の境界を横切らないように制御され得る。例えば、前記ループフィルタがサブピクチャの境界を横切るかどうかは制御され得る。また、前記ループフィルタがタイルの境界を横切るかどうかは制御され得る。これと共に、前記ループフィルタが仮想境界を横切るかどうかは制御され得る。ここで、仮想境界はラインバッファの可用性に基づいてCTU上に定義され得る。

【0123】

仮想境界を横切ってインループフィルタリング手順が実行されるかどうかと関連し、インループフィルタリング関連情報はSPSの仮想境界可用フラグ（SPS内の仮想境界可用フラグ）、SPSの仮想境界存在フラグ、ピクチャヘッダの仮想境界存在フラグ、SPSピクチャヘッダの仮想境界存在フラグ、仮想境界の位置に関する情報の少なくとも1つを含み得る。

【0124】

本文書に含まれた実施形態において、仮想境界の位置に関する情報は、垂直の仮想境界

10

20

30

40

50

の x 座標に関する情報及び／又は水平の仮想境界の y 座標に関する情報を含み得る。具体的に、仮想境界の位置に関する情報は、ルマサンプル単位の垂直の仮想境界の x 座標及び／又は水平の仮想境界の y 座標に関する情報を含み得る。また、仮想境界の位置に関する情報は、SPS に存在する垂直の仮想境界の x 座標に関する情報（シンタックス要素）の数に関する情報を含み得る。また、仮想境界の位置に関する情報は、SPS に存在する水平の仮想境界の y 座標に関する情報（シンタックス要素）の数に関する情報を含み得る。又は、仮想境界の位置に関する情報は、ピクチャヘッダに存在する垂直の仮想境界の x 座標に関する情報（シンタックス要素）の数に関する情報を含み得る。また、仮想境界の位置に関する情報は、ピクチャヘッダに存在する水平の仮想境界の y 座標に関する情報（シンタックス要素）の数に関する情報を含み得る。

10

【 0 1 2 5 】

次の表は、本実施形態による SPS (sequence parameter set) の例示的なシンタックス及びセマンティクスを示す。

【 0 1 2 6 】

【表 1】

scq_parameter_sct_rbsp() {	Descriptor
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
...	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
...	
}	

20

30

40

【 0 1 2 7 】

50

【表 2】

subpics_present_flag equal to 1 specifies that subpicture parameters are present in the SPS RBSP syntax.	
subpics_present_flag equal to 0 specifies that subpicture parameters are not present in the SPS RBSP syntax.	
sps_num_subpics_minus1 plus 1 specifies the number of subpictures. sps_num_subpics_minus1 shall be in the range of 0 to 254. When not present, the value of sps_num_subpics_minus1 is inferred to be equal to 0.	
subpic_ctu_top_left_x[i] specifies horizontal position of top left CTU of i-th subpicture in unit of CtbSizeY. The length of the syntax element is $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} / \text{CtbSizeY}))$ bits. When not present, the value of subpic_ctu_top_left_x[i] is inferred to be equal to 0.	
subpic_ctu_top_left_y[i] specifies vertical position of top left CTU of i-th subpicture in unit of CtbSizeY. The length of the syntax element is $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} / \text{CtbSizeY}))$ bits. When not present, the value of subpic_ctu_top_left_y[i] is inferred to be equal to 0.	
subpic_width_minus1[i] plus 1 specifies the width of the i-th subpicture in units of CtbSizeY. The length of the syntax element is $\text{Ceil}(\text{Log2}(\text{pic_width_max_in_luma_samples} / \text{CtbSizeY}))$ bits. When not present, the value of subpic_width_minus1[i] is inferred to be equal to $\text{Ceil}(\text{pic_width_max_in_luma_samples} / \text{CtbSizeY}) - 1$.	10
subpic_height_minus1[i] plus 1 specifies the height of the i-th subpicture in units of CtbSizeY. The length of the syntax element is $\text{Ceil}(\text{Log2}(\text{pic_height_max_in_luma_samples} / \text{CtbSizeY}))$ bits. When not present, the value of subpic_height_minus1[i] is inferred to be equal to $\text{Ceil}(\text{pic_height_max_in_luma_samples} / \text{CtbSizeY}) - 1$.	
subpic_treated_as_pic_flag[i] equal to 1 specifies that the i-th subpicture of each coded picture in the CLVS is treated as a picture in the decoding process excluding in-loop filtering operations. subpic_treated_as_pic_flag[i] equal to 0 specifies that the i-th subpicture of each coded picture in the CLVS is not treated as a picture in the decoding process excluding in-loop filtering operations. When not present, the value of subpic_treated_as_pic_flag[i] is inferred to be equal to 0.	
loop_filter_across_subpic_enabled_flag[i] equal to 1 specifies that in-loop filtering operations may be performed across the boundaries of the i-th subpicture in each coded picture in the CLVS. loop_filter_across_subpic_enabled_flag[i] equal to 0 specifies that in-loop filtering operations are not performed across the boundaries of the i-th subpicture in each coded picture in the CLVS. When not present, the value of loop_filter_across_subpic_enabled_flag[i] is inferred to be equal to 1.	20
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS.	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.	
sps_sao_enabled_flag equal to 1 specifies that the sample adaptive offset process is applied to the reconstructed picture after the deblocking filter process. sps_sao_enabled_flag equal to 0 specifies that the sample adaptive offset process is not applied to the reconstructed picture after the deblocking filter process.	
sps_alf_enabled_flag equal to 0 specifies that the adaptive loop filter is disabled. sps_alf_enabled_flag equal to 1 specifies that the adaptive loop filter is enabled.	
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0.	30
sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. The value of sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.	
sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.	
sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. The value of sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.	

【0 1 2 8】

次の表は、本実施形態によるPPS (picture parameter set) の例示的なシンタックス及びセマンティクスを示す。

【0 1 2 9】

10

20

30

40

50

【表 3】

pic_parameter_set_rbsp() {	Descriptor
...	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
...	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
...	
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
...	
}	

10

20

【 0 1 3 0 】

【表 4】

no_pic_partition_flag equal to 1 specifies that no picture partitioning applied to each picture referring to the PPS. no_pic_partition_flag equal to 0 specifies each picture referring to the PPS may be partitioned into more than one tile or slice.
loop_filter_across_tiles_enabled_flag equal to 1 specifies that in-loop filtering operations may be performed across tile boundaries in pictures referring to the PPS. loop_filter_across_tiles_enabled_flag equal to 0 specifies that in-loop filtering operations are not performed across tile boundaries in pictures referring to the PPS. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
loop_filter_across_slices_enabled_flag equal to 1 specifies that in-loop filtering operations may be performed across slice boundaries in pictures referring to the PPS. loop_filter_across_slices_enabled_flag equal to 0 specifies that in-loop filtering operations are not performed across slice boundaries in pictures referring to the PPS. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.
deblocking_filter_control_present_flag equal to 1 specifies the presence of deblocking filter control syntax elements in the PPS. deblocking_filter_control_present_flag equal to 0 specifies the absence of deblocking filter control syntax elements in the PPS.
deblocking_filter_override_enabled_flag equal to 1 specifies the presence of pic_deblocking_filter_override_flag in the PHs referring to the PPS or slice_deblocking_filter_override_flag in the slice headers referring to the PPS. deblocking_filter_override_enabled_flag equal to 0 specifies the absence of pic_deblocking_filter_override_flag in PHs referring to the PPS or slice_deblocking_filter_override_flag in slice headers referring to the PPS. When not present, the value of deblocking_filter_override_enabled_flag is inferred to be equal to 0.
pps_deblocking_filter_disabled_flag equal to 1 specifies that the operation of deblocking filter is not applied for slices referring to the PPS in which slice_deblocking_filter_disabled_flag is not present. pps_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for slices referring to the PPS in which slice_deblocking_filter_disabled_flag is not present. When not present, the value of pps_deblocking_filter_disabled_flag is inferred to be equal to 0.
pps_beta_offset_div2 and pps_tc_offset_div2 specify the default deblocking parameter offsets for β and tC (divided by 2) that are applied for slices referring to the PPS, unless the default deblocking parameter offsets are overridden by the deblocking parameter offsets present in the slice headers of the slices referring to the PPS. The values of pps_beta_offset_div2 and pps_tc_offset_div2 shall both be in the range of -6 to 6, inclusive. When not present, the value of pps_beta_offset_div2 and pps_tc_offset_div2 are inferred to be equal to 0.

30

40

【 0 1 3 1 】

次の表は、本実施形態によるピクチャヘッダの例示的なシンタックス及びセマンティク

50

スを示す。

【 0 1 3 2 】

【表 5－1】

picture_header_rbsp() {	Descriptor
...	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_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(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
...	
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)

【 0 1 3 3 】

10

20

30

40

50

【表 5 - 2】

if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
...	
}	

10

【 0 1 3 4】

20

30

40

50

【表 6 – 1】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.	
ph_num_ver_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_x[i] syntax elements that are present in the PH.	
ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. The value of ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.	
ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH.	10
ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. The value of ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.	
pic_sao_enabled_present_flag equal to 1 specifies that pic_sao_luma_flag and pic_sao_chroma_flag are present in the PH. pic_sao_enabled_present_flag equal to 0 specifies that pic_sao_luma_flag and pic_sao_chroma_flag are not present in the PH. When pic_sao_enabled_present_flag is not present, it is inferred to be equal to 0.	
pic_sao_luma_enabled_flag equal to 1 specifies that SAO is enabled for the luma component in all slices associated with the PH; pic_sao_luma_enabled_flag equal to 0 specifies that SAO for the luma component may be disabled for one, or more, or all slices associated with the PH.	
pic_sao_chroma_enabled_flag equal to 1 specifies that SAO is enabled for the chroma component in all slices associated with the PH; pic_sao_chroma_enabled_flag equal to 0 specifies that SAO for chroma component may be disabled for one, or more, or all slices associated with the PH.	20
pic_alf_enabled_present_flag equal to 1 specifies that pic_alf_enabled_flag, pic_num_alf_aps_ids_luma, pic_alf_aps_id_luma[i], pic_alf_chroma_idc, and pic_alf_aps_id_chroma are present in the PH.	
pic_alf_enabled_present_flag equal to 0 specifies that pic_alf_enabled_flag, pic_num_alf_aps_ids_luma, pic_alf_aps_id_luma[i], pic_alf_chroma_idc, and pic_alf_aps_id_chroma are not present in the PH. When pic_alf_enabled_present_flag is not present, it is inferred to be equal to 0.	
pic_alf_enabled_flag equal to 1 specifies that adaptive loop filter is enabled for all slices associated with the PH and may be applied to Y, Cb, or Cr colour component in the slices. pic_alf_enabled_flag equal to 0 specifies that adaptive loop filter may be disabled for one, or more, or all slices associated with the PH. When not present, pic_alf_enabled_flag is inferred to be equal to 0.	
pic_num_alf_aps_ids_luma specifies the number of ALF APSs that the slices associated with the PH refers to.	
pic_alf_aps_id_luma[i] specifies the adaptation_parameter_set_id of the i-th ALF APS that the luma component of the slices associated with the PH refers to.	
The value of alf_luma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to pic_alf_aps_id_luma[i] shall be equal to 1.	30
pic_alf_chroma_idc equal to 0 specifies that the adaptive loop filter is not applied to Cb and Cr colour components.	
pic_alf_chroma_idc equal to 1 indicates that the adaptive loop filter is applied to the Cb colour component.	
pic_alf_chroma_idc equal to 2 indicates that the adaptive loop filter is applied to the Cr colour component.	
pic_alf_chroma_idc equal to 3 indicates that the adaptive loop filter is applied to Cb and Cr colour components. When pic_alf_chroma_idc is not present, it is inferred to be equal to 0.	
pic_alf_aps_id_chroma specifies the adaptation_parameter_set_id of the ALF APS that the chroma component of the slices associated with the PH refers to.	
pic_deblocking_filter_override_present_flag equal to 1 specifies that pic_deblocking_filter_override_flag is present in the PH. pic_deblocking_filter_override_present_flag equal to 0 specifies that pic_deblocking_filter_override_flag is not present in the PH. When pic_deblocking_filter_override_present_flag is not present, it is inferred to be equal to 0.	
pic_deblocking_filter_override_flag equal to 1 specifies that deblocking parameters are present in the PH.	
pic_deblocking_filter_override_flag equal to 0 specifies that deblocking parameters are not present in the PH. When not present, the value of pic_pic_deblocking_filter_override_flag is inferred to be equal to 0.	40

【 0 1 3 5 】

【表 6 – 2】

pic_deblocking_filter_disabled_flag equal to 1 specifies that the operation of the deblocking filter is not applied for the slices associated with the PH. pic_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for the slices associated with the PH. When pic_deblocking_filter_disabled_flag is not present, it is inferred to be equal to pps_deblocking_filter_disabled_flag.	
pic_beta_offset_div2 and pic_tc_offset_div2 specify the deblocking parameter offsets for β and tC (divided by 2) for the slices associated with the PH. The values of pic_beta_offset_div2 and pic_tc_offset_div2 shall both be in the range of -6 to 6, inclusive. When not present, the values of pic_beta_offset_div2 and pic_tc_offset_div2 are inferred to be equal to pps_beta_offset_div2 and pps_tc_offset_div2, respectively.	

10

20

30

40

50

【 0 1 3 6 】

次の表は、本実施形態によるスライスヘッダの例示的なシンタックス及びセマンティクスを示す。

【 0 1 3 7 】

【表 7】

slice_header() {	Descriptor
...	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !pic_sao_enabled_present_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !pic_alf_enabled_present_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(deblocking_filter_override_enabled_flag && !pic_deblocking_filter_override_present_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)
}	
}	
...	
}	

【 0 1 3 8 】

10

20

30

40

50

【表 8】

cu_chroma_qp_offset_enabled_flag equal to 1 specifies that the cu_chroma_qp_offset_flag may be present in the transform unit and palette coding syntax. cu_chroma_qp_offset_enabled_flag equal to 0 specifies that the cu_chroma_qp_offset_flag is not present in the transform unit or palette coding syntax. When not present, the value of cu_chroma_qp_offset_enabled_flag is inferred to be equal to 0. slice_sao_luma_flag equal to 1 specifies that SAO is enabled for the luma component in the current slice; slice_sao_luma_flag equal to 0 specifies that SAO is disabled for the luma component in the current slice. When slice_sao_luma_flag is not present, it is inferred to be equal to pic_sao_luma_enabled_flag. slice_sao_chroma_flag equal to 1 specifies that SAO is enabled for the chroma component in the current slice; slice_sao_chroma_flag equal to 0 specifies that SAO is disabled for the chroma component in the current slice. When slice_sao_chroma_flag is not present, it is inferred to be equal to pic_sao_chroma_enabled_flag. slice_alf_enabled_flag equal to 1 specifies that adaptive loop filter is enabled and may be applied to Y, Cb, or Cr colour component in a slice. slice_alf_enabled_flag equal to 0 specifies that adaptive loop filter is disabled for all colour components in a slice. When not present, the value of slice_alf_enabled_flag is inferred to be equal to pic_alf_enabled_flag. slice_num_alf_aps_ids_luma specifies the number of ALF APSs that the slice refers to. When slice_alf_enabled_flag is equal to 1 and slice_num_alf_aps_ids_luma is not present, the value of slice_num_alf_aps_ids_luma is inferred to be equal to the value of pic_num_alf_aps_ids_luma. slice_alf_aps_id_luma[i] specifies the adaptation_parameter_set_id of the i-th ALF APS that the luma component of the slice refers to. The TemporalId of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_luma[i] shall be less than or equal to the TemporalId of the coded slice NAL unit. When slice_alf_enabled_flag is equal to 1 and slice_alf_aps_id_luma[i] is not present, the value of slice_alf_aps_id_luma[i] is inferred to be equal to the value of pic_alf_aps_id_luma[i]. The value of alf_luma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_luma[i] shall be equal to 1. slice_alf_chroma_idc equal to 0 specifies that the adaptive loop filter is not applied to Cb and Cr colour components. slice_alf_chroma_idc equal to 1 indicates that the adaptive loop filter is applied to the Cb colour component. slice_alf_chroma_idc equal to 2 indicates that the adaptive loop filter is applied to the Cr colour component. slice_alf_chroma_idc equal to 3 indicates that the adaptive loop filter is applied to Cb and Cr colour components. When slice_alf_chroma_idc is not present, it is inferred to be equal to pic_alf_chroma_idc. slice_alf_aps_id_chroma specifies the adaptation_parameter_set_id of the ALF APS that the chroma component of the slice refers to. The TemporalId of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_chroma shall be less than or equal to the TemporalId of the coded slice NAL unit. When slice_alf_enabled_flag is equal to 1 and slice_alf_aps_id_chroma is not present, the value of slice_alf_aps_id_chroma is inferred to be equal to the value of pic_alf_aps_id_chroma. The value of alf_chroma_filter_signal_flag of the APS NAL unit having aps_params_type equal to ALF_APS and adaptation_parameter_set_id equal to slice_alf_aps_id_chroma shall be equal to 1. slice_deblocking_filter_override_flag equal to 1 specifies that deblocking parameters are present in the slice header. slice_deblocking_filter_override_flag equal to 0 specifies that deblocking parameters are not present in the slice header. When not present, the value of slice_deblocking_filter_override_flag is inferred to be equal to pic_deblocking_filter_override_flag. slice_deblocking_filter_disabled_flag equal to 1 specifies that the operation of the deblocking filter is not applied for the current slice. slice_deblocking_filter_disabled_flag equal to 0 specifies that the operation of the deblocking filter is applied for the current slice. When slice_deblocking_filter_disabled_flag is not present, it is inferred to be equal to pic_deblocking_filter_disabled_flag. slice_beta_offset_div2 and slice_tc_offset_div2 specify the deblocking parameter offsets for β and tC (divided by 2) for the current slice. The values of slice_beta_offset_div2 and slice_tc_offset_div2 shall both be in the range of -6 to 6, inclusive. When not present, the values of slice_beta_offset_div2 and slice_tc_offset_div2 are inferred to be equal to pic_beta_offset_div2 and pic_tc_offset_div2, respectively.	10
	20
	30

【0139】

以下では、インループフィルタリングで利用できる仮想境界に対する情報のシグナリングに関して説明する。

【0140】

既存の設計において、仮想境界を横切るループフィルタをディセーブルするために、i) SPSの仮想境界存在フラグ (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) は0に設定され、また、全てのピクチャヘッダに対してPHの仮想境界存在フラグ (ph_loop_filter_across_virtual_boundaries_disabled_present_flag) は存在し、0に設定され、又はii) SPSの仮想境界存在フラグ (sps_loop_filter_across_virtua

10

20

30

40

50

`l_boundaries_disabled_present_flag`)は1に設定され、また、SPSの垂直仮想境界の数に関する情報(`sps_num_vertical_boundaries`)及びSPSの水平仮想境界の数に関する情報(`sps_num_horizontal_boundaries`)はいずれも0に設定され得る。

【0141】

既存の設計においては、前記ii)によって、SPSの仮想境界存在フラグ(`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`)が1に設定され、デコーダが仮想境界の位置に対するシグナリングを予想することにより、デコード手順での問題が引き起こされることがある。

10

【0142】

以下の段落で記述される実施形態は、前述した問題を解決するための解決策を提案することができる。実施形態は独立に適用してもよい。或いは、少なくとも2つ以上の実施形態を組み合わせ適用してもよい。

【0143】

本文書の一実施形態において、仮想境界を示すためのシンタックス要素がSPSに含まれるかどうかは、フラグにより制御され得る。例えば、前記フラグの数は2個であり得る(例えば、SPSの仮想境界可用フラグ(`SPS_virtual_boundaries_enabled_flag`)、SPSの仮想境界存在フラグ(`SPS_virtual_boundaries_present_flag`))。

20

【0144】

本実施形態による一例において、前記SPSの仮想境界可用フラグは、`sps_loop_filter_across_virtual_boundaries_disabled_flag`(又は`sps_virtual_boundaries_enabled_flag`)と称され得る。前記SPSの仮想境界可用フラグは、仮想境界を横切ってループフィルタをディセーブルするための特徴がイネーブルされるかどうかを示すことができる。

【0145】

本実施形態による一例において、前記SPSの仮想境界存在フラグは、`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`(又は`sps_virtual_boundaries_present_flag`)と称され得る。前記SPSの仮想境界存在フラグは、仮想境界のためのシグナリング情報がSPS又はピクチャヘッダ(`picture_header`、PH)に含まれるかどうかを示すことができる。

30

【0146】

本実施形態による一例において、前記SPSの仮想境界可用フラグ(`sps_loop_filter_across_virtual_boundaries_disabled_flag`)が1であり、かつ、前記SPSの仮想境界存在フラグ(`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`)が0である場合、仮想境界を横切るループフィルタをディセーブルするシグナリング情報がピクチャヘッダに含まれ得る。

40

【0147】

本実施形態による一例において、仮想境界の位置に関する情報(例えば、垂直の仮想境界、水平の仮想境界)がSPSに含まれる場合、垂直の仮想境界の数と水平の仮想境界の数との和が0より大きいように制限され得る。

【0148】

本実施形態による一例において、現在ピクチャに対して仮想境界でフィルタディセーブルされるかどうかを示す変数が導出され得る。例えば、前記変数は`VirtualBoundariesDisabledFlag`を含み得る。

50

【0149】

本例示における一つの場合として、前記SPSの仮想境界可用フラグ (`sps_loop_filter_across_virtual_boundaries_disabled_flag`) が1であり、かつ、前記SPSの仮想境界存在フラグ (`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`) が1である場合、`VirtualBoundairesDisabledFlag` は1であり得る。

【0150】

本例示における別の場合として、前記SPSの仮想境界可用フラグ (`sps_loop_filter_across_virtual_boundaries_disabled_flag`) が1であり、前記SPSの仮想境界存在フラグ (`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`) が0であり、かつ、垂直の仮想境界の数に関する情報 (`ex.ph_num_ver_virtual_boundaries`) と水平の仮想境界の数に関する情報 (`ex.ph_num_hor_virtual_boundaries`) との和が0より大きい場合、`VirtualBoundairesDisabledFlag` は1であり得る。

【0151】

本例示におけるその他の場合においては、`VirtualBoundairesDisabledFlag` は0であり得る。

【0152】

次の表は、本実施形態によるSPSの例示的なシンタックスを示す。

【0153】

【表9】

<code>seq_parameter_set_rbsp()</code> {	Descriptor
<code>sps_decoding_parameter_set_id</code>	u(4)
...	
<code>sps_loop_filter_across_virtual_boundaries_disabled_flag</code>	u(1)
if(<code>sps_loop_filter_across_virtual_boundaries_disabled_flag</code>) {	
<code>sps_loop_filter_across_virtual_boundaries_disabled_present_flag</code>	u(1)
if(<code>sps_loop_filter_across_virtual_boundaries_disabled_present_flag</code>) {	
<code>sps_num_ver_virtual_boundaries</code>	u(2)
for(<code>i = 0; i < sps_num_ver_virtual_boundaries; i++</code>)	
<code>sps_virtual_boundaries_pos_x[i]</code>	u(13)
<code>sps_num_hor_virtual_boundaries</code>	u(2)
for(<code>i = 0; i < sps_num_hor_virtual_boundaries; i++</code>)	
<code>sps_virtual_boundaries_pos_y[i]</code>	u(13)
}	
}	
...	
}	

【0154】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【0155】

10

20

30

40

50

【表 1 0】

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

When **gdr_enabled_flag** is equal to 1, **sps_loop_filter_across_virtual_boundaries_disabled_flag** is constraint to be 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is constraint to be 0.

sps_num_ver_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_x[i]** syntax elements that are present in the SPS. When **sps_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0.

sps_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the i-th vertical virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_x[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

sps_num_hor_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_y[i]** syntax elements that are present in the SPS. When **sps_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **sps_num_ver_virtual_boundaries** and **sps_num_hor_virtual_boundaries** shall be greater than 0.

sps_virtual_boundaries_pos_y[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the i-th horizontal virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_y[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

10

20

【 0 1 5 6】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

【 0 1 5 7】

【表 1 1】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag && ! sps_loop_filter_across_virtual_boundaries_disabled_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]	ue(v)
}	
...	
}	

40

【 0 1 5 8】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

50

【 0 1 5 9 】

【表 1 2】

ph_num_ver_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_x[i]** syntax elements that are present in the PH. When **ph_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0.
ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    if( sps_loop_filter_across_virtual_boundaries_disabled_present_flag ||
        ( ph_num_ver_virtual_boundaries + ph_num_ver_virtual_boundaries > 0 ) )
        VirtualBoundariesDisabledFlag = 1
```

10

The parameter **VirtualBoundariesNumVer** is derived as follows:

```
VirtualBoundariesNumVer = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesNumVer =
    sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_num_ver_virtual_boundaries : ph_num_ver_virtual_boundaries
```

(7-43)

ph_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the *i*-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

The location of the vertical virtual boundary **VirtualBoundariesPosX[i]** is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosX[ i ] =
    (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ]) * 8
```

(7-44)

20

The distance between any two vertical virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

The parameter **VirtualBoundariesNumHor** is derived as follows:

```
VirtualBoundariesNumHor = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesNumHor =
    sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_num_hor_virtual_boundaries : ph_num_hor_virtual_boundaries
```

(7-45)

30

ph_virtual_boundaries_pos_y[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the *i*-th horizontal virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

The location of the horizontal virtual boundary **VirtualBoundariesPosY[i]** is derived as follows:

```
VirtualBoundariesPosY[ i ]
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosY[ i ] =
    (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_virtual_boundaries_pos_y[ i ] : ph_virtual_boundaries_pos_y[ i ]) * 8
```

(7-46)

The distance between any two horizontal virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

40

【 0 1 6 0 】

表 9 乃至表 1 2 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (**sequence parameter set**、**SPS**) 及びピクチャヘッダ (**picture header**、**PH**) を含み得る。前記 **SPS** は、仮想境界可用フラグ (**sps_loop_filter_across_virtual_boundaries_disabled_flag**) を含み得る。前記 **SPS** は、前記仮想境界可用フラグに基づいて **SPS** の仮想境界存在フ

50

ラグ (`sps__loop__filter__across__virtual__boundaries__disabled__present__flag`) を含み得る。

【0161】

例えば、前記SPSは、前記仮想境界可用フラグの値が1である場合、前記SPSの仮想境界存在フラグを含み得る。前記SPSは、前記仮想境界可用フラグ及び前記SPSの仮想境界存在フラグに基づいて、SPSの垂直仮想境界の数に関する情報 (`sps__num__ver__virtual__boundaries`)、SPSの垂直仮想境界の位置に関する情報 (`sps__virtual__boundaries__pos__x[i]`)、SPSの水平仮想境界の数に関する情報 (`sps__num__hor__virtual__boundaries`)、SPSの水平仮想境界の位置に関する情報 (`sps__virtual__boundaries__pos__y[i]`) を含み得る。例えば、前記SPSは、前記仮想境界可用フラグの値が1であり、かつ、前記SPSの仮想境界存在フラグの値が1である場合、前記SPSの垂直仮想境界の数に関する情報、前記SPSの垂直仮想境界の位置に関する情報、前記SPSの水平仮想境界の数に関する情報、前記SPSの水平仮想境界の位置に関する情報を含み得る。

10

【0162】

一例において、前記SPSの垂直仮想境界の位置に関する情報の数は、前記SPSの垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記SPSの水平仮想境界の位置に関する情報の数は、前記SPSの水平仮想境界の数に関する情報に基づいて決定されることができる。前記ピクチャヘッダは、前記仮想境界可用フラグ及びSPSの仮想境界存在フラグに基づいて、PHの垂直仮想境界の数に関する情報 (`ph__num__ver__virtual__boundaries`)、PHの垂直仮想境界の位置に関する情報 (`ph__virtual__boundaries__pos__x[i]`)、PHの水平仮想境界の数に関する情報 (`ph__num__hor__virtual__boundaries`)、PHの水平仮想境界の位置に関する情報 (`ph__virtual__boundaries__pos__y[i]`) を含み得る。

20

【0163】

例えば、前記ピクチャヘッダは、前記仮想境界可用フラグの値が1であり、かつ、前記SPSの仮想境界存在フラグの値が0である場合、前記PHの垂直仮想境界の数に関する情報、前記PHの垂直仮想境界の位置に関する情報、前記PHの水平仮想境界の数に関する情報、前記PHの水平仮想境界の位置に関する情報を含み得る。一例において、前記PHの垂直仮想境界の位置に関する情報の数は、前記PHの垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記PHの水平仮想境界の位置に関する情報の数は、前記PHの水平仮想境界の数に関する情報に基づいて決定されることができる。

30

【0164】

本文書の別の実施形態において、SPSを参照するピクチャのヘッダ情報 (ピクチャヘッダ) の各々は、PHの仮想境界存在フラグ `ph__loop__filter__across__virtual__boundaries__disabled__present__flag` (又は `ph__virtual__boundaries__present__flag`) を含み得る。本実施形態も、以前の実施形態のようにSPSの仮想境界可用フラグ (`sps__loop__filter__across__virtual__boundaries__disabled__flag`) 及びSPSの仮想境界存在フラグ (`sps__loop__filter__across__virtual__boundaries__disabled__present__flag`) と共に説明されることができる。

40

【0165】

本実施形態による一例において、前記SPSの仮想境界可用フラグ (`sps__loop__filter__across__virtual__boundaries__disabled__flag`) が1であり、かつ、前記SPSの仮想境界存在フラグ (`sps__loop__filter__across__virtual__boundaries__disabled__present__flag`) が0である場合、SPSを参照するピクチャのヘッ

50

ダ情報（ピクチャヘッダ）の各々は、PHの仮想境界存在フラグ `ph_loop_filter_across_virtual_boundaries_disabled_present_flag`（又は `ph_virtual_boundaries_present_flag`）を含み得る。

【0166】

本実施形態による一例において、仮想境界の位置に関する情報（例えば、垂直の仮想境界、水平の仮想境界）がSPSに含まれる場合、垂直の仮想境界の数と水平の仮想境界の数との和が0より大きいように制限され得る。

【0167】

本実施形態による一例において、現在ピクチャに対して仮想境界でフィルタがディセーブルされるかどうかを示す変数が導出され得る。例えば、前記変数は、`VirtualBoundairesDisabledFlag`を含み得る。

10

【0168】

本例示における一つの場合として、前記SPSの仮想境界可用フラグ（`sps_loop_filter_across_virtual_boundaries_disabled_flag`）が1であり、かつ、前記SPSの仮想境界存在フラグ（`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`）が1である場合、`VirtualBoundairesDisabledFlag`は1であり得る。

【0169】

20

本例示における別の場合として、前記SPSの仮想境界可用フラグ（`sps_loop_filter_across_virtual_boundaries_disabled_flag`）が1であり、前記PHの仮想境界存在フラグ（`ph_loop_filter_across_virtual_boundaries_disabled_present_flag`）が1である場合、`VirtualBoundairesDisabledFlag`は1であり得る。

【0170】

本例示におけるその他の場合においては、`VirtualBoundairesDisabledFlag`は0であり得る。

【0171】

30

次の表は、本実施形態によるSPSの例示的なシンタックスを示す。

【0172】

40

50

【表 1 3】

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
...	
}	

10

20

【 0 1 7 3】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 1 7 4】

【表 1 4】

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0. sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0. sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive. sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0. When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0. sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.
--

30

40

50

【 0 1 7 5 】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

【 0 1 7 6 】

【表 1 5】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag && !sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_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)
}	
}	
...	
}	

【 0 1 7 7 】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 1 7 8 】

10

20

30

40

50

【表 1 6 – 1】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
    sps_loop_filter_across_virtual_boundaries_disabled_present_flag
    || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

10

Alternatively, the following constraint may be specified:

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 0, **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1

ph_num_ver_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_x[i]** syntax elements that are present in the PH. When **ph_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0. The parameter **VirtualBoundariesNumVer** is derived as follows:

20

```
VirtualBoundariesNumVer = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesNumVer =
    sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_num_ver_virtual_boundaries : ph_num_ver_virtual_boundaries (7-43)
```

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

The distance between any two vertical virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

The location of the vertical virtual boundary **VirtualBoundariesPosX[i]** is derived as follows:

30

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesPosX[ i ] =
    (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?
    sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ]) * 8
```

【0 1 7 9】

40

50

【表 1 6 - 2】

ph_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive. The location of the vertical virtual boundary VirtualBoundariesPosX[i] is derived as follows: <pre> VirtualBoundariesPosX[i] = 0 if(sps_loop_filter_across_virtual_boundaries_disabled_flag) VirtualBoundariesPosX[i] = (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ? sps_virtual_boundaries_pos_x[i] : ph_virtual_boundaries_pos_x[i]) * 8 </pre> (7-44)	10
The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples. ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0. The parameter VirtualBoundariesNumHor is derived as follows: <pre> VirtualBoundariesNumHor = 0 if(sps_loop_filter_across_virtual_boundaries_disabled_flag) VirtualBoundariesNumHor = sps_loop_filter_across_virtual_boundaries_disabled_present_flag ? sps_num_hor_virtual_boundaries : ph_num_hor_virtual_boundaries </pre> (7-45)	20
ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive. The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows: <pre> VirtualBoundariesPosY[i] = 0 if(sps_loop_filter_across_virtual_boundaries_disabled_flag) VirtualBoundariesPosY[i] = (sps_loop_filter_across_virtual_boundaries_disabled_present_flag ? sps_virtual_boundaries_pos_y[i] : ph_virtual_boundaries_pos_y[i]) * 8 </pre> (7-46)	30
The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.	40

【0 1 8 0】

表 1 3 乃至表 1 6 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (sequence parameter set、SPS) 及びピクチャヘッダ (picture header、PH) を含み得る。前記 SPS は、仮想境界可用フラグ (sps_loop_filter_across_virtual_boundaries_disabled_flag) を含み得る。前記 SPS は、前記仮想境界可用フラグに基づいて、SPS の仮想境界存在フラグ (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) を含み得る。例えば、前記 SPS は、前記仮想境界可用フラグの値が 1 である場合、前記 SPS の仮想境界存在フラグを含み得る。前記 SPS は、前記仮想境界可用フラグ及び前記 SPS の仮想境界存在フラグに基づいて、SPS の垂直仮想境界の数に関する情報 (sps_num_ver_virtual_boundaries)、SPS の垂直仮想境界の位置に関する情報 (sps_virtual_boundaries_pos_x[i])、SPS の水平仮想境界の数に関する情報 (sps_num_hor_virtual_boundari

es)、SPSの水平仮想境界の位置に関する情報(sps_virtual_boundaries_pos_y[i])を含み得る。

【0181】

例えば、前記SPSは、前記仮想境界可用フラグの値が1であり、かつ、前記SPSの仮想境界存在フラグの値が1である場合、前記SPSの垂直仮想境界の数に関する情報、前記SPSの垂直仮想境界の位置に関する情報、前記SPSの水平仮想境界の数に関する情報、前記SPSの水平仮想境界の位置に関する情報を含み得る。一例において、前記SPSの垂直仮想境界の位置に関する情報の数は、前記SPSの垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記SPSの水平仮想境界の位置に関する情報の数は、前記SPSの水平仮想境界の数に関する情報に基づいて決定されることができる。前記ピクチャヘッダは、前記仮想境界可用フラグ及びSPSの仮想境界存在フラグに基づいて、PHの仮想境界存在フラグを含み得る。

10

【0182】

例えば、前記ピクチャヘッダは、前記仮想境界可用フラグの値が1であり、かつ、前記SPSの仮想境界存在フラグの値が0である場合、前記ピクチャヘッダは前記PHの仮想境界存在フラグを含み得る。前記ピクチャヘッダは、前記PHの仮想境界存在フラグに基づいてPHの垂直仮想境界の数に関する情報(ph_num_ver_virtual_boundaries)、PHの垂直仮想境界の位置に関する情報(ph_virtual_boundaries_pos_x[i])、PHの水平仮想境界の数に関する情報(ph_num_hor_virtual_boundaries)、PHの水平仮想境界の位置に関する情報(ph_virtual_boundaries_pos_y[i])を含み得る。

20

【0183】

例えば、前記ピクチャヘッダは、前記PHの仮想境界存在フラグの値が1である場合、前記PHの垂直仮想境界の数に関する情報、前記PHの垂直仮想境界の位置に関する情報、前記PHの水平仮想境界の数に関する情報、前記PHの水平仮想境界の位置に関する情報を含み得る。一例において、前記PHの垂直仮想境界の位置に関する情報の数は、前記PHの垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記PHの水平仮想境界の位置に関する情報の数は、前記PHの水平仮想境界の数に関する情報に基づいて決定されることができる。

30

【0184】

本文書の別の実施形態において、仮想境界を示すためのシンタックス要素がSPSに含まれるかどうかは、フラグにより制御され得る。例えば、前記フラグの数は2個であり得る(例えば、SPSの仮想境界存在フラグ(SPS_virtual_boundaries_present_flag)、SPS PHの仮想境界存在フラグ(SPS_PH_virtual_boundaries_present_flag))。

【0185】

本実施形態による一例において、前記SPSの仮想境界存在フラグは、sps_loop_filter_across_virtual_boundaries_disabled_present_flag(又はsps_virtual_boundaries_present_flag)と称され得る。前記SPSの仮想境界存在フラグは、仮想境界の情報がSPSに含まれるかどうかを示すことができる。

40

【0186】

本実施形態による一例において、前記SPS PHの仮想境界存在フラグは、sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flagと称され得る。前記SPS PHの仮想境界存在フラグは、仮想境界の情報がピクチャヘッダ(picture_header、PH)に含まれるかどうかを示すことができる。

【0187】

本実施形態による一例において、前記SPSの仮想境界存在フラグ(sps_loop

50

`__filter_across_virtual_boundaries_disabled_present_flag`) が1である場合、前記SPS PHの仮想境界存在フラグ (`sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag`) は存在せず、また、0に推論されるように制限され得る。

【0188】

本実施形態による一例において、前記SPS PHの仮想境界存在フラグ (`sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag`) が1である場合、仮想境界を横切るループフィルタをディセーブルするシグナリング情報がピクチャヘッダに含まれ得る。

10

【0189】

次の表は、本実施形態によるSPSの例示的なシンタックスを示す。

【0190】

【表17】

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code> sps_decoding_parameter_set_id</code>	<code>u(4)</code>
<code> ...</code>	
<code> sps_loop_filter_across_virtual_boundaries_disabled_present_flag</code>	<code>u(1)</code>
<code> if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {</code>	
<code> sps_num_ver_virtual_boundaries</code>	<code>u(2)</code>
<code> for(i = 0; i < sps_num_ver_virtual_boundaries; i++)</code>	
<code> sps_virtual_boundaries_pos_x[i]</code>	<code>u(13)</code>
<code> sps_num_hor_virtual_boundaries</code>	<code>u(2)</code>
<code> for(i = 0; i < sps_num_hor_virtual_boundaries; i++)</code>	
<code> sps_virtual_boundaries_pos_y[i]</code>	<code>u(13)</code>
<code> } else</code>	
<code> sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag</code>	<code>u(1)</code>
<code> ...</code>	
<code> }</code>	

20

30

【0191】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【0192】

40

50

【表 1 8】

sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that virtual boundary locations for disabling in-loop filter operations are present in the SPS. **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that virtual boundary locations for disabling in-loop filter operations are not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

sps_num_ver_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_x[i]** syntax elements that are present in the SPS. When **sps_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0.

sps_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the i-th vertical virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_x[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

sps_num_hor_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_y[i]** syntax elements that are present in the SPS. When **sps_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

When **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **sps_num_ver_virtual_boundaries** and **sps_num_hor_virtual_boundaries** shall be greater than 0.

sps_virtual_boundaries_pos_y[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the i-th horizontal virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_y[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that virtual boundary locations for disabling in-loop filter operations may be present in the picture header of pictures referring to the SPS. **sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that virtual boundary locations for disabling in-loop filter operations are not present in picture header of pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, **sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

When **gdr_enabled_flag** is equal to 1, **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is constrained to be equal to 0 and **sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is constrained to be 1.

10

20

【0 1 9 3】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

30

【0 1 9 4】

【表 1 9】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_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)
}	
}	
...	
}	

40

50

【 0 1 9 5 】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 1 9 6 】

【表 2 0 - 1 】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

$$\begin{aligned} \text{VirtualBoundariesDisabledFlag} = \\ \text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag} \mid \mid \\ \text{sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag} \mid \mid \\ \text{ph_loop_filter_across_virtual_boundaries_disabled_present_flag} \end{aligned} \quad (7-42)$$

ph_num_ver_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_x[i]** syntax elements that are present in the PH. When **ph_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0. . When **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, **ph_num_ver_virtual_boundaries** is constraint to be greater than 0.

The parameter **VirtualBoundariesNumVer** is derived as follows:

$$\begin{aligned} \text{VirtualBoundariesNumVer} = \text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag} ? \\ \text{sps_num_ver_virtual_boundaries} : \\ (\text{sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag} ? \\ \text{ph_num_ver_virtual_boundaries} : 0) \end{aligned} \quad (7-43)$$

ph_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the i-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

The location of the vertical virtual boundary **VirtualBoundariesPosX[i]** is derived as follows:

$$\begin{aligned} \text{VirtualBoundariesPosX}[i] = (\text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag} ? \\ \text{sps_virtual_boundaries_pos_x}[i] : \\ (\text{sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag} ? \\ \text{ph_virtual_boundaries_pos_x}[i] * 8 : 0) \end{aligned} \quad (7-44)$$

The distance between any two vertical virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

【 0 1 9 7 】

10

20

30

40

50

【表 2 0 – 2】

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

When **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **ph_num_ver_virtual_boundaries** and **ph_num_hor_virtual_boundaries** shall be greater than 0.

The parameter **VirtualBoundariesNumHor** is derived as follows:

$$\begin{aligned} \text{VirtualBoundariesNumHor} = & \text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?} \\ & \text{sps_num_hor_virtual_boundaries :} \\ & (\text{sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?} \\ & \text{ph_num_hor_virtual_boundaries : 0}) \end{aligned} \quad (7-45)$$

ph_virtual_boundaries_pos_y[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the *i*-th horizontal virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

The location of the horizontal virtual boundary **VirtualBoundariesPosY[i]** is derived as follows:

$$\begin{aligned} \text{VirtualBoundariesPosY}[i] = & (\text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag ?} \\ & \text{sps_virtual_boundaries_pos_y}[i] : \\ & (\text{sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag ?} \\ & \text{ph_virtual_boundaries_pos_y}[i]) * 8) : 0 \end{aligned} \quad (7-46)$$

The distance between any two horizontal virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples

【0 1 9 8】

表 1 7 乃至表 2 0 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (**sequence parameter set**、**SPS**) 及びピクチャヘッダ (**picture header**、**PH**) を含み得る。前記 **SPS** は、**SPS** の仮想境界存在フラグ (**sps_loop_filter_across_virtual_boundaries_disabled_present_flag**) を含み得る。前記 **SPS** は、前記 **SPS** の仮想境界存在フラグに基づいて、**SPS** の垂直仮想境界の数に関する情報 (**sps_num_ver_virtual_boundaries**)、**SPS** の垂直仮想境界の位置に関する情報 (**sps_virtual_boundaries_pos_x[i]**)、**SPS** の水平仮想境界の数に関する情報 (**sps_num_hor_virtual_boundaries**)、**SPS** の水平仮想境界の位置に関する情報 (**sps_virtual_boundaries_pos_y[i]**) を含み得る。

【0 1 9 9】

例えば、前記 **SPS** は、前記 **SPS** の仮想境界存在フラグの値が 1 である場合、前記 **SPS** の垂直仮想境界の数に関する情報、前記 **SPS** の垂直仮想境界の位置に関する情報、前記 **SPS** の水平仮想境界の数に関する情報、前記 **SPS** の水平仮想境界の位置に関する情報を含み得る。一例において、前記 **SPS** の垂直仮想境界の位置に関する情報の数は、前記 **SPS** の垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記 **SPS** の水平仮想境界の位置に関する情報の数は、前記 **SPS** の水平仮想境界の数に関する情報に基づいて決定されることができ。前記 **SPS** は、前記 **SPS** の仮想境界存在フラグに基づいて **SPS** **PH** の仮想境界存在フラグを含み得る。

【0 2 0 0】

例えば、前記 **SPS** の仮想境界存在フラグの値が 0 である場合、前記 **SPS** は **SPS** **PH** の仮想境界存在フラグを含み得る。前記ピクチャヘッダは、前記 **SPS** **PH** の仮想境界存在フラグに基づいて **PH** の仮想境界存在フラグを含み得る。例えば、前記 **SPS** **PH** の仮想境界存在フラグの値が 1 である場合、前記ピクチャヘッダは前記 **PH** の仮想境

界存在フラグを含み得る。前記ピクチャヘッダは、前記PHの仮想境界存在フラグに基づいて、PHの垂直仮想境界の数に関する情報（`ph_num_ver_virtual_boundaries`）、PHの垂直仮想境界の位置に関する情報（`ph_virtual_boundaries_pos_x[i]`）、PHの水平仮想境界の数に関する情報（`ph_num_hor_virtual_boundaries`）、PHの水平仮想境界の位置に関する情報（`ph_virtual_boundaries_pos_y[i]`）を含み得る。

【0201】

例えば、前記ピクチャヘッダは、前記PHの仮想境界存在フラグの値が1である場合、前記PHの垂直仮想境界の数に関する情報、前記PHの垂直仮想境界の位置に関する情報、前記PHの水平仮想境界の数に関する情報、前記PHの水平仮想境界の位置に関する情報を含み得る。一例において、前記PHの垂直仮想境界の位置に関する情報の数は、前記PHの垂直仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記PHの水平仮想境界の位置に関する情報の数は、前記PHの水平仮想境界の数に関する情報に基づいて決定されることができる。

10

【0202】

本文書の別の実施形態において、漸進的デコードリフレッシュ（`gradual_decoding_refresh`、GDR）が可用である場合（すなわち、`gdr_enabled_flag`の値が1である場合）、ループフィルタが仮想境界でディセーブルされる特徴はイネーブルされ、かつ、仮想境界の情報はピクチャヘッダでシグナリングされ得る（ピクチャヘッダに含まれ得る）。

20

【0203】

本文書の別の実施形態において、仮想境界を横切ってループフィルタをディセーブルする機能がイネーブルされると、仮想境界の位置のシグナリングに関する情報が1つ以上のパラメータセットに含まれ得る。例えば、仮想境界を横切ってループフィルタをディセーブルする機能がイネーブルされると、仮想境界の位置のシグナリングに関する情報がSPSとピクチャヘッダに含まれ得る。

【0204】

本実施形態において、SPSの仮想境界可用フラグ（`sps_loop_filter_across_virtual_boundaries_disabled_flag`）が1であり、かつ、仮想境界の位置に関するシグナリング情報が1つ以上のパラメータセットに含まれる場合、次の通りである。

30

【0205】

a) 仮想境界の位置に関するシグナリング情報がSPSにのみ含まれるか、ピクチャヘッダにのみ含まれるか、又はSPSとピクチャヘッダの両方に含まれ得る。

【0206】

b) 各ピクチャに対する`VirtualBoundariesDisabledFlag`の導出は次の通りである。

【0207】

－ `sps_loop_filter_across_virtual_boundaries_disabled_flag`が0であると、`VirtualBoundariesDisabledFlag`は0に設定され得る。

40

【0208】

－ 本例示における別の場合において、仮想境界の位置に関する情報がSPS又はピクチャと関連したピクチャヘッダの全てでシグナリングされない場合、`VirtualBoundariesDisabledFlag`は0に設定され得る。

【0209】

－ 本例示におけるその他の場合において（仮想境界の位置がSPSでのみシグナリングされるか、ピクチャヘッダでのみシグナリングされるか、又はSPSとピクチャヘッダの両方でシグナリングされる場合に）、`VirtualBoundariesDisab`

50

`LeadFlag` は 1 に設定され得る。

【0210】

c) ピクチャに適用される仮想境界は、ピクチャが直接に又は間接に参照するパラメータセットでシグナリングされる仮想境界の集合 (`union`) を含み得る。例えば、前記仮想境界は `SPS` でシグナリングされる仮想境界を含み得る (存在する場合)。例えば、前記仮想境界はピクチャと関連したピクチャヘッダでシグナリングされる仮想境界を含み得る (存在する場合)。

【0211】

d) ピクチャ当たりの仮想境界の最大数や事前に定義された (`pre-defined`) 値を超えないような制限が適用され得る。例えば、前記事前に定義された値は 8 であり得る。

【0212】

e) ピクチャヘッダにシグナリングされた仮想境界の位置に関する情報は (ある場合)、他のパラメータセット (例えば、`SPS` 又は `PPS`) に含まれた仮想境界の位置に関する情報と同じではないように制限され得る。

【0213】

ー 代案として、現在ピクチャに対して適用されたある仮想境界の位置に対して、前記仮想境界の位置 (例えば、`SPS` とピクチャと関連したピクチャヘッダにシグナリングされる同じ仮想境界の位置) は、2 つの互いに異なるパラメータセットに含まれ得る。

【0214】

f) 前記 `SPS` の仮想境界存在フラグ (`sps_loop_filter_across_virtual_boundaries_disabled_present_flag`) が 1 である場合、前記 `SPS` PH の仮想境界存在フラグ (`sps_ph_loop_filter_across_virtual_boundaries_disabled_present_flag`) は存在せず、また、0 に推論されるように制限され得る。

【0215】

次の表は、本実施形態による `SPS` の例示的なシンタックスを示す。

【0216】

【表 21】

<code>seq_parameter_set_rbsp()</code> {	Descriptor
<code>sps_decoding_parameter_set_id</code>	<code>u(4)</code>
...	
<code>sps_loop_filter_across_virtual_boundaries_disabled_flag</code>	<code>u(1)</code>
<code>if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {</code>	
<code>sps_loop_filter_across_virtual_boundaries_disabled_present_flag</code>	<code>u(1)</code>
<code>if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {</code>	
<code>sps_num_ver_virtual_boundaries</code>	<code>u(2)</code>
<code>for(i = 0; i < sps_num_ver_virtual_boundaries; i++)</code>	
<code>sps_virtual_boundaries_pos_x[i]</code>	<code>u(13)</code>
<code>sps_num_hor_virtual_boundaries</code>	<code>u(2)</code>
<code>for(i = 0; i < sps_num_hor_virtual_boundaries; i++)</code>	
<code>sps_virtual_boundaries_pos_y[i]</code>	<code>u(13)</code>
<code>}</code>	
<code>}</code>	
...	
<code>}</code>	

【0217】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 2 1 8 】

【表 2 2】

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.	10
sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0.	20
sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0. sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive. sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.	20
When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0. sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.	

【 0 2 1 9 】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

【 0 2 2 0 】

30

40

50

【表 2 3】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_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)
}	
}	
...	
}	

10

【 0 2 2 1】

20

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 2 2 2】

30

40

50

【表 2 4 – 1】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
        sps_loop_filter_across_virtual_boundaries_disabled_present_flag
        || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

10

Alternatively, the following constraint may be specified:

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 0, **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1

ph_num_ver_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_x[i]** syntax elements that are present in the PH. When **ph_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0. The parameter **VirtualBoundariesNumVer** is derived as follows:

20

```
VirtualBoundariesNumVer = sps_num_ver_virtual_boundaries +
    ph_num_ver_virtual_boundaries (7-43)
```

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

The distance between any two vertical virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

ph_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the *i*-th vertical virtual boundary in units of luma samples.
ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

30

The list **VirtualBoundariesPosX[i]**, for *I* ranges from 0 to **VirtualBoundariesNumVer** – 1, inclusive, is derived as follows:

```
for( i = 0; i < VirtualBoundariesNumVer; i++ ) {
    if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
        VirtualBoundariesPosX[ i ] = ( i < sps_num_ver_virtual_boundaries ) ?
            sps_virtual_boundaries_pos_x[ i ] :
            ph_virtual_boundaries_pos_x[ i - sps_num_ver_virtual_boundaries ]
        VirtualBoundariesPosX[ i ] *= 8
    } else
        VirtualBoundariesPosX[ i ] = 0
}
```

40

【 0 2 2 3】

【表 2 4 - 2】

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples. ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0. The parameter VirtualBoundariesNumHor is derived as follows: $\text{VirtualBoundariesNumHor} = \text{sps_num_hor_virtual_boundaries} + \text{ph_num_hor_virtual_boundaries} \quad (7-45)$	
ph_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive. The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows: The list VirtualBoundariesPosY[i], for i ranges from 0 to VirtualBoundariesNumHor - 1, inclusive, is derived as follows: <pre> for(i = 0; i < VirtualBoundariesNumHor; i++) { if(sps_loop_filter_across_virtual_boundaries_disabled_flag) { VirtualBoundariesPosY[i] = (i < sps_num_hor_virtual_boundaries) ? sps_virtual_boundaries_pos_y[i] : ph_virtual_boundaries_pos_y[i - sps_num_hor_virtual_boundaries] VirtualBoundariesPosY[i] *= 8 } else VirtualBoundariesPosY[i] = 0 } </pre>	10 20
The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.	

【 0 2 2 4 】

表 2 1 乃至表 2 4 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (sequence parameter set、SPS) 及びピクチャヘッダ (picture header、PH) を含み得る。前記 SPS は仮想境界可用フラグ (sps_loop_filter_across_virtual_boundaries_disabled_flag) を含み得る。前記 SPS は、前記仮想境界可用フラグに基づいて、SPS の仮想境界存在フラグ (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) を含み得る。例えば、前記 SPS は、前記仮想境界可用フラグの値が 1 である場合、前記 SPS の仮想境界存在フラグを含み得る。前記 SPS は、前記仮想境界可用フラグ及び前記 SPS の仮想境界存在フラグに基づいて、SPS の垂直仮想境界の数に関する情報 (sps_num_ver_virtual_boundaries)、SPS の垂直仮想境界の位置に関する情報 (sps_virtual_boundaries_pos_x[i])、SPS の水平仮想境界の数に関する情報 (sps_num_hor_virtual_boundaries)、SPS の水平仮想境界の位置に関する情報 (sps_virtual_boundaries_pos_y[i]) を含み得る。

【 0 2 2 5 】

例えば、前記 SPS は、前記仮想境界可用フラグの値が 1 であり、かつ、前記 SPS の仮想境界存在フラグの値が 1 である場合、前記 SPS の垂直仮想境界の数に関する情報、前記 SPS の垂直仮想境界の位置に関する情報、前記 SPS の水平仮想境界の数に関する

【 0 2 2 6 】

20

本文書の別の実施形態においては、前述した実施形態によるが、垂直仮想境界の数と水平仮想境界の数との和が0より大きいという制限をしないことによって、ループフィルタリングが実行できる。

30

40

50

【表 2 5】

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
...	
}	

10

【 0 2 3 2 】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

20

【 0 2 3 3 】

【表 2 6】

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When gdr_enabled_flag is equal to 1, sps_loop_filter_across_virtual_boundaries_disabled_flag is constraint to be 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is constraint to be 0. sps_num_ver_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_x[i] syntax elements that are present in the SPS. When sps_num_ver_virtual_boundaries is not present, it is inferred to be equal to 0. sps_virtual_boundaries_pos_x[i] is used to compute the value of VirtualBoundariesPosX[i], which specifies the location of the i-th vertical virtual boundary in units of luma samples. sps_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive. sps_num_hor_virtual_boundaries specifies the number of sps_virtual_boundaries_pos_y[i] syntax elements that are present in the SPS. When sps_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0. When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and sps_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries and sps_num_hor_virtual_boundaries shall be greater than 0. sps_virtual_boundaries_pos_y[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples. sps_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.	
--	--

30

40

【 0 2 3 4 】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

50

【 0 2 3 5 】

【表 2 7】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
for(i = 0; i < sps_num_ver_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_x_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_x_delta[i] > 0)	
ph_virtual_boundaries_pos_x_sign[i]	u(1)
}	
for(i = 0; i < sps_num_hor_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_y_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_y_delta[i] > 0)	
ph_virtual_boundaries_pos_y_sign[i]	u(1)
}	
}	
}	
}	
...	
}	

10

20

【 0 2 3 6 】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 2 3 7 】

30

40

50

【表 2 8 – 1】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

$$\text{VirtualBoundariesDisabledFlag} = (\text{sps_num_ver_virtual_boundaries} + \text{sps_num_hor_virtual_boundaries} > 0) ? 1 : 0$$

10

ph_virtual_boundaries_pos_x_delta[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the i-th vertical virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

ph_virtual_boundaries_pos_x_sign[i] specifies the sign of the i-th virtual boundary, which specifies the location of the i-th vertical virtual boundary in units of luma samples. When not present, the value of **ph_virtual_boundaries_pos_x_sign[i]** is inferred to be equal to 0.

The variable **phVirtualBoundariesPosX[i]** for i ranges from 0 to **sps_num_ver_virtual_boundaries** – 1, inclusive, is initialized as follows:

$$\text{phVirtualBoundariesPosX}[i] = \text{ph_virtual_boundaries_pos_x_delta}[i] * (1 - 2 * \text{ph_virtual_boundaries_pos_x_sign}[i])$$

20

The location of the vertical virtual boundary **VirtualBoundariesPosX[i]** is derived as follows:

$$\begin{aligned} &\text{VirtualBoundariesPosX}[i] = 0 \\ &\text{if}(\text{sps_loop_filter_across_virtual_boundaries_disabled_flag}) \{ \\ &\quad \text{if}(\text{ph_loop_filter_across_virtual_boundaries_disabled_present_flag}) \\ &\quad \quad \text{VirtualBoundariesPosX}[i] = (\text{sps_virtual_boundaries_pos_x}[i] + \text{phVirtualBoundariesPosX}[i]) * 8 \\ &\quad \text{else} \\ &\quad \quad \text{VirtualBoundariesPosX}[i] = (\text{sps_virtual_boundaries_pos_x}[i]) * 8 \\ &\quad \} \end{aligned} \quad (7-44)$$

The distance between any two vertical virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples.

30

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

ph_virtual_boundaries_pos_y_delta[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the i-th horizontal virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

The location of the horizontal virtual boundary **VirtualBoundariesPosY[i]** is derived as follows:

【0 2 3 8】

40

【表 2 8 - 2】

ph_virtual_boundaries_pos_y_sign [i] specifies the sign of the i-th virtual boundary, which specifies the location of the i-th horizontal virtual boundary in units of luma samples. When not present, the value of **ph_virtual_boundaries_pos_y_sign**[i] is inferred to be equal to 0.
The variable **phVirtualBoundariesPosY**[i] for i ranges from 0 to **sps_num_hor_virtual_boundaries** - 1, inclusive, is initialized as follows:

```

phVirtualBoundariesPosY[ i ] = ph_virtual_boundaries_pos_y_delta[ i ] *
    ( 1 - 2 * ph_virtual_boundaries_pos_y_sign [ i ] )

VirtualBoundariesPosY[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
    if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag ) {
        VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] +
            phVirtualBoundariesPosY[ i ] ) * 8
    }
    else
        VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_pos_y[ i ] ) * 8
}

```

(7-46)

The distance between any two horizontal virtual boundaries shall be greater than or equal to **CtbSizeY** luma samples

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **ph_num_ver_virtual_boundaries** and **ph_num_hor_virtual_boundaries** shall be greater than 0.

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **sps_num_ver_virtual_boundaries**, **sps_num_hor_virtual_boundaries**, **ph_num_ver_virtual_boundaries** and **ph_num_hor_virtual_boundaries** shall not be greater than 8.

【0 2 3 9】

表 2 5 乃至表 2 8 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (**sequence parameter set**、**SPS**) 及びピクチャヘッダ (**picture header**、**PH**) を含み得る。前記 **SPS** は、仮想境界可用フラグ (**sps_loop_filter_across_virtual_boundaries_disabled_flag**) を含み得る。前記 **SPS** は、前記仮想境界可用フラグに基づいて、**SPS** の垂直仮想境界の数に関する情報 (**sps_num_ver_virtual_boundaries**)、**SPS** の垂直仮想境界の位置に関する情報 (**sps_virtual_boundaries_pos_x** [i])、**SPS** の水平仮想境界の数に関する情報 (**sps_num_hor_virtual_boundaries**)、**SPS** の水平仮想境界の位置に関する情報 (**sps_virtual_boundaries_pos_y** [i]) を含み得る。例えば、前記 **SPS** は、前記仮想境界可用フラグの値が 1 である場合、前記 **SPS** の水平仮想境界の数に関する情報、前記 **SPS** の水平仮想境界の位置に関する情報、前記 **SPS** の垂直仮想境界の数に関する情報、前記 **SPS** の垂直仮想境界の位置に関する情報を含み得る。

【0 2 4 0】

一例において、前記 **SPS** の水平仮想境界の位置に関する情報の数は、前記 **SPS** の水平仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記 **SPS** の垂直仮想境界の位置に関する情報の数は、前記 **SPS** の垂直仮想境界の数に関する情報に基づいて決定されることができる。前記ピクチャヘッダは、前記仮想境界可用フラグに基づいて **PH** の仮想境界存在フラグを含み得る。例えば、前記仮想境界可用フラグの値が 1 である場合、前記ピクチャヘッダは、前記 **PH** の仮想境界存在フラグを含み得る。前記ピクチャヘッダは、前記 **PH** の仮想境界存在フラグに基づいて、**PH** の水平仮想境界の位置のデルタ値に関する情報 (**ph_virtual_boundaries_pos_x_delta**

l t a [i]) 、 P H の水平仮想境界の位置の符号に関する情報 (p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ x _ s i g n [i]) 、 P H の垂直仮想境界の位置のデルタ値に関する情報 (p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ y _ d e l t a [i]) 、 P H の垂直仮想境界の位置の符号に関する情報 (p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ y _ s i g n [i]) を含み得る。

【 0 2 4 1 】

例えば、前記ピクチャヘッダは、前記 P H の仮想境界存在フラグの値が 1 である場合、前記 P H の垂直仮想境界の位置のデルタ値に関する情報、前記 P H の垂直仮想境界の位置の符号に関する情報、前記 P H の水平仮想境界の位置のデルタ値に関する情報、前記 P H の水平仮想境界の位置の符号に関する情報を含み得る。一例において、前記 S P S の垂直仮想境界の数に関する情報に基づいて、前記 P H の垂直仮想境界の位置のデルタ値に関する情報の数及び前記 P H の垂直仮想境界の位置の符号に関する情報の数が決定されることができ、かつ、前記 S P S の水平仮想境界の数に関する情報に基づいて、前記 P H の水平仮想境界の位置のデルタ値に関する情報の数及び前記 P H の水平仮想境界の位置の符号に関する情報の数が決定できる。

【 0 2 4 2 】

本文書のまた別の実施形態においては、各ピクチャに対する仮想境界の位置に関する情報のシグナリングを説明する。一例において、仮想境界の位置に関する情報が S P S に含まれ、仮想境界の位置のデルタ値に関する情報がピクチャヘッダに含まれない場合、ループフィルタリングのために S P S に含まれた仮想境界の位置に関する情報が用いられ得る。仮想境界の位置に関する情報が S P S に含まれず、仮想境界の位置のデルタ値に関する情報がピクチャヘッダに含まれる場合、ループフィルタリングのためにピクチャヘッダに含まれた仮想境界の位置に関する情報が用いられ得る。仮想境界の位置に関する情報が S P S に含まれ、仮想境界の位置のデルタ値に関する情報がピクチャヘッダに含まれる場合、S P S でシグナリングされた仮想境界の位置に関する情報と、これと関連したデルタ値間の合算に基づいて仮想境界の位置が導出され得る。仮想境界の位置に関する情報が S P S に含まれず、仮想境界の位置のデルタ値に関する情報がピクチャヘッダに含まれない場合、ピクチャに対して仮想境界が適用されないことがある。

【 0 2 4 3 】

次の表は、本実施形態による S P S の例示的なシンタックスを示す。

【 0 2 4 4 】

10

20

30

40

50

【表 2 9】

seq_parameter_set_rbsp() {	Descriptor
sps_decoding_parameter_set_id	u(4)
...	
sps_loop_filter_across_virtual_boundaries_disabled_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
sps_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
...	
}	

10

20

【 0 2 4 5】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 2 4 6】

【表 3 0】

sps_loop_filter_across_virtual_boundaries_disabled_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures referring to the SPS. **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures referring to the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

sps_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the syntax elements for the in-loop filtering operations for the virtual boundaries is present in the SPS. **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** equal to 0 specifies that the syntax elements for the virtual boundaries in-loop filtering operations is not present in the SPS. In-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations.

When **gdr_enabled_flag** is equal to 1, **sps_loop_filter_across_virtual_boundaries_disabled_flag** is constraint to be 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is constraint to be 0.

sps_num_ver_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_x[i]** syntax elements that are present in the SPS. When **sps_num_ver_virtual_boundaries** is not present, it is inferred to be equal to 0.

sps_virtual_boundaries_pos_x[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the i-th vertical virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_x[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

sps_num_hor_virtual_boundaries specifies the number of **sps_virtual_boundaries_pos_y[i]** syntax elements that are present in the SPS. When **sps_num_hor_virtual_boundaries** is not present, it is inferred to be equal to 0.

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1, the sum of **sps_num_ver_virtual_boundaries** and **sps_num_hor_virtual_boundaries** shall be greater than 0.

sps_virtual_boundaries_pos_y[i] is used to compute the value of **VirtualBoundariesPosY[i]**, which specifies the location of the i-th horizontal virtual boundary in units of luma samples. **sps_virtual_boundaries_pos_y[i]** shall be in the range of 1 to $\text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$, inclusive.

30

40

50

【 0 2 4 7 】

次の表は、本実施形態によるヘッダ情報（ピクチャヘッダ）の例示的なシンタックスを示す。

【 0 2 4 8 】

【表 3 1】

picture_header_rbsp() {	Descriptor
non_reference_picture_flag	u(1)
...	
if(sps_loop_filter_across_virtual_boundaries_disabled_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
if(!sps_num_ver_virtual_boundaries)	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_x_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_x_delta[i] > 0)	
ph_virtual_boundaries_pos_x_sign[i]	u(1)
}	
if(!sps_num_hor_virtual_boundaries)	
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++) {	
ph_virtual_boundaries_pos_y_delta[i]	ue(v)
if(ph_virtual_boundaries_pos_y_delta[i] > 0)	
ph_virtual_boundaries_pos_y_sign[i]	u(1)
}	
}	
}	
}	
...	
}	

【 0 2 4 9 】

次の表は、前記シンタックスに含まれたシンタックス要素に関する例示的なセマンティクスを示す。

【 0 2 5 0 】

10

20

30

40

50

【表 3 2 – 1】

ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 1 specifies that the in-loop filtering operations are disabled across the virtual boundaries in pictures associated to the PH.
ph_loop_filter_across_virtual_boundaries_disabled_present_flag equal to 0 specifies that no such disabling of in-loop filtering operations is applied in pictures associated to the PH. The in-loop filtering operations include the deblocking filter, sample adaptive offset filter, and adaptive loop filter operations. When not present, the value of **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is inferred to be equal to 0.

The parameter **VirtualBoundariesDisabledFlag** is derived as follows:

```
VirtualBoundariesDisabledFlag = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag )
    VirtualBoundariesDisabledFlag =
        sps_loop_filter_across_virtual_boundaries_disabled_present_flag
        || ph_loop_filter_across_virtual_boundaries_disabled_present_flag
```

10

Alternatively, the following constraint may be specified:

When **sps_loop_filter_across_virtual_boundaries_disabled_flag** is equal to 1 and **sps_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 0, **ph_loop_filter_across_virtual_boundaries_disabled_present_flag** is equal to 1

ph_num_ver_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_x_delta[i]** syntax elements that are present in the PH. When **ph_num_ver_virtual_boundaries** is not present, it is inferred to be equal to **sps_num_ver_virtual_boundaries**.

ph_num_hor_virtual_boundaries specifies the number of **ph_virtual_boundaries_pos_y_delta[i]** syntax elements that are present in the PH. When **ph_num_hor_virtual_boundaries** is not present, it is inferred to be equal to **sps_num_hot_virtual_boundaries**.

20

ph_virtual_boundaries_pos_x_delta[i] is used to compute the value of **VirtualBoundariesPosX[i]**, which specifies the location of the *i*-th vertical virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_x[i] shall be in the range of 1 to $\text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$, inclusive.

ph_virtual_boundaries_pos_x_sign[i] specifies the sign of the *i*-th virtual boundary, which specifies the location of the *i* th vertical virtual boundary in units of luma samples. When not present, the value of **ph_virtual_boundaries_pos_x_sign[i]** is inferred to be equal to 0.

The variable **phVirtualBoundariesPosX[i]** for *i* ranges from 0 to **sps_num_ver_virtual_boundaries** – 1, inclusive, is initialized as follows:

```
phVirtualBoundariesPosX[ i ] = ph_virtual_boundaries_pos_x_delta[ i ] *
    ( 1 - 2 * ph_virtual_boundaries_pos_x_sign[ i ] )
```

30

The location of the vertical virtual boundary **VirtualBoundariesPosX[i]** is derived as follows:

```
VirtualBoundariesPosX[ i ] = 0
if( sps_loop_filter_across_virtual_boundaries_disabled_flag ) {
    if( sps_loop_filter_across_virtual_boundaries_disabled_present_flag ) {
        if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )
            VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] +
                phVirtualBoundariesPosC[ i ] ) * 8
        else
            VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_pos_x[ i ] ) * 8
    } else if( ph_loop_filter_across_virtual_boundaries_disabled_present_flag )
        VirtualBoundariesPosX[ i ] = phVirtualBoundariesPosX[ i ] * 8
}
```

40

【 0 2 5 1 】

【表 3 2 - 2】

The distance between any two vertical virtual boundaries shall be greater than or equal to CtbSizeY luma samples.

ph_num_hor_virtual_boundaries specifies the number of ph_virtual_boundaries_pos_y[i] syntax elements that are present in the PH. When ph_num_hor_virtual_boundaries is not present, it is inferred to be equal to 0.

ph_virtual_boundaries_pos_y_delta[i] is used to compute the value of VirtualBoundariesPosY[i], which specifies the location of the i-th horizontal virtual boundary in units of luma samples.

ph_virtual_boundaries_pos_y[i] shall be in the range of 1 to Ceil(pic_height_in_luma_samples ÷ 8) - 1, inclusive.

The location of the horizontal virtual boundary VirtualBoundariesPosY[i] is derived as follows:

ph_virtual_boundaries_pos_y_sign[i] specifies the sign of the i-th virtual boundary, which specifies the location of the i-th horizontal virtual boundary in units of luma samples. When not present, the value of ph_virtual_boundaries_pos_y_sign[i] is inferred to be equal to 0.

The variable phVirtualBoundariesPosY[i] for i ranges from 0 to sps_num_hor_virtual_boundaries - 1, inclusive, is initialized as follows:

$$\begin{aligned} \text{phVirtualBoundariesPosY}[i] &= \text{ph_virtual_boundaries_pos_y_delta}[i] * \\ &\quad (1 - 2 * \text{ph_virtual_boundaries_pos_y_sign}[i]) \\ \\ \text{VirtualBoundariesPosY}[i] &= 0 \\ \text{if}(\text{sps_loop_filter_across_virtual_boundaries_disabled_flag}) \\ \quad \text{if}(\text{sps_loop_filter_across_virtual_boundaries_disabled_present_flag}) \\ \quad \quad \text{if}(\text{ph_loop_filter_across_virtual_boundaries_disabled_present_flag}) \quad (7-46) \\ \quad \quad \quad \text{VirtualBoundariesPosY}[i] &= (\text{sps_virtual_boundaries_pos_y}[i] + \\ &\quad \quad \quad \text{phVirtualBoundariesPosY}[i]) * 8 \\ \quad \quad \text{else} \\ \quad \quad \quad \text{VirtualBoundariesPosY}[i] &= (\text{sps_virtual_boundaries_pos_y}[i]) * 8 \\ \quad \quad \text{else if}(\text{ph_loop_filter_across_virtual_boundaries_disabled_present_flag}) \\ \quad \quad \quad \text{VirtualBoundariesPosY}[i] &= \text{phVirtualBoundariesPosY}[i] * 8 \\ \quad \text{;} \end{aligned}$$

The distance between any two horizontal virtual boundaries shall be greater than or equal to CtbSizeY luma samples

When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall be greater than 0.

When sps_loop_filter_across_virtual_boundaries_disabled_flag is equal to 1 and ph_loop_filter_across_virtual_boundaries_disabled_present_flag is equal to 1, the sum of sps_num_ver_virtual_boundaries, sps_num_hor_virtual_boundaries, ph_num_ver_virtual_boundaries and ph_num_hor_virtual_boundaries shall not be greater than 8.

【0 2 5 2】

表 2 9 乃至表 3 2 と関連した実施形態において、エンコード装置によりエンコードされた画像情報及び／又はエンコード装置からデコード装置に受信されたビットストリームを介して取得された画像情報は、シーケンスパラメータセット (sequence parameter set、SPS) 及びピクチャヘッダ (picture header、PH) を含み得る。

【0 2 5 3】

前記 SPS は、仮想境界可用フラグ (sps_loop_filter_across_virtual_boundaries_disabled_flag) を含み得る。前記 SPS は、前記仮想境界可用フラグに基づいて、SPS の仮想境界存在フラグ (sps_loop_filter_across_virtual_boundaries_disabled_present_flag) を含み得る。例えば、前記 SPS は、前記仮想境界可用フラグの値が 1 である場合、前記 SPS の仮想境界存在フラグを含み得る。前記 SPS は、前記仮想境界可用フラグ及び前記 SPS の仮想境界存在フラグに基づいて、SPS の垂直仮想境界の数に関する情報 (sps_num_ver_virtual_boundaries)、SPS の垂直仮想境界の位置に関する情報 (sps_vir

`t u a l _ b o u n d a r i e s _ p o s _ x [ i ]`)、SPSの水平仮想境界の数に関する情報 (`s p s _ n u m _ h o r _ v i r t u a l _ b o u n d a r i e s`)、SPSの水平仮想境界の位置に関する情報 (`s p s _ v i r t u a l _ b o u n d a r i e s _ p o s _ y [ i ]`) を含み得る。

【0254】

例えば、前記SPSは、前記仮想境界可用フラグの値が1であり、かつ、前記SPSの仮想境界存在フラグの値が1である場合、前記水平仮想境界の数に関する情報、前記水平仮想境界の位置に関する情報、前記垂直仮想境界の数に関する情報、前記垂直仮想境界の位置に関する情報を含み得る。一例において、前記水平仮想境界の位置に関する情報の数は、前記水平仮想境界の数に関する情報に基づいて決定されることができ、かつ、前記垂直仮想境界の位置に関する情報の数は、前記垂直仮想境界の数に関する情報に基づいて決定されることができる。前記ピクチャヘッダは、前記仮想境界可用フラグに基づいてPHの仮想境界存在フラグを含み得る。

【0255】

例えば、前記仮想境界可用フラグの値が1である場合、前記ピクチャヘッダは前記PHの仮想境界存在フラグを含み得る。前記ピクチャヘッダは、前記PHの仮想境界存在フラグ及び前記SPSの垂直仮想境界の数に関する情報に基づいて、PHの垂直仮想境界の数に関する情報 (`p h _ n u m _ v e r _ v i r t u a l _ b o u n d a r i e s`) を含み得る。例えば、前記PHの仮想境界存在フラグの値が1であり、かつ、前記SPSの垂直仮想境界の数に関する情報の値が0である場合、前記ピクチャヘッダは、前記PHの垂直仮想境界の数に関する情報を含み得る。一例において、前記ピクチャヘッダは、前記PHの垂直仮想境界の数に関する情報に基づいてPHの垂直仮想境界の位置のデルタ値に関する情報 (`p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ x _ d e l t a [ i ]`)、PHの垂直仮想境界の位置の符号に関する情報 (`p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ x _ s i g n [ i ]`) を含み得る。一例において、前記PHの垂直仮想境界の数に関する情報に基づいて、前記PHの垂直仮想境界の位置のデルタ値に関する情報の数及び前記PHの垂直仮想境界の位置の符号に関する情報の数が決定できる。前記ピクチャヘッダは、前記PHの仮想境界存在フラグ及び前記SPSの水平仮想境界の数に関する情報に基づいてPHの水平仮想境界の数 (`p h _ n u m _ h o r _ v i r t u a l _ b o u n d a r i e s`) に関する情報を含み得る。

【0256】

例えば、前記PHの仮想境界存在フラグの値が1であり、かつ、前記SPSの水平仮想境界の数に関する情報の値が0である場合、前記ピクチャヘッダは、前記PHの水平仮想境界の数に関する情報を含み得る。一例において、前記ピクチャヘッダは、前記PHの水平仮想境界の数に関する情報に基づいて、PHの水平仮想境界の位置のデルタ値に関する情報 (`p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ y _ d e l t a [ i ]`)、PHの水平仮想境界の位置の符号に関する情報 (`p h _ v i r t u a l _ b o u n d a r i e s _ p o s _ y _ s i g n [ i ]`) を含み得る。一例において、前記PHの水平仮想境界の数に関する情報に基づいて、前記PHの水平仮想境界の位置のデルタ値に関する情報の数及び前記PHの水平仮想境界の位置の符号に関する情報の数が決定できる。

【0257】

前記表と共に、本文書の実施形態によると、仮想境界を横切ってインループフィルタリングを実行するための情報を効率的にシグナリングできる。例えば、仮想境界を横切ってインループフィルタリングが可用であるかどうかと関連した情報のシグナリングに基づいて、インループフィルタリングが実行できる。

【0258】

図7及び図8は、本文書の実施形態によるビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

【0259】

図7で開示された方法は、図2又は図8で開示されたエンコード装置により実行される

10

20

30

40

50

ことができる。具体的に、例えば、図7のS700及びS710は、図8の前記エンコード装置のレジデュアル処理部230により実行されることができ、図7のS720は、図8の前記エンコード装置のフィルタリング部260により実行されることができ、図7のS730は、図8の前記エンコード装置のエントロピーエンコード部240により実行されることができ。また、図7で示していないが、図7で前記エンコード装置の予測部220により予測サンプル又は予測関連情報を導出することができ、前記エンコード装置のエントロピーエンコード部240によりレジデュアル情報又は予測関連情報からビットストリームが生成できる。図7で開示された方法は、本文書で前述した実施形態を含み得る。

【0260】

図7を参照すると、エンコード装置はレジデュアルサンプルを導出することができる（S700）。エンコード装置は現在ブロックに対するレジデュアルサンプルを導出することができ、前記現在ブロックに対するレジデュアルサンプルは、現在ブロックの原本サンプル及び予測サンプルに基づいて導出できる。具体的に、エンコード装置は、予測モードに基づいて前記現在ブロックの予測サンプルを導出することができる。この場合、インター予測又はイントラ予測など、本文書で開示された多様な予測方法が適用できる。前記予測サンプルと原本サンプルに基づいてレジデュアルサンプルを導出することができる。

10

【0261】

エンコード装置は変換係数を導出することができる。エンコード装置は、前記レジデュアルサンプルに対する変換手順に基づいて変換係数を導出することができる。例えば、変換手順は、DCT、DST、GBT、又はCNTの少なくとも1つを含み得る。

20

【0262】

エンコード装置は量子化された変換係数を導出することができる。エンコード装置は、前記変換係数に対する量子化手順に基づいて量子化された変換係数を導出することができる。量子化された変換係数は、係数スキャン順序に基づいて1次元ベクトルの形態を有し得る。

【0263】

エンコード装置はレジデュアル情報を生成することができる（S710）。エンコード装置は、前記現在ブロックに対する前記レジデュアルサンプルに基づいてレジデュアル情報を生成することができる。エンコード装置は、前記量子化された変換係数を示すレジデュアル情報を生成することができる。レジデュアル情報は、指数ゴロム、CAVLC、CABACなどのような多様なエンコード方法を通じて生成できる。

30

【0264】

エンコード装置は復元サンプルを生成することができる。エンコード装置は、前記レジデュアル情報に基づいて復元サンプルを生成することができる。復元サンプルはレジデュアル情報に基づいたレジデュアルサンプルを予測サンプルと加算して生成できる。具体的に、エンコード装置は現在ブロックに対する予測（イントラ又はインター予測）を実行し、原本サンプルと予測から生成された予測サンプルに基づいて復元サンプルを生成することができる。

【0265】

復元サンプルは、復元ルマサンプル及び復元クロマサンプルを含み得る。具体的に、レジデュアルサンプルは、レジデュアルルマサンプル及びレジデュアルクロマサンプルを含み得る。レジデュアルルマサンプルは、原本ルマサンプル及び予測ルマサンプルに基づいて生成できる。レジデュアルクロマサンプルは、原本クロマサンプル及び予測クロマサンプルに基づいて生成できる。エンコード装置は、前記レジデュアルルマサンプルに対する変換係数（ルマ変換係数）及び／又は前記レジデュアルクロマサンプルに対する変換係数（クロマ変換係数）を導出することができる。量子化された変換係数は、量子化されたルマ変換係数及び／又は量子化されたクロマ変換係数を含み得る。

40

【0266】

エンコード装置は、現在ピクチャの復元サンプルのためのインループフィルタリング関連情報を生成することができる（S720）。エンコード装置は、復元サンプルに対する

50

インループフィルタリング手順を実行することができ、前記インループフィルタリング手順に基づいてインループフィルタリング関連情報を生成することができる。例えば、インループフィルタリング関連情報は、本文書で前述した仮想境界に関する情報（SPSの仮想境界可用フラグ、ピクチャヘッダの仮想境界可用フラグ、SPSの仮想境界存在フラグ、ピクチャヘッダの仮想境界存在フラグ、仮想境界の位置に関する情報など）を含み得る。
【0267】

エンコード装置はビデオ／画像情報をエンコードすることができる（S730）。前記画像情報は、レジデュアル情報、予測関連情報、及び／又はインループフィルタリング関連情報を含み得る。エンコードされたビデオ／画像情報は、ビットストリームの形態で出力されることができる。前記ビットストリームはネットワーク又は格納媒体を介してデコード装置に送信されることができる。

10

【0268】

前記画像／ビデオ情報は、本文書の実施形態による多様な情報を含むことができる。例えば、前記画像／ビデオ情報は、前述した表1乃至32の少なくとも1つに開示された情報を含み得る。

【0269】

一実施形態において、前記インループフィルタリング関連情報は、インループフィルタリング手順が仮想境界を横切って実行されるかどうかと関連したSPSの仮想境界可用フラグを含み得る。例えば、SPSの仮想境界可用フラグに基づいて、SPS又はピクチャヘッダ情報に仮想境界関連情報のシグナリングが存在するかどうか決定できる。SPSの仮想境界可用フラグは、仮想境界を横切るインループフィルタリング手順のディセーブルが可用であるかどうかを示すことができる。

20

【0270】

一実施形態において、前記画像情報はSPS（sequence parameter set）を含み、前記SPSは、前記SPSの仮想境界可用フラグ及びSPSの仮想境界存在フラグを含み、かつ、前記SPSの仮想境界存在フラグに基づいて、前記仮想境界の位置に関する情報及び前記仮想境界の数に関する情報が前記SPSに含まれるかどうか決定できる。

【0271】

一実施形態において、前記SPSの仮想境界存在フラグの値が1であることに基づいて、前記SPSは垂直仮想境界の数に関する情報を含み得る。

30

【0272】

一実施形態において、前記SPSは垂直仮想境界の位置に関する情報を含み得る。また、前記垂直仮想境界の数に関する情報に基づいて、前記垂直仮想境界の位置に関する情報の数が決定できる。

【0273】

一実施形態において、前記SPSの仮想境界存在フラグの値が1であることに基づいて、前記SPSは水平仮想境界の数に関する情報を含み得る。

【0274】

一実施形態において、前記SPSは水平仮想境界の位置に関する情報を含み得る。また、前記水平仮想境界の数に関する情報に基づいて、前記水平仮想境界の位置に関する情報の数が決定できる。

40

【0275】

一実施形態において、前記画像情報はピクチャヘッダ情報を含み、並びに、前記SPSの仮想境界可用フラグの値が1であり、及び、前記SPSの仮想境界存在フラグの値が0であることに基づいて、前記ピクチャヘッダ情報はピクチャヘッダの仮想境界存在フラグを含み得る。

【0276】

一実施形態において、前記ピクチャヘッダの仮想境界存在フラグの値が1であることに基づいて、前記ピクチャヘッダ情報は垂直仮想境界の数に関する情報を含み得る。

50

【0277】

一実施形態において、前記ピクチャヘッダ情報は垂直仮想境界の位置に関する情報を含み得る。また、前記垂直仮想境界の数に関する情報に基づいて、前記垂直仮想境界の位置に関する情報の数が決定できる。

【0278】

一実施形態において、前記ピクチャヘッダの仮想境界存在フラグの値が1であることに基づいて、前記ピクチャヘッダ情報は水平仮想境界の数に関する情報を含み得る。

【0279】

一実施形態において、前記ピクチャヘッダ情報は水平仮想境界の位置に関する情報を含み得る。また、前記水平仮想境界の数に関する情報に基づいて、前記水平仮想境界の位置に関する情報の数が決定できる。

10

【0280】

一実施形態において、前記SPSが垂直仮想境界の位置に関する情報と水平仮想境界の位置に関する情報とを含むことに基づいて、前記垂直仮想境界の数と前記水平仮想境界の数との和は0より大きくてもよい。

【0281】

一実施形態において、前記インループフィルタリング関連情報（及び／又は仮想境界関連情報）は、SPSの仮想境界存在フラグ、ピクチャヘッダの仮想境界存在フラグ、及び漸進的デコードリフレッシュ（*gradual decoding refresh*、GDR）可用フラグをさらに含み得る。例えば、前記GDR可用フラグの値が1であることに基づいて、前記SPSの仮想境界可用フラグの値は1であり、前記SPSの仮想境界存在フラグの値は0であり、かつ、前記ピクチャヘッダの仮想境界存在フラグの値は1であり得る（仮想境界情報のシグナリングはピクチャヘッダに存在し得る）。

20

【0282】

図9及び図10は、本文書の実施形態によるビデオ／画像デコード方法及び関連コンポーネントの一例を概略的に示す。

【0283】

図9で開示された方法は、図3又は図10で開示されたデコード装置により実行されることができる。具体的に、例えば、図9のS900は、前記デコード装置のエントロピーデコード部310により実行されることができ、S910は、前記デコード装置のレジデュアル処理部320及び／又は加算部340により実行されることができ、かつ、S920は、前記デコード装置のフィルタリング部350により実行されることができる。図9で開示された方法は、本文書で前述した実施形態を含み得る。

30

【0284】

図9を参照すると、デコード装置はビデオ／画像情報を受信／取得することができる（S900）。ビデオ／画像情報は、レジデュアル情報、予測関連情報、及び／又はインループフィルタリング関連情報を含み得る。デコード装置は、ビットストリームを介して前記画像／ビデオ情報を受信／取得することができる。

【0285】

前記画像／ビデオ情報は、本文書の実施形態による多様な情報を含み得る。例えば、前記画像／ビデオ情報は、前述した表1乃至32の少なくとも1つに開示された情報を含み得る。

40

【0286】

デコード装置は、量子化された変換係数を導出することができる。デコード装置は、前記レジデュアル情報に基づいて量子化された変換係数を導出することができる。量子化された変換係数は、係数スキャン順序に基づいて1次元ベクトルの形態を有し得る。量子化された変換係数は、量子化されたルマ変換係数及び／又は量子化されたクロマ変換係数を含み得る。

【0287】

デコード装置は変換係数を導出することができる。デコード装置は、前記量子化された

50

変換係数に対する逆量子化手順に基づいて変換係数を導出することができる。デコード装置は、量子化されたルマ変換係数に基づいて、逆量子化を介してルマ変換係数を導出することができる。デコード装置は、量子化されたクロマ変換係数に基づいて、逆量子化を介してクロマ変換係数を導出することができる。

【0288】

デコード装置はレジデュアルサンプルを生成／導出することができる。デコード装置は、前記変換係数に対する逆変換手順に基づいて、レジデュアルサンプルを導出することができる。デコード装置は、ルマ変換係数に基づいて、逆変換手順を介してレジデュアルルマサンプルを導出することができる。デコード装置は、クロマ変換係数に基づいて、逆変換手順を介してレジデュアルクロマサンプルを導出することができる。

10

【0289】

デコード装置は復元サンプルを生成／導出することができる（S910）。例えば、デコード装置は復元ルマサンプル及び／又は復元クロマサンプルを生成／導出することができる。デコード装置は、前記レジデュアル情報に基づいて復元ルマサンプル及び／又は復元クロマサンプルを生成することができる。デコード装置は、前記レジデュアル情報に基づいて復元サンプルを生成することができる。復元サンプルは、復元ルマサンプル及び／又は復元クロマサンプルを含み得る。復元サンプルのルマ成分が復元ルマサンプルに対応し、復元サンプルのクロマ成分が復元クロマサンプルに対応し得る。デコード装置は予測手順を介して予測ルマサンプル及び／又は予測クロマサンプルを生成することができる。デコード装置は予測ルマサンプルとレジデュアルルマサンプルに基づいて復元ルマサンプルを生成することができる。デコード装置は、予測クロマサンプルとレジデュアルクロマサンプルに基づいて復元クロマサンプルを生成することができる。

20

【0290】

デコード装置は、修正された（フィルタリングされた）復元サンプルを生成することができる（S920）。デコード装置は、前記復元サンプルに対するインループフィルタリング手順に基づいて、修正された復元サンプルを生成することができる。デコード装置は、インループフィルタリング関連情報に基づいて、修正された復元サンプルを生成することができる。デコード装置は、修正された復元サンプルを生成するために、デブロッキング手順、SAO手順、及び／又はALF手順を用いることができる。

【0291】

一実施形態において、前記画像情報はSPSを含み得る。前記SPSはSPSの仮想境界可用フラグを含み、かつ、前記インループフィルタリング手順は、前記SPSの仮想境界可用フラグに基づいて、仮想境界を横切って実行されることができる（又は仮想境界を横切って実行されないことがある）。例えば、SPSの仮想境界可用フラグに基づいて、SPS又はピクチャヘッダ情報に仮想境界関連情報のシグナリングが存在するかどうか決定できる。SPSの仮想境界可用フラグは、仮想境界を横切るインループフィルタリング手順のディセーブルが可用であるかどうかを示すことができる。

30

【0292】

一実施形態において、前記SPSは、SPSの仮想境界存在フラグをさらに含み得る。前記SPSの仮想境界存在フラグに基づいて、前記仮想境界の位置に関する情報及び前記仮想境界の数に関する情報が前記SPSに含まれるかどうか決定できる。

40

【0293】

一実施形態において、前記SPSの仮想境界存在フラグの値が1であることに基いて、前記SPSは垂直仮想境界の数に関する情報を含み得る。

【0294】

一実施形態において、前記SPSは垂直仮想境界の位置に関する情報を含み得る。また、前記垂直仮想境界の数に関する情報に基づいて、前記垂直仮想境界の位置に関する情報の数が決定できる。

【0295】

一実施形態において、前記SPSの仮想境界存在フラグの値が1であることに基いて

50

、前記 S P S は水平仮想境界の数に関する情報を含み得る。

【0296】

一実施形態において、前記 S P S は水平仮想境界の位置に関する情報を含み得る。また、前記水平仮想境界の数に関する情報に基づいて、前記水平仮想境界の位置に関する情報の数が決定できる。

【0297】

一実施形態において、前記画像情報はピクチャヘッダ情報を含み、並びに、前記 S P S の仮想境界可用フラグの値が 1 であり、及び、前記 S P S の仮想境界存在フラグの値が 0 であることに基づいて、前記ピクチャヘッダ情報はピクチャヘッダの仮想境界存在フラグを含み得る。

【0298】

一実施形態において、前記ピクチャヘッダの仮想境界存在フラグの値が 1 であることに基づいて、前記ピクチャヘッダ情報は垂直仮想境界の数に関する情報を含み得る。

【0299】

一実施形態において、前記ピクチャヘッダ情報は垂直仮想境界の位置に関する情報を含み得る。また、前記垂直仮想境界の数に関する情報に基づいて、前記垂直仮想境界の位置に関する情報の数が決定できる。

【0300】

一実施形態において、前記ピクチャヘッダの仮想境界存在フラグの値が 1 であることに基づいて、前記ピクチャヘッダ情報は水平仮想境界の数に関する情報を含み得る。

【0301】

一実施形態において、前記ピクチャヘッダ情報は水平仮想境界の位置に関する情報を含み得る。また、前記水平仮想境界の数に関する情報に基づいて、前記水平仮想境界の位置に関する情報の数が決定できる。

【0302】

一実施形態において、前記 S P S が垂直仮想境界の位置に関する情報と水平仮想境界の位置に関する情報とを含むことに基づいて、前記垂直仮想境界の数と前記水平仮想境界の数との和は 0 より大きくてもよい。

【0303】

一実施形態において、前記インループフィルタリング関連情報（及び／又は仮想境界関連情報）は、S P S の仮想境界存在フラグ、ピクチャヘッダの仮想境界存在フラグ、及び漸進的デコードリフレッシュ（*gradual decoding refresh*、GDR）可用フラグをさらに含み得る。例えば、前記 GDR 可用フラグの値が 1 であることに基づいて、前記 S P S の仮想境界可用フラグの値は 1 であり、前記 S P S の仮想境界存在フラグの値は 0 であり、かつ、前記ピクチャヘッダの仮想境界存在フラグの値は 1 であり得る（仮想境界情報のシグナリングはピクチャヘッダに存在し得る）。

【0304】

デコード装置は、現在ブロックに対するレジデュアルサンプルが存在する場合、現在ブロックに対するレジデュアルに関する情報を受信することができる。レジデュアルに関する情報は、レジデュアルサンプルに関する変換係数を含むことができる。デコード装置は、レジデュアル情報に基づいて現在ブロックに対するレジデュアルサンプル（又は、レジデュアルサンプルアレイ）を導出することができる。具体的に、デコード装置は、レジデュアル情報に基づいて量子化された変換係数を導出することができる。量子化された変換係数は、係数スキャン順序に基づいて 1 次元ベクトル形態を有することができる。デコード装置は、前記量子化された変換係数に対する逆量子化手順に基づいて変換係数を導出することができる。デコード装置は、変換係数に基づいてレジデュアルサンプルを導出することができる。

【0305】

デコード装置は、（イントラ）予測サンプルとレジデュアルサンプルとに基づいて復元サンプルを生成することができ、前記復元サンプルに基づいて復元ブロック又は復元ピク

10

20

30

40

50

チャを導出することができる。具体的に、デコード装置は、（イントラ）予測サンプルとレジデュアルサンプルとの和に基づいて復元サンプルを生成することができる。以後、デコード装置は、必要によって、主観的／客観的画質を向上させるために、デブロッキングフィルタリング及び／又はSAO手順のようなインループフィルタリング手順を前記復元ピクチャに適用できることは、前述の通りである。

【0306】

例えば、デコード装置は、ビットストリーム又はエンコードされた情報をデコードし、前述した情報（又はシンタックス要素）の全部又は一部を含む画像情報を獲得することができる。また、前記ビットストリーム又はエンコードされた情報は、コンピュータ読み取り可能な格納媒体に格納されることができ、前述したデコード方法が行われるように引き起こすことができる。

10

【0307】

前述した実施形態において、方法は、一連のステップ又はブロックとしてフローチャートに基づいて説明されているが、該当実施形態は、ステップの順序に限定されるわけではなく、あるステップは、前述したところと異なるステップと異なる順序で、又は同時に発生し得る。また、当業者であれば、フローチャートに示されているステップが排他的ではなく、異なるステップが含まれるか、フローチャートの1つ又はそれ以上のステップが本文書の実施形態の範囲に影響を与えずに削除され得ることを理解することができる。

【0308】

前述した本文書の実施形態に係る方法は、ソフトウェアの形態で実現されることができ、本文書に係るエンコード装置及び／又はデコード装置は、例えば、TV、コンピュータ、スマートフォン、セットトップボックス、ディスプレイ装置等の画像処理を行う装置に含まれ得る。

20

【0309】

本文書で、実施形態がソフトウェアで実現される際、前述した方法は、前述した機能を行うモジュール（過程、機能等）で実現されることができる。モジュールはメモリに格納され、プロセッサにより実行されることができる。メモリは、プロセッサの内部又は外部にあってもよく、よく知られている様々な手段でプロセッサと連結されてもよい。プロセッサは、ASIC (application-specific integrated circuit)、他のチップセット、論理回路及び／又はデータ処理装置を含むことができる。メモリは、ROM (read-only memory)、RAM (random access memory)、フラッシュメモリ、メモリカード、格納媒体及び／又は他の格納装置を含むことができる。即ち、本文書で説明した実施形態は、プロセッサ、マイクロプロセッサ、コントローラ又はチップ上で実現されて行われ得る。例えば、各図面で示している機能ユニットは、コンピュータ、プロセッサ、マイクロプロセッサ、コントローラ又はチップ上で実現されて行われ得る。この場合、実現のための情報 (ex. information on instructions) 又はアルゴリズムがデジタル格納媒体に格納されることができる。

30

【0310】

また、本文書の実施形態が適用されるデコード装置及びエンコード装置は、マルチメディア放送受信装置、モバイル通信端末、ホームシネマビデオ装置、デジタルシネマビデオ装置、監視用カメラ、ビデオ対話装置、ビデオ通信のようなリアルタイム通信装置、モバイルストリーミング装置、格納媒体、カムコーダ、オーダーメイド型ビデオ (VoD) サービス提供装置、OTTビデオ (Over the top video) 装置、インターネットストリーミングサービス提供装置、3次元 (3D) ビデオ装置、VR (virtual reality) 装置、AR (argumente reality) 装置、画像電話ビデオ装置、運送手段端末 (ex. 車両 (自律走行車両含む) 端末、飛行機端末、船舶端末など) 及び医療用ビデオ装置等に含まれ得、ビデオ信号又はデータ信号を処理するために使用され得る。例えば、OTTビデオ (Over the top video) 装置としては、ゲームコンソール、ブルーレイプレーヤー、インターネットアクセスTV、

40

50

ホームシアターシステム、スマートフォン、タブレットPC、DVR (Digital Video Recorder) 等を含み得る。

【0311】

また、本文書の実施形態が適用される処理方法は、コンピュータで実行されるプログラムの形態で生産されることができ、コンピュータが読み取ることができる記録媒体に格納されることができる。本文書の実施形態に係るデータ構造を有するマルチメディアデータもまた、コンピュータが読み取ることができる記録媒体に格納されることができる。前記コンピュータが読み取ることができる記録媒体は、コンピュータで読み取られるデータが格納される全ての種類の格納装置及び分散格納装置を含む。前記コンピュータが読み取ることができる記録媒体は、例えば、ブルーレイディスク (BD)、汎用直列バス (USB)、ROM、PROM、EPROM、EEPROM、RAM、CD-ROM、磁気テープ、フロッピディスク、及び光学的データ格納装置を含み得る。また、前記コンピュータが読み取ることができる記録媒体は、搬送波 (例えば、インターネットを介した送信) の形態で実現されたメディアを含む。また、エンコード方法で生成されたビットストリームが、コンピュータが読み取られる記録媒体に格納されるか、有無線通信ネットワークを介して送信されることができる。

10

【0312】

また、本文書の実施形態は、プログラムコードによるコンピュータプログラム製品で実現されることができ、前記プログラムコードは、本文書の実施形態によってコンピュータで実行されることができる。前記プログラムコードは、コンピュータによって読み取り可能なキャリア上に格納されることができる。

20

【0313】

図11は、本文書に開示された実施形態が適用されることができるコンテンツストリーミングシステムの例を示す。

【0314】

図11を参照すると、本文書の実施形態が適用されるコンテンツストリーミングシステムは、大きくエンコードサーバ、ストリーミングサーバ、ウェブサーバ、メディアストレージ、ユーザ装置、及びマルチメディア入力装置を含むことができる。

【0315】

前記エンコードサーバは、スマートフォン、カメラ、カムコーダ等のようなマルチメディア入力装置から入力されたコンテンツをデジタルデータに圧縮してビットストリームを生成し、これを前記ストリーミングサーバに送信する役割をする。別の例として、スマートフォン、カメラ、カムコーダ等のようなマルチメディア入力装置がビットストリームを直接生成する場合、前記エンコードサーバは省略され得る。

30

【0316】

前記ビットストリームは、本文書の実施形態が適用されるエンコード方法又はビットストリームの生成方法により生成されることができ、前記ストリーミングサーバは、前記ビットストリームを送信又は受信する過程で、一時的に前記ビットストリームを格納することができる。

【0317】

前記ストリーミングサーバは、ウェブサーバを介してのユーザ要請に基づいてマルチメディアデータをユーザ装置に送信し、前記ウェブサーバは、ユーザにどのようなサービスがあるかを知らせる媒介体役割をする。ユーザが前記ウェブサーバに所望のサービスを要請すれば、前記ウェブサーバは、これをストリーミングサーバに伝達し、前記ストリーミングサーバは、ユーザにマルチメディアデータを送信する。このとき、前記コンテンツストリーミングシステムは、別の制御サーバを含むことができ、この場合、前記制御サーバは、前記コンテンツストリーミングシステム内の各装置間の命令／応答を制御する役割をする。

40

【0318】

前記ストリーミングサーバは、メディア格納所及び／又はエンコードサーバからコンテ

50

ンツを受信することができる。例えば、前記エンコードサーバからコンテンツを受信するようになる場合、前記コンテンツをリアルタイムに受信することができる。この場合、円滑なストリーミングサービスを提供するために、前記ストリーミングサーバは、前記ビットストリームを一定時間の間、格納することができる。

【0319】

前記ユーザ装置の例では、携帯電話、スマートフォン (smart phone)、ノートブックコンピュータ (laptop computer)、デジタル放送用端末機、PDA (personal digital assistants)、PMP (portable multimedia player)、ナビゲーション、スレートPC (slate PC)、タブレットPC (tablet PC)、ウルトラブック (ultrabook)、ウェアラブルデバイス (wearable device、例えば、ウォッチ型端末機 (smartwatch)、グラス型端末機 (smart glass)、HMD (head mounted display))、デジタルTV、デスクトップコンピュータ、デジタルサイニジなどがありうる。

【0320】

前記コンテンツストリーミングシステム内の各サーバは、分散サーバとして運営されることができ、この場合、各サーバで受信するデータは、分散処理されることができる。

【0321】

本明細書に記載された請求項は、多様な方式で組み合わせることができる。例えば、本明細書の方法請求項の技術的特徴が組み合わせられて装置として実現されることができ、本明細書の装置請求項の技術的特徴が組み合わせられて方法として実現されることができる。また、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴とが組み合わせられて装置として実現されることができ、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴とが組み合わせられて方法として実現されることができる。

10

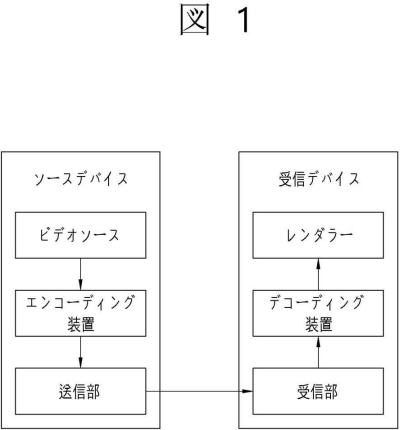
20

30

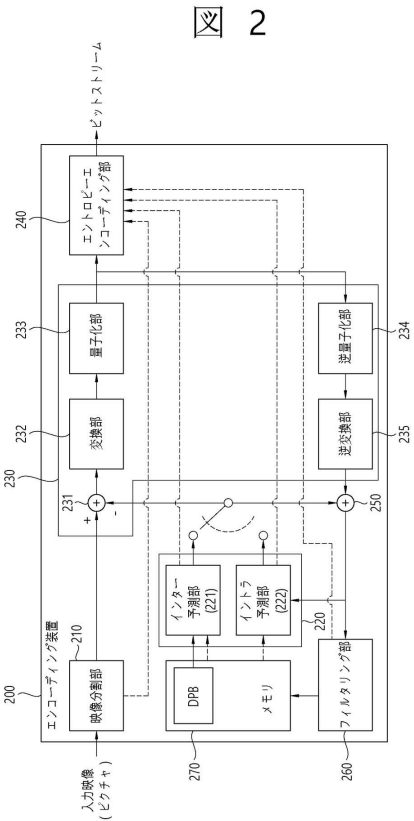
40

50

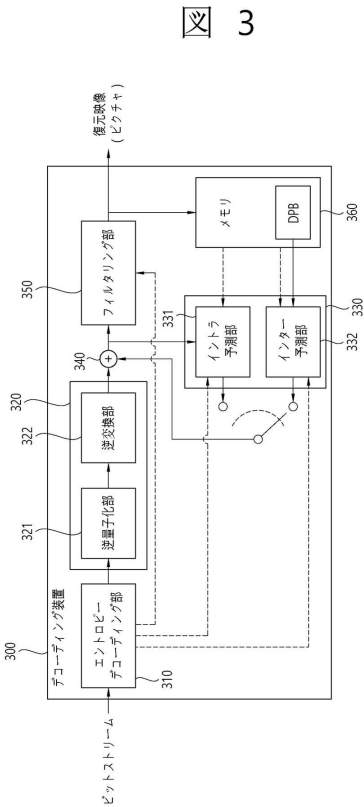
【図面】
【図 1】



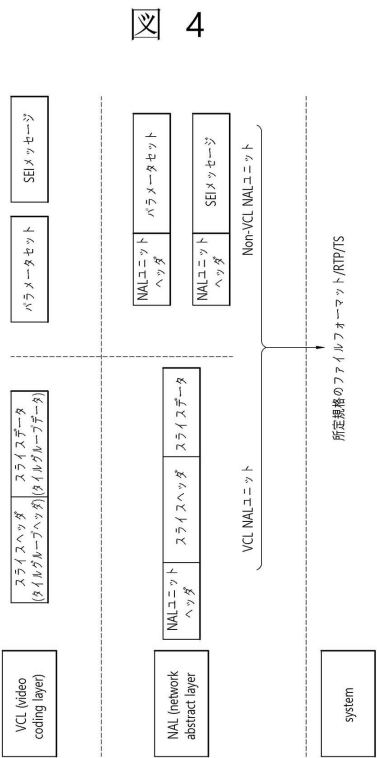
【図 2】



【図 3】



【図 4】



10

20

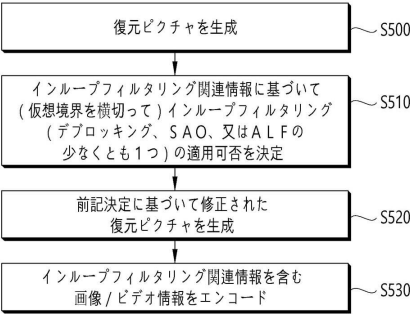
30

40

50

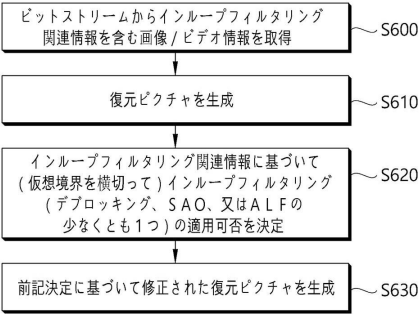
【図 5】

図 5



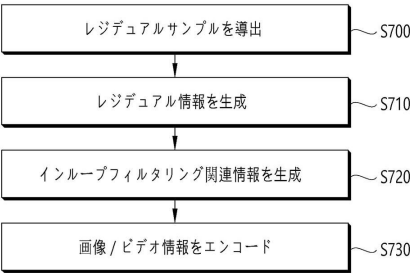
【図 6】

図 6



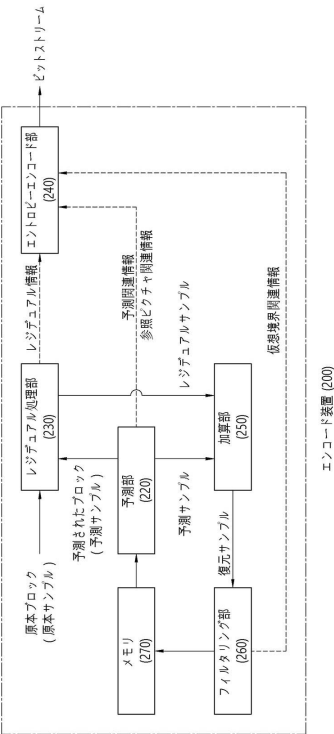
【図 7】

図 7



【図 8】

図 8



10

20

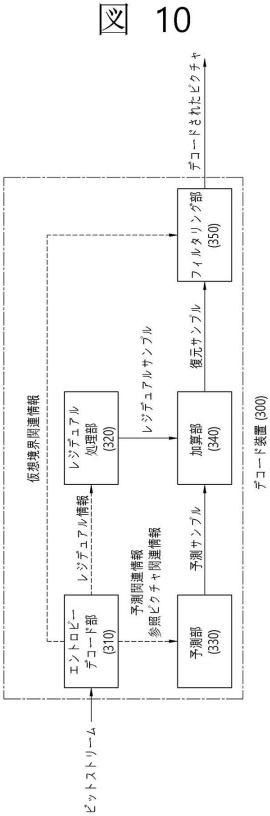
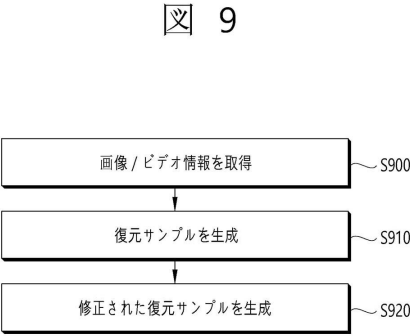
30

40

50

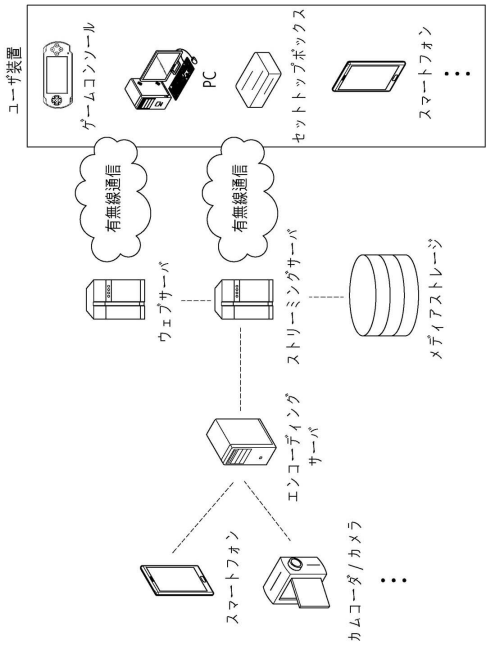
【図 9】

【図 10】



【図 11】

図 11



10

20

30

40

50

フロントページの続き

- 弁理士 河合 章
(74)代理人 100114018
弁理士 南山 知広
(74)代理人 100159259
弁理士 竹本 実
(72)発明者 ヘンドリー ヘンドリー
大韓民国, ソウル 06772, ソチョーク, ヤンジェーデロ 11-ギル, 19, エルジー エレクトロニクス インコーポレイティド, アイピー センター
(72)発明者 シータル パルリ
大韓民国, ソウル 06772, ソチョーク, ヤンジェーデロ 11-ギル, 19, エルジー エレクトロニクス インコーポレイティド, アイピー センター
(72)発明者 キム スンファン
大韓民国, ソウル 06772, ソチョーク, ヤンジェーデロ 11-ギル, 19, エルジー エレクトロニクス インコーポレイティド, アイピー センター
審査官 岩井 健二
(56)参考文献 国際公開第2020/263769 (WO, A1)
国際公開第2020/249123 (WO, A1)
国際公開第2020/011796 (WO, A1)
Benjamin Bross, Jianle Chen, Shan Liu, and Ye-Kui Wang, Versatile Video Coding (Draft 7), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-P2001 (version 14), 16th Meeting: Geneva, CH, 2019年11月14日, pp.38-42,45-48,96-106,113-120
Sheng-Yen Lin, et al., AHG12: Loop filter disabled across virtual boundaries, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N0438-v3, 14th Meeting: Geneva, CH, 2019年03月, pp.1-8
Chih-Yao Chiu, et al., AHG9: On signalling of virtual boundary, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-Q0181-v2, 17th Meeting: Brussels, BE, 2020年01月, pp.1-6
Seethal Paluri, Hendry, and Seung Hwan Kim, [AHG9]: On virtual boundary signalling, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-Q0246-v3, 17th Meeting: Brussels, BE, 2020年01月, pp.1-4
Jonatan Samuelsson, Sachin Deshpande, and Andrew Segall, On virtual boundary signalling, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-Q0258, 17th Meeting: Brussels, BE, 2019年12月, pp.1-2
(58)調査した分野 (Int.Cl., DB名)
H04N 19/00 - 19/98