

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7704491号
(P7704491)

(45)発行日 令和7年7月8日(2025.7.8)

(24)登録日 令和7年6月30日(2025.6.30)

(51)国際特許分類	F I
H 0 4 N 19/513(2014.01)	H 0 4 N 19/513
H 0 4 N 19/117(2014.01)	H 0 4 N 19/117
H 0 4 N 19/176(2014.01)	H 0 4 N 19/176
H 0 4 N 19/70 (2014.01)	H 0 4 N 19/70

請求項の数 22 (全76頁)

(21)出願番号	特願2021-555037(P2021-555037)	(73)特許権者	503433420 華為技術有限公司 HUAWEI TECHNOLOGIES CO., LTD. 中華人民共和國 5 1 8 1 2 9 広東省深 チェン 市龍崗区坂田 華為総部 ベ ン 公樓 Huawei Administrat ion Building, Banti an, Longgang Distri ct, Shenzhen, Guang dong 5 1 8 1 2 9, P. R. C hina
(86)(22)出願日	令和2年3月11日(2020.3.11)	(74)代理人	100107766 弁理士 伊東 忠重
(65)公表番号	特表2022-524627(P2022-524627 A)		
(43)公表日	令和4年5月9日(2022.5.9)		
(86)国際出願番号	PCT/US2020/022083		
(87)国際公開番号	WO2020/185885		
(87)国際公開日	令和2年9月17日(2020.9.17)		
審査請求日	令和3年10月20日(2021.10.20)		
審判番号	不服2023-19931(P2023-19931/J 1)		
審判請求日	令和5年11月24日(2023.11.24)		
(31)優先権主張番号	62/816,751		
(32)優先日	平成31年3月11日(2019.3.11)		
(33)優先権主張国・地域又は機関			
	最終頁に続く		最終頁に続く

(54)【発明の名称】 エンコーダ、デコーダ及び対応する方法

(57)【特許請求の範囲】

【請求項1】

デコーダにおいて実現される方法であって、

前記デコーダの受信機により、インター予測に従ってコーディングされたサブピクチャを含む現在のピクチャを含むビットストリームを受信するステップであり、前記サブピクチャは、ピクチャ内の2つ以上のスライス/タイルグループの長方形又は正方形の領域である、ステップと、

前記デコーダのプロセッサにより、前記サブピクチャのブロックについての参照ブロックを決定するステップと、

前記プロセッサにより、前記サブピクチャがピクチャとして扱われることをフラグが示し、前記ブロックについての動きベクトルが前記サブピクチャ外を指すとき、前記参照ブロック内のサンプル位置の整数部にクリッピング関数を適用して前記整数部が前記サブピクチャ内にあるように制限するステップであり、前記サンプル位置は補間フィルタに使用され、前記補間フィルタはルマサンプルバイリニア補間プロセスを含み、前記ブロックはルマサンプルのブロックを含む、ステップと、

前記プロセッサにより、予測サンプル値を取得するために、前記クリッピング関数の結果に前記補間フィルタを適用するステップであり、前記予測サンプル値は予測ルマサンプル値を含む、ステップと、

前記プロセッサにより、前記予測サンプル値に基づいて、前記ブロックを復号するステップと

10

20

を含む方法。

【請求項 2】

前記ルマサンプルバイリニア補間プロセスは、全サンプルユニット(xInt_L,yInt_L)におけるルマ位置を含む入力を受信し、前記ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLX_L)を出力し、前記クリッピング関数は、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、xInt_i=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xInt_L+i)、且つ yInt_i=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yInt_L+i)

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、xInt_i及びyInt_iはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、Clip3は以下に従う前記クリッピング関数であり、

【数 4 9】

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

ここで、x、y及びzは数値入力値であることに従って、前記整数部に適用される、請求項 1 に記載の方法。

【請求項 3】

前記補間フィルタはルマサンプル8タップ補間フィルタリングプロセスを含み、前記ブロックはルマサンプルのブロックを含み、前記予測サンプル値は予測ルマサンプル値を含む、請求項 1 又は 2 に記載の方法。

【請求項 4】

前記ルマサンプル8タップ補間フィルタリングプロセスは、全サンプルユニット(xInt_L,yInt_L)におけるルマ位置を含む入力を受信し、前記ルマサンプル8タップ補間フィルタリングプロセスは、予測ルマサンプル値(predSampleLX_L)を出力し、前記クリッピング関数は、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

xInt_i=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xInt_L+i-3)、且つ yInt_i=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yInt_L+i-3)

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、xInt_i及びyInt_iはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、Clip3は以下に従う前記クリッピング関数であり、

【数 5 0】

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

ここで、x、y及びzは数値入力値であることに従って、前記整数部に適用される、請求項 3 に記載の方法。

【請求項 5】

前記補間フィルタはクロマサンプル補間プロセスを含み、前記ブロックはクロマサンプル

ルのブロックを含み、前記予測サンプル値は予測クロマサンプル値を含む、請求項 1 乃至 4 のうちいずれか 1 項に記載の方法。

【請求項 6】

前記クロマサンプル補間プロセスは、全サンプルユニット(xInt_C,yInt_C)におけるクロマ位置を含む入力を受信し、前記クロマサンプル補間プロセスは、予測クロマサンプル値(predSampleLX_C)を出力し、前記クリッピング関数は、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、
 $xInt_i = Clip3(SubPicLeftBoundaryPos/SubWidthC, SubPicRightBoundaryPos/SubWidthC, xInt_C + i)$ 、且つ
 $yInt_i = Clip3(SubPicTopBoundaryPos/SubHeightC, SubPicBotBoundaryPos/SubHeightC, yInt_C + i)$

10

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、xInt_i及びyInt_iはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、SubWidthC及びSubHeightCはルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、Clip3は以下に従う前記クリッピング関数であり、

20

【数 5 1】

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

ここで、x、y及びzは数値入力値であることに従って、前記整数部に適用される、請求項 5 に記載の方法。

【請求項 7】

エンコーダにおいて実現される方法であって、

前記エンコーダのプロセッサにより、現在のピクチャをサブピクチャに区分し、前記サブピクチャをブロックに区分するステップであり、前記サブピクチャは、ピクチャ内の2つ以上のスライス/タイルグループの長方形又は正方形の領域である、ステップと、

30

前記プロセッサにより、前記ブロックを符号化するための参照ブロックを取得するステップと、

前記プロセッサにより、前記サブピクチャがピクチャとして扱われることをフラグが示し、前記ブロックについての動きベクトルが前記サブピクチャ外を指すとき、前記参照ブロック内のサンプル位置の整数部にクリッピング関数を適用して前記整数部が前記サブピクチャ内にあるように制限するステップであり、前記サンプル位置は補間フィルタに使用され、前記補間フィルタはルマサンプルバイリニア補間プロセスを含み、前記ブロックはルマサンプルのブロックを含む、ステップと、

40

前記プロセッサにより、予測サンプル値を取得するために、前記クリッピング関数の結果に前記補間フィルタを適用するステップであり、前記予測サンプル値は予測ルマサンプル値を含む、ステップと、

前記プロセッサにより、前記予測サンプル値及び動きベクトルに基づいて、前記ブロックをビットストリームに符号化するステップと

を含む方法。

【請求項 8】

前記ルマサンプルバイリニア補間プロセスは、全サンプルユニット(xInt_L,yInt_L)におけるルマ位置を含む入力を受信し、前記ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLX_L)を出力し、前記クリッピング関数は、以下に従って、すな

50

わち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、
 $xInt_i = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i)$ 、且つ
 $yInt_i = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i)$

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、 $xInt_i$ 及び $yInt_i$ はインデックス*i*におけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、Clip3は以下に従う前記クリッピング関数であり、

【数52】

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

ここで、 x 、 y 及び z は数値入力値であることに従って、前記整数部に適用される、請求項7に記載の方法。

【請求項9】

前記補間フィルタはルマサンプル8タップ補間フィルタリングプロセスを含み、前記ブロックはルマサンプルのブロックを含み、前記予測サンプル値は予測ルマサンプル値を含む、請求項7又は8に記載の方法。

【請求項10】

前記ルマサンプル8タップ補間フィルタリングプロセスは、全サンプルユニット($xInt_L$, $yInt_L$)におけるルマ位置を含む入力を受信し、前記ルマサンプル8タップ補間フィルタリングプロセスは、予測ルマサンプル値(predSampleLXL)を出力し、前記クリッピング関数は、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

$xInt_i = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i - 3)$ 、且つ
 $yInt_i = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i - 3)$

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、 $xInt_i$ 及び $yInt_i$ はインデックス*i*におけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、Clip3は以下に従う前記クリッピング関数であり、

【数53】

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

ここで、 x 、 y 及び z は数値入力値であることに従って、前記整数部に適用される、請求項9に記載の方法。

【請求項11】

前記補間フィルタはクロマサンプル補間プロセスを含み、前記ブロックはクロマサンプルのブロックを含み、前記予測サンプル値は予測クロマサンプル値を含む、請求項7乃至10のうちいずれか1項に記載の方法。

【請求項12】

前記クロマサンプル補間プロセスは、全サンプルユニット($xInt_C, yInt_C$)におけるクロマ位置を含む入力を受信し、前記クロマサンプル補間プロセスは、予測クロマサンプル値(pr

10

20

30

40

50

edSampleLXC)を出力し、前記クリッピング関数は、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、
 $xInti = Clip3(SubPicLeftBoundaryPos/SubWidthC, SubPicRightBoundaryPos/SubWidthC, xIntC+i)$ 、且つ
 $yInti = Clip3(SubPicTopBoundaryPos/SubHeightC, SubPicBotBoundaryPos/SubHeightC, yIntC+i)$

ここで、subpic_treated_as_pic_flagは前記サブピクチャがピクチャとして扱われることを示すために設定される前記フラグであり、SubPicIdxは前記サブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosは前記サブピクチャの右境界の位置であり、SubPicLeftBoundaryPosは前記サブピクチャの左境界の位置であり、SubPicTopBoundaryPosは前記サブピクチャの上境界の位置であり、SubPicBotBoundaryPosは前記サブピクチャの下境界の位置であり、SubWidthC及びSubHeightCはルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、Clip3は以下に従う前記クリッピング関数であり、

【数54】

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

ここで、x、y及びzは数値入力値であることに従って、前記整数部に適用される、請求項11に記載の方法。

【請求項13】

ビデオ復号デバイスであって、

プロセッサと、前記プロセッサに結合された受信機と、前記プロセッサに結合されたメモリと、前記プロセッサに結合された送信機とを含み、前記プロセッサ、受信機、メモリ及び送信機は、請求項1乃至6のうちいずれか1項の方法を実行するように構成される、ビデオ復号デバイス。

【請求項14】

ビデオ符号化デバイスであって、

プロセッサと、前記プロセッサに結合された受信機と、前記プロセッサに結合されたメモリと、前記プロセッサに結合された送信機とを含み、前記プロセッサ、受信機、メモリ及び送信機は、請求項7乃至12のうちいずれか1項の方法を実行するように構成される、ビデオ符号化デバイス。

【請求項15】

ビデオ復号デバイスにより使用するためのコンピュータプログラム製品を含む非一時的なコンピュータ読み取り可能媒体であって、

前記コンピュータプログラム製品は、プロセッサにより実行されたとき、前記ビデオ復号デバイスに請求項1乃至6のうちいずれか1項に記載の方法を実行させるように、前記非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ実行可能命令を含む、非一時的なコンピュータ読み取り可能媒体。

【請求項16】

ビデオ符号化デバイスにより使用するためのコンピュータプログラム製品を含む非一時的なコンピュータ読み取り可能媒体であって、

前記コンピュータプログラム製品は、プロセッサにより実行されたとき、前記ビデオ符号化デバイスに請求項7乃至12のうちいずれか1項に記載の方法を実行させるように、前記非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ実行可能命令を含む、非一時的なコンピュータ読み取り可能媒体。

【請求項17】

デコーダであって、

10

20

30

40

50

インター予測に従ってコーディングされたサブピクチャを含む現在のピクチャを含むビットストリームを受信するように構成された受信ユニットであり、前記サブピクチャは、ピクチャ内の2つ以上のスライス/タイルグループの長方形又は正方形の領域である、受信ユニットと、

前記サブピクチャのブロックについての参照ブロックを決定するように構成された決定ユニットと、

前記サブピクチャがピクチャとして扱われることをフラグが示し、前記ブロックについての動きベクトルが前記サブピクチャ外を指すとき、前記参照ブロック内のサンプル位置の整数部にクリッピング関数を適用して前記整数部が前記サブピクチャ内にあるように制限するように構成され、前記サンプル位置は補間フィルタに使用され、前記補間フィルタはルマサンプルバイリニア補間プロセスを含み、前記ブロックはルマサンプルのブロックを含み、予測ルマサンプル値を含む予測サンプル値を取得するために、前記クリッピング関数の結果に前記補間フィルタを適用するように構成された適用ユニットと、

前記予測サンプル値に基づいて、前記ブロックを復号するように構成された復号ユニットと

を含むデコーダ。

【請求項 18】

前記デコーダは、請求項 1 乃至 6 のうちいずれか 1 項に記載の方法を実行するように更に構成される、請求項 17 に記載のデコーダ。

【請求項 19】

エンコーダであって、

現在のピクチャをサブピクチャに区分し、前記サブピクチャをブロックに区分するように構成された区分ユニットであり、前記サブピクチャは、ピクチャ内の2つ以上のスライス/タイルグループの長方形又は正方形の領域である、区分ユニットと、

前記ブロックを符号化するための参照ブロックを取得するように構成された取得ユニットと、

前記サブピクチャがピクチャとして扱われることをフラグが示し、前記ブロックについての動きベクトルが前記サブピクチャ外を指すとき、前記参照ブロック内のサンプル位置の整数部にクリッピング関数を適用して前記整数部が前記サブピクチャ内にあるように制限するように構成され、前記サンプル位置は補間フィルタに使用され、前記補間フィルタはルマサンプルバイリニア補間プロセスを含み、前記ブロックはルマサンプルのブロックを含み、予測ルマサンプル値を含む予測サンプル値を取得するために、前記クリッピング関数の結果に前記補間フィルタを適用するように構成された適用ユニットと、

前記予測サンプル値及び動きベクトルに基づいて、前記ブロックをビットストリームに符号化するように構成された符号化ユニットと

を含むエンコーダ。

【請求項 20】

前記エンコーダは、請求項 7 乃至 12 のうちいずれか 1 項に記載の方法を実行するように更に構成される、請求項 19 に記載のエンコーダ。

【請求項 21】

コンピュータ又はプロセッサ上で実行されたとき、請求項 1 乃至 6 のうちいずれか 1 項に記載の方法を実行するためのプログラムコードを含むコンピュータプログラム。

【請求項 22】

コンピュータ又はプロセッサ上で実行されたとき、請求項 7 乃至 12 のうちいずれか 1 項に記載の方法を実行するためのプログラムコードを含むコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

〔技術分野〕

本開示は、一般的にビデオコーディングに関し、具体的にはビデオコーディングにおい

10

20

30

40

50

てピクチャのサブピクチャをコーディングすることに関する。

【背景技術】

【0002】

比較的短いビデオですら描写するために必要とされるビデオデータの量は相当になる可能性があり、これは、データがストリーミングされる時、或いは制限された帯域幅容量を有する通信ネットワークを横切って通信される時、困難を生じることがある。したがって、ビデオデータは、一般的に、現代の電気通信ネットワークを横切って通信される前に圧縮される。メモリリソースが制限されることがあるので、ビデオが記憶デバイスに記憶されるとき、ビデオのサイズも問題になる可能性がある。ビデオ圧縮デバイスは、しばしば、伝送又は記憶の前にビデオデータをコーディングするために送信元においてソフトウェア及び/又はハードウェアを使用し、それにより、デジタルビデオ画像を表すのに必要とされるデータの量を減少させる。次いで、圧縮されたデータは、ビデオデータを復号するビデオ解凍デバイスにより宛先において受信される。制限されたネットワークリソース及びより高いビデオ品質の絶えず増加する要求のため、画像品質にほとんど或いは全く犠牲を払わずに圧縮比を改善する改善された圧縮及び解凍技術が望ましい。

10

【発明の概要】

【0003】

実施形態では、本開示は、デコーダにおいて実現される方法を含み、当該方法は、デコーダの受信機により、インター予測に従ってコーディングされたサブピクチャを含む現在のピクチャを含むビットストリームを受信するステップと、デコーダのプロセッサにより、サブピクチャのブロックについての動きベクトルを決定するステップと、プロセッサにより、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用するステップと、プロセッサにより、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するステップと、プロセッサにより、予測サンプル値に基づいて、ブロックを復号するステップとを含む。いくつかのインター予測モードのうち1つに従ってインター予測が実行されてもよい。特定のインター予測モードは、エンコーダ及びデコーダの双方で動きベクトル予測子の候補リストを生成する。これは、エンコーダが全体の動きベクトルをシグナリングする代わりに、候補リストからインデックスをシグナリングすることにより、動きベクトルをシグナリングすることを可能にする。さらに、いくつかのシステムは、独立した抽出のためにサブピクチャを符号化する。これは、現在のサブピクチャが他のサブピクチャからの情報を復号することなく復号されて表示されることを可能にする。これは、サブピクチャ外を指す動きベクトルが使用されるとき、エラーを生じることがある。この理由は、動きベクトルにより指されるデータが復号されないことがあり、したがって、利用可能ではないことがあるからである。本開示は、サブピクチャがピクチャとして扱われるべきであることを示すフラグを含む。現在のサブピクチャがピクチャのように扱われるとき、現在のサブピクチャは、他のサブピクチャを参照することなく抽出されるべきである。具体的には、本例は、補間フィルタを適用するときに適用されるクリッピング機能を使用する。このクリッピング機能は、別々の抽出をサポートするためのサブピクチャの間の分離を維持するために、補間フィルタが隣接サブピクチャからのデータに依存しないことを確保する。したがって、クリッピング機能は、フラグが設定され、動きベクトルが現在のサブピクチャ外を指すときに適用される。次いで、補間フィルタはクリッピング機能の結果に適用される。したがって、本例は、サブピクチャ抽出を実行するときのエラーを防止することにより、更なる機能をビデオコーデックに提供する。

20

30

40

【0004】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがルマサンプルバイリニア補間プロセスを含み、ブロックがルマサンプルのブロックを含み、予測サンプル値が予測ルマサンプル値を含むことを提供する。

【0005】

50

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、ルマサンプルバイリニア補間プロセスが、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信し、ルマサンプルバイリニア補間プロセスが、予測ルマサンプル値(predSampleLXL)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i)、且つ
yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i)

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

【数1】

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

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【0006】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがルマサンプル8タップ補間フィルタリングプロセスを含み、ブロックがルマサンプルのブロックを含み、予測サンプル値が予測ルマサンプル値を含むことを提供する。

【0007】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、ルマサンプル8タップ補間フィルタリングプロセスが、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信し、ルマサンプルバイリニア補間プロセスが、予測ルマサンプル値(predSampleLXL)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3)、且つ
yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3)

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

【数2】

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

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【0008】

10

20

30

40

50

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがクロマサンプル補間プロセスを含み、ブロックがクロマサンプルのブロックを含み、予測サンプル値が予測クロマサンプル値を含むことを提供する。

【 0 0 0 9 】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、クロマサンプル補間プロセスが、全サンプルユニット(xIntC,yIntC)におけるクロマ位置を含む入力を受信し、クロマサンプル補間プロセスが、予測クロマサンプル値(predSampleLXC)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

xInti=Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i)、且つ

yInti=Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i)

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、SubWidthC及びSubHeightCはルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、Clip3は以下に従うクリッピング機能であり、

【 数 3 】

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

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【 0 0 1 0 】

実施形態では、本開示は、エンコーダにおいて実現される方法を含み、当該方法は、エンコーダのプロセッサにより、現在のピクチャをサブピクチャに区分し、サブピクチャをブロックに区分するステップと、プロセッサにより、インター予測に従ってブロックを符号化することを決定するステップと、プロセッサにより、ブロックを符号化するための動きベクトルを選択するステップと、プロセッサにより、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用するステップと、プロセッサにより、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するステップと、プロセッサにより、予測サンプル値及び動きベクトルに基づいて、ブロックをビットストリームに符号化するステップと、プロセッサに結合されたメモリにより、デコーダに向けて通信するためにビットストリームを記憶するステップとを含む。いくつかのインター予測モードのうち1つに従ってインター予測が実行されてもよい。特定のインター予測モードは、エンコーダ及びデコーダの双方で動きベクトル予測子の候補リストを生成する。これは、エンコーダが全体の動きベクトルをシグナリングする代わりに、候補リストからインデックスをシグナリングすることにより、動きベクトルをシグナリングすることを可能にする。さらに、いくつかのシステムは、独立した抽出のためにサブピクチャを符号化する。これは、現在のサブピクチャが他のサブピクチャからの情報を復号することなく復号されて表示されることを可能にする。これは、サブピクチャ外を指す動きベクトルが使用されるとき、エラーを生じることがある。この理由は、動きベクトルにより指されるデータが復号されな

いことがあり、したがって、利用可能ではないことがあるからである。本開示は、サブピクチャがピクチャとして扱われるべきであることを示すフラグを含む。現在のサブピクチャがピクチャのように扱われるとき、現在のサブピクチャは、他のサブピクチャを参照することなく抽出されるべきである。具体的には、本例は、補間フィルタを適用するときに適用されるクリッピング機能を使用する。このクリッピング機能は、別々の抽出をサポートするためのサブピクチャの間の分離を維持するために、補間フィルタが隣接サブピクチャからのデータに依存しないことを確保する。したがって、クリッピング機能は、フラグが設定され、動きベクトルが現在のサブピクチャ外を指すときに適用される。次いで、補間フィルタはクリッピング機能の結果に適用される。したがって、本例は、サブピクチャ抽出を実行するときのエラーを防止することにより、更なる機能をビデオコーデックに提供

10

【0011】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがルマサンプルバイリニア補間プロセスを含み、ブロックがルマサンプルのブロックを含み、予測サンプル値が予測ルマサンプル値を含むことを提供する。

【0012】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、ルマサンプルバイリニア補間プロセスが、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信し、ルマサンプルバイリニア補間プロセスが、予測ルマサンプル値(predSampleLXL)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、

20

$xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL+i)$ 、且つ
 $yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL+i)$

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

30

【数4】

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

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【0013】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがルマサンプル8タップ補間フィルタリングプロセスを含み、ブロックがルマサンプルのブロックを含み、予測サンプル値が予測ルマサンプル値を含むことを提供する。

40

【0014】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、ルマサンプル8タップ補間フィルタリングプロセスが、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信し、ルマサンプルバイリニア補間プロセスが、予測ルマサンプル値(predSampleLXL)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、
 $xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL+i-3)$ 、且つ
 $yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL+i-3)$

50

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

【数5】

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{otherwise} \end{cases} \quad 10$$

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【0015】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、補間フィルタがクロマサンプル補間プロセスを含み、ブロックがクロマサンプルのブロックを含み、予測サンプル値が予測クロマサンプル値を含むことを提供する。

【0016】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、クロマサンプル補間プロセスが、全サンプルユニット(xIntC,yIntC)におけるクロマ位置を含む入力を受信し、クロマサンプル補間プロセスが、予測クロマサンプル値(predSampleLXC)を出力し、クリッピング機能が、以下に従って、すなわち、subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまり、
 $xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos}/\text{SubWidthC}, \text{SubPicRightBoundaryPos}/\text{SubWidthC}, xIntC+i)$ 、且つ
 $yInti = \text{Clip3}(\text{SubPicTopBoundaryPos}/\text{SubHeightC}, \text{SubPicBotBoundaryPos}/\text{SubHeightC}, yIntC+i)$

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、SubWidthC及びSubHeightCはルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、Clip3は以下に従うクリッピング機能であり、

【数6】

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{otherwise} \end{cases} \quad 40$$

ここで、x、y及びzは数値入力値であることに従って、サンプル位置に適用されることを提供する。

【0017】

実施形態では、本開示は、プロセッサと、プロセッサに結合された受信機と、プロセッサに結合されたメモリと、プロセッサに結合された送信機とを含むビデオコーディングデバイスを含み、プロセッサ、受信機、メモリ及び送信機は、上記の態様のいずれかの方法を実行するように構成される。

【0018】

実施形態では、本開示は、ビデオコーディングデバイスにより使用するためのコンピュータプログラム製品を含む非一時的なコンピュータ読み取り可能媒体を含み、コンピュータプログラム製品は、プロセッサにより実行されたとき、ビデオコーディングデバイスに上記の態様のいずれかの方法を実行させるように、非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ実行可能命令を含む。

【0019】

実施形態では、本開示は、インター予測に従ってコーディングされたサブピクチャを含む現在のピクチャを含むビットストリームを受信するための受信手段と、サブピクチャのブロックについての動きベクトルを決定するための決定手段と、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用し、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するための適用手段と、予測サンプル値に基づいて、ブロックを復号するための復号手段と、復号ビデオシーケンスの一部として表示するために、ブロックを転送するための転送手段とを含むデコーダを含む。

10

【0020】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、デコーダが上記の態様のいずれかの方法を実行するように更に構成されることを提供する。

【0021】

実施形態では、本開示は、現在のピクチャをサブピクチャに区分し、サブピクチャをブロックに区分するための区分手段と、インター予測に従ってブロックを符号化することを決定するための決定手段と、ブロックを符号化するための動きベクトルを選択するための選択手段と、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用し、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するための適用手段と、予測サンプル値及び動きベクトルに基づいて、ブロックをビットストリームに符号化するための符号化手段と、デコーダに向けて通信するためにビットストリームを記憶するための記憶手段とを含むエンコーダを含む。

20

30

【0022】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、エンコーダが上記の態様のいずれかの方法を実行するように更に構成されることを提供する。

【0023】

任意選択で、上記の態様のいずれかにおいて、当該態様の他の実現方式は、エンコーダが上記の態様のいずれかの方法を実行するように更に構成されることを提供する。

【0024】

明確性の目的で、本開示の範囲内で新たな実施形態を作成するために、上記の実施形態のいずれか1つは、他の上記の実施形態のいずれか1つ以上と組み合わせられてもよい。

【0025】

これら及び他の特徴は、添付の図面及び特許請求の範囲に関連して考慮される以下の詳細な説明からより明確に理解される。

40

【図面の簡単な説明】

【0026】

本開示のより完全な理解のために、ここで、添付の図面及び詳細な説明に関連して考慮される以下の簡単な説明に参照が行われ、ここで、同様の参照符号は同様の部分を表す。

【図1】ビデオ信号をコーディングする例示的な方法のフローチャートである。

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

【図3】例示的なビデオエンコーダを示す概略図である。

50

【図 4】例示的なビデオデコーダを示す概略図である。

【図 5 A】サブピクチャに区分された例示的なピクチャを示す概略図である。

【図 5 B】スライスに区分された例示的なサブピクチャを示す概略図である。

【図 5 C】タイルに区分された例示的なスライスを示す概略図である。

【図 5 D】コーディングツリーユニット(coding tree unit, CTU)に区分された例示的なスライスを示す概略図である。

【図 6】一方向インター予測の例を示す概略図である。

【図 7】双方向インター予測の例を示す概略図である。

【図 8】隣接するコーディングされたブロックからの候補動きベクトルに基づいて現在のブロックをコーディングする例を示す概略図である。

10

【図 9】動きベクトルの候補リストを決定するための例示的なパターンを示す概略図である。

【図 10】例示的なループ内フィルタを示すブロック図である。

【図 11】ピクチャのサブピクチャを復号することをサポートするためのコーディングツールパラメータを含む例示的なビットストリームを示す概略図である。

【図 12】例示的なビデオコーディングデバイスの概略図である。

【図 13】サブピクチャがピクチャとして扱われるとき、補間フィルタにおいてクリッピング機能を適用しつつ、ビデオシーケンスをビットストリームに符号化する例示的な方法のフローチャートである。

【図 14】サブピクチャがピクチャとして扱われるとき、補間フィルタにおいてクリッピング機能を適用しつつ、ビットストリームからビデオシーケンスを復号する例示的な方法のフローチャートである。

20

【図 15】サブピクチャがピクチャとして扱われるとき、補間フィルタにおいてクリッピング機能を適用しつつ、ビットストリームにおいて画像のビデオシーケンスをコーディングするための例示的なシステムの概略図である。

【発明を実施するための形態】

【0027】

まず、1つ以上の実施形態の例示的な実現方式が以下に提供されるが、開示のシステム及び/又は方法は、現在既知であっても存在していても、いずれかの数の技術を使用して実現されてもよいことが理解されるべきである。本開示は、ここに例示及び記載される例示的な設計及び実現方式を含む、以下に示す例示的な実現方式、図面及び技術に決して限定されるべきでなく、添付の特許請求の範囲の範囲内で、これらの均等物の全範囲と共に変更されてもよい。

30

【0028】

以下の略称、すなわち、適応ループフィルタ(Adaptive Loop Filter, ALF)、コーディングツリーブロック(Coding Tree Block, CTB)、コーディングツリーユニット(Coding Tree Unit, CTU)、コーディングユニット(Coding Unit, CU)、コーディングされたビデオシーケンス(Coded Video Sequence, CVS)、共同ビデオ作業部会(Joint Video Experts Team, JVET)、動き制約タイルセット(Motion-Constrained Tile Set, MCTS)、最大転送ユニット(Maximum Transfer Unit, MTU)、ネットワーク抽象化レイヤ(Network Abstraction Layer, NAL)、ピクチャ順序カウント(Picture Order Count, POC)、生バイトシーケンスペイロード(Raw Byte Sequence Payload, RBSP)、サンプル適応オフセット(Sample Adaptive Offset, SAO)、シーケンスパラメータセット(Sequence Parameter Set, SPS)、時間動きベクトル予測(Temporal Motion Vector Prediction, TMVP)、バーサタイルビデオコーディング(Versatile Video Coding, VVC)及びワーキングドラフト(Working Draft, WD)が、ここで使用される。

40

【0029】

多くのビデオ圧縮技術が、最小のデータ損失でビデオファイルのサイズを低減するために使用できる。例えば、ビデオ圧縮技術は、ビデオシーケンスにおけるデータ冗長性を低

50

減又は除去するために、空間(例えば、イントラピクチャ)予測及び/又は時間(例えば、インターピクチャ)予測を実行することを含むことができる。ブロックベースのビデオコーディングのために、ビデオスライス(例えば、ビデオピクチャ又はビデオピクチャの一部)は、ビデオブロックに区分されてもよく、これはまた、ツリーブロック、コーディングツリーブロック(coding tree block, CTB)、コーディングツリーユニット(coding tree unit, CTU)、コーディングユニット(coding unit, CU)及び/又はコーディングノードとも呼ばれてもよい。ピクチャのイントラコーディングされる(I)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに関する空間予測を使用してコーディングされる。ピクチャのインターコーディングされる一方向予測(P)又は双方向予測(B)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに関する空間予測、又は他の参照ピクチャ内の参照サンプルに関する時間予測を使用することによりコーディングされてもよい。ピクチャは、フレーム及び/又は画像と呼ばれてもよく、参照ピクチャは、参照フレーム及び/又は参照画像と呼ばれてもよい。空間又は時間予測は、画像ブロックを表す予測ブロックを生じる。残差データは、元の画像ブロックと予測ブロックとの間の画素差を表す。したがって、インターコーディングされるブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルと、コーディングされるブロックと予測ブロックとの間の差を示す残差データとに従って符号化される。イントラコーディングされるブロックは、イントラコーディングモードと残差データとに従って符号化される。更なる圧縮のために、残差データは、画素ドメインから変換ドメインに変換されてもよい。これらは残差変換係数を生じ、これらは量子化されてもよい。量子化された変換係数は、最初に二次元配列に配置されてもよい。量子化された変換係数は、変換係数の一次元ベクトルを生成するためにスキャンされてもよい。エントロピーコーディングは、更に多くの圧縮を達成するために適用されてもよい。このようなビデオ圧縮技術について、以下により詳細に説明する。

【 0 0 3 0 】

符号化されたビデオが正確に復号できることを確保するために、ビデオは、対応するビデオコーディング標準に従って符号化及び復号される。ビデオコーディング標準は、国際電気通信連合(International Telecommunication Union, ITU)標準化部門(ITU Standardization Sector, ITU-T)のH.261、国際標準化機構/国際電気標準会議(International Organization for Standardization/International Electrotechnical Commission, ISO/IEC)の動画像専門家集団(Motion Picture Experts Group, MPEG)-1パート2、ITU-TのH.262又はISO/IECのMPEG-2パート2、ITU-TのH.263、ISO/IECのMPEG-4パート2、ITU-TのH.264又はISO/IECのMPEG-4パート10としても知られる高度ビデオコーディング(Advanced Video Coding, AVC)、及びITU-TのH.265又はMPEG-Hパート2としても知られる高効率ビデオコーディング(High Efficiency Video Coding, HEVC)を含む。AVCは、スケーラブルビデオコーディング(Scalable Video Coding, SVC)、マルチビュービデオコーディング(Multiview Video Coding, MVC)及びマルチビュービデオコーディング・プラス・深度(Multiview Video Coding plus Depth, MVC+D)及び3次元(three dimensional, 3D)AVC(3D-AVC)のような拡張を含む。HEVCは、スケーラブルHEVC(Scalable HEVC, SHVC)、マルチビューHEVC(Multiview HEVC, MV-HEVC)及び3D HEVC(3D-HEVC)のような拡張を含む。ITU-T及びISO/IECの共同ビデオ作業部会(joint video experts team, JVET)は、バーサタイルビデオコーディング(Versatile Video Coding, VVC)と呼ばれるビデオコーディング標準を開発し始めている。VVCはワーキングドラフト(Working Draft, WD)に含まれており、これは、アルゴリズム記述、VVC WDのエンコーダ側記述及び参照ソフトウェアを提供するJVET-M1001-v6を含む。

【 0 0 3 1 】

ビデオ画像をコーディングするために、画像はまず区分され、区分はビットストリームにコーディングされる。様々なピクチャ区分方式が利用可能である。例えば、画像は、通常スライス、従属スライス、タイルに、及び/又は波面並列処理(Wavefront Parallel P

10

20

30

40

50

rocessing, WPP)に従って区分できる。簡単のために、HEVCは、ビデオコーディングのためにスライスをCTBのグループに区分するとき、通常スライス、従属スライス、タイル、WPP及びこれらの組み合わせのみが使用できるようにエンコーダを制限する。このような区分は、最大転送ユニット(Maximum Transfer Unit, MTU)サイズのマッチング、並列処理、及び低減したエンドツーエンド遅延をサポートするために適用できる。MTUは、単一パケットで伝送できるデータの最大量を示す。パケットペイロードがMTUを超える場合、そのペイロードはフラグメンテーションと呼ばれるプロセスを通じて2つのパケットに分割される。

【0032】

単にスライスとも呼ばれる通常スライスは、ループフィルタリング動作のため、いくつかの相互依存性にもかかわらず、同じピクチャ内の他の通常スライスと独立して再現できる画像の区分された部分である。各通常スライスは、伝送のためにそれ自身のネットワーク抽象化レイヤ(Network Abstraction Layer, NAL)ユニットにカプセル化される。さらに、スライス境界を横切るピクチャ内予測(イントラサンプル予測、動き情報予測、コーディングモード予測)及びエントロピーコーディング依存性は、独立した再現をサポートするために無効にされてもよい。このような独立した再現は並列化をサポートする。例えば、通常スライススペースの並列化は、最小のプロセッサ間又はコア間通信を使用する。しかし、各通常スライスは独立しているため、各スライスは個々のスライスヘッダに関連付けられる。通常スライスの使用は、各スライスについてのスライスヘッダのビットコストのため、且つ、スライス境界を横切る予測の欠如のため、実質的なコーディングオーバーヘッドを招く可能性がある。さらに、通常スライスは、MTUサイズ要件のためのマッチングをサポートするために使用されてもよい。具体的には、通常スライスが個々のNALユニットにカプセル化されて独立してコーディングできるので、スライスを複数のパケットに分かれるのを回避するために、各通常スライスはMTU方式におけるMTUよりも小さくなるべきである。したがって、並列化の目標及びMTUサイズのマッチングの目標は、ピクチャ内のスライスレイアウトに矛盾する要求を課すことがある。

【0033】

従属スライスは通常スライスと同様であるが、短縮されたスライスヘッダを有し、ピクチャ内予測を壊すことなく画像ツリーブロック境界の区分を可能にする。したがって、従属スライスは、通常スライスが複数のNALユニットにフラグメント化されることを可能にし、これは、全体の通常スライスの符号化が完了する前に通常スライスの一部が送出されることを可能にすることにより、低減したエンドツーエンド遅延を提供する。

【0034】

タイルは、タイルの列及び行を作成する水平及び垂直境界により作成された画像の区分された部分である。タイルは、ラスタスキャン順(右から左及び上から下)にコーディングされてもよい。CTBのスキャン順はタイル内でローカルである。したがって、第1のタイル内のCTBは、次のタイル内のCTBに進む前に、ラスタスキャン順にコーディングされる。通常スライスと同様に、タイルは、ピクチャ内予測依存性及びエントロピー復号依存性を壊す。しかし、タイルは個々のNALユニットに含まれなくてもよく、したがって、タイルはMTUサイズのマッチングに使用されなくてもよい。各タイルは、1つのプロセッサ/コアにより処理される可能性があり、隣接タイルを復号する処理ユニットの間のピクチャ内予測に使用されるプロセッサ間/コア間通信は、(隣接タイルが同じスライス内にあるとき)共有スライスヘッダを伝達し、再現されたサンプル及びメタデータのループフィルタリング関連の共有を実行することに制限され得る。1つよりも多くのタイルがスライスに含まれるとき、スライス内の最初のエントリーポイントオフセット以外の各タイルについてのエントリーポイントバイトオフセットが、スライスヘッダでシグナリングされてもよい。各スライス及びタイルについて、以下の条件、すなわち、1)スライス内の全てのコーディングされたツリーブロックが同じタイルに属すること、及び2)タイル内の全てのコーディングされたツリーブロックが同じスライスに属すること、の少なくとも1つが満たされるべきである。

10

20

30

40

50

【 0 0 3 5 】

WPPでは、画像はCTBの単一行に区分される。エントロピー復号及び予測メカニズムは、他の行内のCTBからのデータを使用してもよい。CTB行の並列復号を通じて並列処理が可能になる。例えば、現在の行は、前の行と並列に復号されてもよい。しかし、現在の行の復号は、2つのCTBだけ前の行の復号プロセスから遅延される。この遅延は、現在のCTBがコーディングされる前に、現在の行内の現在のCTBの上のCTB及び右上のCTBに関連するデータが利用可能であることを確保する。この手法は、グラフィックで表されるとき、波面として現れる。このスタガー型開始(staggered start)は、画像がCTB行を含むのと同じ数までのプロセッサ/コアでの並列化を可能にする。ピクチャ内の隣接ツリーブロック行の間のピクチャ内予測が許容されるので、ピクチャ内予測を可能にするためのプロセッサ間/コア間通信は相当になる可能性がある。WPP区分はNALユニットサイズを考慮する。したがって、WPPはMTUサイズのマッチングをサポートしない。しかし、所望の通りMTUサイズのマッチングを実現するために、通常スライスが特定のコーディングオーバーヘッドでWPPに関して使用できる。

10

【 0 0 3 6 】

タイルはまた、動き制約タイルセットを含んでもよい。動き制約タイルセット(motion constrained tile set, MCTS)は、関連する動きベクトルがMCTS内の全サンプル位置、及び補間のためにMCTS内の全サンプル位置のみを必要とする分数サンプル位置に制約されるように設計されたタイルセットである。さらに、MCTS外のブロックから導出された時間動きベクトル予測のための動きベクトル候補の使用は禁止される。このように、各MCTSは、MCTSに含まれないタイルの存在なしに、独立して復号されてもよい。時間MCTSの補足強化情報(supplemental enhancement information, SEI)メッセージは、ビットストリーム内のMCTSの存在を示し、MCTSをシグナリングするために使用されてもよい。MCTSのSEIメッセージは、MCTSセットについての適合ビットストリームを生成するために、(SEIメッセージのセマンティクスの一部として指定される)MCTSサブビットストリーム抽出で使用できる補足情報を提供する。当該情報は、多数の抽出情報セットを含み、それぞれが多数のMCTSセットを定義し、MCTSサブビットストリーム抽出プロセス中に使用されるべき置換ビデオパラメータセット(video parameter set, VPS)、シーケンスパラメータセット(sequence parameter set, SPS)及びピクチャパラメータセット(picture parameter set, PPS)の生バイトシーケンスペイロード(raw bytes sequence payload, RBSP)バイトを含む。MCTSサブビットストリーム抽出プロセスに従ってサブビットストリームを抽出するとき、パラメータセット(VPS、SPS及びPPS)は書き換えられてもよく或いは置換されてもよく、スライスアドレス関連のシンタックスエレメント(first_slice_segment_in_pic_flag及びslice_segment_addressを含む)の1つ又は全てが抽出されたサブビットストリームにおいて異なる値を使用してもよいので、スライスヘッダは更新されてもよい。

20

30

【 0 0 3 7 】

ピクチャはまた、1つ以上のサブピクチャに区分されてもよい。ピクチャをサブピクチャに区分することは、ピクチャの異なる部分がコーディングの観点から異なって扱われることを可能にし得る。例えば、他のサブピクチャを抽出せずに、サブピクチャが抽出されて表示できる。他の例として、異なるサブピクチャは、異なる解像度で表示でき、(例えば、テレビ会議アプリケーションにおいて)互いに相対的に再配置でき、或いは、サブピクチャが共通のピクチャからのデータを併せて含むとしても、別々のピクチャとしてコーディングできる。

40

【 0 0 3 8 】

サブピクチャの例示的な実現方式は以下の通りである。ピクチャは1つ以上のサブピクチャに区分できる。サブピクチャは、0に等しいアドレスを有するスライス/タイルグループで始まるスライス/タイルグループの長方形又は正方形のセットである。各サブピクチャは、異なるPPSを参照してもよく、したがって、各サブピクチャは、異なる区分メカニズムを使用してもよい。サブピクチャは、復号プロセスにおいてピクチャのように扱われて

50

もよい。現在のサブピクチャを復号するために使用される現在の参照ピクチャは、復号ピクチャバッファ内の参照ピクチャから現在のサブピクチャと同一位置の領域を抽出することにより生成されてもよい。抽出された領域は、復号されたサブピクチャでもよく、したがって、ピクチャ内の同じサイズ及び同じ位置のサブピクチャの間でインター予測が行われてもよい。タイルグループは、サブピクチャのタイルラスタスキャンにおけるタイルのシーケンスでもよい。以下が、ピクチャ内のサブピクチャの位置を決定するために導出されてもよい。各サブピクチャは、ピクチャ境界内のサブピクチャに適合するのに十分大きいピクチャ内のCTUラスタスキャン順における次の非占有位置に含まれてもよい。

【0039】

様々なビデオコーディングシステムにより使用されるサブピクチャ方式は、コーディング効率及び/又は機能を低減する様々な問題を含む。本開示は、このような問題に対する様々な解決策を含む。第1の例示的な問題では、いくつかのインター予測モードのうち1つに従ってインター予測が実行されてもよい。特定のインター予測モードは、エンコーダ及びデコーダの双方で動きベクトル予測子の候補リストを生成する。これは、エンコーダが全体の動きベクトルをシグナリングする代わりに、候補リストからインデックスをシグナリングすることにより、動きベクトルをシグナリングすることを可能にする。さらに、いくつかのシステムは、独立した抽出のためにサブピクチャを符号化する。これは、現在のサブピクチャが他のサブピクチャからの情報を復号することなく復号されて表示されることを可能にする。これは、サブピクチャ外を指す動きベクトルが使用されるとき、エラーを生じることがある。この理由は、動きベクトルにより指されるデータが復号されないこと

【0040】

したがって、第1の例では、サブピクチャがピクチャとして扱われるべきであることを示すフラグがここに開示される。このフラグは、サブピクチャの別々の抽出をサポートするように設定される。フラグが設定されたとき、同一位置のブロックから取得された動きベクトル予測子は、サブピクチャ内を指す動きベクトルのみを含む。サブピクチャ外を指すいずれかの動きベクトル予測子は除外される。これは、サブピクチャ外を指す動きベクトルが選択されず、関連するエラーが回避されることを確保する。同一位置のブロックは、現在のピクチャとは異なるピクチャからのブロックである。現在のピクチャ内のブロック(同一位置でないブロック)からの動きベクトル予測子は、補間フィルタのような他のプロセスがこのような動きベクトル予測子についてのエラーを防止し得るので、サブピクチャ外を指してもよい。したがって、本例は、サブピクチャ抽出を実行するときのエラーを防止することにより、更なる機能をビデオエンコーダ/デコーダ(コーデック)に提供する。

【0041】

第2の例では、サブピクチャがピクチャとして扱われるべきであることを示すフラグがここに開示される。現在のサブピクチャがピクチャのように扱われるとき、現在のサブピクチャは、他のサブピクチャを参照することなく抽出されるべきである。具体的には、本例は、補間フィルタを適用するとき適用されるクリッピング機能を使用する。このクリッピング機能は、別々の抽出をサポートするためのサブピクチャの間の分離を維持するために、補間フィルタが隣接サブピクチャからのデータに依存しないことを確保する。したがって、クリッピング機能は、フラグが設定され、動きベクトルが現在のサブピクチャ外を指すときに適用される。次いで、補間フィルタはクリッピング機能の結果に適用される。したがって、本例は、サブピクチャ抽出を実行するときのエラーを防止することにより、更なる機能をビデオコーデックに提供する。したがって、第1の例及び第2の例は、第1の例示的な問題に対処する。

【0042】

第2の例示的な問題では、ビデオコーディングシステムは、ピクチャをサブピクチャ、スライス、タイル及び/又はコーディングツリーユニットに区分し、次いで、これらはブロックに区分される。次いで、このようなブロックは、デコーダに向けて伝送するために符号化される。このようなブロックを復号することは、様々なタイプのノイズを含む復号画

像を生じることがある。このような問題を修正するために、ビデオコーディングシステムは、ブロック境界を横切って様々なフィルタを適用してもよい。これらのフィルタは、ブロッキング、量子化ノイズ及び他の望ましくないコーディングアーチファクトを除去できる。上記のように、いくつかのシステムは、独立した抽出のためにサブピクチャを符号化する。これは、他のサブピクチャからの情報を復号することなく、現在のサブピクチャが復号されて表示されることを可能にする。このようなシステムでは、サブピクチャは、符号化のためにブロックに区分されてもよい。したがって、サブピクチャエッジに沿ったブロック境界は、サブピクチャ境界と整列してもよい。いくつかの場合、ブロック境界はまた、タイル境界と整列してもよい。フィルタは、このようなブロック境界を横切って適用されてもよく、したがって、サブピクチャ境界及び/又はタイル境界を横切って適用されてもよい。これは、隣接サブピクチャからのデータが利用不可能であるとき、フィルタリングプロセスが予期しない方式で動作することがあるので、現在のサブピクチャが独立して抽出されるときにエラーを引き起こすことがある。

10

【0043】

第3の例では、サブピクチャレベルでフィルタリングを制御するフラグがここに開示される。フラグがサブピクチャについて設定されたとき、フィルタは、サブピクチャ境界を横切って適用できる。フラグが設定されていないとき、フィルタは、サブピクチャ境界を横切って適用されない。このように、フィルタは、別々の抽出のために符号化されるサブピクチャについてオフにすることができ、或いは、グループとしての表示のために符号化されるサブピクチャについてオンにすることができる。したがって、本例は、サブピクチャ抽出を実行するときのフィルタ関連のエラーを防止することにより、更なる機能をビデオコーデックに提供する。

20

【0044】

第4の例では、タイルレベルでフィルタリングを制御するように設定できるフラグがここに開示される。タイルについてフラグが設定されたとき、フィルタはタイル境界を横切って適用できる。フラグが設定されていないとき、フィルタはタイル境界を横切って適用されない。このように、フィルタは、(例えば、タイルの内部をフィルタリングし続けつつ)タイル境界で使用するためにオフ又はオンにすることができる。したがって、本例は、タイル境界を横切る選択的フィルタリングをサポートすることにより、更なる機能をビデオコーデックに提供する。したがって、第3の例及び第4の例は、第2の例示的な問題に対処する。

30

【0045】

第3の例示的な問題では、ビデオコーディングシステムは、ピクチャをサブピクチャに区分してもよい。これは、ビデオをコーディングするとき、異なるサブピクチャが異なって扱われることを可能にする。例えば、サブピクチャは別々に抽出されて表示でき、アプリケーションレベルの変更に基づいて独立してサイズ変更できる等である。いくつかの場合、ピクチャをタイルに区分してタイルをサブピクチャに割り当てることにより、サブピクチャが作成されてもよい。いくつかのビデオコーディングシステムは、サブピクチャに含まれるタイルに関してサブピクチャ境界を記述する。しかし、タイル方式は、いくつかのピクチャで使用されなくてもよい。したがって、このような境界記述は、サブピクチャの使用を、タイルを使用するピクチャに制限し得る。

40

【0046】

第5の例では、CTB及び/又はCTUに関してサブピクチャ境界をシグナリングするためのメカニズムがここに開示される。具体的には、サブピクチャの幅及び高さは、CTBの単位でシグナリングできる。また、サブピクチャの左上CTUの位置は、CTBで測定されるようにピクチャの左上CTUからのオフセットとしてシグナリングできる。CTU及びCTBサイズは、所定の値に設定されてもよい。したがって、CTB及びCTUに関してサブピクチャ寸法及び位置をシグナリングすることは、デコーダが表示のためにサブピクチャを位置決めするのに十分な情報を提供する。これは、タイルが使用されないときであっても、サブピクチャが使用されることを可能にする。また、このシグナリングメカニズムは、複雑さを回

50

避け、比較的少数のビットを使用してコーディングできる。したがって、本例は、サブピクチャがタイルと独立して使用されることを可能にすることにより、更なる機能をビデオコーデックに提供する。さらに、本例は、コーディング効率を増加させ、したがって、エンコーダ及び/又はデコーダにおけるプロセッサ、メモリ及び/又はネットワークリソースの使用を低減する。したがって、第5の例は、第3の例示的な問題に対処する。

【0047】

第4の例示的な問題では、ピクチャは、符号化のために複数のスライスに区分できる。いくつかのビデオコーディングシステムでは、スライスは、ピクチャに対するこれらの位置に基づいてアドレス指定される。更に他のビデオコーディングシステムは、サブピクチャの概念を使用する。上記のように、サブピクチャは、コーディングの観点から、他のサブピクチャと異なって扱われることができる。例えば、サブピクチャは、他のサブピクチャと独立して抽出されて表示できる。このような場合、ピクチャ位置に基づいて生成されるスライスアドレスは、相当の数の想定スライスアドレスが省略されるので、適切に動作しなくなることがある。いくつかのビデオコーディングシステムは、サブピクチャ抽出をサポートするためにスライスアドレスを変更するための要求によってスライスヘッダを動的に書き換えることにより、この問題に対処する。このようなプロセスは、リソース集約的になる可能性がある。この理由は、ユーザがサブピクチャを見ることを要求する毎にこのプロセスが発生し得るからである。

【0048】

第6の例では、スライスを含むサブピクチャに対してアドレス指定されるスライスがここに開示される。例えば、スライスヘッダは、サブピクチャ識別子(identifier, ID)と、サブピクチャに含まれる各スライスのアドレスとを含んでもよい。さらに、シーケンスパラメータセット(sequence parameter set, SPS)は、サブピクチャIDにより参照され得るサブピクチャの寸法を含んでもよい。したがって、サブピクチャの別々の抽出が要求されるとき、スライスヘッダは書き換えられる必要がない。スライスヘッダ及びSPSは、表示のためにサブピクチャ内のスライスを位置決めすることをサポートするのに十分な情報を含む。したがって、本例は、コーディング効率を増加させ、及び/又はスライスヘッダの冗長な書き換えを回避し、したがって、エンコーダ及び/又はデコーダにおけるプロセッサ、メモリ及び/又はネットワークリソースの使用を低減する。したがって、第6の例は、第4の例示的な問題に対処する。

【0049】

図1は、ビデオ信号をコーディングする例示的な動作方法100のフローチャートである。具体的には、ビデオ信号はエンコーダにおいて符号化される。符号化プロセスは、ビデオファイルサイズを低減するために様々なメカニズムを使用することによりビデオ信号を圧縮する。より小さいファイルサイズは、圧縮ビデオファイルがユーザに向けて伝送されることを可能にし、一方、関連する帯域幅オーバーヘッドを低減する。次いで、デコーダは、エンドユーザへの表示のために元のビデオ信号を再現するために、圧縮ビデオファイルを復号する。復号プロセスは、一般的に、デコーダがビデオ信号を一貫して再現することを可能にするように符号化プロセスをミラーリングする。

【0050】

ステップ101において、ビデオ信号がエンコーダに入力される。例えば、ビデオ信号は、メモリに記憶された非圧縮ビデオファイルでもよい。他の例として、ビデオファイルは、ビデオカメラのようなビデオキャプチャデバイスによりキャプチャされ、ビデオのライブストリーミングをサポートするように符号化されてもよい。ビデオファイルは、オーディオ成分及びビデオ成分の双方を含んでもよい。ビデオ成分は、順に見られるときに動きの視覚的な印象を与える一連の画像フレームを含む。フレームは、ここではルマ成分(又はルマサンプル)と呼ばれる光に関して表現される画素と、クロマ成分(又はカラーサンプル)と呼ばれる色とを含む。いくつかの例では、フレームはまた、三次元表示をサポートするための深度値を含んでもよい。

【0051】

ステップ103において、ビデオはブロックに区分される。区分は、圧縮のために、各フレーム内の画素を正方形及び/又は長方形のブロックに細分することを含む。例えば、高効率ビデオコーディング(High Efficiency Video Coding, HEVC)(H.265及びMPEG-Hパート2としても知られる)では、フレームは、まず、所定のサイズ(例えば、64画素×64画素)のブロックであるコーディングツリーユニット(coding tree unit, CTU)に分割できる。CTUはルマサンプル及びクロマサンプルの双方を含む。コーディングツリーは、CTUをブロックに分割し、次いで、更なる符号化をサポートする構成が達成されるまで、ブロックを再帰的に細分するために使用されてもよい。例えば、フレームのルマ成分は、個々のブロックが比較的均一な照明値を含むまで細分されてもよい。さらに、フレームのクロマ成分は、個々のブロックが比較的均一な色値を含むまで細分されてもよい。したがって、区分メカニズムはビデオフレームの内容に依存して変化する。

10

【0052】

ステップ105において、ステップ103で区分された画像ブロックを圧縮するために、様々な圧縮メカニズムが使用される。例えば、インター予測及び/又はイントラ予測が使用されてもよい。インター予測は、共通のシーン内のオブジェクトが連続したフレームに現れる傾向があるという事実を利用するように設計されている。したがって、参照フレーム内のオブジェクトを示すブロックは、隣接フレーム内で繰り返し記述される必要はない。具体的には、テーブルのようなオブジェクトは、複数のフレームにわたって一定の位置に留まり得る。したがって、テーブルは一回記述され、隣接フレームは参照フレームに戻って参照できる。複数のフレームにわたってオブジェクトをマッチングするために、パターンマッチングメカニズムが使用されてもよい。さらに、移動するオブジェクトは、例えば、オブジェクトの動き又はカメラの動きのため、複数のフレームを横切って表されてもよい。特定の例として、ビデオは、複数のフレームにわたってスクリーンを横切って移動する自動車を示してもよい。このような動きを記述するために、動きベクトルが使用できる。動きベクトルは、フレーム内のオブジェクトの座標から参照フレーム内のオブジェクトの座標へのオフセットを提供する二次元ベクトルである。したがって、インター予測は、参照フレーム内の対応するブロックからのオフセットを示す動きベクトルのセットとして、現在のフレーム内の画像ブロックを符号化できる。

20

【0053】

イントラ予測は、共通のフレーム内のブロックを符号化する。イントラ予測は、ルマ成分及びクロマ成分がフレーム内で集中する傾向があるという事実を利用する。例えば、樹木の一部における緑色の断片は、同様の緑色の断片に隣接して位置する傾向がある。イントラ予測は、複数の方向予測モード(例えば、HEVCにおける33個)、プラナーモード及び直流(direct current, DC)モードを使用する。方向モードは、現在のブロックが、対応する方向の隣接ブロックのサンプルと同様/同じであることを示す。プラナーモードは、行/列に沿った一連のブロック(例えば、平面)が、行の端における隣接ブロックに基づいて補間できることを示す。プラナーモードは、実質的に、値を変化させる際に比較的一定の傾きを使用することにより、行/列を横切る光/色の滑らかな遷移を示す。DCモードは境界平滑化のために使用され、ブロックが方向予測モードの角度方向に関連する全ての隣接ブロックのサンプルに関連する平均値と同様/同じであることを示す。したがって、イントラ予測ブロックは、実際の値の代わりに、様々な関係予測モード値として画像ブロックを表すことができる。さらに、インター予測ブロックは、実際の値の代わりに、動きベクトル値として画像ブロックを表すことができる。いずれの場合でも、予測ブロックは、いくつかの場合に画像ブロックを正確に表さないことがある。いずれかの差は残差ブロックに記憶される。ファイルを更に圧縮するために、変換が残差ブロックに適用されてもよい。

30

40

【0054】

ステップ107において、様々なフィルタリング技術が適用されてもよい。HEVCでは、フィルタは、ループ内フィルタリング方式に従って適用される。上記のブロックベースの予測は、デコーダにおいてブロック状の画像の生成を生じることがある。さらに、ブロックベースの予測方式は、ブロックを符号化し、次いで、参照ブロックとして後で使用する

50

ために、符号化ブロックを再現してもよい。ループ内フィルタリング方式は、ノイズ抑制フィルタ、デブロッキングフィルタ、適応ループフィルタ及びサンプル適応オフセット(sample adaptive offset, SAO)フィルタをブロック/フレームに繰り返し適用する。これらのフィルタは、符号化ファイルが正確に再現できるように、このようなブロッキングアーチファクトを軽減する。さらに、これらのフィルタは、アーチファクトが再現された参照ブロックに基づいて符号化される後続のブロック内に更なるアーチファクトを作成する可能性が低くなるように、再現された参照ブロック内のアーチファクトを軽減する。

【0055】

一旦ビデオ信号が区分され、圧縮され、フィルタリングされると、ステップ109において、結果のデータはビットストリームに符号化される。ビットストリームは、上記のデータと、デコーダにおける適切なビデオ信号の再現をサポートするのに望ましいいずれかのシグナリングデータとを含む。例えば、このようなデータは、区分データ、予測データ、残差ブロック、及びコーディング命令をデコーダに提供する様々なフラグを含んでもよい。ビットストリームは、要求によってデコーダに向けて伝送するためにメモリに記憶されてもよい。ビットストリームはまた、複数のデコーダに向けてブロードキャスト及び/又はマルチキャストされてもよい。ビットストリームの作成は反復プロセスである。したがって、ステップ101、103、105、107及び109は、多くのフレーム及びブロックにわたって連続的に及び/又は同時に発生してもよい。図1に示す順序は、説明の明確さ及び容易さのために提示されており、ビデオコーディングプロセスを特定の順序に限定することを意図しない。

【0056】

デコーダはビットストリームを受信し、ステップ111において復号プロセスを開始する。具体的には、デコーダは、ビットストリームを対応するシンタックス及びビデオデータに変換するためにエントロピー復号方式を使用する。ステップ111において、デコーダは、フレームの区分を決定するために、ビットストリームからのシンタックスデータを使用する。区分は、ステップ103におけるブロック区分の結果と一致すべきである。ここで、ステップ111において使用されるエントロピー符号化/復号について説明する。エンコーダは、入力画像内の値の空間位置決めに基づいていくつかの可能な選択肢からブロック区分方式を選択することのように、圧縮プロセスの間に多くの選択を行う。正確な選択肢をシグナリングすることは、多数のピンを使用することがある。ここで使用されるとき、ピンは、変数として扱われるバイナリ値(例えば、コンテキストに依存して変化し得るビット値)である。エントロピーコーディングは、エンコーダが、特定の場合に明らかに実行不可能ないずれかのオプションを破棄することを可能にし、許容可能なオプションのセットを残す。次いで、それぞれの許容可能なオプションには、コードワードが割り当てられる。コードワードの長さは、許容可能なオプションの数に基づく(例えば、2つのオプションについて1つのピン、3つ~4つのオプションについて2つのピン等)。次いで、エンコーダは、選択されたオプションについてコードワードを符号化する。この方式は、全ての可能なオプションの潜在的に大きいセットからの選択を一意に示すこととは対照的に、コードワードが、許容可能なオプションの小さいサブセットからの選択を一意に示すことが望ましいほどの大きさになるので、コードワードのサイズを低減する。次いで、デコーダは、エンコーダと同様の方式で許容可能なオプションのセットを決定することにより、選択を復号する。許容可能なオプションのセットを決定することにより、デコーダは、コードワードを読み取り、エンコーダにより行われる選択を決定できる。

【0057】

ステップ113において、デコーダはブロック復号を実行する。具体的には、デコーダは、残差ブロックを生成するために逆変換を使用する。次いで、デコーダは、区分に従って画像ブロックを再現するために、残差ブロック及び対応する予測ブロックを使用する。予測ブロックは、ステップ105においてエンコーダで生成されたようにイントラ予測ブロック及びインター予測ブロックの双方を含んでもよい。次いで、再現された画像ブロックは、ステップ111において決定された区分データに従って、再現されたビデオ信号のフレー

10

20

30

40

50

ムに位置決めされる。ステップ113についてのシンタックスはまた、上記のようにエンクロピーコーディングを介してビットストリームでシグナリングされてもよい。

【0058】

ステップ115において、エンコーダにおけるステップ107と同様の方式で、再現されたビデオ信号のフレームに対してフィルタリングが実行される。例えば、ブロッキングアーチファクトを除去するために、ノイズ抑制フィルタ、デブロッキングフィルタ、適応ループフィルタ及びSAOフィルタがフレームに適用されてもよい。一旦フレームがフィルタリングされると、ビデオ信号は、エンドユーザにより見るために、ステップ117においてディスプレイに出力できる。

【0059】

図2は、ビデオコーディングのための例示的なコーディング及び復号(コーデック)システム200の概略図である。具体的には、コーデックシステム200は、動作方法100の実現をサポートするための機能を提供する。コーデックシステム200は、エンコーダ及びデコーダの双方で使用されるコンポーネントを示すように一般化されている。コーデックシステム200は、動作方法100におけるステップ101及び103に関して説明したように、ビデオ信号を受信及び区分し、これは、区分されたビデオ信号201を生じる。次いで、コーデックシステム200は、方法100におけるステップ105、107及び109に関して説明したように、エンコーダとして機能するとき、区分されたビデオ信号201をコーディングされたビットストリームに圧縮する。デコーダとして機能するとき、コーデックシステム200は、動作方法100におけるステップ111、113、115及び117に関して説明したように、ビットストリームから出力ビデオ信号を生成する。コーデックシステム200は、全体コーデック制御コンポーネント211と、変換スケーリング及び量子化コンポーネント213と、イントラピクチャ推定コンポーネント215と、イントラピクチャ予測コンポーネント217と、動き補償コンポーネント219と、動き推定コンポーネント221と、スケーリング及び逆変換コンポーネント229と、フィルタ制御解析コンポーネント227と、ループ内フィルタコンポーネント225と、復号ピクチャバッファコンポーネント223と、ヘッダフォーマット及びコンテキスト適応型バイナリ算術コーディング(context adaptive binary arithmetic coding, CABAC)コンポーネント231とを含む。このようなコンポーネントは、図示のように結合される。図2において、黒線は符号化/復号されるべきデータの移動を示し、破線は他のコンポーネントの動作を制御する制御データの移動を示す。コーデックシステム200のコンポーネントは、全てエンコーダに存在してもよい。デコーダは、コーデックシステム200のコンポーネントのサブセットを含んでもよい。例えば、デコーダは、イントラピクチャ予測コンポーネント217と、動き補償コンポーネント219と、スケーリング及び逆変換コンポーネント229と、ループ内フィルタコンポーネント225と、復号ピクチャバッファコンポーネント223とを含んでもよい。ここで、これらのコンポーネントについて説明する。

【0060】

区分されたビデオ信号201は、コーディングツリーにより画素のブロックに区分されている、キャプチャされたビデオシーケンスである。コーディングツリーは、画素のブロックを画素のより小さいブロックに細分するために、様々な分割モードを使用する。次いで、これらのブロックは、より小さいブロックに更に細分できる。ブロックは、コーディングツリー上のノードと呼ばれてもよい。より大きい親ノードは、より小さい子ノードに分割される。ノードが細分される回数は、ノード/コーディングツリーの深さと呼ばれる。いくつかの場合、分割されたブロックは、コーディングユニット(coding unit, CU)に含まれることができる。例えば、CUは、CUについての対応するシンタックス命令と共に、ルマブロック、赤色差クロマ(Cr)ブロック及び青色差クロマ(Cb)ブロックを含むCTUのサブ部分とすることができる。分割モードは、使用される分割モードに依存して、変化する形状のそれぞれ2つ、3つ又は4つの子ノードにノードを区分するために使用される二分木(binary tree, BT)、三分木(triple tree, TT)及び四分木(quad tree, QT)を含んでもよい。区分されたビデオ信号201は、圧縮のために、全体コーデック制御コンポーネント21

10

20

30

40

50

1、変換スケーリング及び量子化コンポーネント213、イントラピクチャ推定コンポーネント215、フィルタ制御解析コンポーネント227及び動き推定コンポーネント221に転送される。

【0061】

全体コーデック制御コンポーネント211は、アプリケーションの制約に従って、ビットストリームへのビデオシーケンスの画像のコーディングに関連する判断を行うように構成される。例えば、全体コーデック制御コンポーネント211は、再現品質に対するビットレート/ビットストリームサイズの最適化を管理する。このような判断は、記憶空間/帯域幅の可用性及び画像解像度要求に基づいて行われてもよい。全体コーデック制御コンポーネント211はまた、バッファのアンダーラン及びオーバーランの問題を軽減するために、伝送速度を考慮してバッファの利用を管理する。これらの問題を管理するために、全体コーデック制御コンポーネント211は、他のコンポーネントによる区分、予測及びフィルタリングを管理する。例えば、全体コーデック制御コンポーネント211は、解像度を増加させて帯域幅の使用を増加させるために圧縮の複雑さを動的に増加させてもよく、或いは、解像度及び帯域幅の使用を減少させるために圧縮の複雑さを減少させてもよい。したがって、全体コーデック制御コンポーネント211は、ビットレートの懸念とビデオ信号再現品質とのバランスを取るために、コーデックシステム200の他のコンポーネントを制御する。全体コーデック制御コンポーネント211は、他のコンポーネントの動作を制御する制御データを作成する。制御データはまた、デコードにおいて復号するためのパラメータをシグナリングするためのビットストリームに符号化されるために、ヘッダフォーマット及びCABACコンポーネント231に転送される。

【0062】

区分されたビデオ信号201はまた、インター予測のために動き推定コンポーネント221及び動き補償コンポーネント219に送信される。区分されたビデオ信号201のフレーム又はスライスは、複数のビデオブロックに分割されてもよい。動き推定コンポーネント221及び動き補償コンポーネント219は、時間予測を提供するために、1つ以上の参照フレーム内の1つ以上のブロックに対して受信したビデオブロックのインター予測コーディングを実行する。コーデックシステム200は、例えばビデオデータの各ブロックについて適切なコーディングモードを選択するために、複数のコーディングパスを実行してもよい。

【0063】

動き推定コンポーネント221及び動き補償コンポーネント219は、高度に統合されてもよいが、概念的な目的のために別々に図示されている。動き推定コンポーネント221により実行される動き推定は、ビデオブロックについての動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、例えば、予測ブロックに対してコーディングされたオブジェクトの変位を示してもよい。予測ブロックは、画素差に関して、コーディングされるべきブロックに密接にマッチングすることが見出されたブロックである。予測ブロックはまた、参照ブロックとも呼ばれてもよい。このような画素差は、差分絶対値和(sum of absolute difference, SAD)、差分二乗和(sum of square difference, SSD)又は他の差分メトリックにより決定されてもよい。HEVCは、CTU、コーディングツリーブロック(coding tree block, CTB)及びCUを含むいくつかのコーディングされたオブジェクトを使用する。例えば、CTUはCTBに分割でき、次いで、これはCUに含めるためにCBに分割できる。CUは、予測データを含む予測ユニット(prediction, unit PU)及び/又はCUについての変換残差データを含む変換ユニット(transform unit, TU)として符号化できる。動き推定コンポーネント221は、レート歪み最適化プロセスの一部としてレート歪み解析を使用することにより、動きベクトル、PU及びTUを生成する。例えば、動き推定コンポーネント221は、現在のブロック/フレームについての複数の参照ブロック、複数の動きベクトル等を決定してもよく、最良のレート歪み特性を有する参照ブロック、動きベクトル等を選択してもよい。最良のレート歪み特性は、ビデオ再現の品質(例えば、圧縮によるデータ損失の量)とコーディング効率(例えば、最終的な符号化のサイズ)の双方をバランスさせる。

10

20

30

40

50

【 0 0 6 4 】

いくつかの例では、コーデックシステム200は、復号ピクチャバッファコンポーネント223に記憶された参照ピクチャのサブ整数画素位置についての値を計算してもよい。例えば、ビデオコーデックシステム200は、参照ピクチャの1/4画素位置、1/8画素位置又は他の分数画素位置の値を補間してもよい。したがって、動き推定コンポーネント221は、全画素位置及び分数画素位置に対する動き探索を実行し、分数画素精度で動きベクトルを出力してもよい。動き推定コンポーネント221は、PUの位置を参照ピクチャの予測ブロックの位置と比較することにより、インターコーディングされたスライス内のビデオブロックのPUについての動きベクトルを計算する。動き推定コンポーネント221は、符号化のために、計算された動きベクトルを動きデータとしてヘッダフォーマット及びCABACコンポーネント231に出力し、動きを動き補償コンポーネント219に出力する。

10

【 0 0 6 5 】

動き補償コンポーネント219により実行される動き補償は、動き推定コンポーネント221により決定された動きベクトルに基づいて予測ブロックをフェッチ又は生成することを伴ってもよい。同様に、いくつかの例では、動き推定コンポーネント221及び動き補償コンポーネント219は機能的に統合されてもよい。現在のビデオブロックのPUについての動きベクトルを受信すると、動き補償コンポーネント219は、動きベクトルが指す予測ブロックを特定してもよい。次いで、コーディングされている現在のビデオブロックの画素値から予測ブロックの画素値を減算することにより、残差ビデオブロックが形成され、画素差分値を形成する。一般的に、動き推定コンポーネント221は、ルマ成分に対して動き推定を実行し、動き補償コンポーネント219は、クロマ成分及びルマ成分の双方について、ルマ成分に基づいて計算された動きベクトルを使用する。予測ブロック及び残差ブロックは、変換スケールリング及び量子化コンポーネント213に転送される。

20

【 0 0 6 6 】

区分されたビデオ信号201はまた、イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217に送信される。動き推定コンポーネント221及び動き補償コンポーネント219と同様に、イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217は、高度に統合されてもよいが、概念的な目的のために別々に図示されている。イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217は、上記のように、フレームの間で動き推定コンポーネント221及び動き補償コンポーネント219により実行されるインター予測の代わりとして、現在のフレーム内のブロックに対して現在のブロックをイントラ予測する。特に、イントラピクチャ推定コンポーネント215は、現在のブロックを符号化するために使用するイントラ予測モードを決定する。いくつかの例では、イントラピクチャ推定コンポーネント215は、複数のテストされたイントラ予測モードから現在のブロックを符号化するための適切なイントラ予測モードを選択する。次いで、選択されたイントラ予測モードは、符号化のためにヘッダフォーマット及びCABACコンポーネント231に転送される。

30

【 0 0 6 7 】

例えば、イントラピクチャ推定コンポーネント215は、様々なテストされたイントラ予測モードについてレート歪み解析を使用してレート歪み値を計算し、テストされたモードの中で最良のレート歪み特性を有するイントラ予測モードを選択する。レート歪み解析は、一般的に、符号化ブロックを生成するために符号化された元の非符号化ブロックと符号化ブロックとの間の歪み(又はエラー)の量と、符号化ブロックを生成するために使用されるビットレート(例えば、ビット数)とを決定する。イントラピクチャ推定コンポーネント215は、どのイントラ予測モードがブロックについて最良のレート歪み値を示すかを決定するために、様々な符号化ブロックについての歪み及びレートから比を計算する。さらに、イントラピクチャ推定コンポーネント215は、レート歪み最適化(rate-distortion optimization, RDO)に基づく深度モデリングモード(depth modeling mode, DMM)を使用して深度マップの深度ブロックをコーディングするように構成されてもよい。

40

【 0 0 6 8 】

50

イントラピクチャ予測コンポーネント217は、エンコーダで実現されるとき、イントラピクチャ推定コンポーネント215により決定された、選択されたイントラピクチャ予測モードに基づいて、予測ブロックから残差ブロックを生成してもよく、或いは、デコーダで実現されるとき、ビットストリームから残差ブロックを読み取ってもよい。残差ブロックは、行列として表される、予測ブロックと元のブロックとの間の値の差を含む。次いで、残差ブロックは、変換スケーリング及び量子化コンポーネント213に転送される。イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217は、ルマ成分及びクロマ成分の双方に対して実行してもよい。

【0069】

変換スケーリング及び量子化コンポーネント213は、残差ブロックを更に圧縮するように構成される。変換スケーリング及び量子化コンポーネント213は、離散コサイン変換(discrete cosine transform, DCT)、離散サイン変換(discrete sine transform, DST)又は概念的に同様の変換のような変換を残差ブロックに適用し、残差変換係数値を含むビデオブロックを生成する。ウェーブレット変換、整数変換、サブバンド変換又は他のタイプの変換も使用されてもよい。変換は、残差情報を画素値ドメインから周波数ドメインのような変換ドメインに変換してもよい。変換スケーリング及び量子化コンポーネント213はまた、例えば、周波数に基づいて、変換された残差情報をスケーリングするように構成される。このようなスケーリングは、異なる周波数情報が異なる粒度で量子化されるように、スケールファクタを残差情報に適用することを伴い、これは、再現されたビデオの最終的な視覚品質に影響を及ぼしてもよい。変換スケーリング及び量子化コンポーネント213はまた、ビットレートを更に低減するために変換係数を量子化するように構成される。量子化プロセスは、係数の一部又は全部に関連するビット深度を低減してもよい。量子化の程度は、量子化パラメータを調整することにより修正されてもよい。いくつかの例では、次いで、変換スケーリング及び量子化コンポーネント213は、量子化された変換係数を含む行列のスキャンを実行してもよい。量子化された変換係数は、ビットストリームに符号化されるためにヘッダフォーマット及びCABACコンポーネント231に転送される。

【0070】

スケーリング及び逆変換コンポーネント229は、動き推定をサポートするために変換スケーリング及び量子化コンポーネント213の逆の動作を適用する。スケーリング及び逆変換コンポーネント229は、例えば、他の現在のブロックについての予測ブロックになり得る参照ブロックとして後で使用するために、画素ドメインにおける残差ブロックを再現するために逆スケーリング、変換及び/又は量子化を適用する。動き推定コンポーネント221及び/又は動き補償コンポーネント219は、後のブロック/フレームの動き推定において使用するために、残差ブロックを対応する予測ブロックに戻して加算することにより参照ブロックを計算してもよい。フィルタは、スケーリング、量子化及び変換の間に作成されたアーチファクトを軽減するために、再現された参照ブロックに適用される。そうでなければ、このようなアーチファクトは、後続のブロックが予測されるときに不正確な予測を引き起こす(且つ更なるアーチファクトを作成する)可能性がある。

【0071】

フィルタ制御解析コンポーネント227及びループ内フィルタコンポーネント225は、フィルタを残差ブロック及び/又は再現された画像ブロックに適用する。例えば、スケーリング及び逆変換コンポーネント229からの変換された残差ブロックは、元の画像ブロックを再現するために、イントラピクチャ予測コンポーネント217及び/又は動き補償コンポーネント219からの対応する予測ブロックと結合されてもよい。次いで、フィルタは、再現された画像ブロックに適用されてもよい。いくつかの例では、フィルタは、代わりに残差ブロックに適用されてもよい。図2における他のコンポーネントと同様に、フィルタ制御解析コンポーネント227及びループ内フィルタコンポーネント225は、高度に統合されており、一緒に実現されてもよいが、概念的な目的のために別々に図示されている。再現された参照ブロックに適用されるフィルタは、特定の空間領域に適用され、このようなフィルタがどのように適用されるかを調整するために複数のパラメータを含む。フィルタ制御解

10

20

30

40

50

析コンポーネント227は、このようなフィルタがどこに適用されるべきかを決定するために、再現された参照ブロックを解析し、対応するパラメータを設定する。このようなデータは、符号化のためのフィルタ制御データとしてヘッダフォーマット及びCABACコンポーネント231に転送される。ループ内フィルタコンポーネント225は、フィルタ制御データに基づいてこのようなフィルタを適用する。フィルタは、デブロッキングフィルタ、ノイズ抑制フィルタ、SAOフィルタ及び適応ループフィルタを含んでもよい。このようなフィルタは、例に依存して、空間/画素ドメイン(例えば、再現された画素ブロック)又は周波数ドメインにおいて適用されてもよい。

【0072】

エンコーダとして動作するとき、フィルタリング後の再現された画像ブロック、残差ブロック及び/又は予測ブロックは、上記のように動き推定において後で使用するために、復号ピクチャバッファコンポーネント223に記憶される。デコーダとして動作するとき、復号ピクチャバッファコンポーネント223は、再現及びフィルタリングされたブロックを記憶し、出力ビデオ信号の一部としてディスプレイに向けて転送する。復号ピクチャバッファコンポーネント223は、予測ブロック、残差ブロック及び/又は再現された画像ブロックを記憶できるいずれかのメモリデバイスでもよい。

【0073】

ヘッダフォーマット及びCABACコンポーネント231は、コーデックシステム200の様々なコンポーネントからデータを受信し、デコーダに向けて伝送するために、このようなデータをコーディングされたビットストリームに符号化する。具体的には、ヘッダフォーマット及びCABACコンポーネント231は、全体制御データ及びフィルタ制御データのような制御データを符号化するための様々なヘッダを生成する。さらに、イントラ予測及び動きデータを含む予測データ並びに量子化された変換係数データの形式の残差データは、全てビットストリームに符号化される。最終的なビットストリームは、元の区分されたビデオ信号201を再現するためにデコーダにより望まれる全ての情報を含む。このような情報はまた、イントラ予測モードインデックステーブル(コードワードマッピングテーブルとも呼ばれる)、様々なブロックについての符号化コンテキストの定義、最確イントラ予測モードの指示、区分情報の指示等を含んでもよい。このようなデータは、エントロピーコーディングを使用することにより符号化されてもよい。例えば、コンテキスト適応型可変長コーディング(context adaptive variable length coding, CAVLC)、CABAC、シンタクススペースのコンテキスト適応型バイナリ算術コーディング(syntax-based context-adaptive binary arithmetic coding, SBAC)、確率区間区分エントロピー(probability interval partitioning entropy, PIPE)コーディング又は他のエントロピーコーディング技術を使用することにより、情報が符号化されてもよい。エントロピーコーディングに続いて、コーディングされたビットストリームは、他のデバイス(例えば、ビデオデコーダ)に伝送されてもよく、或いは、後の伝送又は取得のためにアーカイブされてもよい。

【0074】

図3は、例示的なビデオエンコーダ300を示すブロック図である。ビデオエンコーダ300は、コーデックシステム200の符号化機能を実現するため、及び/又は動作方法100のステップ101、103、105、107及び/又は109を実現するために使用されてもよい。エンコーダ300は、入力ビデオ信号を区分し、区分されたビデオ信号301を生じ、これは、区分されたビデオ信号201と実質的に同様である。次いで、区分されたビデオ信号301は、エンコーダ300のコンポーネントにより圧縮され、ビットストリームに符号化される。

【0075】

具体的には、区分されたビデオ信号301は、イントラ予測のためにイントラピクチャ予測コンポーネント317に転送される。イントラピクチャ予測コンポーネント317は、イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217と実質的に同様でもよい。区分されたビデオ信号301はまた、復号ピクチャバッファコンポーネント323内の参照ブロックに基づくインター予測のために、動き補償コンポーネント32

10

20

30

40

50

1に転送される。動き補償コンポーネント321は、動き推定コンポーネント221及び動き補償コンポーネント219と実質的に同様でもよい。イントラピクチャ予測コンポーネント317及び動き補償コンポーネント321からの予測ブロック及び残差ブロックは、残差ブロックの変換及び量子化のために変換及び量子化コンポーネント313に転送される。変換及び量子化コンポーネント313は、変換スケールリング及び量子化コンポーネント213と実質的に同様でもよい。変換及び量子化された残差ブロック並びに対応する予測ブロックは(関連する制御データと共に)、ビットストリームにコーディングするためにエントロピーコーディングコンポーネント331に転送される。エントロピーコーディングコンポーネント331は、ヘッダフォーマット及びCABACコンポーネント231と実質的に同様でもよい。

【0076】

変換及び量子化された残差ブロック及び/又は対応する予測ブロックはまた、動き補償コンポーネント321により使用するための参照ブロックに再現するために、変換及び量子化コンポーネント313から逆変換及び量子化コンポーネント329に転送される。逆変換及び量子化コンポーネント329は、スケールリング及び逆変換コンポーネント229と実質的に同様でもよい。ループ内フィルタコンポーネント325内のループ内フィルタはまた、例に依存して、残差ブロック及び/又は再現された参照ブロックに適用される。ループ内フィルタコンポーネント325は、フィルタ制御解析コンポーネント227及びループ内フィルタコンポーネント225と実質的に同様でもよい。ループ内フィルタコンポーネント325は、ループ内フィルタコンポーネント225に関して説明したように、複数のフィルタを含んでもよい。次いで、フィルタリングされたブロックは、動き補償コンポーネント321による参照ブロックとして使用するために、復号ピクチャバッファコンポーネント323に記憶される。復号ピクチャバッファコンポーネント323は、復号ピクチャバッファコンポーネント223と実質的に同様でもよい。

【0077】

図4は、例示的なビデオデコーダ400を示すブロック図である。ビデオデコーダ400は、コーデックシステム200の復号機能を実現するため、及び/又は動作方法100のステップ111、113、115及び/又は117を実現するために使用されてもよい。デコーダ400は、例えば、エンコーダ300からビットストリームを受信し、エンドユーザに表示するために、ビットストリームに基づいて再現された出力ビデオ信号を生成する。

【0078】

ビットストリームは、エントロピー復号コンポーネント433により受信される。エントロピー復号コンポーネント433は、CAVLC、CABAC、SBAC、PIPEコーディング又は他のエントロピーコーディング技術のようなエントロピー復号方式を実現するように構成される。例えば、エントロピー復号コンポーネント433は、ビットストリーム内のコードワードとして符号化された更なるデータを解釈するためのコンテキストを提供するために、ヘッダ情報を使用してよい。復号情報は、全体制御データ、フィルタ制御データ、区分情報、動きデータ、予測データ及び残差ブロックからの量子化された変換係数のような、ビデオ信号を復号するためのいずれかの所望の情報を含む。量子化された変換係数は、残差ブロックへの再現のために、逆変換及び量子化コンポーネント429に転送される。逆変換及び量子化コンポーネント429は、逆変換及び量子化コンポーネント329と同様でもよい。

【0079】

再現された残差ブロック及び/又は予測ブロックは、イントラ予測動作に基づいて画像ブロックに再現するために、イントラピクチャ予測コンポーネント417に転送される。イントラピクチャ予測コンポーネント417は、イントラピクチャ推定コンポーネント215及びイントラピクチャ予測コンポーネント217と同様でもよい。具体的には、イントラピクチャ予測コンポーネント417は、フレーム内の参照ブロックを特定するために予測モードを使用し、イントラ予測画像ブロックを再現するために残差ブロックをその結果に適用する。再現されたイントラ予測画像ブロック及び/又は残差ブロック並びに対応するインター予測データは、ループ内フィルタコンポーネント425を介して復号ピクチャバッファコンポ

10

20

30

40

50

ーメント423に転送され、これらのコンポーネントは、復号ピクチャバッファコンポーネント223及びループ内フィルタコンポーネント225とそれぞれ実質的に同様でもよい。ループ内フィルタコンポーネント425は、再現された画像ブロック、残差ブロック及び/又は予測ブロックをフィルタリングし、このような情報が復号ピクチャバッファコンポーネント423に記憶される。復号ピクチャバッファコンポーネント423からの再現された画像ブロックは、インター予測のために動き補償コンポーネント421に転送される。動き補償コンポーネント421は、動き推定コンポーネント221及び/又は動き補償コンポーネント219と実質的に同様でもよい。具体的には、動き補償コンポーネント421は、予測ブロックを生成するために参照ブロックからの動きベクトルを使用し、画像ブロックを再現するために残差ブロックをその結果に適用する。結果の再現されたブロックはまた、ループ内フィルタコンポーネント425を介して復号ピクチャバッファコンポーネント423に転送されてもよい。復号ピクチャバッファコンポーネント423は、更なる再現された画像ブロックを記憶し続け、これらは、区分情報を介してフレームに再現できる。このようなフレームはまた、シーケンスに配置されてもよい。シーケンスは、再現された出力ビデオ信号としてディスプレイに向けて出力される。

【0080】

図5Aは、サブピクチャ510に区分された例示的なピクチャ500を示す概略図である。例えば、ピクチャ500は、コーデックシステム200及び/又はエンコーダ300により符号化するために区分でき、コーデックシステム200及び/又はデコーダ400により復号するために区分できる。他の例として、ピクチャ500は、ステップ111においてデコーダにより使用するために、方法100のステップ103においてエンコーダにより区分されてもよい。

【0081】

ピクチャ500は、指定の時間位置におけるビデオシーケンスの完全な視覚部分を示す画像である。ピクチャ500はまた、画像及び/又はフレームとも呼ばれてもよい。ピクチャ500は、ピクチャ順序カウント(picture order count, POC)により指定されてもよい。POCは、ビデオシーケンスにおけるピクチャ500の出力/表示順序を示すインデックスである。ピクチャ500は、サブピクチャ510に区分できる。サブピクチャ510は、ピクチャ500内の1つ以上のスライス/タイルグループの長方形又は正方形の領域である。サブピクチャ510は任意選択であり、したがって、いくつかのビデオシーケンスはサブピクチャ510を含むが、他のものは含まない。4つのサブピクチャ510が示されているが、ピクチャ500は如何なる数のサブピクチャ510にも区分できる。サブピクチャ510の区分は、全体のコーディングされたビデオシーケンス(coded video sequence)にわたって一貫してもよい。

【0082】

サブピクチャ510は、ピクチャ500の異なる領域が異なって扱われることを可能にするために使用されてもよい。例えば、指定のサブピクチャ510は、独立して抽出され、デコーダに伝送されてもよい。具体的な例として、仮想現実(virtual reality, VR)ヘッドセットを使用するユーザは、ピクチャ500のサブセットを見てもよく、これは、ピクチャ500に示されるような空間に物理的に存在するという印象をユーザに提供してもよい。このような場合、ユーザに表示される可能性のあるサブピクチャ510のみをストリーミングすることは、コーディング効率を増加させ得る。他の例として、異なるサブピクチャ510は、特定のアプリケーションにおいて異なって扱われてもよい。具体的な例として、テレビ会議アプリケーションは、現在話していないユーザよりも目立つ位置でより高い解像度において活動的な話者を表示してもよい。異なるサブピクチャ510内で異なるユーザを位置決めすることは、この機能をサポートするために表示画像のリアルタイムの再構成をサポートする。

【0083】

各サブピクチャ510は、一意のサブピクチャIDにより識別でき、これは、全体のCVSについて一貫していてもよい。例えば、ピクチャ500の左上のサブピクチャ510は、0のサブピクチャIDを有してもよい。このような場合、シーケンス内のいずれかのピクチャ500

10

20

30

40

50

の左上サブピクチャ510は、0のサブピクチャIDにより参照できる。さらに、各サブピクチャ510は、定義された構成を含んでもよく、これは、全体のCVSについて一貫していてもよい。例えば、サブピクチャ510は、高さ、幅及び/又はオフセットを含んでもよい。高さ及び幅は、サブピクチャ510のサイズを記述し、オフセットは、サブピクチャ510の位置を記述する。例えば、行におけるサブピクチャ510の全ての幅の和は、ピクチャ500の幅である。さらに、列におけるサブピクチャ510の全ての高さの和は、ピクチャ500の高さである。さらに、オフセットは、ピクチャ500の左上角に対するサブピクチャ510の左上角の位置を示す。サブピクチャ510の高さ、幅及びオフセットは、ピクチャ500内で対応するサブピクチャ510を位置決めするのに十分な情報を提供する。サブピクチャ510の区分は、全体のCVSにわたって一貫していてもよいので、サブピクチャに関連するパラメータは、シーケンスパラメータセット(sequence parameter set, SPS)に含まれてもよい。

10

【0084】

図5Bは、スライス515に区分された例示的なサブピクチャ510を示す概略図である。図示のように、ピクチャ500のサブピクチャ510は、1つ以上のスライス515を含んでもよい。スライス515は、単一のネットワーク抽象化レイヤ(network abstraction layer, NAL)ユニットに排他的に含まれるピクチャのタイル内の整数個の完全なタイル又は整数個の連続する完全なCTU行である。4つのスライス515が示されているが、サブピクチャ510は如何なる数のスライス515を含んでもよい。スライス515は、指定のPOCのピクチャ500に特有の視覚データを含む。したがって、スライス515に関連するパラメータは、ピクチャパラメータセット(picture parameter set, PPS)及び/又はスライスヘッダに含まれてもよい。

20

【0085】

図5Cは、タイル517に区分された例示的なスライス515を示す概略図である。図示のように、ピクチャ500のスライス515は、1つ以上のタイル517を含んでもよい。タイル517は、ピクチャ500を長方形の行及び列に区分することにより作成されてもよい。したがって、タイル517は、ピクチャ内の特定のタイル列及び特定のタイル行内のCTUの長方形又は正方形の領域である。タイル化は任意選択であり、したがって、いくつかのビデオシーケンスはタイル517を含むが、他のものは含まない。4つのタイル517が示されているが、スライス515は如何なる数のタイル517を含んでもよい。タイル517は、指定のPOCのピクチャ500のスライス515に特有の視覚データを含んでもよい。いくつかの場合、スライス515はまた、タイル517に含まれてもよい。したがって、タイル517に関連するパラメータは、PPS及び/又はスライスヘッダに含まれてもよい。

30

【0086】

図5Dは、CTU519に区分された例示的なスライス515を示す概略図である。図示のように、ピクチャ500のスライス515(又はスライス515のタイル517)は、1つ以上のCTU519を含んでもよい。CTU519は、符号化/復号されるコーディングブロックを作成するためにコーディングツリーにより細分されたピクチャ500の領域である。CTU519は、モノクロピクチャ500のためのルマサンプル、又はカラーピクチャ500のためのルマサンプル及びクロマサンプルの組み合わせを含んでもよい。コーディングツリーにより区分できるルマサンプル又はクロマサンプルのグループ化は、コーディングツリーブロック(coding tree block, CTB)518と呼ばれる。したがって、CTU519は、ルマサンプルのCTB518と、3つのサンプル配列を有するピクチャ500のクロマサンプルの2つの対応するCTB518とを含むか、或いは、モノクロピクチャ、又は3つの別々の色平面及びサンプルをコーディングするために使用されるシンタックス構造を使用してコーディングされたピクチャのサンプルのCTB518を含む。

40

【0087】

上記に示すように、ピクチャ500は、サブピクチャ510、スライス515、タイル517、CTU519及び/又はCTB518に区分されてもよく、次いで、これらはブロックに区分される。次いで、このようなブロックは、デコーダに向けた伝送のために符号化される。このよ

50

うなブロックを復号することは、様々なタイプのノイズを含む復号画像を生じることがある。このような問題を修正するために、ビデオコーディングシステムは、ブロック境界を横切って様々なフィルタを適用してもよい。これらのフィルタは、ブロッキング、量子化ノイズ及び他の望ましくないコーディングアーチファクトを除去できる。上記のように、サブピクチャ510は、独立した抽出を実行するときにも使用されてもよい。この場合、現在のサブピクチャ510は、他のサブピクチャ510からの情報を復号せずに復号されて表示されてもよい。したがって、サブピクチャ510エッジに沿ったブロック境界は、サブピクチャ境界と整列してもよい。いくつかの場合、ブロック境界はまた、タイル境界と整列してもよい。フィルタは、このようなブロック境界を横切って適用されてもよく、したがって、サブピクチャ境界及び/又はタイル境界を横切って適用されてもよい。これは、隣接サブピクチャ510からのデータが利用不可能であるときにフィルタリングプロセスが予期しない方式で動作することがあるので、現在のサブピクチャ510が独立して抽出されるときにエラーを引き起こすことがある。

【0088】

これらの問題に対処するために、サブピクチャ510レベルでフィルタリングを制御するフラグが使用されてもよい。例えば、フラグはloop_filter_across_subpic_enabled_flagとして示されてもよい。サブピクチャ510についてフラグが設定されたとき、フィルタは、対応するサブピクチャ境界を横切って適用できる。フラグが設定されていないとき、フィルタは、対応するサブピクチャ境界を横切って適用されない。このように、フィルタは、別々の抽出のために符号化されるサブピクチャ510についてオフにすることができ、或いは、グループとしての表示のために符号化されるサブピクチャ510についてオンにすることができる。タイル517レベルでフィルタリングを制御するために、他のフラグが設定できる。フラグは、loop_filter_across_tiles_enabled_flagとして示されてもよい。タイル517についてフラグが設定されたとき、フィルタは、タイル境界を横切って適用できる。フラグが設定されていないとき、フィルタは、タイル境界を横切って適用されない。このように、フィルタは、(例えば、タイルの内部をフィルタリングし続けつつ)タイル境界で使用するためにオフ又はオンにすることができる。ここで使用されるとき、フィルタは、フィルタが境界の両側のサンプルに適用されるとき、サブピクチャ510又はタイル517境界を横切って適用される。

【0089】

また、上記のように、タイル化は任意選択である。しかし、いくつかのビデオコーディングシステムは、サブピクチャ510に含まれるタイル517に関してサブピクチャ境界を記述する。このようなシステムでは、タイル517に関するサブピクチャ境界記述は、サブピクチャ510の使用を、タイル517を使用するピクチャ500に制限する。サブピクチャ510の適用性を拡大するために、サブピクチャ510は、CTB518及び/又はCTU519に関して、境界に関して記述されてもよい。具体的には、サブピクチャ510の幅及び高さは、CTB518の単位でシグナリングできる。さらに、サブピクチャ510の左上CTU519の位置は、CTB518で測定されるようにピクチャ500の左上CTU519からのオフセットとしてシグナリングできる。CTU519及びCTB518サイズは、所定の値に設定されてもよい。したがって、CTB518及びCTU519に関してサブピクチャ寸法及び位置をシグナリングすることは、デコーダが表示のためにサブピクチャ510を位置決めするのに十分な情報を提供する。これは、タイル517が使用されないときであっても、サブピクチャ510が使用されることを可能にする。

【0090】

さらに、いくつかのビデオコーディングシステムは、スライス515を、ピクチャ500に対するこれらの位置に基づいてアドレス指定する。これは、サブピクチャ510が独立した抽出及び表示のためにコーディングされるときに問題を生じる。このような場合、省略されたサブピクチャ510に関連するスライス515及び対応するアドレスも省略される。スライス515のアドレスの省略は、デコーダがスライス515を適切に位置決めすることを妨げることがある。いくつかのビデオコーディングシステムは、スライス515に関連するスラ

10

20

30

40

50

イスヘッダ内のアドレスを動的に書き換えることにより、この問題に対処する。ユーザは、いずれかのサブピクチャを要求してもよいので、このような書き換えは、ユーザがビデオを要求する毎に発生し、これは極めてリソース集約的である。この問題を克服するために、スライス515は、サブピクチャ510が使用されるとき、スライス515を含むサブピクチャ510に対してアドレス指定される。例えば、スライス515は、スライス515を含むサブピクチャ510に特有のインデックス又は他の値により識別できる。スライスアドレスは、スライス515に関連するスライスヘッダにコーディングできる。スライス515を含むサブピクチャ510のサブピクチャIDもまた、スライスヘッダに符号化できる。さらに、サブピクチャ510の寸法/構成は、サブピクチャIDと共にSPSにコーディングできる。したがって、デコーダは、完全なピクチャ500を参照することなく、サブピクチャIDに基づいてSPSからサブピクチャ510構成を取得し、スライス515をサブピクチャ510に位置決めすることができる。したがって、サブピクチャ510が抽出されるときに、スライスヘッダの書き換えが省略でき、これは、エンコーダ、デコーダ及び/又は対応するスライサにおけるリソース使用をかなり低減する。

【0091】

一旦ピクチャ500がCTB518及び/又はCTU519に区分されると、CTB518及び/又はCTU519はコーディングブロックに更に分割できる。次いで、コーディングブロックは、イントラ予測及び/又はインター予測に従ってコーディングできる。本開示はまた、インター予測メカニズムに関連する改良を含む。インター予測は、一方向インター予測及び/又は双方向インター予測に従って動作できるいくつかの異なるモードで実行できる。

【0092】

図6は、例えば、ブロック圧縮ステップ105、ブロック復号ステップ113、動き推定コンポーネント221、動き補償コンポーネント219、動き補償コンポーネント321及び/又は動き補償コンポーネント421において、動きベクトル(motion vector, MV)を決定するために実行される、一方向インター予測600の例を示す概略図である。例えば、一方向インター予測600は、ピクチャ500のようなピクチャを区分するとき作成される符号化及び/又は復号ブロックについての動きベクトルを決定するために使用できる。

【0093】

一方向インター予測600は、現在のフレーム610内の現在のブロック611を予測するために、参照ブロック631を有する参照フレーム630を使用する。参照フレーム630は、図示のように、(例えば、後続の参照フレームとして)現在のフレーム610の後に時間的に配置されてもよいが、いくつかの例では、(例えば、先行の参照フレームとして)現在のフレーム610の前に時間的に配置されてもよい。現在のフレーム610は、特定の時間に符号化/復号される例示的なフレーム/ピクチャである。現在のフレーム610は、参照フレーム630の参照ブロック631内のオブジェクトとマッチングする現在のブロック611内のオブジェクトを含む。参照フレーム630は、現在のフレーム610を符号化するための参照として使用されるフレームであり、参照ブロック631は、現在のフレーム610の現在のブロック611にも含まれるオブジェクトを含む参照フレーム630内のブロックである。

【0094】

現在のブロック611は、コーディングプロセスにおける指定の点で符号化/復号されているいずれかのコーディングユニットである。現在のブロック611は、全体の区分されたブロックでもよく、或いは、アフィンインター予測モードを使用するときのサブブロックでもよい。現在のフレーム610は、何らかの時間距離(temporal distance, TD)633だけ参照フレーム630から分離している。TD633は、ビデオシーケンスにおいて、現在のフレーム610と参照フレーム630との間の時間量を示し、フレームの単位で測定されてもよい。現在のブロック611についての予測情報は、フレームの間の方向及び時間距離を示す参照インデックスにより、参照フレーム630及び/又は参照ブロック631を参照してもよい。TD633により表される期間にわたって、現在のブロック611内のオブジェクトは、現在のフレーム610内の位置から参照フレーム630内の他の位置(例えば、参照ブロック631の位置)に移動する。例えば、オブジェクトは、時間と共にオブジェクトの動きの方向である動

10

20

30

40

50

き軌跡613に沿って移動してもよい。動きベクトル635は、TD633にわたる動き軌跡613に沿ったオブジェクトの動きの方向及び大きさを記述する。したがって、符号化された動きベクトル635、参照ブロック631、及び現在のブロック611と参照ブロック631との間の差を含む残差は、現在のブロック611を再現して現在のフレーム610内で現在のブロック611を位置決めするのに十分な情報を提供する。

【0095】

図7は、例えば、ブロック圧縮ステップ105、ブロック復号ステップ113、動き推定コンポーネント221、動き補償コンポーネント219、動き補償コンポーネント321及び/又は動き補償コンポーネント421において、MVを決定するために実行される、双方向インター予測700の例を示す概略図である。例えば、双方向インター予測700は、ピクチャ500

10

【0096】

双方向インター予測700は、一方向インター予測600と同様であるが、現在のフレーム710内の現在のブロック711を予測するために一對の参照フレームを使用する。したがって、現在のフレーム710及び現在のブロック711は、それぞれ現在のフレーム610及び現在のブロック611と実質的に同様である。現在のフレーム710は、ビデオシーケンスにおいて現在のフレーム710の前に生じる先行の参照フレーム720と、ビデオシーケンスにおいて現在のフレーム710の後に生じる後続の参照フレーム730との間に時間的に配置される。先行の参照フレーム720及び後続の参照フレーム730は、他の点では参照フレーム630と実質的に同様である。

20

【0097】

現在のブロック711は、先行の参照フレーム720内の先行の参照ブロック721と、後続の参照フレーム730内の後続の参照ブロック731とマッチングされる。このようなマッチングは、ビデオシーケンスの過程において、オブジェクトが、先行の参照ブロック721の位置から動き軌跡713に沿って現在のブロック711を介して後続の参照ブロック731の位置に移動することを示す。現在のフレーム710は、何らかの先行の時間距離(TD0)723だけ先行の参照フレーム720から分離しており、何らかの後続の時間距離(TD1)733だけ後続の参照フレーム730から分離している。TD0 723は、フレームの単位で、ビデオシーケンスにおける先行の参照フレーム720と現在のフレーム710との間の時間量を示す。TD1 733は、フレームの単位で、ビデオシーケンスにおける現在のフレーム710と後続の参照フレーム730との間の時間量を示す。したがって、オブジェクトは、TD0 723により示される期間にわたって、動き軌跡713に沿って先行の参照ブロック721から現在のブロック711に移動する。オブジェクトはまた、TD1 733により示される期間にわたって、動き軌跡713に沿って現在のブロック711から後続の参照ブロック731に移動する。現在のブロック711についての予測情報は、フレームの間の方向及び時間距離を示す一對の参照インデックスにより、先行の参照フレーム720及び/又は先行の参照ブロック721と、後続の参照フレーム730及び/又は後続の参照ブロック731とを参照してもよい。

30

【0098】

先行の動きベクトル(MV0)725は、TD0 723にわたる(例えば、先行の参照フレーム720と現在のフレーム710との間の)動き軌跡713に沿ったオブジェクトの動きの方向及び大きさを記述する。後続の動きベクトル(MV1)735は、TD1 733にわたる(例えば、現在のフレーム710と後続の参照フレーム730との間の)動き軌跡713に沿ったオブジェクトの動きの方向及び大きさを記述する。したがって、双方向インター予測700において、現在のブロック711は、先行の参照ブロック721及び/又は後続の参照ブロック731、MV0 725及びMV1 735を使用することによりコーディング及び再現できる。

40

【0099】

マージモード及び高度動きベクトル予測(advanced motion vector prediction, AMVP)モードの双方において、候補リストは、候補リスト決定パターンにより定義された順に候補動きベクトルを候補リストに追加することにより生成される。このような候補動

50

きベクトルは、一方向インター予測600、双方向インター予測700又はこれらの組み合わせによる動きベクトルを含んでもよい。具体的には、動きベクトルは、隣接ブロックについて、このようなブロックが符号化されるときに生成される。このような動きベクトルは、現在のブロックについての候補リストに追加され、現在のブロックについての動きベクトルは、候補リストから選択される。次いで、動きベクトルは、候補リスト内の選択された動きベクトルのインデックスとしてシグナリングできる。デコーダは、エンコーダと同じプロセスを使用して候補リストを構築でき、シグナリングされたインデックスに基づいて候補リストから選択された動きベクトルを決定できる。したがって、候補動きベクトルは、このような隣接ブロックが符号化されるときにどの手法が使用されるかに依存して、一方向インター予測600及び/又は双方向インター予測700に従って生成された動きベクトルを含む。

10

【0100】

図8は、隣接するコーディングされたブロック802からの候補動きベクトルに基づいて現在のブロック801をコーディングする例800を示す概略図である。エンコーダ300及び/又はデコーダ400の動作方法100、及び/又はコーデックシステム200の機能の使用は、候補リストを生成するために隣接ブロック802を使用できる。このような候補リストは、一方向インター予測600及び/又は双方向インター予測700によるインター予測において使用できる。次いで、候補リストは、現在のブロック801を符号化/復号するために使用でき、これは、ピクチャ500のようなピクチャを区分することにより生成されてもよい。

【0101】

現在のブロック801は、例に依存して、指定の時間においてエンコーダで符号化されているか或いはデコーダで復号されているブロックである。コーディングされたブロック802は、指定の時間において既に符号化されているブロックである。したがって、コーディングされたブロック802は、候補リストを生成するとき使用するのに潜在的に利用可能である。現在のブロック801及びコーディングされたブロック802は、共通のフレームに含まれてもよく、及び/又は、時間的に隣接するフレームに含まれてもよい。コーディングされたブロック802が現在のブロック801と共に共通のフレームに含まれるとき、コーディングされたブロック802は、現在のブロック801の境界にすぐ隣接する(例えば、接触する)境界を含む。コーディングされたブロック802が時間的に隣接するフレームに含まれるとき、コーディングされたブロック802は、時間的に隣接するフレーム内で、現在のフレーム内の現在のブロック801の位置と同じ位置に位置する。候補リストは、コーディングされたブロック802からの動きベクトルを候補動きベクトルとして追加することにより生成できる。次いで、候補リストから候補動きベクトルを選択し、選択された候補動きベクトルのインデックスをシグナリングすることにより、現在のブロック801がコーディングできる。

20

30

【0102】

図9は、動きベクトルの候補リストを決定するための例示的なパターン900を示す概略図である。具体的には、エンコーダ300及び/又はデコーダ400の動作方法100、及び/又はコーデックシステム200の機能の使用は、ピクチャ500から区分された現在のブロック801を符号化するために、候補リスト911を生成する際に使用する候補リスト決定パターン900を使用できる。結果の候補リスト911は、マージ候補リスト又はAMVP候補リストでもよく、これは、一方向インター予測600及び/又は双方向インター予測700によるインター予測において使用できる。

40

【0103】

現在のブロック901を符号化するとき、候補リスト決定パターン900は、有効な候補動きベクトルを求めて、現在のブロック901と同じピクチャ/フレーム内で、A0、A1、B0、B1及び/又はB2として示される位置905を探索する。候補リスト決定パターン900はまた、有効な候補動きベクトルを求めて、同一位置のブロック909を探索してもよい。同一位置のブロック909は、現在のブロック901と同じ位置にあるブロックであるが、時間的に隣接するピクチャ/フレームに含まれる。次いで、候補動きベクトルは、所定の検査順に候

50

補リスト911に配置できる。したがって、候補リスト911は、インデックス付き候補動きベクトルの手続き的に生成されたリストである。

【0104】

候補リスト911は、現在のブロック901についてインター予測を実行するための動きベクトルを選択するために使用できる。例えば、エンコーダは、候補リスト911からの候補動きベクトルにより指される参照ブロックのサンプルを取得できる。次いで、エンコーダは、現在のブロック901に最も近くマッチングする参照ブロックを指す候補動きベクトルを選択できる。次いで、選択された候補動きベクトルのインデックスは、現在のブロック901を表すように符号化できる。いくつかの場合、候補動きベクトルは、部分的な参照サンプル915を含む参照ブロックを指す。この場合、動きベクトル選択をサポートするための完全な参照サンプル915を再現するために、補間フィルタ913が使用できる。補間フィルタ913は、信号をアップサンプリング可能なフィルタである。具体的には、補間フィルタ913は、部分的な/より低品質の信号を入力として受け付け、より完全な/より高品質の信号の近似値を決定することが可能なフィルタである。したがって、補間フィルタ913は、特定の場合に、現在のブロック901についての参照ブロックを選択する際に、したがって、現在のブロック901を符号化するための動きベクトルを選択する際に使用するための、参照サンプル915の完全なセットを取得するために使用できる。

10

【0105】

候補リストを使用することによりインター予測に基づいてブロックをコーディングするための上記のメカニズムは、サブピクチャ510のようなサブピクチャが使用される時、特定のエラーを引き起こすことがある。具体的には、現在のブロック901が現在のサブピクチャに含まれるが、動きベクトルが、隣接サブピクチャに少なくとも部分的に位置する参照ブロックを指すとき、問題が生じることがある。このような場合、現在のサブピクチャは、隣接サブピクチャなしで提示するために抽出されてもよい。これが生じるとき、隣接サブピクチャ内の参照ブロックの部分はデコーダに伝送されなくてもよく、したがって、参照ブロックは現在のブロック901を復号するために利用可能でないことがある。これが生じるとき、デコーダは、現在のブロック901を復号するのに十分なデータへのアクセスを有さない。

20

【0106】

本開示は、この問題に対処するためのメカニズムを提供する。一例では、現在のサブピクチャがピクチャとして扱われるべきであることを示すフラグが使用される。このフラグは、サブピクチャの別々の抽出をサポートするように設定できる。具体的には、フラグが設定されたとき、現在のサブピクチャは、他のサブピクチャ内のデータを参照することなく符号化されるべきである。この場合、現在のサブピクチャが他のサブピクチャとは別々にコーディングされ、別々のピクチャとして表示できるという点で、現在のサブピクチャはピクチャのように扱われる。したがって、このフラグは、`subpic_treated_as_pic_flag[i]`として示されてもよく、`i`は現在のサブピクチャのインデックスである。フラグが設定されたとき、同一位置のブロック909から取得された動きベクトル候補(動きベクトル予測子としても知られる)は、現在のサブピクチャ内を指す動きベクトルのみを含む。現在のサブピクチャ外を指すいずれかの動きベクトル予測子は、候補リスト911から除外される。これは、現在のサブピクチャ外を指す動きベクトルが選択されず、関連するエラーが回避されることを確保する。この例は、特に、同一位置のブロック909からの動きベクトルに当てはまる。同じピクチャ/フレーム内の探索位置905からの動きベクトルは、以下に記載のように、別々のメカニズムにより補正されてもよい。

30

40

【0107】

他の例は、現在のサブピクチャがピクチャとして扱われるとき(例えば、`subpic_treated_as_pic_flag[i]`が設定されたとき)、探索位置905をアドレス指定するために使用されてもよい。現在のサブピクチャがピクチャのように扱われるとき、現在のサブピクチャは、他のサブピクチャを参照することなく抽出されるべきである。例示的なメカニズムは、補間フィルタ913に関する。補間フィルタ913は、他の位置における関連するサンプルを

50

補間(例えば、予測)するために、1つの位置におけるサンプルに適用できる。本例では、探索位置905におけるコーディングされたブロックからの動きベクトルは、補間フィルタ913が現在のサブピクチャからの参照サンプル915のみに基づいて現在のサブピクチャ外の参照サンプル915を補間できる限り、このような参照サンプル915を指してもよい。したがって、本例は、同じピクチャからの探索位置905からの動きベクトル候補に補間フィルタ913を適用するときに適用されるクリッピング機能を使用する。このクリッピング機能は、隣接サブピクチャからのデータをクリッピングし、したがって、動きベクトル候補により指される参照サンプル915を決定するとき、補間フィルタ913への入力としてこのようなデータを除去する。この手法は、サブピクチャがピクチャとして扱われるとき、別々の抽出及び復号をサポートするように、符号化の間にサブピクチャの間の分離を維持する。クリッピング機能は、ルマサンプルバイリニア補間プロセス、ルマサンプル8タップ補間フィルタリングプロセス及び/又はクロマサンプル補間プロセスに適用されてもよい。

10

【0108】

図10は、例示的なループ内フィルタ1000を示すブロック図である。ループ内フィルタ1000は、ループ内フィルタ225、325及び/又は425を実現するために使用されてもよい。さらに、ループ内フィルタ1000は、方法100を実行するときにエンコーダ及びデコーダにおいて適用されてもよい。さらに、ループ内フィルタ1000は、ピクチャ500から区分された現在のブロック801をフィルタリングするために適用でき、これは、パターン900に従って生成された候補リストに基づいて、一方向インター予測600及び/又は双方向インター予測700に従ってコーディングされてもよい。ループ内フィルタ1000は、デブロッキングフィルタ1043、SAOフィルタ1045及び適応ループフィルタ(adaptive loop filter, ALF)1047を含む。ループ内フィルタ1000のフィルタは、(例えば、参照ブロックとして使用する前に)エンコーダにおいて且つ表示する前にデコーダにおいて、再現された画像ブロックに順に適用される。

20

【0109】

デブロッキングフィルタ1043は、ブロックベースのインター及びイントラ予測により作成されたブロック形状のエッジを除去するように構成される。デブロッキングフィルタ1043は、区分境界で生じるクロマ及び/又はルマ値の不連続性を求めて画像部分(例えば、画像スライス)をスキャンする。次いで、デブロッキングフィルタ1043は、このような不連続性を除去するために、平滑化機能をブロック境界に適用する。デブロッキングフィルタ1043の強度は、ブロック境界に隣接する領域で生じる空間的アクティビティ(例えば、ルマ/クロマ成分の分散)に依存して変更されてもよい。

30

【0110】

SAOフィルタ1045は、符号化プロセスにより引き起こされるサンプル歪みに関連するアーチファクトを除去するように構成される。エンコーダにおけるSAOフィルタ1045は、相対的なデブロッキングエッジ形状及び/又は方向に基づいて、再現された画像のデブロッキングされたサンプルをいくつかのカテゴリに分類する。次いで、カテゴリに基づいてオフセットが決定され、サンプルに追加される。次いで、オフセットはビットストリームに符号化され、デコーダにおいてSAOフィルタ1045により使用される。SAOフィルタ1045は、バンディングアーチファクト(banding artefact)(滑らかな遷移ではなく値の帯)及びリングングアーチファクト(ringing artefact)(鋭いエッジ近くのスプリアス信号)を除去する。

40

【0111】

ALF1047は、エンコーダにおいて、再現された画像を元の画像と比較するように構成される。ALF1047は、例えば、ウィナー(Wiener)ベースの適応フィルタを介して、再現された画像と元の画像との間の差を記述する係数を決定する。このような係数は、ビットストリームに符号化され、再現された画像と元の画像との間の差を除去するためにデコーダにおけるALF1047により使用される。

【0112】

ループ内フィルタ1000によりフィルタリングされた画像データは、復号ピクチャパッ

50

ファ223、323及び/又は423と実質的に同様のピクチャバッファ1023に出力される。上記のように、デブロッキングフィルタ1043、SAOフィルタ1045及び/又はALF1047は、それぞれloop_filter_across_subpic_enabledフラグ及び/又はloop_filter_across_tiles_enabled_flagのようなフラグにより、サブピクチャ境界及び/又はタイル境界でオフにすることができる。

【0113】

図11は、ピクチャのサブピクチャを復号することをサポートするためのコーディングツールパラメータを含む例示的なビットストリーム1100を示す概略図である。例えば、ビットストリーム1100は、コーデックシステム200及び/又はデコーダ400により復号するために、コーデックシステム200及び/又はエンコーダ300により生成できる。他の例として、ビットストリーム1100は、ステップ111においてデコーダにより使用するために、方法100のステップ109においてエンコーダにより生成されてもよい。さらに、ビットストリーム1100は、符号化ピクチャ500、対応するサブピクチャ510、及び/又は現在のブロック801及び/又は901のような関連するコーディングされたブロックを含んでもよく、これは、パターン900に従って生成された候補リストに基づいて一方向インター予測600及び/又は双方向インター予測700に従ってコーディングされてもよい。ビットストリーム1100はまた、ループ内フィルタ1000を構成するためのパラメータを含んでもよい。

【0114】

ビットストリーム1100は、シーケンスパラメータセット(sequence parameter set, SPS)1110、複数のピクチャパラメータセット(picture parameter set, PPS)1111、複数のスライスヘッダ1115及び画像データ1120を含む。SPS1110は、ビットストリーム1100に含まれるビデオシーケンス内の全てのピクチャに共通のシーケンスデータを含む。このようなデータは、ピクチャサイズ、ビット深度、コーディングツールパラメータ、ビットレート制限等を含むことができる。PPS1111は、全体のピクチャに適用するパラメータを含む。したがって、ビデオシーケンス内の各ピクチャは、PPS1111を参照してもよい。各ピクチャはPPS1111を参照するが、いくつかの例では、単一のPPS1111は複数のピクチャのためのデータを含むことができる点に留意すべきである。例えば、複数の同様のピクチャは、同様のパラメータに従ってコーディングされてもよい。このような場合、単一のPPS1111は、このような同様のピクチャのためのデータを含んでもよい。PPS1111は、対応するピクチャ内のスライスに利用可能なコーディングツール、量子化パラメータ、オフセット等を示すことができる。スライスヘッダ1115は、ピクチャ内の各スライスに特有のパラメータを含む。したがって、ビデオシーケンスに、スライス当たり1つのスライスヘッダ1115が存在してもよい。スライスヘッダ1115は、スライスタイプ情報、ピクチャ順序カウント(picture order count, POC)、参照ピクチャリスト、予測重み、タイルエントリポイント、デブロッキングパラメータ等を含んでもよい。スライスヘッダ1115はまた、いくつかの状況においてタイルグループヘッダとも呼ばれてもよい点に留意すべきである。

【0115】

画像データ1120は、インター予測及び/又はイントラ予測に従って符号化されたビデオデータと、対応する変換及び量子化された残差データとを含む。例えば、ビデオシーケンスは、画像データとしてコーディングされた複数のピクチャを含む。ピクチャは、ビデオシーケンスの単一のフレームであり、したがって、ビデオシーケンスを表示するときに、一般的に単一のユニットとして表示される。しかし、仮想現実、ピクチャ・イン・ピクチャ等のような特定の技術を実現するために、サブピクチャが表示されてもよい。ピクチャはPPS1111をそれぞれ参照する。ピクチャは、上記のように、サブピクチャ、タイル及び/又はスライスに分割される。いくつかのシステムでは、スライスは、タイルを含むタイルグループと呼ばれる。スライス及び/又はタイルのタイルグループは、スライスヘッダ1115を参照する。スライスは、CTU及び/又はCTBに更に分割される。CTU/CTBは、コーディングツリーに基づいてコーディングブロックに更に分割される。次いで、コーディングブロックは、予測メカニズムに従って符号化/復号できる。

10

20

30

40

50

【 0 1 1 6 】

ビットストリーム1100内のパラメータセットは、ここに記載の例を実現するために使用できる様々なデータを含む。第1の例の実現方式をサポートするために、ビットストリーム1100のSPS1110は、指定のサブピクチャに関する、ピクチャとして扱われるサブピクチャのフラグ1131を含む。いくつかの例では、ピクチャとして扱われるサブピクチャのフラグ1131は、subpic_treated_as_pic_flag[i]として示され、iはフラグに関連するサブピクチャのインデックスである。例えば、ピクチャとして扱われるサブピクチャのフラグ1131は、(画像データ1120内の)コーディングされたビデオシーケンス内のそれぞれのコーディングされたピクチャの第iのサブピクチャが、ループ内フィルタリング動作を除く復号プロセスにおいてピクチャとして扱われることを指定するために、1に等しく設定されてもよい。ピクチャとして扱われるサブピクチャのフラグ1131は、現在のピクチャ内の現在のサブピクチャがインター予測に従ってコーディングされているときに使用されてもよい。ピクチャとして扱われるサブピクチャのフラグ1131が、現在のサブピクチャがピクチャとして扱われることを示すように設定されたとき、現在のブロックについての候補動きベクトルの候補リストは、候補リストから、同一位置のブロックに含まれ且つ現在のサブピクチャ外を指す同一位置の動きベクトルを除外することにより決定できる。これは、現在のサブピクチャが他のサブピクチャから別々に抽出されるとき、現在のサブピクチャ外を指す動きベクトルが選択されず、関連するエラーが回避されることを確保する。

10

【 0 1 1 7 】

いくつかの例では、現在のブロックについての動きベクトルの候補リストは、時間ルマ動きベクトル予測に従って決定される。例えば、現在のブロックがルマサンプルのルマブロックであり、現在のブロックについての選択された現在の動きベクトルが参照ブロック内の参照ルマサンプルを指す時間ルマ動きベクトルであり、現在のブロックが参照ルマサンプルに基づいてコーディングされるとき、時間ルマ動きベクトル予測が使用されてもよい。このような場合、時間ルマ動きベクトル予測は以下に従って実行される。

20

```
xColBr=xCb+cbWidth;
```

```
yColBr=yCb+cbHeight;
```

```
rightBoundaryPos=subpic_treated_as_pic_flag[SubPicIdx]?
```

```
SubPicRightBoundaryPos:pic_width_in_luma_samples-1;且つ
```

```
botBoundaryPos=subpic_treated_as_pic_flag[SubPicIdx]?
```

30

```
SubPicBotBoundaryPos:pic_height_in_luma_samples-1
```

ここで、xColBr及びyColBRは同一位置のブロックの位置を指定し、xCb及びyCbは現在のピクチャの左上サンプルに対する現在のブロックの左上サンプルを指定し、cbWidthは現在のブロックの幅であり、cbHeightは現在のブロックの高さであり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、pic_width_in_luma_samplesはルマサンプルで測定された現在のピクチャの幅であり、pic_height_in_luma_samplesはルマサンプルで測定された現在のピクチャの高さであり、botBoundaryPosはサブピクチャの下境界の計算された位置であり、rightBoundaryPosはサブピクチャの右境界の計算された位置であり、SubPicIdxはサブピクチャのインデックスであり、yCb CtbLog2SizeYがyColBr CtbLog2SizeYに等しくないとき、同一位置の動きベクトルは除外され、CtbLog2SizeYはコーディングツリーブロックのサイズを示す。

40

【 0 1 1 8 】

ピクチャとして扱われるサブピクチャのフラグ1131はまた、第2の例の実現方式のために使用されてもよい。第1の例のように、ピクチャとして扱われるサブピクチャのフラグ1131は、現在のピクチャ内の現在のサブピクチャがインター予測に従ってコーディングされているときに使用されてもよい。この例では、サブピクチャの現在のブロックについて(例えば、候補リストから)動きベクトルが決定できる。ピクチャとして扱われるサブピクチャのフラグ1131が設定されたとき、クリッピング機能が参照ブロック内のサンプル位置に適用できる。サンプル位置は、ルマ値及び/又は一対のクロマ値を含む単一のサンプルを

50

含むことができるピクチャ内の位置である。次いで、動きベクトルが現在のサブピクチャ外を指すとき、補間フィルタが適用できる。このクリッピング機能は、別々の抽出をサポートするためのサブピクチャの間の分離を維持するために、補間フィルタが隣接サブピクチャからのデータに依存しないことを確保する。

【0119】

クリッピング機能は、ルマサンプルバイリニア補間プロセスにおいて適用できる。ルマサンプルバイリニア補間プロセスは、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信してもよい。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLXL)を出力する。クリッピング機能は、以下のようにサンプル位置に適用される。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。

xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i)、且つ
yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i)
ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能である。

【数7】

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

ここで、x、y及びzは数値入力値である。

【0120】

クリッピング機能はまた、ルマサンプル8タップ補間フィルタリングプロセスにおいて適用できる。ルマサンプル8タップ補間フィルタリングプロセスは、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信する。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLXL)を出力する。クリッピング機能は、以下のようにサンプル位置に適用される。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。

xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3)、且つ
yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3)
ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は上記の通りである。

【0121】

クリッピング機能はまた、クロマサンプル補間プロセスにおいて適用できる。クロマサンプル補間プロセスは、全サンプルユニット(xIntC,yIntC)におけるクロマ位置を含む入力を受信する。クロマサンプル補間プロセスは、予測クロマサンプル値(predSampleLXC)を出力する。クリッピング機能は、以下のようにサンプル位置に適用される。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。

xInti=Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/Sub

10

20

30

40

50

WidthC,xIntC+i)、且つ

$yInti = Clip3(SubPicTopBoundaryPos/SubHeightC, SubPicBotBoundaryPos/SubHeightC, yIntC+i)$

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、SubWidthC及びSubHeightCはルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、Clip3は上記の通りである。

10

【0122】

SPS1110内のサブピクチャを横切るループフィルタ有効化のフラグ1132は、第3の例の実現方式のために使用されてもよい。サブピクチャを横切るループフィルタ有効化のフラグ1132は、フィルタリングが指定のサブピクチャの境界を横切って使用されるか否かを制御するために設定されてもよい。例えば、サブピクチャを横切るループフィルタ有効化のフラグ1132は、loop_filter_across_subpic_enabled_flagとして示されてもよい。サブピクチャを横切るループフィルタ有効化のフラグ1132は、ループ内フィルタリング動作がサブピクチャの境界を横切って実行できることを指定するとき、1に設定されてもよく、或いは、ループ内フィルタリング動作がサブピクチャの境界を横切って実行されないことを指定するとき、0に設定されてもよい。したがって、フィルタリング動作は、サブピクチャを横切るループフィルタ有効化のフラグ1132の値に基づいて、サブピクチャ境界を横切って実行されてもよく或いは実行されなくてもよい。フィルタリング動作は、デブロッキングフィルタ1043、ALF1047及び/又はSAOフィルタ1045を適用することを含んでもよい。このように、フィルタは、別々の抽出のために符号化されるサブピクチャについてオフにすることができ、或いは、グループとしての表示のために符号化されるサブピクチャについてオンにすることができる。

20

【0123】

PPS1111内のタイルを横切るループフィルタ有効化のフラグ1134は、第4の例の実現方式のために使用されてもよい。タイルを横切るループフィルタ有効化のフラグ1134は、フィルタリングが指定のタイルの境界を横切って使用されるか否かを制御するために設定されてもよい。例えば、タイルを横切るループフィルタ有効化のフラグ1134は、loop_filter_across_tiles_enabled_flagとして示されてもよい。タイルを横切るループフィルタ有効化のフラグ1134は、ループ内フィルタリング動作がタイルの境界を横切って実行できることを指定するとき、1に設定されてもよく、或いは、ループ内フィルタリング動作がタイルの境界を横切って実行されないことを指定するとき、0に設定されてもよい。したがって、フィルタリング動作は、タイルを横切るループフィルタ有効化のフラグ1134の値に基づいて、指定のタイル境界を横切って実行されてもよく或いは実行されなくてもよい。フィルタリング動作は、デブロッキングフィルタ1043、ALF1047及び/又はSAOフィルタ1045を適用することを含んでもよい。

30

40

【0124】

SPS1110内のサブピクチャデータ1133は、第5の例の実現方式のために使用されてもよい。サブピクチャデータ1133は、画像データ1120内の各サブピクチャについての幅、高さ及びオフセットを含んでもよい。例えば、各サブピクチャの幅及び高さは、CTBの単位でサブピクチャデータ1133に記述できる。いくつかの例では、サブピクチャの幅及び高さは、それぞれsubpic_width_minus1及びsubpic_height_minus1として、サブピクチャデータ1133に記憶される。さらに、各サブピクチャのオフセットは、CTUの単位でサブピクチャデータ1133に記述できる。例えば、各サブピクチャのオフセットは、サブピクチャの左上CTUの垂直位置及び水平位置として指定できる。具体的には、サブピクチャのオフセットは、ピクチャの左上CTUとサブピクチャの左上CTUとの差として指定でき

50

る。いくつかの例では、サブピクチャの左上CTUの垂直位置及び水平位置は、それぞれsubpic_ctu_top_left_y及びsubpic_ctu_top_left_xとしてサブピクチャデータ1133に記憶される。この例の実現方式は、タイルに関してではなく、CTB/CTUに関してサブピクチャデータ1133内のサブピクチャを記述する。これは、対応するピクチャ/サブピクチャにおいてタイルが使用されていないときであっても、サブピクチャが使用されることを可能にする。

【0125】

SPS1110内のサブピクチャデータ1133、スライスヘッダ1115内のスライスアドレス1136及びスライスヘッダ1115内のスライスサブピクチャID1135は、第6の例の実現方式のために使用されてもよい。サブピクチャデータ1133は、第5の例の実現方式において記載されるように実現されてもよい。スライスアドレス1136は、スライスヘッダ1115に関連するスライス(例えば、画像データ1120)のサブピクチャレベルのスライスインデックスを含んでもよい。例えば、スライスは、ピクチャ内のスライスの位置に基づくのではなく、サブピクチャ内のスライスの位置に基づいてインデックス付けされる。スライスアドレス1136は、slice_address変数に記憶されてもよい。スライスサブピクチャID1135は、スライスヘッダ1115に関連するスライスを含むサブピクチャのIDを含む。具体的には、スライスサブピクチャID1135は、サブピクチャデータ1133内の対応するサブピクチャの記述(例えば、幅、高さ及びオフセット)を参照してもよい。スライスサブピクチャID1135は、slice_subpic_id変数に記憶されてもよい。したがって、スライスアドレス1136は、サブピクチャデータ1133に記述されるように、サブピクチャID1135により示されるサブピクチャ内のスライスの位置に基づくインデックスとしてシグナリングされる。このように、サブピクチャが別々に抽出され、他のサブピクチャがビットストリーム1100から省略されるときであっても、サブピクチャ内のスライスの位置が決定できる。これは、このアドレス指定方式が各サブピクチャのアドレスを他のサブピクチャから分離するからである。したがって、ピクチャ内のスライスの位置に基づいてスライスがアドレス指定されるアドレス指定方式において必要とされるように、スライスヘッダ1115は、サブピクチャが抽出されるときに書き換えられる必要がない。この手法は、スライスが(ラスタスキャンスライスとは対照的に)長方形/正方形スライスであるときに使用されてもよい点に留意すべきである。例えば、PPS1111内のrect_slice_flagは、スライスが長方形スライスであることを示すために、1に等しく設定できる。

【0126】

いくつかのビデオコーディングシステムにおいて使用されるサブピクチャの例示的な実現方式は以下の通りである。CVSに存在し得るサブピクチャに関連する情報は、SPSでシグナリングされてもよい。このようなシグナリングは以下の情報を含んでもよい。CVSの各ピクチャに存在するサブピクチャの数はSPSに含まれてもよい。SPS又はCVSのコンテキストにおいて、全てのアクセスユニット(access unit, AU)についての同一位置のサブピクチャは、併せてサブピクチャシーケンスと呼ばれてもよい。各サブピクチャの特性に関連する情報を更に指定するためのループもまた、SPSに含まれてもよい。このような情報は、サブピクチャ識別、サブピクチャの位置(例えば、サブピクチャの左上角ルマサンプルとピクチャの左上角ルマサンプルとの間のオフセット距離)及びサブピクチャのサイズを含んでもよい。さらに、SPSはまた、サブピクチャのそれぞれが動き制約サブピクチャであるか否かをシグナリングするために使用されてもよく、動き制約サブピクチャは、MCTSを含むサブピクチャである。各サブピクチャについてのプロファイル、階層及びレベル情報は、このような情報が別途導出可能でない限り、ビットストリームに含まれてもよい。このような情報は、全体のピクチャを含む元のビットストリームからサブピクチャを抽出することから作成された抽出ビットストリームについてのプロファイル、階層及びレベル情報に使用されてもよい。各サブピクチャのプロファイル及び階層は、元のプロファイル及び階層と同じであると導出されてもよい。各サブピクチャについてのレベルは、明示的にシグナリングされてもよい。このようなシグナリングは、上記のループに存在してもよい。シーケンスレベルの仮想参照デコーダ(hypothetical reference decoder, HRD)

パラメータは、各サブピクチャ(又は等価的に各サブピクチャシーケンス)についてのSPSのビデオユーザビリティ情報(video usability information, VUI)部分でシグナリングされてもよい。

【0127】

ピクチャが2つ以上のサブピクチャに区分されていないとき、サブピクチャの特性(例えば、位置、サイズ等)は、サブピクチャIDを除いて、ビットストリームでシグナリングされなくてもよい。CVS内のピクチャ内のサブピクチャが抽出されるとき、新たなビットストリーム内の各アクセスユニットはサブピクチャを含まなくてもよい。この理由は、新たなビットストリーム内の各AU内の結果の画像データが複数のサブピクチャに区分されないからである。したがって、位置及びサイズのようなサブピクチャ特性は、このような情報がピクチャ特性から導出できるので、SPSから省略されてもよい。しかし、サブピクチャ識別は、抽出されたサブピクチャに含まれるビデオコーディングレイヤ(video coding layer, VCL)NALユニット/タイルグループによりこのIDが参照されることがあるので、依然としてシグナリングされる。リソース使用を低減するために、サブピクチャを抽出するとき、サブピクチャIDを変更することは回避されるべきである。

10

【0128】

ピクチャ内のサブピクチャの位置(xオフセット及びyオフセット)は、ルマサンプルの単位でシグナリングでき、サブピクチャの左上角ルマサンプルとピクチャの左上角ルマサンプルとの間の距離を表してもよい。他の例では、ピクチャ内のサブピクチャの位置は、最小コーディングルマブロックサイズ(MinCbSizeY)の単位でシグナリングでき、サブピクチャの左上角ルマサンプルとピクチャの左上角ルマサンプルとの間の距離を表してもよい。他の例では、サブピクチャ位置オフセットの単位は、パラメータセット内のシンタックスエレメントにより明示的に示されてもよく、単位は、CtbSizeY、MinCbSizeY、ルマサンプル又は他の値でもよい。コーデックは、サブピクチャの右境界がピクチャの右境界と一致しないとき、サブピクチャの幅がルマCTUサイズ(CtbSizeY)の整数倍とすることを要求してもよい。同様に、コーデックは、サブピクチャの下境界がピクチャの下境界と一致しないとき、サブピクチャの高さがCtbSizeYの整数倍とすることを更に要求してもよい。コーデックはまた、サブピクチャの幅がルマCTUサイズの整数倍でないとき、サブピクチャがピクチャの最も右の位置に位置することを要求してもよい。同様に、コーデックは、サブピクチャの高さがルマCTUサイズの整数倍でないとき、サブピクチャがピクチャの最も下の位置に位置することを要求してもよい。サブピクチャの幅がルマCTUサイズの単位でシグナリングされ、サブピクチャの幅がルマCTUサイズの整数倍でないとき、ルマサンプルにおける実際の幅は、サブピクチャのオフセット位置、ルマCTUサイズにおけるサブピクチャの幅、及びルマサンプルにおけるピクチャの幅に基づいて導出されてもよい。同様に、サブピクチャの高さがルマCTUサイズの単位でシグナリングされ、サブピクチャの高さがルマCTUサイズの整数倍でないとき、ルマサンプルにおける実際の高さは、サブピクチャのオフセット位置、ルマCTUサイズにおけるサブピクチャの高さ、及びルマサンプルにおけるピクチャの高さに基づいて導出できる。

20

30

【0129】

いずれかのサブピクチャについて、サブピクチャIDは、サブピクチャインデックスと異なってもよい。サブピクチャインデックスは、SPS内のサブピクチャのループでシグナリングされるようなサブピクチャのインデックスでもよい。代替として、サブピクチャインデックスは、ピクチャに対するサブピクチャラスタスキャン順に割り当てられたインデックスでもよい。各サブピクチャのサブピクチャIDの値がそのサブピクチャインデックスと同じであるとき、サブピクチャIDは、シグナリング又は導出されてもよい。各サブピクチャのサブピクチャIDがそのサブピクチャインデックスと異なるとき、サブピクチャIDは、明示的にシグナリングされる。サブピクチャIDをシグナリングするためのビット数は、サブピクチャ特性を含む同じパラメータセットで(例えば、SPSで)シグナリングされてもよい。サブピクチャIDについてのいくつかの値は、特定の目的のために留保されてもよい。このような値の留保は以下の通りとすることができる。タイルグループ/スライスヘッダが

40

50

、どのサブピクチャがタイルグループを含むかを指定するためのサブピクチャIDを含むとき、タイルグループ/スライスヘッダの最初の数ビットが全て0でないことを確保し、エミュレーション防止コードを生成することを回避するために、値0が留保されてもよく、サブピクチャに使用されなくてもよい。ピクチャのサブピクチャが重複及びギャップなしにピクチャの全領域をカバーしないとき、値(例えば、値1)は、サブピクチャのいずれの部分でもないタイルグループのために留保されてもよい。代替として、残りの領域のサブピクチャIDが明示的にシグナリングされてもよい。サブピクチャIDをシグナリングするためのビット数は、以下のように制約されてもよい。値の範囲は、サブピクチャIDの留保値を含み、ピクチャ内の全てのサブピクチャを一意に識別するのに十分であるべきである。例えば、サブピクチャIDについての最小ビット数は $\text{Ceil}(\text{Log}_2(\text{ピクチャ内のサブピクチャの数} + \text{留保されたサブピクチャIDの数}))$ の値とすることができる。

10

【0130】

ループ内のサブピクチャのまとまりは、全体のピクチャをギャップなしに且つ重複なしにカバーするために必要とされてもよい。この制約が適用されるとき、サブピクチャが、サブピクチャが抽出できることを意味する動き制約サブピクチャであるか否かを指定するために、各サブピクチャについてフラグが存在する。代替として、サブピクチャのまとまりが全体のピクチャをカバーしなくてもよい。しかし、ピクチャのサブピクチャの間に重複が存在しなくてもよい。

【0131】

サブピクチャIDは、サブピクチャ抽出プロセスを支援するために、抽出器がNALユニットビットの残りを理解する必要がないように、NALユニットヘッダの直後に存在してもよい。VCL NALユニットについて、サブピクチャIDはタイルグループヘッダの最初のビットに存在してもよい。非VCL NALユニットについて、以下の条件が当てはまってもよい。サブピクチャIDは、SPSについてのNALユニットヘッダの直後に存在する必要がなくてもよい。PPSに関して、同じピクチャの全てのタイルグループが同じPPSを参照するように制約されているとき、NALユニットヘッダの直後にサブピクチャIDが存在する必要はない。他方、同じピクチャのタイルグループが異なるPPSを参照することが許容される場合、サブピクチャIDはPPSの最初のビット(例えば、PPS NALユニットヘッダの直後)に存在してもよい。この場合、1つのピクチャの2つの異なるタイルグループが同じPPSを共有することは許容されない。代替として、同じピクチャのタイルグループが異なるPPSを参照することが許容され、同じピクチャの異なるタイルグループも同じPPSを共有することが許容されるとき、サブピクチャIDはPPSシンタックスに存在しない。代替として、同じピクチャのタイルグループが異なるPPSを参照することが許容され、同じピクチャの異なるタイルグループも同じPPSを共有することが許容されるとき、サブピクチャIDのリストがPPSシンタックスに存在する。当該リストは、PPSが適用するサブピクチャを示す。他の非VCL NALユニットについて、非VCLユニットがピクチャレベル(例えば、アクセスユニットデリミタ、シーケンスの終了、ビットストリームの終了等)以上に適用する場合、サブピクチャIDがそのNALユニットヘッダの直後に存在する必要はない。そうでない場合、サブピクチャIDはNALユニットヘッダの直後に存在してもよい。

20

30

【0132】

個々のサブピクチャ内のタイル区分は、PPSでシグナリングされてもよいが、同じピクチャ内のタイルグループは、異なるPPSを参照することが許容される。この場合、タイルは、ピクチャを横切るのではなく、各サブピクチャ内でグループ化される。したがって、このような場合のタイルグループ化の概念は、サブピクチャをタイルに区分することを含む。代替として、サブピクチャパラメータセット(Sub-Picture Parameter Set, SPPS)が、個々のサブピクチャ内のタイル区分を記述するために使用されてもよい。SPPSは、SPS IDを参照するシンタックスエレメントを使用することによりSPSを参照する。SPPSはサブピクチャIDを含んでもよい。サブピクチャ抽出の目的のため、サブピクチャIDを参照するシンタックスエレメントは、SPPS内の最初のシンタックスエレメントである。SPPSは、列の数、行の数、均一なタイル間隔等を示すタイル構造を含む。SPPSは、ループフィ

40

50

ルタが関連するサブピクチャ境界を横切って有効であるか否かを示すためのフラグを含んでもよい。代替として、各サブピクチャについてのサブピクチャ特性は、SPSの代わりにSPPSでシグナリングされてもよい。個々のサブピクチャ内のタイル区分は、PPSでシグナリングされてもよいが、同じピクチャ内のタイルグループは、異なるPPSを参照することが許容される。一旦アクティブにされると、SPPSは、復号順で連続するAUのシーケンスの間にわたって続いてよいが、CVSの開始ではないAUにおいて非アクティブ/アクティブにされてもよい。複数のSPPSは、複数のサブピクチャを有する単一レイヤのビットストリームの復号プロセス中のいずれかの時点でアクティブでもよく、SPPSは、AUの異なるサブピクチャにより共有されてもよい。代替として、SPPS及びPPSは1つのパラメータセットにマージできる。これが生じるために、同じサブピクチャに含まれる全てのタイルグループは、SPPSとPPSとの間のマージから生じる同じパラメータセットを参照するように制約されてもよい。

【0133】

サブピクチャIDをシグナリングするために使用されるビット数は、NALユニットヘッダでシグナリングされてもよい。このような情報は、存在するとき、NALユニットのペイロードの開始(例えば、NALユニットヘッダの直後の最初の数ビット)についてサブピクチャID値を解析する際のサブピクチャ抽出プロセスを支援する。このようなシグナリングのために、NALユニットヘッダ内の留保ビットのいくつかは、NALユニットヘッダの長さを増加させるのを回避するために使用されてもよい。このようなシグナリングのためのビット数は、sub-picture-ID-bit-lenの値をカバーすべきである。例えば、VVCのNALユニットヘッダ内の7つの留保ビットのうち4つのビットがこの目的のために使用されてもよい。

【0134】

サブピクチャを復号するとき、垂直CTB位置(xCtb)及び水平CTB位置(yCtb)として示される各コーディングツリーブロックの位置は、サブピクチャ内のルマサンプル位置ではなく、ピクチャ内の実際のルマサンプル位置に調整される。このように、全てがサブピクチャではなくピクチャに位置するかのよう復号されるので、各参照ピクチャからの同一位置のサブピクチャの抽出は回避できる。コーディングツリーブロックの位置を調整するために、変数SubpictureXOffset及びSubpictureYOffsetは、サブピクチャ位置(subpic_x_offset及びsubpic_y_offset)に基づいて導出される。変数の値は、サブピクチャ内の各コーディングツリーブロックのルマサンプル位置x及びy座標の値にそれぞれ加算される。サブピクチャ抽出プロセスは以下のように定義できる。プロセスへの入力は、抽出されるべきターゲットサブピクチャを含む。これは、サブピクチャID又はサブピクチャ位置の形式で入力できる。入力がサブピクチャの位置であるとき、関連するサブピクチャIDは、SPS内のサブピクチャ情報を解析することにより解決できる。非VCL NALユニットについて、以下が当てはまる。ピクチャサイズ及びレベルに関連するSPS内のシンタックスエレメントは、サブピクチャのサイズ及びレベル情報で更新される。以下の非VCL NALユニット、すなわち、PPS、アクセスユニットデリミタ(access unit delimiter, AUD)、シーケンスの終了(end of sequence, EOS)、ビットストリームの終了(end of bit stream, EOB)、及びピクチャレベル以上に適用可能ないずれかの他の非VCL NALユニットは、抽出によって変更されない。サブピクチャIDがターゲットサブピクチャIDに等しくない残りの非VCL NALユニットは除去される。サブピクチャIDがターゲットサブピクチャIDに等しくないVCL NALユニットも除去される。

【0135】

サブピクチャネスト化SEIメッセージは、サブピクチャのセットについてのAUレベル又はサブピクチャレベルのSEIメッセージのネスト化に使用されてもよい。サブピクチャネスト化SEIメッセージで搬送されるデータは、バッファリング周期、ピクチャタイミング及び非HRD SEIメッセージを含んでもよい。このSEIメッセージのシンタックス及びセマンティクスは以下の通りとすることができる。全方向性メディアフォーマット(omnidirectional media format, OMAF)環境のようなシステム動作のために、ビューポートをカバーするサブピクチャシーケンスのセットが、OMAFプレイヤーにより要求されて復号され

10

20

30

40

50

てもよい。したがって、シーケンスレベルのSEIメッセージは、長方形又は正方形のピクチャ領域を併せて含むサブピクチャシーケンスのセットの情報を搬送してもよい。当該情報はシステムにより使用でき、当該情報は、最小復号能力と、サブピクチャシーケンスのセットのビットレートとを示す。当該情報は、サブピクチャシーケンスのセットのみを含むビットストリームのレベルと、ビットストリームのビットレートと、任意選択でサブピクチャシーケンスのセットについて指定されたサブビットストリーム抽出プロセスとを含む。

【0136】

上記の実現方式は、いくつかの問題を含む。ピクチャの幅及び高さ、及び/又はサブピクチャの幅/高さ/オフセットのシグナリングは効率的ではない。このような情報をシグナリングするために、より多くのビットが節約できる。サブピクチャサイズ及び位置情報がSPSでシグナリングされる時、PPSはタイル構成を含む。さらに、PPSは、同じピクチャの複数のサブピクチャにより共有されることが許容される。したがって、num_tile_columns_minus1及びnum_tile_rows_minus1についての値の範囲はより明確に指定されるべきである。さらに、サブピクチャが動き制約されているか否かを示すフラグのセマンティクスは明確に指定されていない。レベルは、各サブピクチャシーケンスについて必須にシグナリングされる。しかし、サブピクチャシーケンスが独立して復号できないとき、サブピクチャのレベルをシグナリングすることは有用ではない。さらに、いくつかのアプリケーションでは、いくつかのサブピクチャシーケンスは、少なくとも1つの他のサブピクチャシーケンスと一緒に復号されてレンダリングされるべきである。したがって、このようなサブピクチャシーケンスの単一のものについてレベルをシグナリングすることは、有用ではないことがある。さらに、各サブピクチャについてレベル値を決定することは、エンコーダに負担をかけることがある。

【0137】

独立して復号可能なサブピクチャシーケンスの導入によって、ピクチャの特定の領域の独立した抽出及び復号を必要とするシナリオは、タイルグループに基づいて機能しないことがある。したがって、タイルグループIDの明示的なシグナリングは有用ではないことがある。さらに、PPSシンタックスエレメントpps_seq_parameter_set_id及びloop_filter_across_tiles_enabled_flagのそれぞれの値は、コーディングされたピクチャのタイルグループヘッダにより参照される全てのPPSで同じであるべきである。これは、アクティブなSPSがCVS内で変化するべきでなく、loop_filter_across_tiles_enabled_flagの値がタイルに基づいて並列処理するためにピクチャ内の全てのタイルについて同じであるべきであるからである。ピクチャ内で長方形及びラスタスキャンタイルグループの混合を可能にするか否かは明確に指定されるべきである。異なるタイルグループモードを使用するために、異なるピクチャの一部であり且つCVS内で同じサブピクチャIDを使用するサブピクチャを許容するか否かもまた、指定されるべきである。時間ルマ動きベクトル予測のための導出プロセスは、時間動きベクトル予測(temporal motion vector prediction, TMVP)におけるピクチャ境界としてサブピクチャ境界を扱うことを可能にしないことがある。さらに、ルマサンプルバイリニア補間プロセス、ルマサンプル8タップ補間フィルタリングプロセス及びクロマサンプル補間プロセスは、動き補償におけるピクチャ境界としてサブピクチャ境界を扱うように構成されないことがある。また、サブピクチャ境界におけるデブロッキング、SAO及びALFフィルタリング動作の制御のためのメカニズムも指定されるべきである。

【0138】

独立して復号可能なサブピクチャシーケンスの導入によって、loop_filter_across_tile_groups_enabled_flagはあまり有用でないことがある。これは、並列処理の目的でループ内フィルタリング動作をオフにすることが、loop_filter_across_tile_groups_enabled_flagを0に等しく設定することによっても満たされることがあるからである。さらに、ピクチャの特定の領域の独立した抽出及び復号を可能にするためのループ内フィルタリング動作をオフにすることは、loop_filter_across_sub_pic_enabled_flagを0に等し

10

20

30

40

50

く設定することによっても満たされることがある。したがって、`loop_filter_across_tile_groups_enabled_flag`に基づいてタイルグループ境界を横切るループ内フィルタリング動作をオフにするためのプロセスを更に指定することは、不必要にデコーダに負担をかけ、ビットを浪費する。さらに、上記の復号プロセスは、タイル境界を横切るALFフィルタリング動作をオフにすることを可能にしないことがある。

【0139】

したがって、本開示は、サブピクチャベースのビデオコーディングをサポートするための設計を含む。サブピクチャは、ピクチャと同じ復号プロセスを使用して独立して復号されてもよく或いは復号されなくてもよい、ピクチャ内の長方形又は正方形の領域である。技術の説明は、バーサタイルビデオコーディング(Versatile Video Coding, VVC)標準に基づく。しかし、当該技術はまた、他のビデオコーデック仕様に適用してもよい。

10

【0140】

いくつかの例では、ピクチャ幅及び高さのシンタックスエレメントと、サブピクチャ幅/高さ/`offset_x`/`offset_y`のシンタックスエレメントのリストについて、サイズ単位がシグナリングされる。全てのシンタックスエレメントは、`xxx_minus1`の形式でシグナリングされる。例えば、サイズ単位が64ルマサンプルであるとき、99の幅の値は、6400ルマサンプルのピクチャ幅を指定する。同じ例は、これらのシンタックスエレメントの他のものにも当てはまる。他の例では、以下の1つ以上が当てはまる。サイズ単位は、`xxx_minus1`の形式でピクチャ幅及び高さのシンタックスエレメントについてシグナリングされてもよい。サブピクチャ幅/高さ/`offset_x`/`offset_y`のシンタックスエレメントのリストについてシグナリングされるこのようなサイズ単位は、`xxx_minus1`の形式でもよい。他の例では、以下の1つ以上が当てはまる。ピクチャ幅及びサブピクチャ幅/`offset_x`のシンタックスエレメントのリストについてのサイズ単位は、`xxx_minus1`の形式でシグナリングされてもよい。ピクチャ高さ及びサブピクチャ高さ/`offset_y`のシンタックスエレメントのリストについてのサイズ単位は、`xxx_minus1`の形式でシグナリングされてもよい。他の例では、以下の1つ以上が当てはまる。`xxx_minus1`の形式のピクチャ幅及び高さのシンタックスエレメントは、最小コーディングユニットの単位でシグナリングされてもよい。`xxx_minus1`の形式のサブピクチャ幅/高さ/`offset_x`/`offset_y`のシンタックスエレメントは、CTU又はCTBの単位でシグナリングされてもよい。右ピクチャ境界における各サブピクチャについてのサブピクチャ幅は導出されてもよい。下ピクチャ境界における各サブピクチャについてのサブピクチャ高さは導出されてもよい。サブピクチャ幅/高さ/`offset_x`/`offset_y`の他の値は、全てビットストリームでシグナリングされてもよい。他の例では、サブピクチャの幅及び高さ並びにピクチャ内のこれらの位置のシグナリングのためのモードは、サブピクチャが均一なサイズを有する場合について追加されてもよい。サブピクチャは、これらが同じサブピクチャ行及びサブピクチャ列を含むとき、均一なサイズを有する。このモードでは、サブピクチャ行の数、サブピクチャ列の数、各サブピクチャ列の幅及び各サブピクチャ行の高さは、全てシグナリングされてもよい。

20

30

【0141】

他の例では、サブピクチャ幅及び高さのシグナリングは、PPSに含まれなくてもよい。`num_tile_columns_minus1`及び`num_tile_rows_minus1`は、0以上1024のような1つの整数値以下の範囲にあるべきである。他の例では、PPSを参照するサブピクチャが1つよりも多くのタイルを有するとき、存在フラグにより条件付けられた2つのシンタックスエレメントは、PPSでシグナリングされてもよい。これらのシンタックスエレメントは、CTBの単位でサブピクチャ幅及び高さをシグナリングするために使用され、PPSを参照する全てのサブピクチャのサイズを指定する。

40

【0142】

他の例では、個々のサブピクチャを記述する更なる情報もまたシグナリングされてもよい。`sub_pic_treated_as_pic_flag[i]`のようなフラグは、サブピクチャシーケンスのサブピクチャがループ内フィルタリング動作以外の目的で復号プロセスにおいてピクチャとして扱われるか否かを示すために、各サブピクチャシーケンスについてシグナリングされ

50

てもよい。サブピクチャシーケンスが適合するレベルは、sub_pic_treated_as_pic_flag[i]が1に等しいときのみシグナリングされてもよい。サブピクチャシーケンスは、同じサブピクチャIDを有するサブピクチャのCVSである。sub_pic_treated_as_pic_flag[i]が1に等しいとき、サブピクチャシーケンスのレベルもまたシグナリングされてもよい。これは、全てのサブピクチャシーケンスについてのフラグにより、或いは、各サブピクチャシーケンス毎に1つのフラグにより制御できる。他の例では、VCL NALユニットを変更せずにサブビットストリーム抽出が有効にされてもよい。これは、PPSから明示的なタイルグループIDのシグナリングを除去することにより達成できる。tile_group_addressのセマンティクスは、rect_tile_group_flagが長方形タイルグループを示すものと等しいときに指定される。tile_group_addressは、サブピクチャ内のタイルグループの中のタイルグループのタイルグループインデックスを含んでもよい。

10

【0143】

他の例では、PPSシンタックスエレメントpps_seq_parameter_set_id及びloop_filter_across_tiles_enabled_flagのそれぞれの値は、コーディングされたピクチャのタイルグループヘッダにより参照される全てのPPSにおいて同じであるものとする。他のPPSシンタックスエレメントは、コーディングされたピクチャのタイルグループヘッダに参照される異なるPPSについて異なってもよい。single_tile_in_pic_flagの値は、コーディングされたピクチャのタイルグループヘッダに参照される異なるPPSについて異なってもよい。このように、CVS内のいくつかのピクチャは、1つのタイルのみを有してもよいが、CVS内のいくつかの他のピクチャは複数のタイルを有してもよい。これはまた、ピクチャのいくつかのサブピクチャ(例えば、非常に大きいもの)が複数のタイルを有することを可能にし、一方、同じピクチャの他のサブピクチャ(例えば、非常に小さいもの)は1つのタイルのみを有する。

20

【0144】

他の例では、ピクチャは、長方形及びラスタスキャンタイルグループの混合を含んでもよい。したがって、ピクチャのいくつかのサブピクチャは長方形タイルグループモードを使用し、一方、他のサブピクチャはラスタスキャンタイルグループモードを使用する。この柔軟性は、ビットストリームマージシナリオに有益である。代替として、制約は、ピクチャの全てのサブピクチャが同じタイルグループモードを使用するものとするを要求してもよい。CVS内の同じサブピクチャIDを有する異なるピクチャからのサブピクチャは、異なるタイルグループモードを使用しなくてもよい。CVS内の同じサブピクチャIDを有する異なるピクチャからのサブピクチャは、異なるタイルグループモードを使用してもよい。

30

【0145】

他の例では、サブピクチャについてのsub_pic_treated_as_pic_flag[i]が1に等しいとき、サブピクチャについての時間動きベクトル予測のための同一位置の動きベクトルは、サブピクチャの境界内から生じるように制限される。したがって、サブピクチャについての時間動きベクトル予測は、サブピクチャ境界がピクチャ境界であるかのように扱われる。さらに、サブピクチャ境界をsub_pic_treated_as_pic_flag[i]が1に等しいサブピクチャについての動き補償においてピクチャ境界として扱うことを可能にするために、クリッピング動作は、ルマサンプルバイリニア補間プロセス、ルマサンプル8タップ補間フィルタリングプロセス及びクロマサンプル補間プロセスの一部として指定される。

40

【0146】

他の例では、各サブピクチャは、loop_filter_across_sub_pic_enabled_flagのようなシグナリングされたフラグに関連付けられる。フラグは、サブピクチャの境界におけるループ内フィルタリング動作の制御、及び対応する復号プロセスにおけるフィルタリング動作の制御に使用される。デブロッキングフィルタプロセスは、loop_filter_across_sub_pic_enabled_flagが0に等しいサブピクチャの境界と一致するコーディングサブブロックエッジ及び変換ブロックエッジに適用されなくてもよい。代替として、デブロッキングフィルタプロセスは、loop_filter_across_sub_pic_enabled_flagが0に等しいサブピ

50

クチャの上又は左境界と一致するコーディングサブブロックエッジ及び変換ブロックエッジに適用されない。代替として、デブロッキングフィルタプロセスは、sub_pic_treated_as_pic_flag[i]が1又は0に等しいサブピクチャの境界と一致するコーディングサブブロックエッジ及び変換ブロックエッジに適用されない。代替として、デブロッキングフィルタプロセスは、サブピクチャの上又は左境界と一致するコーディングサブブロックエッジ及び変換ブロックエッジに適用されない。クリッピング動作は、サブピクチャについてのloop_filter_across_sub_pic_enabled_flagが0に等しいとき、サブピクチャの境界を横切るSAOフィルタリング動作をオフにするように指定されてもよい。クリッピング動作は、loop_filter_across_sub_pic_enabled_flagがサブピクチャについて0に等しいとき、サブピクチャの境界を横切るALFフィルタリング動作をオフにするように指定されてもよい。loop_filter_across_tile_groups_enabled_flagはまた、PPSから除去されてもよい。したがって、loop_filter_across_tiles_enabled_flagが0に等しいとき、サブピクチャ境界ではないタイルグループ境界を横切るループ内フィルタリング動作はオフにされない。ループフィルタ動作は、デブロッキング、SAO及びALFを含んでもよい。他の例では、クリッピング動作は、タイルについてのloop_filter_across_tiles_enabled_flagが0に等しいとき、タイルの境界を横切るALFフィルタリング動作をオフにするように指定される。

10

【 0 1 4 7 】

上記の例の1つ以上は、以下のように実現されてもよい。サブピクチャは、ピクチャ内の1つ以上のタイルグループ又はスライスの長方形又は正方形の領域として定義されてもよい。処理要素の以下の分割、すなわち、各ピクチャの成分への分割、各成分のCTBへの分割、各ピクチャのサブピクチャへの分割、各サブピクチャのサブピクチャ内のタイル列への分割、各サブピクチャのサブピクチャ内のタイル行への分割、サブピクチャ内の各タイル列のタイルへの分割、サブピクチャ内の各タイル行のタイルへの分割、及び各サブピクチャのタイルグループへの分割は、空間又は成分毎の区分を形成してもよい。

20

【 0 1 4 8 】

サブピクチャ内のCTBラスタ及びタイルスキャンプロセスについてのプロセスは、以下の通りでもよい。CTBの単位で第iのタイル列の幅を指定する、0以上num_tile_columns_minus1以下の範囲のiについてのリストColWidth[i]は、以下のように導出されてもよい。

30

【 数 8 】

```

if( uniform_tile_spacing_flag )
for( i = 0; i <= num_tile_columns_minus1; i++ )
    ColWidth[ i ] = ( ( i + 1 ) * SubPicWidthInCtbsY ) /
( num_tile_columns_minus1 + 1 ) -
    ( i * SubPicWidthInCtbsY ) / ( num_tile_columns_minus1 + 1 )
else {
ColWidth[ num_tile_columns_minus1 ] = SubPicWidthInCtbsY
for( i = 0; i < num_tile_columns_minus1; i++ ) {
    ColWidth[ i ] = tile_column_width_minus1[ i ] + 1
    ColWidth[ num_tile_columns_minus1 ] -= ColWidth[ i ]
}
}

```

40

(6-1)

50

【 0 1 4 9 】

CTBの単位で第jのタイル行の高さを指定する、0以上num_tile_rows_minus1以下の範囲のjについてのリストRowHeight[j]は、以下のように導出される。

【数 9】

```

if( uniform_tile_spacing_flag )
for( j = 0; j <= num_tile_rows_minus1; j++ )
    RowHeight[ j ] = ( ( j + 1 ) * SubPicHeightInCtbsY ) /
( num_tile_rows_minus1 + 1 ) -
    ( j * SubPicHeightInCtbsY ) / ( num_tile_rows_minus1 + 1 )
else {
RowHeight[ num_tile_rows_minus1 ] = SubPicHeightInCtbsY           (6-2)
for( j = 0; j < num_tile_rows_minus1; j++ ) {
    RowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    RowHeight[ num_tile_rows_minus1 ] -= RowHeight[ j ]
}
}

```

【 0 1 5 0 】

CTBの単位で第iのタイル列境界の位置を指定する、0以上num_tile_columns_minus1+1以下の範囲のiについてのリストColBd[i]は、以下のように導出される。

【数 1 0】

```

for( ColBd[ 0 ] = 0, i = 0; i <= num_tile_columns_minus1; i++ )
    ColBd[ i + 1 ] = ColBd[ i ] + ColWidth[ i ]           (6-3)

```

【 0 1 5 1 】

CTBの単位で第jのタイル行境界の位置を指定する、0以上num_tile_rows_minus1+1以下の範囲のjについてのリストRowBd[j]は、以下のように導出される。

【数 1 1】

```

for( RowBd[ 0 ] = 0, j = 0; j <= num_tile_rows_minus1; j++ )
RowBd[ j + 1 ] = RowBd[ j ] + RowHeight[ j ]           (6-4)

```

【 0 1 5 2 】

サブピクチャのCTBラスタスキャンにおけるCTBアドレスからサブピクチャのタイルスキャンにおけるCTBアドレスへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrRsについてのリストCtbAddrRsToTs[ctbAddrRs]は、以下のように導出される。

【数 1 2】

```

for( ctbAddrRs = 0; ctbAddrRs < SubPicSizeInCtbsY; ctbAddrRs++ ) {
  tbX = ctbAddrRs % SubPicWidthInCtbsY
  tbY = ctbAddrRs / SubPicWidthInCtbsY
  for( i = 0; i <= num_tile_columns_minus1; i++ )
    if( tbX >= ColBd[ i ] )
      tileX = i
  for( j = 0; j <= num_tile_rows_minus1; j++ )
    if( tbY >= RowBd[ j ] )
      tileY = j
  CtbAddrRsToTs[ ctbAddrRs ] = 0
  for( i = 0; i < tileX; i++ )
    CtbAddrRsToTs[ ctbAddrRs ] += RowHeight[ tileY ] * ColWidth[ i ]
  for( j = 0; j < tileY; j++ )
    CtbAddrRsToTs[ ctbAddrRs ] += SubPicWidthInCtbsY * RowHeight[ j ]
  CtbAddrRsToTs[ ctbAddrRs ] +=
  ( tbY - RowBd[ tileY ] ) * ColWidth[ tileX ] + tbX - ColBd[ tileX ]
}

```

10

(6-5)

20

【 0 1 5 3 】

タイルスキャンにおけるCTBアドレスからサブピクチャのCTBラストスキャンにおけるCTBアドレスへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrTsについてのリストCtbAddrTsToRs[ctbAddrTs]は、以下のように導出される。

30

【 数 1 3 】

```

for( ctbAddrRs = 0; ctbAddrRs < SubPicSizeInCtbsY; ctbAddrRs++ )
  CtbAddrTsToRs[ CtbAddrRsToTs[ ctbAddrRs ] ] = ctbAddrRs

```

(6-6)

【 0 1 5 4 】

サブピクチャのタイルスキャンにおけるCTBアドレスからタイルIDへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrTsについてのリストTileId[ctbAddrTs]は、以下のように導出される。

【 数 1 4 】

```

for( j = 0, tileIdx = 0; j <= num_tile_rows_minus1; j++ )
  for( i = 0; i <= num_tile_columns_minus1; i++, tileIdx++ )
    for( y = RowBd[ j ]; y < RowBd[ j + 1 ]; y++ )
      for( x = ColBd[ i ]; x < ColBd[ i + 1 ]; x++ )
        TileId[ CtbAddrRsToTs[ y * SubPicWidthInCtbsY + x ] ] = tileIdx

```

(6-7)

40

【 0 1 5 5 】

タイルインデックスからタイル内のCTUの数への変換を指定する、0以上NumTilesInS

50

ubPic-1の範囲のtileIdxについてのリストNumCtusInTile[tileIdx]は、以下のように導出される。

【数 1 5】

```
for(j=0, tileIdx=0; j <= num_tile_rows_minus1; j++)
for(i=0; i <= num_tile_columns_minus1; i++, tileIdx++)           (6-8)
    NumCtusInTile[ tileIdx ] = ColWidth[ i ] * RowHeight[ j ]
```

【 0 1 5 6】

10

タイルIDからタイル内の最初のCTBのタイルスキャンにおけるCTBアドレスへの変換を指定する、0以上NumTilesInSubPic-1以下の範囲のtileIdxについてのリストFirstCtbAddrTs[tileIdx]は、以下のように導出される。

【数 1 6】

```
for( ctbAddrTs = 0, tileIdx = 0, tileStartFlag = 1; ctbAddrTs < SubPicSizeInCtbsY;
ctbAddrTs++) {
if( tileStartFlag ) {
    FirstCtbAddrTs[ tileIdx ] = ctbAddrTs           (6-9)
    tileStartFlag = 0
}
tileEndFlag = ctbAddrTs == SubPicSizeInCtbsY - 1 ||
TileId[ ctbAddrTs + 1 ] != TileId[ ctbAddrTs ]
if( tileEndFlag ) {
    tileIdx++
    tileStartFlag = 1
}
}
```

20

30

【 0 1 5 7】

ルマサンプルの単位で第iのタイル列の幅を指定するColumnWidthInLumaSamples[i]の値は、0以上num_tile_columns_minus1以下の範囲のiについてColWidth[i] CtbLog2SizeYに等しく設定される。ルマサンプルの単位で第jのタイル行の高さを指定するRowHeightInLumaSamples[j]の値は、0以上num_tile_rows_minus1以下の範囲のjについてRowHeight[j] CtbLog2SizeYに等しく設定される。

40

【 0 1 5 8】

例示的なシーケンスパラメータセットRBSPのシンタックスは以下の通りである。

50

【表 1】

seq_parameter_set_rbsp() {	記述子	
sps_max_sub_layers_minus1	u(3)	
sps_reserved_zero_5bits	u(5)	
profile_tier_level(sps_max_sub_layers_minus1)		
sps_seq_parameter_set_id	ue(v)	
chroma_format_idc	ue(v)	
if(chroma_format_idc == 3)		10
separate_colour_plane_flag	u(1)	
pic_width_in_luma_samples	ue(v)	
pic_height_in_luma_samples	ue(v)	
bit_depth_luma_minus8	ue(v)	
bit_depth_chroma_minus8	ue(v)	
num_sub_pics_minus1	ue(v)	
sub_pic_id_len_minus1	ue(v)	
if(num_sub_pics_minus1 > 0)		20
sub_pic_level_present_flag	u(1)	
for (i = 0; i <= num_sub_pics_minus1; i++) {		
sub_pic_id[i]	u(v)	
if(num_subpics_minus1 > 0) {		
sub_pic_treated_as_pic_flag[i]	u(1)	
if(sub_pic_treated_as_pic_flag[i] && sub_pic_level_present_flag)		
sub_pic_level_idc[i]	u(8)	
sub_pic_x_offset[i]	ue(v)	30
sub_pic_y_offset[i]	ue(v)	
sub_pic_width_in_luma_samples[i]	ue(v)	
sub_pic_height_in_luma_samples[i]	ue(v)	
}		
}		
log2_max_pic_order_cnt_lsb_minus4	ue(v)	
...		40

【 0 1 5 9 】

例示的なピクチャパラメータセットRBSPのシンタックスは以下の通りである。

【表 2】

pic_parameter_set_rbsp() {	記述子	
pps_pic_parameter_set_id	ue(v)	
pps_seq_parameter_set_id	ue(v)	
loop_filter_across_sub_pic_enabled_flag	u(1)	
single_tile_in_sub_pic_flag	u(1)	
if(!single_tile_in_sub_pic_flag) {		
num_tile_columns_minus1	ue(v)	10
num_tile_rows_minus1	ue(v)	
uniform_tile_spacing_flag	u(1)	
if(!uniform_tile_spacing_flag) {		
for(i = 0; i < num_tile_columns_minus1; i++)		
tile_column_width_minus1[i]	ue(v)	
for(i = 0; i < num_tile_rows_minus1; i++)		
tile_row_height_minus1[i]	ue(v)	20
}		
single_tile_per_tile_group_flag	u(1)	
if(!single_tile_per_tile_group_flag)		
rect_tile_group_flag	u(1)	
if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {		
num_tile_groups_in_sub_pic_minus1	ue(v)	
for(i = 0; i <= num_tile_groups_in_sub_pic_minus1; i++) {		
if(i > 0)		
top_left_tile_idx[i]	u(v)	30
bottom_right_tile_idx[i]	u(v)	
}		
}		
loop_filter_across_tiles_enabled_flag	u(1)	
}		
for(i = 0; i < 2; i++)		
num_ref_idx_default_active_minus1[i]	ue(v)	40

【 0 1 6 0 】

例示的な一般タイルグループヘッダのシンタックスは以下の通りである。

【表 3】

tile_group_header() {	記述子
tile_group_pic_parameter_set_id	ue(v)
tile_group_sub_pic_id	u(v)
if(NumTilesInSubPic > 1 !(rect_tile_group_flag && num_tile_group_in_sub_pic_minus1 == 0))	
tile_group_address	u(v)

10

【0 1 6 1】

例示的なコーディングツリーユニットのシンタックスは以下の通りである。

【表 4】

coding_tree_unit() {	記述子
xCtb = ((CtbAddrInRs % SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_x_offset[SubPicIdx[tile_group_subpic_id]]	
yCtb = ((CtbAddrInRs / SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_y_offset[SubPicIdx[tile_group_subpic_id]]	
...	

20

【0 1 6 2】

例示的なシーケンスパラメータセットRBSPのセマンティクスは以下の通りである。

【0 1 6 3】

bit_depth_chroma_minus8は、クロマ配列BitDepthCのサンプルのビット深度を指定し、クロマ量子化パラメータ範囲オフセットQpBdOffsetCの値は以下の通りである。

30

【数 1 7】

$$\text{BitDepthC} = 8 + \text{bit_depth_chroma_minus8} \quad (7-4)$$

$$\text{QpBdOffsetC} = 6 * \text{bit_depth_chroma_minus8} \quad (7-5)$$

bit_depth_chroma_minus8は、0以上8以下の範囲にあるものとする。

【0 1 6 4】

num_sub_pics_minus1に1を加えたものは、CVS内のそれぞれのコーディングされたピクチャにおけるサブピクチャの数を指定する。num_sub_pics_minus1の値は、0以上1024以下の範囲にあるものとする。sub_pic_id_len_minus1に1を加えたものは、SPS内のシンタックスエレメントsub_pic_id[i]及びタイルグループヘッダ内のシンタックスエレメントtile_group_sub_pic_idを表すために使用されるビット数を指定する。sub_pic_id_len_minus1の値はCeil(Log2(num_sub_pic_minus1+1))-1以上9以下の範囲にあるものとする。sub_pic_level_present_flagは、シンタックスエレメントsub_pic_level_idc[i]が存在してもよいことを指定するために、1に設定される。sub_pic_level_present_flagは、シンタックスエレメントsub_pic_level_idc[i]が存在しないことを指定するために、0に設定される。sub_pic_id[i]は、CVS内のそれぞれのコーディングされたピクチャの第iのサブピクチャのサブピクチャIDを指定する。sub_pic_id[i]の長さは、sub_pic_id_len_minus1+1ビットである。

40

50

【 0 1 6 5 】

sub_pic_treated_as_pic_flag[i]は、CVS内のそれぞれのコーディングされたピクチャの第iのサブピクチャが、ループ内フィルタリング動作を除いて、復号プロセスにおいてピクチャとして扱われることを指定するために、1に等しく設定される。sub_pic_treated_as_pic_flag[i]は、CVS内のそれぞれのコーディングされたピクチャの第iのサブピクチャが、ループ内フィルタリング動作を除いて、復号プロセスにおいてピクチャとして扱われないことを指定するために、0に等しく設定される。sub_pic_level_idc[i]は、第iのサブピクチャシーケンスが適合するレベルを示し、第iのサブピクチャシーケンスは、CVS内のsub_pic_id[i]に等しいサブピクチャIDを有するサブピクチャのVCL NALユニットと、これらに関連する非VCL NALユニットのみから構成される。sub_pic_x_offset[i]は、CVS内の各ピクチャの左上角ルマサンプルに対する第iのサブピクチャの左上角ルマサンプルの水平オフセットを、ルマサンプルの単位で指定する。存在しないとき、sub_pic_x_offset[i]の値は0に等しいと推定される。sub_pic_y_offset[i]は、CVS内の各ピクチャの左上角ルマサンプルに対する第iのサブピクチャの左上角ルマサンプルの垂直オフセットを、ルマサンプルの単位で指定する。存在しないとき、sub_pic_y_offset[i]の値は0に等しいと推定される。sub_pic_width_in_luma_samples[i]は、CVS内の各ピクチャの第iのサブピクチャの幅を、ルマサンプルの単位で指定する。sub_pic_x_offset[i]とsub_pic_width_in_luma_samples[i]との和がpic_width_in_luma_samples未満であるとき、sub_pic_width_in_luma_samples[i]の値はCtbSizeYの整数倍とする。存在しないとき、sub_pic_width_in_luma_samples[i]の値は、pic_width_in_luma_samplesと等しいと推定される。sub_pic_height_in_luma_samples[i]は、CVS内の各ピクチャの第iのサブピクチャの高さを、ルマサンプルの単位で指定する。sub_pic_y_offset[i]とsub_pic_height_in_luma_samples[i]との和がpic_height_in_luma_samples未満であるとき、sub_pic_height_in_luma_samples[i]の値はCtbSizeYの整数倍とする。存在しないとき、sub_pic_height_in_luma_samples[i]の値は、pic_height_in_luma_samplesと等しいと推定される。

10

20

【 0 1 6 6 】

ビットストリーム適合性について、以下の制約が当てはまる。i及びjのいずれかの整数値について、iがjに等しいとき、sub_pic_id[i]及びsub_pic_id[j]の値は同じでないものとする。いずれか2つのサブピクチャsubpicA及びsubpicBについて、subpicAのサブピクチャIDがsubpicBのサブピクチャIDよりも小さいとき、subPicAのいずれかのコーディングされたタイルグループNALユニットは、復号順でsubPicBのいずれかのコーディングされたタイルグループNALユニットに続くものとする。サブピクチャの形状は、各サブピクチャが、復号されたとき、ピクチャ境界から構成されるか或いは以前に復号されたサブピクチャの境界から構成される、その全体の左境界及び全体の上境界を有するものとする。

30

【 0 1 6 7 】

サブピクチャIDからサブピクチャインデックスへの変換を指定する、0以上num_sub_pics_minus1以下の範囲のiでのsub_pic_id[i]に等しいspId値についてのリストSubPicIdx[spId]は、以下のように導出される。

【 数 1 8 】

```
for(i=0; i <= num_sub_pics_minus1; i++)
```

```
SubPicIdx[sub_pic_id[i]] = i (7-5)
```

40

【 0 1 6 8 】

log2_max_pic_order_cnt_lsb_minus4は、以下のようにピクチャ順序カウントについての復号プロセスにおいて使用される変数MaxPicOrderCntLsbの値を指定する。

【 数 1 9 】

```
MaxPicOrderCntLsb = 2(log2_max_pic_order_cnt_lsb_minus4 + 4) (7-5)
```

50

log2_max_pic_order_cnt_lsb_minus4の値は、0以上12以下の範囲にあるものとする。

【0169】

例示的なピクチャパラメータセットRBSPのセマンティクスは以下の通りである。

【0170】

存在するとき、PPSシンタックスエレメントpps_seq_parameter_set_id及びloop_filter_across_tiles_enabled_flagのそれぞれの値は、コーディングされたピクチャのタイルグループヘッダにより参照される全てのPPSにおいて同じであるとする。pps_pic_parameter_set_idは、他のシンタックスエレメントにより参照するPPSを識別する。pps_pic_parameter_set_idの値は、0以上63以下の範囲にあるものとする。pps_seq_parameter_set_idは、アクティブなSPSについてsps_seq_parameter_set_idの値を指定する。pps_seq_parameter_set_idの値は、0以上15以下の範囲にあるものとする。loop_filter_across_sub_pic_enabled_flagは、ループ内フィルタリング動作がPPSを参照するサブピクチャの境界を横切って実行されてもよいことを指定するために、1に等しく設定される。loop_filter_across_sub_pic_enabled_flagは、ループ内フィルタリング動作がPPSを参照するサブピクチャの境界を横切って実行されないことを指定するために、0に等しく設定される。

【0171】

single_tile_in_sub_pic_flagは、PPSを参照する各サブピクチャに1つのタイルのみが存在することを指定するために、1に等しく設定される。single_tile_in_sub_pic_flagは、PPSを参照する各サブピクチャに1つよりも多くのタイルが存在することを指定するために、0に等しく設定される。num_tile_columns_minus1に1を加えたものは、サブピクチャを区分するタイル列の数を指定する。num_tile_columns_minus1は、0以上1024以下の範囲にあるものとする。存在しないとき、num_tile_columns_minus1の値は0に等しいと推定される。num_tile_rows_minus1に1を加えたものは、サブピクチャを区分するタイル行の数を指定する。num_tile_rows_minus1は、0以上1024以下の範囲にあるものとする。存在しないとき、num_tile_rows_minus1の値は0に等しいと推定される。変数NumTilesInSubPicは $(num_tile_columns_minus1+1)*(num_tile_rows_minus1+1)$ に等しく設定される。single_tile_in_sub_pic_flagが0に等しいとき、NumTilesInSubPicは1よりも大きいものとする。

【0172】

uniform_tile_spacing_flagは、タイル列境界及び同様にタイル行境界がサブピクチャの間で均一に分布することを指定するために、1に等しく設定される。uniform_tile_spacing_flagは、タイル列境界及び同様にタイル行境界がサブピクチャの間で均一に分布していないが、シンタックスエレメントtile_column_width_minus1[i]及びtile_row_height_minus1[i]を使用して明示的にシグナリングされることを指定するために、0に等しく設定される。存在しないとき、uniform_tile_spacing_flagの値は1に等しいと推定される。tile_column_width_minus1[i]に1を加えたものは、CTBの単位で第iのタイル列の幅を指定する。tile_row_height_minus1[i]に1を加えたものは、CTBの単位で第iのタイル行の高さを指定する。single_tile_per_tile_groupは、このPPSを参照する各タイルグループが1つのタイルを含むことを指定するために、1に等しく設定される。single_tile_per_tile_groupは、このPPSを参照するタイルグループが1つよりも多くのタイルを含んでもよいことを指定するために、0に等しく設定される。

【0173】

rect_tile_group_flagは、サブピクチャの各タイルグループ内のタイルがラスタスキャン順であり、タイルグループ情報がPPSでシグナリングされないことを指定するために、0に等しく設定される。rect_tile_group_flagは、各タイルグループ内のタイルがサブピクチャの長方形又は正方形の領域をカバーし、タイルグループ情報がPPSでシグナリングされることを指定するために、1に等しく設定される。single_tile_per_tile_group_flagが1に設定されたとき、rect_tile_group_flagは、1に等しいと推定される。num_tile_g

10

20

30

40

50

roups_in_sub_pic_minus1に1を加えたものは、PPSを参照する各サブピクチャ内のタイルグループの数を指定する。num_tile_groups_in_sub_pic_minus1の値は、0以上NumTilesInSubPic-1以下の範囲にあるものとする。存在せず、single_tile_per_tile_group_flagが1に等しいとき、num_tile_groups_in_sub_pic_minus1の値は、NumTilesInSubPic-1に等しいと推定される。

【0174】

top_left_tile_idx[i]は、サブピクチャの第iのタイルグループの左上角に位置するタイルのタイルインデックスを指定する。top_left_tile_idx[i]の値は、jに等しくないいずれかのiについて、top_left_tile_idx[j]の値に等しくないものとする。存在しないとき、top_left_tile_idx[i]の値はiに等しいと推定される。top_left_tile_idx[i]シンタックスエレメントの長さはCeil(Log2(NumTilesInSubPic))ビットである。bottom_right_tile_idx[i]は、サブピクチャの第iのタイルグループの右下角に位置するタイルのタイルインデックスを指定する。single_tile_per_tile_group_flagが1に設定されたとき、bottom_right_tile_idx[i]はtop_left_tile_idx[i]に等しいと推定される。bottom_right_tile_idx[i]シンタックスエレメントの長さは、Ceil(Log2(NumTilesInSubPic))ビットである。

【0175】

いずれかの特定のタイルが1つのタイルグループにのみ含まれるものとして、ビットストリーム適合性の要件である。サブピクチャの第iのタイルグループ内のタイル数を指定する変数NumTilesInTileGroup[i]及び関連する変数は、以下のように導出される。

【数20】

$$\text{deltaTileIdx} = \text{bottom_right_tile_idx}[i] - \text{top_left_tile_idx}[i]$$

$$\text{NumTileRowsInTileGroupMinus1}[i] = \text{deltaTileIdx} / (\text{num_tile_columns_minus1} + 1)$$

$$\text{NumTileColumnsInTileGroupMinus1}[i] = \text{deltaTileIdx} \% (\text{num_tile_columns_minus1} + 1) \quad (7-33)$$

$$\text{NumTilesInTileGroup}[i] = (\text{NumTileRowsInTileGroupMinus1}[i] + 1) * (\text{NumTileColumnsInTileGroupMinus1}[i] + 1)$$

【0176】

loop_filter_across_tiles_enabled_flagは、ループ内フィルタリング動作がPPSを参照するサブピクチャ内のタイル境界を横切って実行されてもよいことを指定するために、1に等しく設定される。loop_filter_across_tiles_enabled_flagは、ループ内フィルタリング動作がPPSを参照するサブピクチャ内のタイル境界を横切って実行されないことを指定するために、0に等しく設定される。ループ内フィルタリング動作は、デブロッキングフィルタ、サンプル適応オフセットフィルタ及び適応ループフィルタ動作を含む。存在しないとき、loop_filter_across_tiles_enabled_flagの値は1に等しいと推定される。num_ref_idx_default_active_minus1[i]に1を加えたものは、iが0に等しいとき、0に等しいnum_ref_idx_active_override_flagを有するP又はBタイルグループについての変数NumRefIdxActive[0]の推定値を指定し、iが1に等しいとき、0に等しいnum_ref_idx_active_override_flagを有するBタイルグループについてのNumRefIdxActive[1]の推定値を指定する。num_ref_idx_default_active_minus1[i]の値は、0以上14以下の範囲にあるものとする。

【0177】

例示的な一般タイルグループヘッダのセマンティクスは以下の通りである。存在するとき、タイルグループヘッダのシンタックスエレメントtile_group_pic_order_cnt_lsb及びtile_group_temporal_mvp_enabled_flagのそれぞれの値は、コーディングされたピクチャの全てのタイルグループヘッダにおいて同じであるものとする。存在するとき、til

10

20

30

40

50

e_group_pic_parameter_set_idの値は、コーディングされたサブピクチャの全てのタイルグループヘッダにおいて同じであるものとする。tile_group_pic_parameter_set_idは、使用中のPPSについてのpps_pic_parameter_set_idの値を指定する。tile_group_pic_parameter_set_idの値は、0以上63以下の範囲にあるものとする。現在のピクチャのTemporalIdの値が、現在のピクチャのタイルグループにより参照される各PPSのTemporalIdの値以上であるものとする。tile_group_sub_pic_idは、タイルグループが属するサブピクチャを識別する。tile_group_sub_pic_idの長さは、sub_pic_id_len_minus1+1ビットである。tile_group_sub_pic_idの値は、コーディングされたサブピクチャの全てのタイルグループヘッダについて同じであるものとする。

10

【0178】

変数SubPicWidthInCtbsY、SubPicHeightInCtbsY及びSubPicSizeInCtbsYは以下のように導出される。

【数21】

$$i = \text{SubPicIdx}[\text{tile_group_subpic_id}]$$

$$\text{SubPicWidthInCtbsY} =$$

$$\text{Ceil}(\text{sub_pic_width_in_luma_samples}[i] \div \text{CtbSizeY}) \quad (7-34)$$

$$\text{SubPicHeightInCtbsY} = \text{Ceil}(\text{sub_pic_height_in_luma_samples}[i] \div \text{CtbSizeY})$$

20

$$\text{SubPicSizeInCtbsY} = \text{SubPicWidthInCtbsY} * \text{SubPicHeightInCtbsY}$$

【0179】

以下の変数、すなわち、CTBの単位で第*i*のタイル列の幅を指定する、0以上num_tile_columns_minus1以下の範囲の*i*についてのリストColWidth[i]、CTBの単位で第*j*のタイル行の高さを指定する、0以上num_tile_rows_minus1以下の範囲の*j*についてのリストRowHeight[j]、CTBの単位で第*i*のタイル列境界の位置を指定する、0以上num_tile_columns_minus1+1以下の範囲の*i*についてのリストColBd[i]、CTBの単位で第*j*のタイル行境界の位置を指定する、0以上num_tile_rows_minus1+1以下の範囲の*j*についてのリストRowBd[j]、サブピクチャのCTBラスタスキャンにおけるCTBアドレスからサブピクチャのタイルスキャンにおけるCTBアドレスへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrRsについてのリストCtbAddrRsToTs[ctbAddrRs]、サブピクチャのタイルスキャンにおけるCTBアドレスからサブピクチャのCTBラスタスキャンにおけるCTBアドレスへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrTsについてのリストCtbAddrTsToRs[ctbAddrTs]、サブピクチャのタイルスキャンにおけるCTBアドレスからタイルIDへの変換を指定する、0以上SubPicSizeInCtbsY-1以下の範囲のctbAddrTsについてのリストTileId[ctbAddrTs]、タイルインデックスからタイル内のCTUの数への変換を指定する、0以上NumTilesInSubPic-1以下の範囲のtileIdxについてのリストNumCtusInTile[tileIdx]、タイルIDからタイル内の最初のCTBのタイルスキャンにおけるCTBアドレスへの変換を指定する、0以上NumTilesInSubPic-1以下の範囲のtileIdxについてのリストFirstCtbAddrTs[tileIdx]、ルマサンプルの単位で第*i*のタイル列の幅を指定する、0以上num_tile_columns_minus1以下の範囲の*i*についてのリストColumnWidthInLumaSamples[i]、及びルマサンプルの単位で第*j*のタイル行の高さを指定する、0以上num_tile_rows_minus1以下の範囲の*j*についてのリストRowHeightInLumaSamples[j]は、CTBラスタ及びタイルスキャン変換プロセスを呼び出すことにより導出される。

30

40

【0180】

0以上num_tile_columns_minus1以下の範囲の*i*についてのColumnWidthInLumaSamples[i]、及び0以上num_tile_rows_minus1以下の範囲の*j*についてのRowHeightInLumaSamples[j]の値は、全て0よりも大きいものとする。変数SubPicLeftBoundaryPos

50

、SubPicTopBoundaryPos、SubPicRightBoundaryPos及びSubPicBotBoundaryPosは以下のように導出される。

【数 2 2】

```

i = SubPicIdx[ tile_group_subpic_id ]
if( sub_pic_treated_as_pic_flag[ i ] ) {
SubPicLeftBoundaryPos = sub_pic_x_offset[ i ]
SubPicRightBoundaryPos =
SubPicLeftBoundaryPos + sub_pic_width_in_luma_samples[ i ] - 1           10
SubPicTopBoundaryPos = sub_pic_y_offset[ i ]
SubPicBotBoundaryPos =
SubPicTopBoundaryPos + sub_pic_height_in_luma_samples[ i ] - 1       (7-34)
}

```

【0 1 8 1】

現在のサブピクチャ内のインデックス $i=0$.NumTilesInSubPic-1を有する各タイルについて、変数TileLeftBoundaryPos[i]、TileTopBoundaryPos[i]、TileRightBoundaryPos[i]及びTileBotBoundaryPos[i]は以下のように導出される。 20

【数 2 3】

```

tileColIdx = i % ( num_tile_columns_minus1 + 1 )
tileRowIdx = i / ( num_tile_columns_minus1 + 1 )
TileLeftBoundaryPos[ i ] = SubPicLeftBoundaryPos + ( ColBd[ tileColIdx ] <<
CtbLog2SizeY )
TileRightBoundaryPos[ i ] = SubPicLeftBoundaryPos +
( ( ColBd[ tileColIdx ] + ColWidth[ tileColIdx ] ) << CtbLog2SizeY ) - 1   30
if( TileRightBoundaryPos[ i ] > pic_width_in_luma_samples - 1 )           (7-41)
TileRightBoundaryPos[ i ] = pic_width_in_luma_samples - 1
TileTopBoundaryPos[ i ] = SubPicTopBoundaryPos + ( RowBd[ tileRowIdx ] <<
CtbLog2SizeY )
TileBotBoundaryPos[ i ] = SubPicTopBoundaryPos +
( ( RowBd[ tileRowIdx ] + RowHeight[ tileRowIdx ] ) << CtbLog2SizeY ) - 1
if( TileBotBoundaryPos[ i ] > pic_height_in_luma_samples - 1 )           40
TileBotBoundaryPos[ i ] = pic_height_in_luma_samples - 1

```

【0 1 8 2】

tile_group_addressは、タイルグループ内の最初のタイルのタイルアドレスを指定する。存在しないとき、tile_group_addressの値は0に等しいと推定される。rect_tile_group_flagが0に等しい場合、以下が当てはまる。タイルアドレスはタイルIDであり、tile_group_addressの長さはCeil(Log2(NumTilesInSubPic))ビットであり、tile_group_addressの値は0以上NumTilesInSubPic-1以下の範囲にあるものとする。そうでない場 50

合(rect_tile_group_flagが1に等しい場合)、以下が当てはまる。タイルアドレスはサブピクチャ内のタイルグループの中のタイルグループのタイルグループインデックスであり、tile_group_addressの長さはCeil(Log2(num_tile_groups_in_sub_pic_minus1+1))ビットであり、tile_group_addressの値は0以上num_tile_groups_in_sub_pic_minus1以下の範囲にあるものとする。

【 0 1 8 3 】

以下の制約が当てはまるのが、ビットストリーム適合性の要件である。tile_group_addressの値は、同じコーディングされたサブピクチャのいずれかの他のコーディングされたタイルグループNALユニットのtile_group_addressの値に等しくないものとする。サブピクチャのタイルグループは、これらのtile_group_address値の増加順であるものとする。サブピクチャのタイルグループの形状は、各タイルが、復号されたとき、サブピクチャ境界から構成されるか或いは以前に復号されたタイルの境界から構成される、その全体の左境界及び全体の上境界を有するものとするようなものとする。

10

【 0 1 8 4 】

num_tiles_in_tile_group_minus1は、存在するとき、タイルグループ内のタイル数から1を引いたものを指定する。num_tiles_in_tile_group_minus1の値は、0以上NumTilesInSubPic-1以下の範囲にあるものとする。存在しないとき、num_tiles_in_tile_group_minus1の値は0に等しいと推定される。現在のタイルグループ内のタイル数を指定する変数NumTilesInCurrTileGroup、及び現在のタイルグループ内の第iのタイルのタイルインデックスを指定する変数TgTileIdx[i]は、以下のように導出される。

20

【 数 2 4 】

```
if( rect_tile_group_flag ) {
```

```
  NumTilesInCurrTileGroup = NumTilesInTileGroup[ tile_group_address ]
```

```
  tileIdx = top_left_tile_idx[ tile_group_address ]
```

```
  for( j = 0, tIdx = 0; j <
```

```
    NumTileRowsInTileGroupMinus1[ tile_group_address ] + 1;           (7-35)
```

```
    j++, tileIdx += num_tile_columns_minus1 + 1 )
```

```
    for( i = 0, currTileIdx = tileIdx; i <
```

```
      NumTileColumnsInTileGroupMinus1[ tile_group_address ] + 1;
```

```
        i++, currTileIdx++, tIdx++ )
```

```
      TgTileIdx[ tIdx ] = currTileIdx
```

```
    } else {
```

```
      NumTilesInCurrTileGroup = num_tiles_in_tile_group_minus1 + 1
```

```
      TgTileIdx[ 0 ] = tile_group_address
```

```
      for( i = 1; i < NumTilesInCurrTileGroup; i++ )
```

```
        TgTileIdx[ i ] = TgTileIdx[ i - 1 ] + 1
```

```
    }
```

30

40

tile_group_typeはタイルグループのコーディングタイプを指定する。

【 0 1 8 5 】

時間ルマ動きベクトル予測のための例示的な導出プロセスは以下の通りである。変数mvLXCol及びavailableFlagLXColは以下のように導出される。tile_group_temporal_mv

50

p_enabled_flagが0に等しい場合、mvLXColの双方の成分は0に等しく設定され、availableFlagLXColは0に等しく設定される。そうでない場合(tile_group_temporal_mv_enabled_flagが1に等しい場合)、以下の順序のステップが当てはまる。右下の同一位置の動きベクトル、並びに下及び右境界サンプル位置は以下のように導出される。

【数 2 5】

$$xColBr = xCb + cbWidth \quad (8-414)$$

$$yColBr = yCb + cbHeight \quad (8-415)$$

$$rightBoundaryPos = sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]] \quad ? \quad 10$$

$$SubPicRightBoundaryPos : pic_width_in_luma_samples - 1 \quad (8-415)$$

$$botBoundaryPos = sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]] \quad ?$$

$$SubPicBotBoundaryPos : pic_height_in_luma_samples - 1 \quad (8-415)$$

yCb CtbLog2SizeYがyColBr CtbLog2SizeYに等しく、yColBrがbotBoundaryPos以下であり、xColBrがrightBoundaryPos以下である場合、以下が当てはまる。変数colCbは、ColPicにより指定された同一位置のピクチャ内の((xColBr - 3) - 3, (yColBr - 3) - 3)により与えられる変更された位置をカバーするルマコーディングブロックを指定する。ルマ位置(xColCb, yColCb)は、ColPicにより指定された同一位置のピクチャの左上ルマサンプルに対する、colCbにより指定された同一位置のルマコーディングブロックの左上サンプルに等しく設定される。同一位置の動きベクトルのための導出プロセスは、入力として、currCb、colCb、(xColCb, yColCb)、refIdxLX及び0に等しく設定されたsbFlagによって呼び出され、出力はmvLXCol及びavailableFlagLXColに割り当てられる。そうでない場合、mvLXColの双方の成分は0に等しく設定され、availableFlagLXColは0に等しく設定される。

【0 1 8 6】

例示的なルマサンプルバイリニア補間プロセスは以下の通りである。全サンプルユニット(xInti, yInti)におけるルマ位置は、i=0..1について以下のように導出される。sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が1に等しい場合、以下が当てはまる。

【数 2 6】

$$xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i) \quad (8-458)$$

$$yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i) \quad (8-458)$$

そうでない場合(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が0に等しい場合)、以下が当てはまる。

【数 2 7】

$$xInti = sps_ref_wraparound_enabled_flag ?$$

$$ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, (xIntL + i)) : \quad (8-459)$$

$$Clip3(0, picW - 1, xIntL + i)$$

$$yInti = Clip3(0, picH - 1, yIntL + i) \quad (8-460)$$

【 0 1 8 7 】

例示的なルマサンプル8タップ補間フィルタリングプロセスは以下の通りである。全サンプルユニット(xInti,yInti)におけるルマ位置は、 $i=0..7$ について以下のように導出される。sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が1に等しい場合、以下が当てはまる。

【 数 2 8 】

$$xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos}, xIntL + i - 3) \quad (8-830)$$

$$yInti = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos}, yIntL + i - 3) \quad (8-830)$$

10

そうでない場合(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が0に等しい場合)、以下が当てはまる。

【 数 2 9 】

$$xInti = \text{sps_ref_wraparound_enabled_flag} ?$$

$$\text{ClipH}((\text{sps_ref_wraparound_offset_minus1} + 1) * \text{MinCbSizeY}, \text{picW}, xIntL + i - 3) :$$

$$\text{Clip3}(0, \text{picW} - 1, xIntL + i - 3)$$

(8-831)

$$yInti = \text{Clip3}(0, \text{picH} - 1, yIntL + i - 3)$$

(8-832)

20

【 0 1 8 8 】

例示的なクロマサンプル補間プロセスは以下の通りである。変数xOffsetは、(sps_ref_wraparound_offset_minus1+1)*MinCbSizeY/SubWidthCに等しく設定される。全サンプルユニット(xInti,yInti)におけるクロマ位置は、 $i=0..3$ について以下のように導出される。

【 0 1 8 9 】

sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が1に等しい場合、以下が当てはまる。

【 数 3 0 】

$$xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos} / \text{SubWidthC}, \text{SubPicRightBoundaryPos} / \text{SubWidthC}, xIntL + i) \quad (8-844)$$

$$yInti = \text{Clip3}(\text{SubPicTopBoundaryPos} / \text{SubHeightC}, \text{SubPicBotBoundaryPos} / \text{SubHeightC}, yIntL + i) \quad (8-844)$$

30

そうでない場合(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]が0に等しい場合)、以下が当てはまる。

【 数 3 1 】

$$xInti = \text{sps_ref_wraparound_enabled_flag} ? \text{ClipH}(xOffset, \text{picWC}, xIntC + i - 1) : (8-845)$$

$$\text{Clip3}(0, \text{picWC} - 1, xIntC + i - 1)$$

$$yInti = \text{Clip3}(0, \text{picHC} - 1, yIntC + i - 1)$$

(8-846)

40

【 0 1 9 0 】

例示的なデブロッキングフィルタプロセスは以下の通りである。デブロッキングフィルタプロセスは、以下のタイプのエッジ、すなわち、ピクチャの境界にあるエッジ、loop_filter_across_sub_pic_enabled_flagが0に等しいサブピクチャの境界に一致するエッジ、loop_filter_across_tiles_enabled_flagが0に等しいタイルの境界に一致するエッジ、tile_group_deblocking_filter_disabled_flagが1に等しいタイルグループの或いは

50

タイルグループ内の上又は左境界に一致するエッジ、考慮される成分の 8×8 サンプルグリッド境界に対応しないエッジ、エッジの双方側がインター予測を使用するクロマ成分内のエッジ、関連する変換ユニットのエッジではないクロマ変換ブロックのエッジ、及びISP_NO_SPLITに等しくないIntraSubPartitionsSplit値を有するコーディングユニットのルマ変換ブロックを横切るエッジを除く、ピクチャの全てのコーディングサブブロックエッジ及び変換ブロックエッジに適用される。

【0191】

一方向の例示的なデブロッキングフィルタプロセスは以下の通りである。コーディングブロック幅 $\log_2 CbW$ 、コーディングブロック高さ $\log_2 CbH$ 及びコーディングブロックの左上サンプルの位置 (xCb, yCb) を有する各コーディングユニットについて、edgeTypeがEDGE_VERに等しく、 $xCb \% 8$ が0に等しいとき、或いは、edgeTypeがEDGE_HORに等しく、 $yCb \% 8$ が0に等しいとき、エッジは以下の順序のステップによりフィルタリングされる。コーディングブロック幅 $nCbW$ は $1 \dots \log_2 CbW$ に等しく設定され、コーディングブロック高さ $nCbH$ は $1 \dots \log_2 CbH$ に等しく設定される。変数filterEdgeFlagは以下のように導出される。edgeTypeがEDGE_VERに等しく、以下の条件の1つ以上が真である場合、filterEdgeFlagは0に等しく設定される。現在のコーディングブロックの左境界はピクチャの左境界である。現在のコーディングブロックの左境界は、サブピクチャの左又は右境界であり、loop_filter_across_sub_pic_enabled_flagは0に等しい。現在のコーディングブロックの左境界はタイルの左境界であり、loop_filter_across_tiles_enabled_flagは0に等しい。そうでなく、edgeTypeがEDGE_HORに等しく、以下の条件の1つ以上が真である場合、変数filterEdgeFlagは0に等しく設定される。現在のルマコーディングブロックの上境界はピクチャの上境界である。現在のコーディングブロックの上境界は、サブピクチャの上又は下境界であり、loop_filter_across_sub_pic_enabled_flagは0に等しい。現在のコーディングブロックの上境界はタイルの上境界であり、loop_filter_across_tiles_enabled_flagは0に等しい。そうでない場合、filterEdgeFlagは1に等しく設定される。

【0192】

例示的なCTB変更プロセスは以下の通りである。recPicture[xSi][ySj]をカバーするコーディングブロックを含むコーディングユニットのpcm_loop_filter_disabled_flag、pcm_flag[xYi][yYj]及びcu_transquant_bypass_flagの値に依存して、 $i=0 \dots nCtbSw-1$ 且つ $j=0 \dots nCtbSh-1$ での全てのサンプル位置 (xSi, ySj) 及び (xYi, yYj) について、以下が当てはまる。 $k=0 \dots 1$ での全てのサンプル位置 $(xSik', ySjk')$ 及び $(xYik', yYjk')$ について以下の条件の1つ以上が真である場合、edgeIdxは0に等しく設定される。位置 $(xSik', ySjk')$ におけるサンプルはピクチャ境界外にある。位置 $(xSik', ySjk')$ におけるサンプルは異なるサブピクチャに属し、サンプルrecPicture[xSi][ySj]が属するタイルグループにおけるloop_filter_across_sub_pic_enabled_flagは0に等しい。loop_filter_across_tiles_enabled_flagは0に等しく、位置 $(xSik', ySjk')$ におけるサンプルは異なるタイルに属する。

【0193】

ルマサンプルについての例示的なコーディングツリーブロックのフィルタリングプロセスは以下の通りである。フィルタリング後の再現されたルマサンプルalfPictureL[x][y]の導出のために、現在のルマコーディングツリーブロックrecPictureL[x][y]内のそれぞれの再現されたルマサンプルは、 $x, y=0 \dots CtbSizeY-1$ によって以下のようにフィルタリングされる。ルマサンプルの所与の配列recPicture内の対応するルマサンプル (x, y) のそれぞれの位置 (hx, vy) は以下のように導出される。位置 (hx, vy) におけるルマサンプルを含むタイルtileAについてのloop_filter_across_tiles_enabled_flagが0に等しい場合、変数tileIdxをtileAのタイルインデックスとすると、以下が当てはまる。

【数32】

10

20

30

40

50

$hx = \text{Clip3}(\text{TileLeftBoundaryPos}[\text{tileIdx}], \text{TileRightBoundaryPos}[\text{tileIdx}], x\text{Ctb} + x)$
(8-1140)

$vy = \text{Clip3}(\text{TileTopBoundaryPos}[\text{tileIdx}], \text{TileBotBoundaryPos}[\text{tileIdx}], y\text{Ctb} + y)$
(8-1141)

そうでなく、位置(hx,vy)におけるルマサンプルを含むサブピクチャ内のloop_filter_across_sub_pic_enabled_flagが0に等しい場合、以下が当てはまる。

【数 3 3】

$hx = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos}, x\text{Ctb} + x)$ (8-1140)

$vy = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos}, y\text{Ctb} + y)$ (8-1141)

そうでない場合、以下が当てはまる。

【数 3 4】

$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x\text{Ctb} + x)$ (8-1140)

$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y\text{Ctb} + y)$ (8-1141)

【0 1 9 4】

ルマサンプルについてのALF転置及びフィルタインデックスのための例示的な導出プロセスは以下の通りである。ルマサンプルの所与の配列recPicture内の対応するルマサンプル(x,y)のそれぞれの位置(hx,vy)は、以下のように導出される。位置(hx,vy)におけるルマサンプルを含むタイルtileAについてのloop_filter_across_tiles_enabled_flagが0に等しい場合、tileIdxをtileAのタイルインデックスとすると、以下が当てはまる。

【数 3 5】

$hx = \text{Clip3}(\text{TileLeftBoundaryPos}[\text{tileIdx}], \text{TileRightBoundaryPos}[\text{tileIdx}], x)$ (8-1140)

$vy = \text{Clip3}(\text{TileTopBoundaryPos}[\text{tileIdx}], \text{TileBotBoundaryPos}[\text{tileIdx}], y)$ (8-1141)

そうでなく、位置(hx,vy)におけるルマサンプルを含むサブピクチャについてのloop_filter_across_sub_pic_enabled_flagが0に等しい場合、以下が当てはまる。

【数 3 6】

$hx = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos}, x)$ (8-1140)

$vy = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos}, y)$ (8-1141)

そうでない場合、以下が当てはまる。

【数 3 7】

$hx = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x)$ (8-1145)

$vy = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y)$ (8-1146)

【0 1 9 5】

クロマサンプルについての例示的なコーディングツリーブロックのフィルタリングプロセスは以下の通りである。フィルタリング後の再現されたクロマサンプルalfPicture[x][y]の導出のために、現在のクロマコーディングツリーブロックrecPicture[x][y]内のそれぞれの再現されたクロマサンプルは、 $x, y = 0..ctbSizeC - 1$ によって以下のようにフィルタリングされる。クロマサンプルの所与の配列recPicture内の対応するクロマサンプル(x,y)

のそれぞれの位置(hx,vy)は、以下のように導出される。位置(hx,vy)におけるクロマサンプルを含むタイルtileAについてのloop_filter_across_tiles_enabled_flagが0に等しい場合、tileIdxをtileAのタイルインデックスとすると、以下が当てはまる。

【数 3 8】

$$\begin{aligned} \text{hx} = & \text{Clip3}(\text{TileLeftBoundaryPos}[\text{tileIdx}] / \text{SubWidthC}, \\ & \text{TileRightBoundaryPos}[\text{tileIdx}] / \text{SubWidthC}, \text{xCtb} + \text{x}) \end{aligned} \quad (8-1140)$$

$$\begin{aligned} \text{vy} = & \text{Clip3}(\text{TileTopBoundaryPos}[\text{tileIdx}] / \text{SubWidthC}, \\ & \text{TileBotBoundaryPos}[\text{tileIdx}] / \text{SubWidthC}, \text{yCtb} + \text{y}) \end{aligned} \quad (8-1141)$$

10

そうでなく、位置(hx,vy)におけるクロマサンプルを含むサブピクチャについてのloop_filter_across_sub_pic_enabled_flagが0に等しい場合、以下が当てはまる。

【数 3 9】

$$\begin{aligned} \text{hx} = & \text{Clip3}(\text{SubPicLeftBoundaryPos} / \text{SubWidthC}, \\ & \text{SubPicRightBoundaryPos} / \text{SubWidthC}, \text{xCtb} + \text{x}) \end{aligned} \quad (8-1140)$$

$$\begin{aligned} \text{vy} = & \text{Clip3}(\text{SubPicTopBoundaryPos} / \text{SubWidthC}, \\ & \text{SubPicBotBoundaryPos} / \text{SubWidthC}, \text{yCtb} + \text{y}) \end{aligned} \quad (8-1141)$$

20

そうでない場合、以下が当てはまる。

【数 4 0】

$$\text{hx} = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, \text{xCtbC} + \text{x}) \quad (8-1177)$$

$$\text{vy} = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, \text{yCtbC} + \text{y}) \quad (8-1178)$$

【0 1 9 6】

変数の和は以下のように導出される。

【数 4 1】

$$\begin{aligned} \text{sum} = & \text{AlfCoeffC}[0] * (\text{recPicture}[\text{hx}, \text{vy} + 2] + \text{recPicture}[\text{hx}, \text{vy} - 2]) + \\ & \text{AlfCoeffC}[1] * (\text{recPicture}[\text{hx} + 1, \text{vy} + 1] + \text{recPicture}[\text{hx} - 1, \text{vy} - 1]) + \\ & \text{AlfCoeffC}[2] * (\text{recPicture}[\text{hx}, \text{vy} + 1] + \text{recPicture}[\text{hx}, \text{vy} - 1]) + \quad (8-1179) \\ & \text{AlfCoeffC}[3] * (\text{recPicture}[\text{hx} - 1, \text{vy} + 1] + \text{recPicture}[\text{hx} + 1, \text{vy} - 1]) + \\ & \text{AlfCoeffC}[4] * (\text{recPicture}[\text{hx} + 2, \text{vy}] + \text{recPicture}[\text{hx} - 2, \text{vy}]) + \\ & \text{AlfCoeffC}[5] * (\text{recPicture}[\text{hx} + 1, \text{vy}] + \text{recPicture}[\text{hx} - 1, \text{vy}]) + \\ & \text{AlfCoeffC}[6] * \text{recPicture}[\text{hx}, \text{vy}] \end{aligned}$$

$$\text{sum} = (\text{sum} + 64) \gg 7 \quad (8-1180)$$

40

修正されたフィルタリング後の再現されたクロマピクチャサンプルalfPicture[xCtbC+x][yCtbC+y]は、以下のように導出される。

【数 4 2】

$$\text{alfPicture}[\text{xCtbC} + \text{x}][\text{yCtbC} + \text{y}] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum}) \quad (8-1181)$$

【0 1 9 7】

図 1 2 は、例示的なビデオコーディングデバイス1200の概略図である。ビデオコーディングデバイス1200は、ここに記載のように、開示される例/実施形態を実現するのに適

50

している。ビデオコーディングデバイス1200は、ネットワーク上でデータをアップストリーム及び/又はダウンストリームに通信するための送信機及び/又は受信機を含むダウンストリームポート1220、アップストリームポート1250及び/又はトランシーバユニット(Tx/Rx)1210を含む。ビデオコーディングデバイス1200はまた、データを処理するための論理ユニット及び/又は中央処理装置(central processing unit, CPU)を含むプロセッサ1230と、データを記憶するためのメモリ1232とを含む。ビデオコーディングデバイス1200はまた、電気、光又は無線通信ネットワークを介してデータを通信するために、アップストリームポート1250及び/又はダウンストリームポート1220に結合された電気、光・電気(optical-to-electrical, OE)コンポーネント、電気・光(electrical-to-optical, EO)コンポーネント及び/又は無線通信コンポーネントを含んでもよい。ビデオコーディングデバイス1200はまた、ユーザに且つユーザからデータを通信するための入力及び/又は出力(input and/or output, I/O)デバイス1260を含んでもよい。I/Oデバイス1260は、ビデオデータを表示するためのディスプレイ、オーディオデータを出力するためのスピーカ等のような出力デバイスを含んでもよい。I/Oデバイス1260はまた、キーボード、マウス、トラックボール等のような入力デバイス、及び/又はこのような出力デバイスと相互作用するための対応するインターフェースを含んでもよい。

10

【0198】

プロセッサ1230は、ハードウェア及びソフトウェアにより実現される。プロセッサ1230は、1つ以上のCPUチップ、(例えば、マルチコアプロセッサとしての)コア、フィールドプログラマブルゲートアレイ(field-programmable gate array, FPGA)、特定用途向け集積回路(application specific integrated circuit, ASIC)及びデジタルシグナルプロセッサ(digital signal processor, DSP)として実現されてもよい。プロセッサ1230は、ダウンストリームポート1220、Tx/Rx1210、アップストリームポート1250及びメモリ1232と通信する。プロセッサ1230は、コーディングモジュール1214を含む。コーディングモジュール1214は、方法100、1300及び1400のような、ここに記載の開示される実施形態を実現し、これらは、ループ内フィルタ1000、ビットストリーム1100、ピクチャ500、及び/又はパターン900に従って生成された候補リストに基づいて一方向インター予測600及び/又は双方向インター予測700に従ってコーディングされ得る現在のブロック801及び/又は901を使用してもよい。コーディングモジュール1214はまた、ここに記載のいずれかの他の方法/メカニズムを実現してもよい。さらに、コーディングモジュール1214は、コーデックシステム200、エンコーダ300及び/又はデコーダ400を実現してもよい。例えば、コーディングモジュール1214は、上記のように、第1、第2、第3、第4、第5及び/又は第6の例の実現方式を実現できる。したがって、コーディングモジュール1214は、ビデオデータをコーディングするとき、ビデオコーディングデバイス1200に更なる機能及び/又はコーディング効率を提供させる。したがって、コーディングモジュール1214は、ビデオコーディングデバイス1200の機能を改善すると共に、ビデオコーディング技術に特有の問題に対処する。さらに、コーディングモジュール1214は、異なる状態へのビデオコーディングデバイス1200の変換をもたらす。代替として、コーディングモジュール1214は、メモリ1232に記憶された命令として実現され、プロセッサ1230により(例えば、非一時的な媒体に記憶されたコンピュータプログラム製品として)実行できる。

20

30

40

【0199】

メモリ1232は、ディスク、テープドライブ、ソリッドステートドライブ、読み取り専用メモリ(read only memory, ROM)、ランダムアクセスメモリ(random access memory, RAM)、フラッシュメモリ、三値連想メモリ(ternary content-addressable memory, TCAM)、スタティックランダムアクセスメモリ(static random-access memory, SRAM)等のような1つ以上のメモリタイプを含む。メモリ1232は、プログラムが実行のために選択されるときにこのようなプログラムを記憶し、プログラム実行中に読み取られる命令及びデータを記憶するために、オーバーフローデータ記憶デバイスとして使用されてもよい。

50

【 0 2 0 0 】

図 1 3 は、サブピクチャ510のようなサブピクチャがピクチャ500のようなピクチャとして扱われるとき、補間フィルタ913のような補間フィルタにおいてクリッピング機能を適用しつつ、ビデオシーケンスをビットストリーム1100のようなビットストリームに符号化する例示的な方法1300のフローチャートである。方法1300は、ループ内フィルタ1000を使用することにより及び/又はパターン900に従って生成された候補リストに対して一方向インター予測600及び/又は双方向インター予測700に従って現在のブロック801及び/又は901を符号化するための方法100を実行するとき、コーデックシステム200、エンコーダ300及び/又はビデオコーディングデバイス1200のようなエンコーダにより使用されてもよい。

10

【 0 2 0 1 】

方法1300は、エンコーダが複数のピクチャを含むビデオシーケンスを受信し、例えばユーザ入力に基づいてそのビデオシーケンスをビットストリームに符号化することを決定したときに開始してもよい。ステップ1301において、エンコーダは、現在のピクチャをサブピクチャに区分する。エンコーダはまた、サブピクチャをブロックに区分する。ステップ1303において、エンコーダは、インター予測に従ってブロックを符号化することを決定する。したがって、エンコーダは、ブロックを符号化するための動きベクトルを選択する。

【 0 2 0 2 】

ステップ1305において、エンコーダは、動きベクトルにより指される参照ブロック内のサンプル位置にクリッピング機能を適用する。クリッピング機能は、動きベクトルがサブピクチャ外を指すとき、補間フィルタの適用をサポートするために適用される。このプロセスは、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたときに発生し、したがって、ピクチャ内の他のサブピクチャから独立した抽出をサポートするようにコーディングされるべきである。この文脈において、サブピクチャが他のサブピクチャ内のデータを参照せずにコーディングされる時、サブピクチャはピクチャとして扱われ、したがって、別々に抽出できる。

20

【 0 2 0 3 】

ステップ1307において、エンコーダは、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用する。一例では、補間フィルタは、ルマサンプルバイリニア補間プロセスを含む。このような場合、ブロックはルマサンプルのブロックを含む。さらに、予測サンプル値は、予測ルマサンプル値を含む。この場合、ルマサンプルバイリニア補間プロセスは、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信する。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLXL)を出力する。この場合、ステップ1305のクリッピング機能は、以下のようにサンプル位置に適用される。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。

30

$xInt_i = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i)$ 、且つ
 $yInt_i = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i)$

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInt_i及びyInt_iはインデックス*i*におけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

40

【 数 4 3 】

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

50

ここで、 x 、 y 及び z は数値入力値である。

【0204】

他の例では、補間フィルタは、ルマサンプル8タップ補間フィルタリングプロセスを含む。この場合、ブロックはルマサンプルのブロックを含み、予測サンプル値は、予測ルマサンプル値を含む。また、ルマサンプル8タップ補間フィルタリングプロセスは、全サンプルユニット($xIntL, yIntL$)におけるルマ位置を含む入力を受信する。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値($predSampleLXL$)を出力する。さらに、ステップ1305のクリッピング機能は、以下のようにサンプル位置に適用される。 $subpic_treated_as_pic_flag[SubPicIdx]$ が1に等しいとき、以下が当てはまる。

$xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i - 3)$ 、且つ
 $yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i - 3)$

10

ここで、 $subpic_treated_as_pic_flag$ はサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、 $SubPicIdx$ はサブピクチャのインデックスであり、 $xInti$ 及び $yInti$ はインデックス i におけるクリッピングされたサンプル位置であり、 $SubPicRightBoundaryPos$ はサブピクチャの右境界の位置であり、 $SubPicLeftBoundaryPos$ はサブピクチャの左境界の位置であり、 $SubPicTopBoundaryPos$ はサブピクチャの上境界の位置であり、 $SubPicBotBoundaryPos$ はサブピクチャの下境界の位置であり、 $Clip3$ は以下に従うクリッピング機能であり、

【数44】

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

20

ここで、 x 、 y 及び z は数値入力値である。

【0205】

他の例では、補間フィルタは、クロマサンプル補間プロセスを含む。この場合、ブロックはクロマサンプルのブロックを含み、予測サンプル値は、予測クロマサンプル値を含む。この場合、クロマサンプル補間プロセスは、全サンプルユニット($xIntC, yIntC$)におけるクロマ位置を含む入力を受信する。さらに、クロマサンプル補間プロセスは、予測クロマサンプル値($predSampleLXC$)を出力する。この場合、ステップ1305のクリッピング機能は、以下のようにサンプル位置に適用される。 $subpic_treated_as_pic_flag[SubPicIdx]$ が1に等しいとき、以下が当てはまる。

30

$xInti = Clip3(SubPicLeftBoundaryPos / SubWidthC, SubPicRightBoundaryPos / SubWidthC, xIntC + i)$ 、且つ

$yInti = Clip3(SubPicTopBoundaryPos / SubHeightC, SubPicBotBoundaryPos / SubHeightC, yIntC + i)$

ここで、 $subpic_treated_as_pic_flag$ はサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、 $SubPicIdx$ はサブピクチャのインデックスであり、 $xInti$ 及び $yInti$ はインデックス i におけるクリッピングされたサンプル位置であり、 $SubPicRightBoundaryPos$ はサブピクチャの右境界の位置であり、 $SubPicLeftBoundaryPos$ はサブピクチャの左境界の位置であり、 $SubPicTopBoundaryPos$ はサブピクチャの上境界の位置であり、 $SubPicBotBoundaryPos$ はサブピクチャの下境界の位置であり、 $SubWidthC$ 及び $SubHeightC$ はルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、 $Clip3$ は以下に従うクリッピング機能であり、

40

【数45】

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

ここで、 x 、 y 及び z は数値入力値であることに従って、サンプル位置に適用される。

50

【 0 2 0 6 】

ステップ1309において、エンコーダは、予測サンプル値及び動きベクトルに基づいて、ブロックをビットストリームに符号化する。次いで、ビットストリームは、デコーダに向けて通信するために記憶されてもよい。

【 0 2 0 7 】

図14は、サブピクチャ510のようなサブピクチャがピクチャ500のようなピクチャとして扱われるとき、補間フィルタ913のような補間フィルタにおいてクリッピング機能を適用しつつ、ビットストリーム1100のようなビットストリームからビデオシーケンスを復号する例示的な方法1400のフローチャートである。方法1400は、ループ内フィルタ1000を使用することにより及び/又はパターン900に従って生成された候補リストに対して一方向インター予測600及び/又は双方向インター予測700に従って現在のブロック801及び/又は901を復号するための方法100を実行するとき、コーデックシステム200、デコーダ400及び/又はビデオコーディングデバイス1200のようなデコーダにより使用されてもよい。

10

【 0 2 0 8 】

方法1400は、デコーダが、例えば方法1300の結果として、ビデオシーケンスを表すコーディングされたデータのビットストリームを受信し始めるときに開始してもよい。ステップ1401において、デコーダは、サブピクチャを含む現在のピクチャを含むビットストリームを受信する。ピクチャ、サブピクチャ、スライス、タイル、CTU及び/又はこれらの他のサブ領域は、インター予測に従ってコーディングされる。ステップ1402において、デコーダは、サブピクチャのブロックについての動きベクトルを決定する。

20

【 0 2 0 9 】

ステップ1403において、デコーダは、動きベクトルにより指される参照ブロック内のサンプル位置にクリッピング機能を適用する。クリッピング機能は、動きベクトルがサブピクチャ外を指すとき、補間フィルタの適用をサポートするために適用される。このプロセスは、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたときに発生し、したがって、ピクチャ内の他のサブピクチャから独立した抽出をサポートするようにコーディングされるべきである。

【 0 2 1 0 】

ステップ1405において、デコーダは、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用する。一例では、補間フィルタは、ルマサンプルバイリニア補間プロセスを含む。このような場合、ブロックはルマサンプルのブロックを含む。さらに、予測サンプル値は、予測ルマサンプル値を含む。この場合、ルマサンプルバイリニア補間プロセスは、全サンプルユニット(xIntL,yIntL)におけるルマ位置を含む入力を受信する。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値(predSampleLXL)を出力する。この場合、ステップ1403のクリッピング機能は、以下のようにサンプル位置に適用される。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。subpic_treated_as_pic_flag[SubPicIdx]が1に等しいとき、以下が当てはまる。

30

$xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos}, xIntL+i)$ 、且つ
 $yInti = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos}, yIntL+i)$

40

ここで、subpic_treated_as_pic_flagはサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、SubPicIdxはサブピクチャのインデックスであり、xInti及びyIntiはインデックスiにおけるクリッピングされたサンプル位置であり、SubPicRightBoundaryPosはサブピクチャの右境界の位置であり、SubPicLeftBoundaryPosはサブピクチャの左境界の位置であり、SubPicTopBoundaryPosはサブピクチャの上境界の位置であり、SubPicBotBoundaryPosはサブピクチャの下境界の位置であり、Clip3は以下に従うクリッピング機能であり、

【 数 4 6 】

50

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

ここで、 x 、 y 及び z は数値入力値である。

【0211】

他の例では、補間フィルタは、ルマサンプル8タップ補間フィルタリングプロセスを含む。この場合、ブロックはルマサンプルのブロックを含み、予測サンプル値は、予測ルマサンプル値を含む。また、ルマサンプル8タップ補間フィルタリングプロセスは、全サンプルユニット($xIntL, yIntL$)におけるルマ位置を含む入力を受信する。ルマサンプルバイリニア補間プロセスは、予測ルマサンプル値($predSampleLXL$)を出力する。さらに、ステップ1403のクリッピング機能は、以下のようにサンプル位置に適用される。 $subpic_treated_as_pic_flag[SubPicIdx]$ が1に等しいとき、以下が当てはまる。

$xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos}, \text{SubPicRightBoundaryPos}, xIntL + i - 3)$ 、且つ
 $yInti = \text{Clip3}(\text{SubPicTopBoundaryPos}, \text{SubPicBotBoundaryPos}, yIntL + i - 3)$

ここで、 $subpic_treated_as_pic_flag$ はサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、 $SubPicIdx$ はサブピクチャのインデックスであり、 $xInti$ 及び $yInti$ はインデックス i におけるクリッピングされたサンプル位置であり、 $SubPicRightBoundaryPos$ はサブピクチャの右境界の位置であり、 $SubPicLeftBoundaryPos$ はサブピクチャの左境界の位置であり、 $SubPicTopBoundaryPos$ はサブピクチャの上境界の位置であり、 $SubPicBotBoundaryPos$ はサブピクチャの下境界の位置であり、 $Clip3$ は以下に従うクリッピング機能であり、

【数47】

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

ここで、 x 、 y 及び z は数値入力値である。

【0212】

他の例では、補間フィルタは、クロマサンプル補間プロセスを含む。この場合、ブロックはクロマサンプルのブロックを含み、予測サンプル値は、予測クロマサンプル値を含む。この場合、クロマサンプル補間プロセスは、全サンプルユニット($xIntC, yIntC$)におけるクロマ位置を含む入力を受信する。さらに、クロマサンプル補間プロセスは、予測クロマサンプル値($predSampleLXC$)を出力する。この場合、ステップ1403のクリッピング機能は、以下のようにサンプル位置に適用される。 $subpic_treated_as_pic_flag[SubPicIdx]$ が1に等しいとき、以下が当てはまる。

$xInti = \text{Clip3}(\text{SubPicLeftBoundaryPos}/\text{SubWidthC}, \text{SubPicRightBoundaryPos}/\text{SubWidthC}, xIntC + i)$ 、且つ

$yInti = \text{Clip3}(\text{SubPicTopBoundaryPos}/\text{SubHeightC}, \text{SubPicBotBoundaryPos}/\text{SubHeightC}, yIntC + i)$

ここで、 $subpic_treated_as_pic_flag$ はサブピクチャがピクチャとして扱われることを示すために設定されるフラグであり、 $SubPicIdx$ はサブピクチャのインデックスであり、 $xInti$ 及び $yInti$ はインデックス i におけるクリッピングされたサンプル位置であり、 $SubPicRightBoundaryPos$ はサブピクチャの右境界の位置であり、 $SubPicLeftBoundaryPos$ はサブピクチャの左境界の位置であり、 $SubPicTopBoundaryPos$ はサブピクチャの上境界の位置であり、 $SubPicBotBoundaryPos$ はサブピクチャの下境界の位置であり、 $SubWidthC$ 及び $SubHeightC$ はルマサンプルとクロマサンプルとの間の水平及び垂直サンプリングレート比を示し、 $Clip3$ は以下に従うクリッピング機能であり、

【数48】

10

20

30

40

50

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

ここで、 x 、 y 及び z は数値入力値であることに従って、サンプル位置に適用される。

【0213】

ステップ1407において、デコーダは、予測サンプル値に基づいてブロックを復号する。次いで、デコーダは、復号ビデオシーケンスの一部として表示するために、ブロックを転送できる。

【0214】

図15は、サブピクチャ510のようなサブピクチャがピクチャ500のようなピクチャとして扱われるとき、補間フィルタ913のような補間フィルタにおいてクリッピング機能を適用しつつ、ビットストリーム1100のようなビットストリームにおいて画像のビデオシーケンスをコーディングするための例示的なシステム1500の概略図である。システム1500は、コーデックシステム200、エンコーダ300、デコーダ400及び/又はビデオコーディングデバイス1200のようなエンコーダ及びデコーダにより実現されてもよい。さらに、システム1500は、ループ内フィルタ1000を使用することにより及び/又はパターン900に従って生成された候補リストに対して一方向インター予測600及び/又は双方向インター予測700に従って現在のブロック801及び/又は901をコーディングするための方法100、1300及び/又は1400を実現するときにも使用されてもよい。

【0215】

システム1500は、ビデオエンコーダ1502を含む。ビデオエンコーダ1502は、現在のピクチャをサブピクチャに区分し、サブピクチャをブロックに区分するための区分モジュール1503を含む。ビデオエンコーダ1502は、インター予測に従ってブロックを符号化することを決定するための決定モジュール1504を更に含む。ビデオエンコーダ1502は、ブロックを符号化するための動きベクトルを選択するための選択モジュール1505を更に含む。ビデオエンコーダ1502は、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用するための適用モジュール1506を更に含む。適用モジュール1506はまた、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するためのものである。ビデオエンコーダ1502は、予測サンプル値及び動きベクトルに基づいて、ブロックをビットストリームに符号化するための符号化モジュール1507を更に含む。ビデオエンコーダ1502は、デコーダに向けて通信するためにビットストリームを記憶するための記憶モジュール1508を更に含む。ビデオエンコーダ1502は、ビットストリームをビデオデコーダ1510に向けて送信するための送信モジュール1509を更に含む。ビデオエンコーダ1502は、方法1300のステップのいずれかを実行するように更に構成されてもよい。

【0216】

システム1500はまた、ビデオデコーダ1510を含む。ビデオデコーダ1510は、インター予測に従ってコーディングされたサブピクチャを含む現在のピクチャを含むビットストリームを受信するための受信モジュール1511を含む。ビデオデコーダ1510は、サブピクチャのブロックについての動きベクトルを決定するための決定モジュール1512を更に含む。ビデオデコーダ1510は、動きベクトルがサブピクチャ外を指すとき、且つ、サブピクチャがピクチャとして扱われることを示すようにフラグが設定されたとき、補間フィルタの適用をサポートするように参照ブロック内のサンプル位置にクリッピング機能を適用するための適用モジュール1513を更に含む。適用モジュール1513はまた、予測サンプル値を取得するために、クリッピング機能の結果に補間フィルタを適用するためのものである。ビデオデコーダ1510は、予測サンプル値に基づいて、ブロックを復号するための復号モジュール1514を更に含む。ビデオデコーダ1510は、復号ビデオシーケンスの一部として表示するために、ブロックを転送するための転送モジュール1515を含む。ビデオデ

10

20

30

40

50

コード1510は、方法1400のステップのいずれかを実行するように更に構成されてもよい。

【0217】

第1のコンポーネントは、第1のコンポーネントと第2のコンポーネントとの間のライン、トレース又は他の媒体を除き、介在するコンポーネントが存在しないとき、第2のコンポーネントに直接的に結合される。第1のコンポーネントは、第1のコンポーネントと第2のコンポーネントとの間のライン、トレース又は他の媒体以外の介在するコンポーネントが存在するとき、第2のコンポーネントに間接的に結合される。「結合」という用語及びその変形は、直接的に結合されること及び間接的に結合されることの双方を含む。「約」という用語の使用は、特に言及のない限り、後続の数の $\pm 10\%$ を含む範囲を意味する。

10

【0218】

ここに記載の例示的な方法のステップは、必ずしも記載の順序で実施される必要はないことも理解されるべきであり、このような方法のステップの順序は、単に例示的なものであると理解されるべきである。同様に、更なるステップがこのような方法に含まれてもよく、特定のステップが本開示の様々な実施形態と一致した方法において省略又は結合されてもよい。

【0219】

本開示においていくつかの実施形態が提供されているが、開示のシステム及び方法は、本開示の真意又は範囲から逸脱することなく、多くの他の特定の形式で具現されてもよいことが理解され得る。本例は、例示的なものであり、限定的なものではないと考えられるべきであり、その意図は、ここに与えられる詳細に限定されないことである。例えば、様々な要素又はコンポーネントは結合されてもよく或いは他のシステムに統合されてもよく、或いは、特定の特徴が省略されてもよく或いは実現されなくてもよい。

20

【0220】

さらに、様々な実施形態において個別又は別々として記載及び図示された技術、システム、サブシステム及び方法は、本開示の範囲から逸脱することなく、結合されてもよく或いは他のシステム、コンポーネント、技術又は方法と統合されてもよい。変形、置換及び変更の他の例は当業者により確認可能であり、ここに開示された真意及び範囲から逸脱することなく行われてもよい。

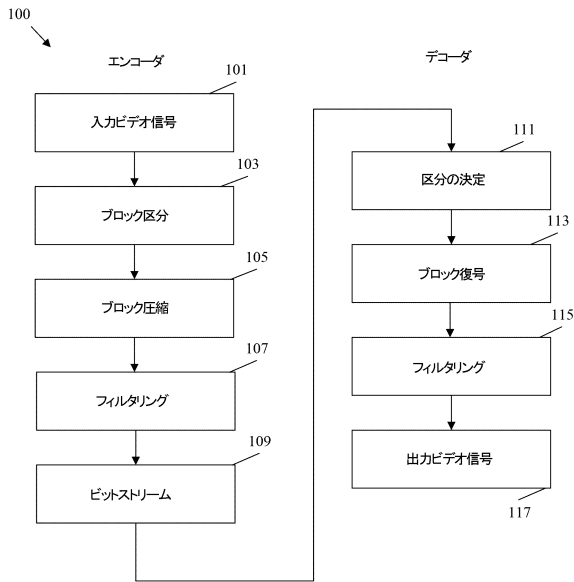
30

40

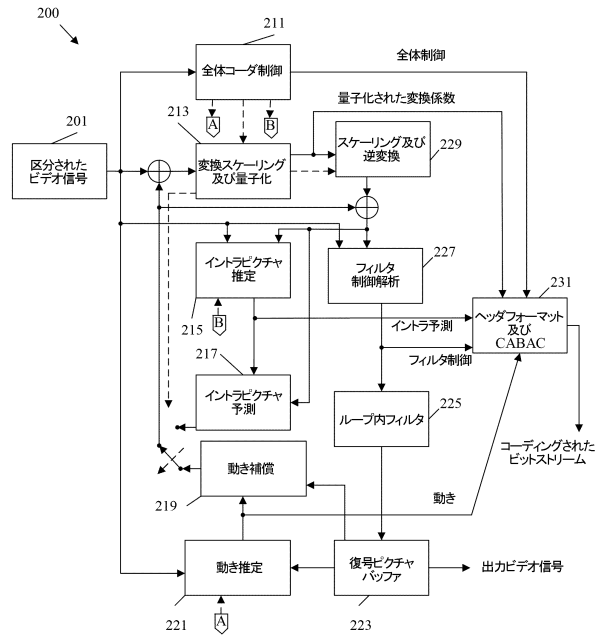
50

【図面】

【図 1】



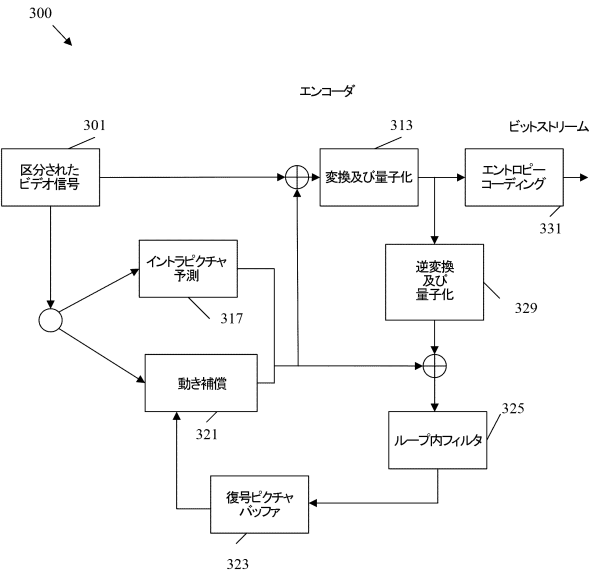
【図 2】



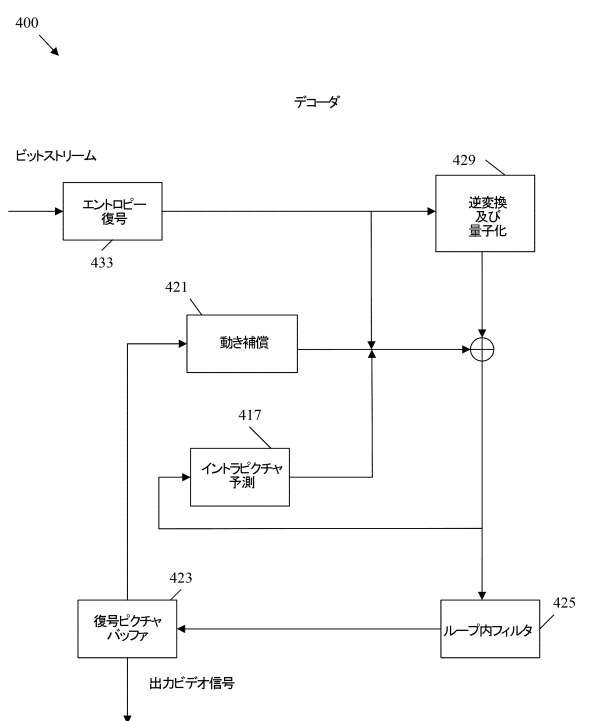
10

20

【図 3】



【図 4】

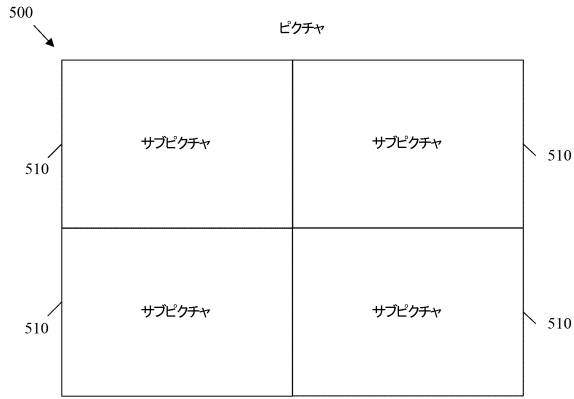


30

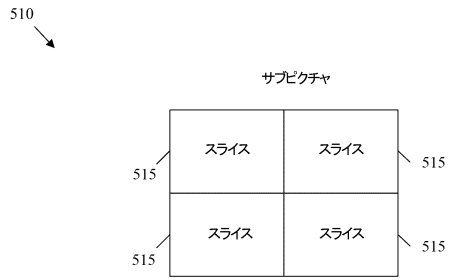
40

50

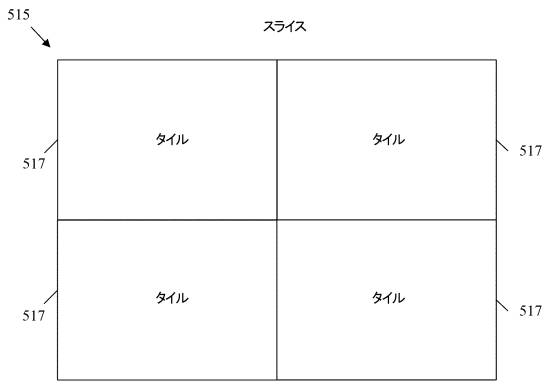
【図 5 A】



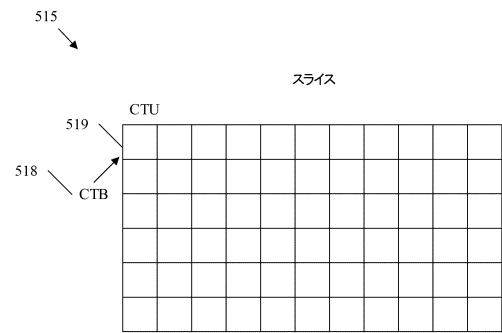
【図 5 B】



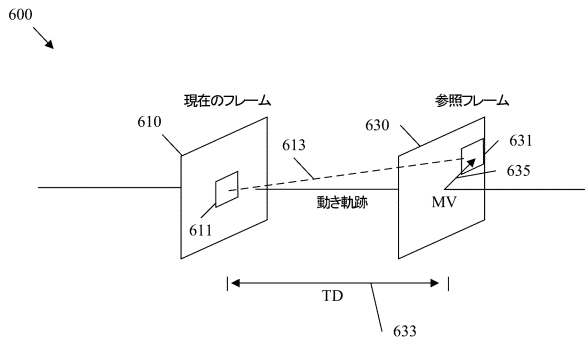
【図 5 C】



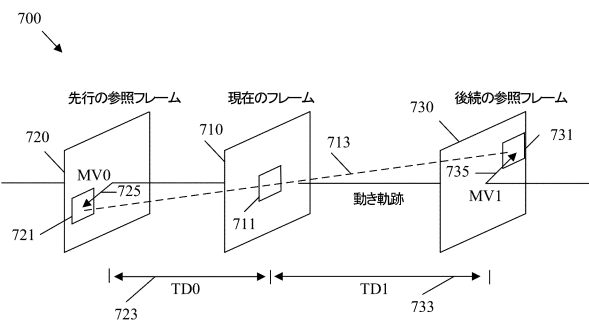
【図 5 D】



【図 6】



【図 7】



10

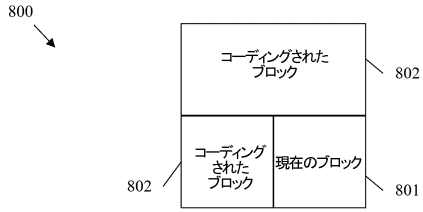
20

30

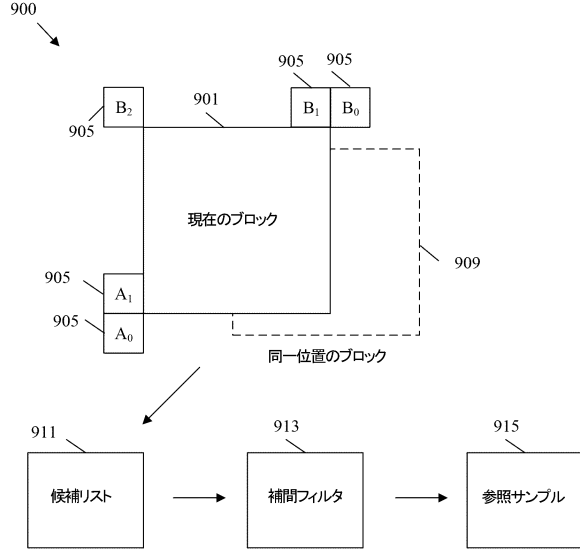
40

50

【図8】

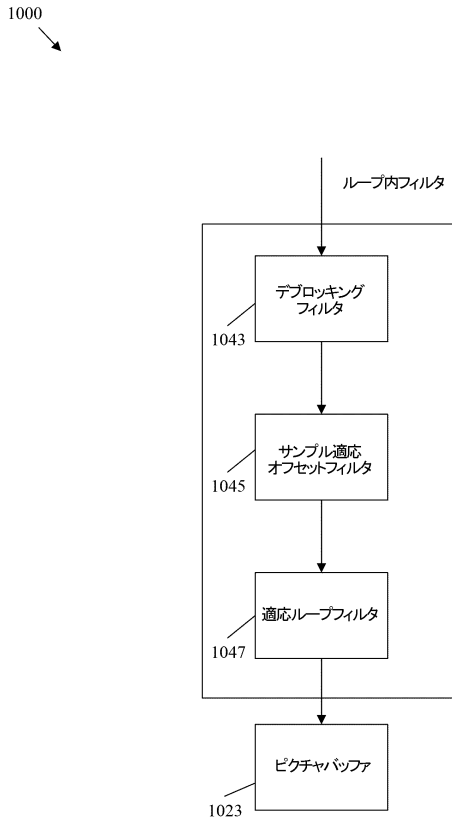


【図9】

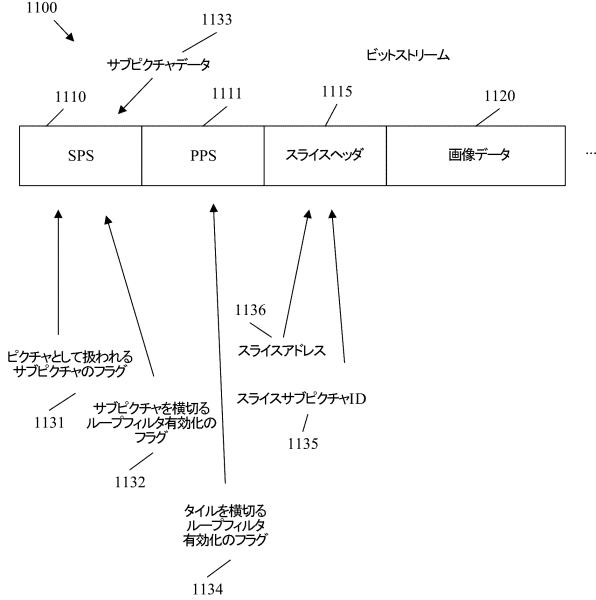


10

【図10】



【図11】



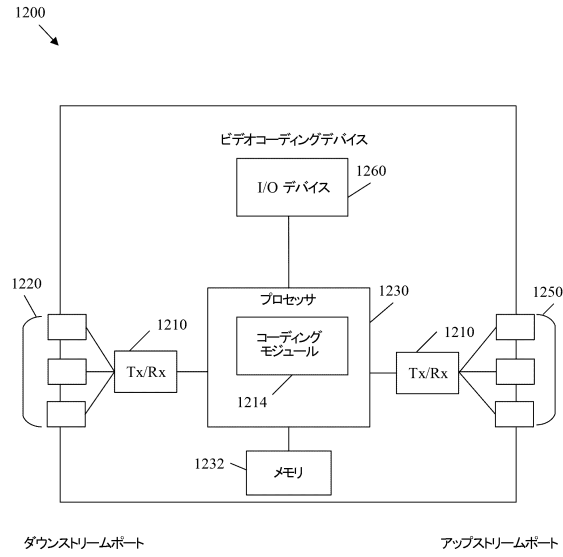
20

30

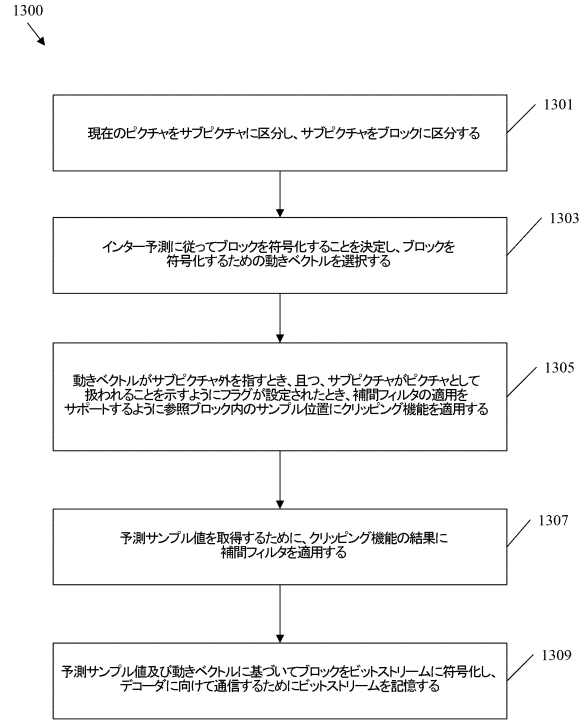
40

50

【図 1 2】



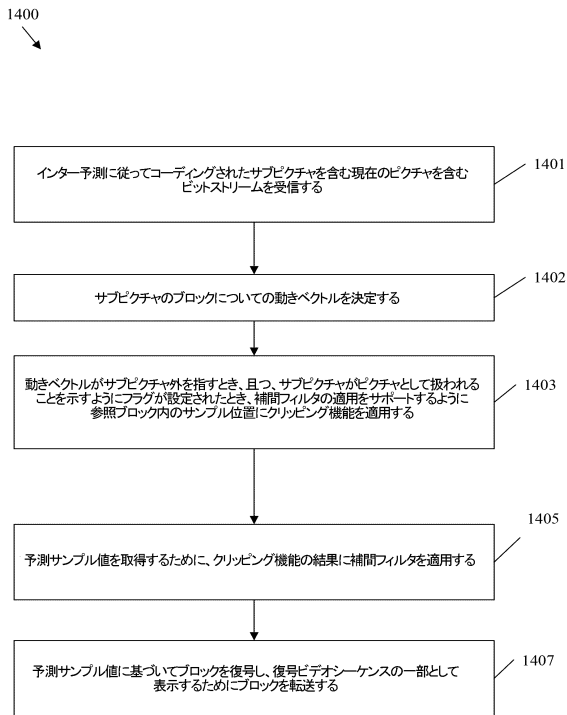
【図 1 3】



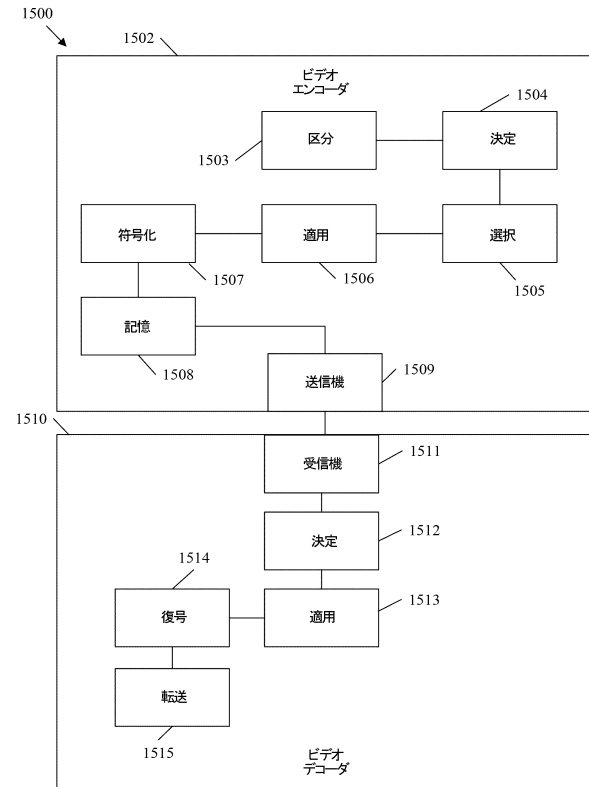
10

20

【図 1 4】



【図 1 5】



30

40

50

フロントページの続き

米国(US)

(31)優先権主張番号 62/826,659

(32)優先日 平成31年3月29日(2019.3.29)

(33)優先権主張国・地域又は機関

米国(US)

(74)代理人 100070150

弁理士 伊東 忠彦

(74)代理人 100135079

弁理士 宮崎 修

(72)発明者 ワン, イェ クォイ

アメリカ合衆国 カリフォルニア州 9 2 1 3 0, サン ディエゴ, サンローズ クレスト ウェイ
6 2 6 4

(72)発明者 チェン, ジャンレ

アメリカ合衆国 カリフォルニア州 9 2 1 3 0, サン ディエゴ, コルテ デ ティプロン 1 0 7 5 6

(72)発明者 ヘンドリー, フヌ

アメリカ合衆国 カリフォルニア州 9 2 1 2 9, サン ディエゴ, フォックスクロフト プレース
8 5 8 4

合議体

審判長 畑中 高行

審判官 櫻本 剛

審判官 圓道 浩史

(56)参考文献 Jianle Chen, et al. , AHG12: Treating tile and tile group boundaries as picture boundaries ,
Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , J
VET-M0136-v1 , 2019年1月3日Benjamin BROSS, et al. , Versatile Video Coding (Draft 4) , Joint Video Experts Team (JVET)
of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-M1001 (version 6) , 2019
年3月9日 , pp.17, 18, 152, 153

(58)調査した分野 (Int.Cl. , D B 名)

H04N 19/00 - 19/98

IEEE Xplore