

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6339183号  
(P6339183)

(45) 発行日 平成30年6月6日(2018.6.6)

(24) 登録日 平成30年5月18日(2018.5.18)

(51) Int.Cl.	F I
HO 4 N 19/597 (2014.01)	HO 4 N 19/597
HO 4 N 19/463 (2014.01)	HO 4 N 19/463
HO 4 N 19/80 (2014.01)	HO 4 N 19/80
HO 4 N 19/52 (2014.01)	HO 4 N 19/52
HO 4 N 19/109 (2014.01)	HO 4 N 19/109

請求項の数 20 (全 69 頁) 最終頁に続く

(21) 出願番号	特願2016-516922 (P2016-516922)
(86) (22) 出願日	平成26年9月26日(2014.9.26)
(65) 公表番号	特表2016-537841 (P2016-537841A)
(43) 公表日	平成28年12月1日(2016.12.1)
(86) 国際出願番号	PCT/US2014/057732
(87) 国際公開番号	W02015/048453
(87) 国際公開日	平成27年4月2日(2015.4.2)
審査請求日	平成29年7月20日(2017.7.20)
(31) 優先権主張番号	61/883,111
(32) 優先日	平成25年9月26日(2013.9.26)
(33) 優先権主張国	米国 (US)
(31) 優先権主張番号	61/887,915
(32) 優先日	平成25年10月7日(2013.10.7)
(33) 優先権主張国	米国 (US)

(73) 特許権者	595020643
	クァアルコム・インコーポレイテッド
	QUALCOMM INCORPORATED
	アメリカ合衆国、カリフォルニア州 92
	121-1714、サン・ディエゴ、モア
	ハウス・ドライブ 5775
(74) 代理人	100108855
	弁理士 蔵田 昌俊
(74) 代理人	100109830
	弁理士 福原 淑弘
(74) 代理人	100158805
	弁理士 井関 守三
(74) 代理人	100112807
	弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 HEVCにおけるサブ予測ユニット (PU) ベースの時間的動きベクトル予測および3D-HEVCにおけるサブPU設計

(57) 【特許請求の範囲】

【請求項1】

ビデオデータを処理する方法であって、

後方ビュー合成予測 (BVSP) モードに従って、ビデオブロックのコーディングユニット (CU) の予測ユニット (PU) のための第1の動き情報を、前記PUの隣接ブロックから決定すること、ここにおいて、前記第1の動き情報は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記PUを2つ以上のサブPUに区分することと、

前記サブPUの各々のための第2の動き情報を決定すること、ここにおいて、前記第2の動き情報は、前記サブPUの各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、と、

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブPUの各々を予測するために、動き補償を実行することと、

後続PUを予測するために使用されることになる、前記PUの前記サブPUの各々のための前記第2の動き情報をメモリに記憶することと、

前記PUの前記サブPUのうちの少なくとも1つのための前記記憶された第2の動き情報に基づいて後続PUを予測することと、

を備え、前記後続PUは、前記PUの隣接ブロックであり、前記後続PUを予測することは、前記後続PUを予測するための動き情報のマージモード候補リストを生成すること

10

20

を備え、前記マージモード候補リストは、前記マージモード候補リスト中のサブPU B V S P 候補として、前記メモリに記憶された前記PUの前記サブPUのうちの少なくとも1つのための前記第2の動き情報を含む

方法。

【請求項2】

前記PUのための前記第1の動き情報を決定することは、前記PUを予測するための動き情報のマージモード候補リストからB V S P 候補として前記第1の動き情報を選択することを備える、

請求項1に記載の方法。

【請求項3】

前記サブPUの各々のための前記第2の動き情報を決定することは、前記サブPUの各々に対応する前記ビュー間参照ピクチャの前記深度ブロックのための4つのコーナーピクセルの最大値に等しい水平成分と、ゼロに等しい垂直成分とを有するように、前記サブPUの各々のための前記少なくとも1つの視差動きベクトルを導出することを備える、

請求項1に記載の方法。

【請求項4】

前記サブPUの各々のための前記第2の動き情報を決定することは、サブPUを予測するための動き情報のマージモード候補リストからサブPU B V S P 候補として前記第2の動き情報を選択することを備える、

請求項1に記載の方法。

【請求項5】

前記メモリは、復号ピクチャバッファを備え、前記第2の動き情報を記憶することは、前記PUのための前記第1の動き情報の前記参照ビューインデックスによって識別された前記ビュー間参照ピクチャを含む参照ピクチャリストとともに、前記サブPUの各々のための前記視差動きベクトルを前記復号ピクチャバッファに記憶することを備える、

請求項1に記載の方法。

【請求項6】

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記PUの前記サブPUの各々のための予測ブロックを生成することと、

前記ビデオブロックと前記サブPUの各々のための前記予測ブロックとに基づいて、残差ブロックを生成することと、

ビデオビットストリーム中に前記PUのための少なくとも前記第1の動き情報のインジケータと、前記残差ブロックとを符号化することと

をさらに備える、請求項1に記載の方法。

【請求項7】

受信されたビデオビットストリームからの前記PUのための少なくとも前記第1の動き情報のインジケータと、残差ブロックとを復号することと、

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記PUの前記サブPUの各々のための予測ブロックを生成することと、

前記残差ブロックと前記サブPUの各々のための前記予測ブロックとに基づいて、前記ビデオブロックの再構築されたバージョンを生成することと

をさらに備える、請求項1に記載の方法。

【請求項8】

ビデオデータを処理する方法であって、

後方ビュー合成予測(B V S P)モードに従って、ビデオブロックのコーディングユニット(CU)の予測ユニット(PU)のための第1の動き情報を、前記PUの隣接ブロックから決定すること、ここにおいて、前記第1の動き情報は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記PUを2つ以上のサブPUに区分することと、

10

20

30

40

50

前記サブPUの各々のための第2の動き情報を決定すること、ここにおいて、前記第2の動き情報は、前記サブPUの各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、と、

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブPUの各々を予測するために、動き補償を実行することと、

前記ビデオブロックを復号ピクチャバッファに記憶するのに先立って、前記ビデオブロックの再構築されたバージョンの各CUにデブロッキングフィルタを適用することと

をさらに備え、前記2つ以上のサブPUを有する前記PUを含む前記ビデオブロックの前記CUに前記デブロッキングフィルタを適用することは、

前記CUに前記デブロッキングフィルタを適用するのに先立って、前記PU中の2つの隣接するサブPU間のサブPU境界において人工的PU境界または人工的変換ユニット(TU)境界を作成することと、

前記人工的PU境界または前記人工的TU境界を含む、前記CUの2つの隣接するPU間のPU境界または前記CUの2つの隣接するTU間のTU境界のうちの少なくとも1つに前記デブロッキングフィルタを適用することと

を備える、請求項1に記載の方法

を備える、方法。

#### 【請求項9】

前記人工的PU境界または前記人工的TU境界を作成することは、前記PUを変換ツリー階層に関連付け、前記サブPUの各々をTUに関連付けるために、前記CUの変換ツリーを変換することを備え、前記サブPU境界は、人工的TU境界を備える、

請求項8に記載の方法。

#### 【請求項10】

前記人工的PU境界または前記人工的TU境界を作成することは、前記PUを変換ツリー階層に関連付け、前記サブPUの各々の少なくとも一部をTUに関連付けるために、前記CUの変換ツリーを変換することを備え、前記サブPU境界は、人工的TUを備える、請求項8に記載の方法。

#### 【請求項11】

前記人工的PU境界または前記人工的TU境界を作成することは、前記PUをCUに関連付け、前記サブPUの各々をPUに関連付けるために、前記CUをコーディングツリーに変換することを備え、前記サブPU境界は、人工的PU境界を備える、

請求項8に記載の方法。

#### 【請求項12】

ビデオ処理デバイスであって、

ビデオデータを記憶するように構成されたメモリと、

前記メモリと通信する1つまたは複数のプロセッサと

を備え、前記1つまたは複数のプロセッサは、

後方ビュー合成予測(BVSP)モードに従って、ビデオブロックのコーディングユニット(CU)の予測ユニット(PU)のための第1の動き情報を、前記PUの隣接ブロックから決定すること、ここにおいて、前記第1の動き情報は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記PUを2つ以上のサブPUに区分することと、

前記サブPUの各々のための第2の動き情報を決定すること、ここにおいて、前記第2の動き情報は、前記サブPUの各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、と、

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブPUの各々を予測するために、動き補償を実行することと、

後続PUを予測するために使用されることになる、前記PUの前記サブPUの各々のための前記第2の動き情報を前記メモリに記憶することと、

10

20

30

40

50

前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記記憶された第 2 の動き情報に基づいて後続 P U を予測することと

を行うように構成され、前記後続 P U は、前記 P U の隣接ブロックであり、前記 1 つまたは複数のプロセッサは、前記後続 P U を予測するための動き情報のマージモード候補リストを生成することを行うように構成され、前記マージモード候補リストは、前記マージモード候補リスト中のサブ P U B V S P 候補として、前記メモリに記憶された前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記第 2 の動き情報を含む

ビデオ処理デバイス。

【請求項 13】

前記 1 つまたは複数のプロセッサは、前記 P U を予測するための動き情報のマージモード候補リストから B V S P 候補として前記第 1 の動き情報を選択することを行うように構成される、

請求項 12 に記載のデバイス。

【請求項 14】

前記 1 つまたは複数のプロセッサは、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの前記深度ブロックのための 4 つのコーナーピクセルの最大値に等しい水平成分と、ゼロに等しい垂直成分とを有するように、前記サブ P U の各々のための前記第 2 の動き情報に含まれた前記少なくとも 1 つの視差動きベクトルを導出することを行うように構成される、

請求項 12 に記載のデバイス。

【請求項 15】

前記 1 つまたは複数のプロセッサは、サブ P U を予測するための動き情報のマージモード候補リストからサブ P U B V S P 候補として前記サブ P U の各々のための前記第 2 の動き情報を選択することを行うように構成される、

請求項 12 に記載のデバイス。

【請求項 16】

前記メモリは、復号ピクチャバッファを備え、前記 1 つまたは複数のプロセッサは、前記 P U のための前記第 1 の動き情報の前記参照ビューインデックスによって識別された前記ビュー間参照ピクチャを含む参照ピクチャリストとともに、前記サブ P U の各々のための前記第 2 の動き情報に含まれた前記視差動きベクトルを前記復号ピクチャバッファに記憶することを行うように構成される、

請求項 12 に記載のデバイス。

【請求項 17】

前記ビデオ処理デバイスは、ビデオ符号化デバイスを備え、前記 1 つまたは複数のプロセッサは、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記ビデオブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、残差ブロックを生成することと、

ビデオビットストリーム中に前記 P U のための少なくとも前記第 1 の動き情報のインジケータと、前記残差ブロックとを符号化することと

を行うように構成される、請求項 12 に記載のデバイス。

【請求項 18】

前記ビデオ処理デバイスは、ビデオ復号デバイスを備え、前記 1 つまたは複数のプロセッサは、

受信されたビデオビットストリームからの前記 P U のための少なくとも前記第 1 の動き情報のインジケータと、残差ブロックとを復号することと、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記残差ブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、前記

10

20

30

40

50

ビデオブロックの再構築されたバージョンを生成することと  
を行うように構成される、請求項 12 に記載のデバイス。

【請求項 19】

後方ビュー合成予測 (BVS P) モードに従って、ビデオブロックのコーディングユニット (C U) の予測ユニット (P U) のための第 1 の動き情報を、前記 P U の隣接ブロックから決定するための手段、ここにおいて、前記第 1 の動き情報は、少なくとも 1 つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記 P U を 2 つ以上のサブ P U に区分するための手段と、

前記サブ P U の各々のための第 2 の動き情報を決定するための手段、ここにおいて、前記第 2 の動き情報が、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも 1 つの視差動きベクトルを含む、と、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブ P U の各々を予測するために、動き補償を実行するための手段と、

後続 P U を予測するために使用されることになる、前記 P U の前記サブ P U の各々のための前記第 2 の動き情報をメモリに記憶するための手段と、

前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記記憶された第 2 の動き情報に基づいて後続 P U を予測するための手段と、

を備え、前記後続 P U は、前記 P U の隣接ブロックであり、前記後続 P U を予測するための手段は、前記後続 P U を予測するための動き情報のマージモード候補リストを生成するための手段をさらに備え、前記マージモード候補リストは、前記マージモード候補リスト中のサブ P U BVS P 候補として、前記メモリに記憶された前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記第 2 の動き情報を含む

ビデオ処理デバイス。

【請求項 20】

ビデオデータを処理するための命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記命令は、実行されると、1 つまたは複数のプロセッサに、

後方ビュー合成予測 (BVS P) モードに従って、ビデオブロックのコーディングユニット (C U) の予測ユニット (P U) のための第 1 の動き情報を、前記 P U の隣接ブロックから決定すること、ここにおいて、前記第 1 の動き情報は、少なくとも 1 つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、

前記 P U を 2 つ以上のサブ P U に区分することと、

前記サブ P U の各々のための第 2 の動き情報を決定すること、ここにおいて、前記第 2 の動き情報は、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも 1 つの視差動きベクトルを含む、と

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブ P U の各々を予測するために、動き補償を実行することと、

後続 P U を予測するために使用されることになる、前記 P U の前記サブ P U の各々のための前記第 2 の動き情報をメモリに記憶することと、

前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記記憶された第 2 の動き情報に基づいて後続 P U を予測することと、

を行わせ、前記後続 P U は、前記 P U の隣接ブロックであり、前記命令は、前記 1 つまたは複数のプロセッサに、前記後続 P U を予測するための動き情報のマージモード候補リストを生成することを行わせ、前記マージモード候補リストは、前記マージモード候補リスト中のサブ P U BVS P 候補として、前記メモリに記憶された前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記第 2 の動き情報を含む

非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【関連出願】

## 【 0 0 0 1 】

[0001]本出願は、各々の内容全体が参照により本明細書に組み込まれる、2013年9月26日に出願された米国仮特許出願第61/883,111号、2013年10月7日に出願された米国仮特許出願第61/887,915号、2013年10月17日に出願された米国仮特許出願第61/892,327号、および2013年10月31日に出願された米国仮特許出願第61/898,385号の利益を主張する。

## 【 技術分野 】

## 【 0 0 0 2 】

[0002]本開示は、ビデオコーディングに関する。

## 【 背景技術 】

10

## 【 0 0 0 3 】

[0003]デジタルビデオ能力は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップコンピュータもしくはデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲームデバイス、ビデオゲーム機、セルラー電話機もしくは衛星無線電話機、いわゆる「スマートフォン」、ビデオ会議デバイス、ビデオストリーミングデバイス、および類似物を含む広範囲のデバイスに組み込まれ得る。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、Part 10、アドバンスドビデオコーディング(AVC: Advanced Video Coding)、高効率ビデオコーディング(HEVC: High Efficiency Video Coding)規格によって定義された規格、およびそのような規格の拡張に記述されているビデオ圧縮技法などのビデオ圧縮技法を実装する。これらのビデオデバイスは、そのようなビデオ圧縮技法を実装することによって、デジタルビデオ情報をより効率的に送信し、受信し、符号化し、復号し、および/または記憶することができる。

20

## 【 0 0 0 4 】

[0004]ビデオ圧縮技法は、ビデオシーケンスに固有の冗長性を低減または除去するために、空間的(ピクチャ内)予測および/または時間的(ピクチャ間)予測を実行する。ブロックベースのビデオコーディングでは、ビデオスライス(すなわち、ビデオフレームまたはビデオフレームの一部)が、ツリーブロック、コーディングユニット(CU)および/またはコーディングノードと呼ばれることもあるビデオブロックに区分され得る。ピクチャのイントラコード化(I)スライスにおけるビデオブロックは、同じピクチャ中の隣接ブロックにおける参照サンプルに対する空間的予測を使用して符号化される。ピクチャのインターコード化(PまたはB)スライスにおけるビデオブロックは、同じピクチャ中の隣接ブロックにおける参照サンプルに対する空間的予測、または他の参照ピクチャ中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

30

## 【 0 0 0 5 】

[0005]空間的予測または時間的予測によって、コーディングされるべきブロックの予測ブロックが生じる。残差データは、コーディングすべき元のブロックと予測ブロックとの間のピクセル差分を表す。インターコード化ブロックは、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に従って符号化され、残差データは、コード化ブロックと予測ブロックとの間の差分を示す。イントラコード化ブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは、ピクセル領域から変換領域に変換され、次いで量子化され得る残差変換係数が生じ得る。最初に2次元アレイで構成される量子化変換係数は、変換係数の1次元ベクトルを生成するためにスキャンされ得、なお一層の圧縮を達成するためにエントロピーコーディングが適用され得る。

40

## 【 発明の概要 】

## 【 0 0 0 6 】

50

[0006]一般に、本開示は、高効率ビデオコーディング（H E V C）規格における、およびH E V C規格に対する3 D拡張における、ビデオコーディングのためのサブ予測ユニット（P U）ベースの動き予測の技法について説明する。一例では、本開示は、動きベクトル改良が可能にされ得るシングルレイヤコーディングにおけるP UのサブP Uを予測するための高度時間的動きベクトル予測（T M V P：temporal motion vector prediction）モードを実行するための技法について説明する。従来、サブP U設計は、予測された動きベクトルのさらなる改良を可能にしないマージインター予測モードを使用したレイヤ間またはビュー間の動き予測のためにのみ可能にされる。高度T M V Pモードは、P UのサブP Uの各々のために異なる動きベクトルおよび参照インデックスを含む、P Uのための動き情報を導出するために、少なくとも2つのステージでP Uのための動きベクトルを決定

10

【0007】

[0007]3 D - H E V C拡張は、マルチビュービデオに加えて深度フォーマットのコーディングをサポートする。一例では、サブP U後方ビュー合成予測（B V S P：sub-PU backward view synthesis prediction）モードでは、現在のP Uのための動き情報は、視差動きベクトルおよび関連付けられたビュー間参照ピクチャから予測され、視差動きベクトルは、ビュー間参照ピクチャの深度ブロックに関連付けられたサブP U動き情報に基づいて改良される。サブP U B V S Pモードでは、別個のサブP U動き情報に基づいてP UのサブP Uの各々を予測するために、動き補償が実行される。従来、サブP Uの各々を予測するために動き補償を実行した後、現在のP Uのための視差動きベクトルだけが記憶される。本開示の技法によれば、サブP U B V S Pモードを使用して予測される各P Uに対して、ビデオコーディングデバイスは、動き補償が実行された後ですら、現在のP UのサブP Uの各々に対して導出された別個の動き情報を記憶する。現在のP Uに対して記憶された追加の動き情報は、次いで、現在のP Uが隣接ブロックである後続P Uを予測するために使用され得る。

20

【0008】

[0008]加えて、本開示は、サブP U境界において人工的P U境界または人工的T U境界を作成することによって、変換ユニット（T U）境界およびコーディングユニット（C U）中のサブP U境界を含むP U境界をフィルタリングするために、ビデオブロックの各C Uにデブロッキングフィルタを適用するための技法について説明する。

30

【0009】

[0009]一例では、本開示は、高度T M V Pモードの第1のステージとして、P Uの隣接ブロックからビデオブロックのC UのP Uのための第1のステージの動きベクトルを決定することと、ここにおいて、第1のステージの動きベクトルが、P Uに対応する参照ピクチャのブロックを識別する、P Uを2つ以上のサブP Uに区分することと、高度T M V Pモードの第2のステージとして、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブP Uの各々のための第2のステージの動き情報を決定することと、ここにおいて、サブP Uの各々のための第2のステージの動き情報が、少なくとも1つの動きベクトルと関連付けられた参照インデックスとを含む、サブP Uの各々のための第2のステージの動き情報に基づいて、サブP Uの各々のための動き補償を別個に実行することとを備える、ビデオデータを処理する方法を対象とする。

40

【0010】

[0010]別の例では、本開示は、ビデオデータを記憶するように構成されたメモリと、メモリと通信する、1つまたは複数のプロセッサとを備えるビデオ処理デバイスを対象とする。1つまたは複数のプロセッサは、高度T M V Pモードの第1のステージとして、P Uの隣接ブロックからビデオブロックのC UのP Uのための第1のステージの動きベクトルを決定することと、ここにおいて、第1のステージの動きベクトルが、P Uに対応する参照ピクチャのブロックを識別する、P Uを2つ以上のサブP Uに区分することと、高度T M V Pモードの第2のステージとして、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブP Uの各々のための第2のステージの動き情報を決定

50

することと、ここにおいて、サブPUの各々のための第2のステージの動き情報が、少なくとも1つの動きベクトルと関連付けられた参照インデックスとを含む、サブPUの各々のための第2のステージの動き情報に基づいて、サブPUの各々のための動き補償を別個に実行することとを行うように構成される。

【0011】

[0011]さらなる例では、本開示は、高度TMVPモードの第1のステージとして、PUの隣接ブロックからビデオブロックのCUのPUのための第1のステージの動きベクトルを決定するための手段と、ここにおいて、第1のステージの動きベクトルが、PUに対応する参照ピクチャのブロックを識別する、PUを2つ以上のサブPUに区分するための手段と、高度TMVPモードの第2のステージとして、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブPUの各々のための第2のステージの動き情報を決定するための手段と、ここにおいて、サブPUの各々のための第2のステージの動き情報が、少なくとも1つの動きベクトルと関連付けられた参照インデックスとを含む、サブPUの各々のための第2のステージの動き情報に基づいて、サブPUの各々のための動き補償を別個に実行するための手段とを備える、ビデオ処理デバイスを対象とする。

10

【0012】

[0012]追加の例では、本開示は、実行されるとき、1つまたは複数のプロセッサに、高度TMVPモードの第1のステージとして、PUの隣接ブロックからビデオブロックのCUのPUのための第1のステージの動きベクトルを決定することと、ここにおいて、第1のステージの動きベクトルが、PUに対応する参照ピクチャのブロックを識別する、PUを2つ以上のサブPUに区分することと、高度TMVPモードの第2のステージとして、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブPUの各々のための第2のステージの動き情報を決定することと、ここにおいて、サブPUの各々のための第2のステージの動き情報が、少なくとも1つの動きベクトルと関連付けられた参照インデックスとを含む、サブPUの各々のための第2のステージの動き情報に基づいて、サブPUの各々のための動き補償を別個に実行することとを行わせる、ビデオデータを処理するための命令を記憶したコンピュータ可読記憶媒体を対象とする。

20

【0013】

[0013]別の例では、本開示は、ビデオデータを処理する方法に関し、本方法は、BVS Pモードに従って、PUの隣接ブロックからビデオブロックのCUのPUのための第1の動き情報を決定することと、ここにおいて、第1の動き情報が、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、PUを2つ以上のサブPUに区分することと、サブPUの各々のための第2の動き情報を決定することと、ここにおいて、第2の動き情報が、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、第2の動き情報に基づいて、ビュー間参照ピクチャからサブPUの各々を予測するために、動き補償を実行することと、後続PUを予測するために使用されるように、PUのサブPUの各々のための第2の動き情報をメモリに記憶することとを備える。

30

40

【0014】

[0014]さらなる例では、本開示は、ビデオデータを記憶するように構成されたメモリと、メモリと通信する、1つまたは複数のプロセッサとを備えるビデオ処理デバイスを対象とする。1つまたは複数のプロセッサは、BVS Pモードに従って、PUの隣接ブロックからビデオブロックのCUのPUのための第1の動き情報を決定することと、ここにおいて、第1の動き情報が、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、PUを2つ以上のサブPUに区分することと、サブPUの各々のための第2の動き情報を決定することと、ここにおいて、第2の動き情報が、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、第2の動き情報に基づ

50



いて、ビュー間参照ピクチャからサブPUの各々を予測するために、動き補償を実行することを行うように構成される。メモリは、後続PUを予測するために使用されるように、PUのサブPUの各々のための第2の動き情報を記憶するように構成される。

【0015】

[0015]追加の例では、本開示は、BVS Pモードに従って、PUの隣接ブロックからビデオブロックのCUのPUのための第1の動き情報を決定するための手段と、ここにおいて、第1の動き情報が、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、PUを2つ以上のサブPUに区分するための手段と、サブPUの各々のための第2の動き情報を決定するための手段と、ここにおいて、第2の動き情報が、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、第2の動き情報に基づいて、ビュー間参照ピクチャからサブPUの各々を予測するために、動き補償を実行するための手段と、後続PUを予測するために使用されるように、PUのサブPUの各々のための第2の動き情報をメモリに記憶するための手段とを備えるビデオ処理デバイスを対象とする。

10

【0016】

[0016]別の例では、本開示は、実行されるとき、1つまたは複数のプロセッサに、BVS Pモードに従って、PUの隣接ブロックからビデオブロックのCUのPUのための第1の動き情報を決定することと、ここにおいて、第1の動き情報が、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、PUを2つ以上のサブPUに区分することと、サブPUの各々のための第2の動き情報を決定することと、ここにおいて、第2の動き情報が、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、第2の動き情報に基づいて、ビュー間参照ピクチャからサブPUの各々を予測するために、動き補償を実行することと、後続PUを予測するために使用されるように、PUのサブPUの各々のための第2の動き情報をメモリに記憶することとを行わせる、ビデオデータを処理するための命令を記憶したコンピュータ可読記憶媒体を対象とする。

20

【0017】

[0017]添付の図面および以下の説明において、1つまたは複数の例の詳細について説明する。他の特徴、目的、および利点は、説明および図面から、ならびに特許請求の範囲から明らかになるであろう。

30

【図面の簡単な説明】

【0018】

【図1】本開示の技法を利用し得る例示的なビデオ符号化および復号システムを示すブロック図。

【図2】コーディングユニット(CU)の予測ユニット(PU)を定義するための8つの区分モードを示す概念図。

【図3】CU内の変換ツリー(transform tree)構造を示す概念図。

【図4A】マージモードのための空間的隣接候補を示す概念図。

40

【図4B】高度動きベクトル予測(AMVP)モードのための空間的隣接候補を示す概念図。

【図5A】時間的動きベクトル予測(TMVP)候補を示す概念図。

【図5B】TMVPのための動きベクトル(MV)スケーリングを示す概念図。

【図6】例示的なデブロッキングフィルタプロセスを示すフローチャート。

【図7】デブロッキングフィルタプロセスのための境界強度(B)計算の例示的な動作を示すフローチャート。

【図8】デブロッキングフィルタプロセスのためのコーディングツリー(coding tree)ユニット(CTU)境界におけるB計算のために再使用または参照される情報を示す概念図。

50

【図 9】デブロッキングフィルタプロセスのためのオン / オフフィルタ決定および強い / 弱いフィルタ選択に関与するピクセルを示す概念図。

【図 10】V3 がベースビューを示し、非ベースビュー（すなわち、V1 または V5）中のピクチャが同じ時間インスタンスのベースビュー中のピクチャから予測され得る、3つのビューの事例に対するマルチビュー予測構造を示す概念図。

【図 11】隣接ブロックベースの視差ベクトル（NB DV）導出のためにアクセスされる空間的隣接ブロックがマージモードおよび AMVP モードのためにアクセスされる空間的隣接ブロックと同じである、NB DV 導出のための空間的隣接ブロックを示す概念図。

【図 12】PU の異なるサブ PU のための異なる動きベクトルを割り当てる後方ビュー合成予測（BVSP）モードを示す概念図。

10

【図 13】サブ PU レベルのビュー間動き予測を示す概念図。

【図 14 A】シングルレイヤコーディングにおける PU 中のサブ PU を予測するための高度 TMVP モードを示す概念図。

【図 14 B】シングルレイヤコーディングにおける PU 中のサブ PU を予測するための高度 TMVP モードを示す概念図。

【図 15】本開示の技法を実装し得るビデオ符号化器の一例を示すブロック図。

【図 16】本開示の技法を実装し得るビデオ復号器の一例を示すブロック図。

【図 17】サブ PU BVSP モードを使用して、現在の PU を予測し、決定されたサブ PU 動き情報記憶を記憶する例示的な動作を示すフローチャート。

【図 18】CU 中のサブ PU 境界を含めて、TU 境界および PU 境界をフィルタリングするために、ビデオブロックの各 CU にデブロッキングフィルタを適用する例示的な動作を示すフローチャート。

20

【図 19】シングルレイヤコーディングにおける PU のサブ PU を予測するための高度 TMVP モードの例示的な動作を示すフローチャート。

【発明を実施するための形態】

【0019】

[0038]本開示は、高効率ビデオコーディング（HEVC）規格に対する 3D 拡張を使用する、ビデオコーディングのためのサブ予測ユニット（PU）レベルの動き予測のための技法について説明する。3D-HEVC 拡張は、マルチビュービデオに加えて深度フォーマットのコーディングをサポートする。マルチビュービデオコーディングでは、現在のビデオピクチャは、同じビュー中の時間的参照ピクチャと参照ビュー中のビュー間参照ピクチャの両方によって予測され得る。

30

【0020】

[0039]一例では、3D-HEVC におけるビュー合成予測は、サブ PU 後方ビュー合成予測（BVSP）モードを使用して実行され得る。サブ PU BVSP モードでは、現在の PU のための動き情報は、視差動きベクトル、およびビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスから予測される。サブ PU BVSP モードでは、現在の PU は 2 つ以上のサブ PU に区分され、視差動きベクトルは、ビュー間参照ピクチャの対応する深度ブロックに基づいて、サブ PU の各々に対して導出されたサブ PU 動き情報に基づいて改良される。別個のサブ PU 動き情報に基づいて PU のサブ PU の各々を予測するために、動き補償が実行される。従来、サブ PU の各々を予測するために動き補償を実行した後、現在の PU のための視差動きベクトルだけが記憶される。

40

【0021】

[0040]本開示の技法によれば、サブ PU BVSP モードを使用して予測される各 PU に対して、ビデオコーディングデバイスは、動き補償が実行された後ですら、現在の PU のサブ PU の各々に対して導出された別個の動き情報を記憶する。現在の PU に対して記憶された追加の動き情報は、次いで、現在の PU が隣接ブロックである後続 PU を予測するために使用され得る。

【0022】

[0041]別の例では、本開示は、動きベクトル改良が可能にされ得るシングルレイヤコー

50

ディングにおけるPUのサブPUを予測するための高度時間的動きベクトル予測(TMP)モードを実行するための技法について説明する。従来、サブPU設計は、予測された動きベクトルのさらなる改良を可能にしないマージインター予測モードを使用したレイヤ間またはビュー間の動き予測に対してのみ可能にされる。高度TMPモードは、PUのサブPUの各々のための異なる動きベクトルおよび参照インデックスを含む、PUのための動き情報を導出するために、少なくとも2つのステージでPUのための動きベクトルを決定することを含む。PU内のサブPUは、 $N \times N$ または $N \times M$ のサイズであり得、 $N$ および $M$ は、任意の整数であり得る。典型的なサブPUサイズは、 $4 \times 4$ 、 $8 \times 8$ 、 $4 \times 8$ 、または $8 \times 4$ であり得る。

【0023】

10

[0042]本開示の技法によれば、現在のPUのための第1のステージの動きベクトルは、そのPUの隣接ブロックから決定され、この場合、第1のステージの動きベクトルは、現在のPUに対応する参照ピクチャのブロックを識別する。3D-HEVCにおけるサブPU動き予測と同様に、より小さいサブPUを含む参照ピクチャの対応する領域(たとえば、ブロック)のサイズは、現在のPUのサイズと同じである。このようにして、現在のPUのサブPUは、一意の動きベクトルを用いて識別された参照ピクチャの対応する領域中に対応するサブPUを有する。第1のステージの動きベクトルは、空間的および/または時間的近隣に基づいて識別され得る。代替的に、第1のステージの動きベクトルは、定数、たとえば、0、または別のあらかじめ規定された値に設定され得、現在のPUのサイズに関連する。別の代替態様では、PUのサブPUの各々は、異なる動きベクトルを用いて参照ピクチャのその対応するブロックを識別することができる。

20

【0024】

[0043]上記で説明したように、現在のPUは2つ以上のサブPUに区分され、サブPUの各々のための第2のステージの動き情報は、第1のステージの動きベクトルによって識別された参照ピクチャの対応する領域、たとえば、ブロックから決定され、この場合、サブPUの各々のための第2のステージの動き情報は、少なくとも1つの動きベクトルと、関連付けられた参照インデックスとを含む。サブPUの各々のための第2のステージの動き情報に基づいて、サブPUの各々を別々に予測するために、動き補償が実行される。この高度TMPプロセスは、すべての他のマージ候補の中でマージ候補をアクティブ化すると、つまり、現在のPUのためにそのようなプロセスが実行される必要があることを示すために、特定のマージ候補が作成されると達成され得る。いくつかの例では、単にマージモードで動作する代わりに、サブPUの各々のための第2のステージの動き情報の少なくとも1つの動きベクトルを改良するために、動きベクトル差分が決定され得る。

30

【0025】

[0044]さらなる例として、本開示は、複数のサブPUを有する、少なくとも1つのPUを含むビデオブロックのコーディングユニット(CU)に適用されるデブロッキングフィルタプロセスに関する技法について説明する。HEVC規格に対するデブロッキングフィルタは、PU内で、すなわち、PUの隣接するサブPU間のサブPU境界に沿って、フィルタするように設計されていないが、これは、HEVCブロックでは、動き補償はPU全体に対して同じであると仮定されるためである。本開示の技法は、HEVCデブロッキングフィルタが3D-HEVCブロックのために引き続き使用され得るように、サブPUをデブロッキングフレンドリーな構造に変換することによって、サブPU境界がデブロッキングされるのを可能にする。提案されるデブロッキング技法は、3D-HEVCを含めて、マルチレイヤコーデックに適用可能であり得る。

40

【0026】

[0045]本開示の技法によれば、サブPUを有するPUを含むビデオブロックのCUにデブロッキングフィルタを適用するのに先立って、サブPU境界において人工的PU境界または人工的TU境界を作成するために、CUは変換される。一例では、サブPU境界が人工的TU境界に変換されるように、PUを変換ツリー階層に関連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーは変換される。別の例では、サブPU境界が

50

人工的PU境界に変換されるように、PUをCUに関連付け、サブPUの各々をPUに関連付けるために、CUはコーディングツリーに変換される。次いで、人工的PU境界と人工的TU境界とを含めて、CUの2つの隣接するPU間のPU境界および/またはCUの2つの隣接するTU間のTU境界にデブロッキングフィルタが適用され得る。

#### 【0027】

[0046]ビデオコーディング規格は、ITU-T H.261と、ISO/IEC MPEG-1 Visualと、ITU-T H.262またはISO/IEC MPEG-2 Visualと、ITU-T H.263と、ISO/IEC MPEG-4 Visualと、そのスケーラブルビデオコーディング(SVC)拡張およびマルチビュービデオコーディング(MVC)拡張を含む(ISO/IEC MPEG-4 AVCとしても知られる)ITU-T H.264とを含む。MVCの最新のジョイントドラフトは、ITU-T H.264、Series H: Audiovisual and Multimedia Systems、Infrastructure of audiovisual service - Coding of moving video、Advanced video coding for generic audiovisual services、Telecommunication Standardization Sector of International Telecommunication Union (ITU)、2010年3月に記載されている。

#### 【0028】

[0047]加えて、新しいビデオコーディング規格、すなわち、高効率ビデオコーディング(HEVC)の設計が、ITU-Tビデオコーディングエキスパートグループ(VEG: Video Coding Experts Group)およびISO/IECモーションピクチャエキスパートグループ(MPEG: Motion Picture Experts Group)のビデオコーディング共同研究部会(JCT-VC: Joint Collaboration Team on Video Coding)によって確定された。「HEVC Working Draft 10」または「WD10」と呼ばれるHEVCドラフト仕様は、Brossら、「High efficiency video coding (HEVC) text specification draft 10 (FDIS & Last Call に対して)」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11のビデオコーディング共同研究部会(JCT-VC)、第12回会合: ジュネーブ、スイス、2013年1月14~23日、JCTVC-L1003v34に記載されており、[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip)から入手可能である。

#### 【0029】

[0048]確定された規格文書は、ITU-T H265、Series H: Audiovisual and Multimedia Systems、Infrastructure of audiovisual services - Coding of moving video、High efficiency video coding、Telecommunication Standardization Sector of International Telecommunication Union (ITU)、2013年4月として公開されている。Il-Koo Kimら、「High Efficiency Video Coding (HEVC) Test Model 10 (HM10) Encoder Description」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコーディング共同研究部会(JCT-VC)、第12回会合: ジュネーブ、スイス、2013年1月14~23日に記載されたHEVCの最近の符号化器記述は、[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1002-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1002-v3.zip)から入手可能である。

#### 【0030】

[0049]3Dサービスをサポートする、HEVCの2つの拡張が、ITU-Tビデオコーディングエキスパートグループ(VEG)およびISO/IECモーションピクチャエキスパートグループ(MPEG)の3Dビデオコーディング共同研究部会(JCT-3V

: Joint Collaboration Team on 3D Video coding) によって開発されている。これらの2つの拡張は、HEVCに対するマルチビュー拡張(MV-HEVC)およびHEVCに対する3D拡張(3D-HEVC)である。

【0031】

[0050] MV-HEVCは、HEVCのブロックレベル設計を変更せずに複数のテクスチャビューのコーディングをサポートする。MV-HEVC Working Draft 5(WD5)と呼ばれ、Techら、「MV-HEVC Draft Text 5」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11との3Dビデオコーディング拡張開発に関する共同研究部会(JCT-3V: Joint Collaborative Team on 3D Video Coding Extension Development)、第5回会議: ウィーン、オーストリア、2013年7月27~8月2日、JCT3V-E1004v6に記載されるMV-HEVCのドラフト仕様は、[http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/5\\_Vienna/wg11/JCT3V-E1004-v6.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip)から入手可能である。

10

【0032】

[0051] 3D-HEVCは、マルチビュービデオに加えて深度フォーマットのコーディングをサポートし、HEVCコーディングモジュールに加えて組み込まれた新しいコーディングツールを含む。新たに導入されたコーディングツールは、テクスチャコーディングと深度コーディングの両方に適用可能である。「3D-HEVC Working Draft 1(WD1)」と呼ばれ、Techら、「3D-HEVC Draft Text 1」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11との3Dビデオコーディング拡張開発に関する共同研究部会(JCT-3V)、第5回会議: ウィーン、オーストリア、2013年7月27~8月2日、JCT3V-E1001v3に記載される3D-HEVCのドラフト仕様は、[http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/5\\_Vienna/wg11/JCT3V-E1001-v3.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip)から入手可能である。

20

【0033】

[0052] 図1は、本開示の技法を利用し得る例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示すように、システム10は、宛先デバイス14によって後で復号されるべき符号化ビデオデータを提供するソースデバイス12を含む。特に、ソースデバイス12は、コンピュータ可読媒体16を介してビデオデータを宛先デバイス14に提供する。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲーム機、ビデオストリーミングデバイスなどを含む、広範囲にわたるデバイスのいずれかを備え得る。場合によっては、ソースデバイス12および宛先デバイス14はワイヤレス通信のために装備され得る。

30

【0034】

[0053] 宛先デバイス14は、コンピュータ可読媒体16を介して復号されるべき符号化ビデオデータを受信することができる。コンピュータ可読媒体16は、符号化ビデオデータをソースデバイス12から宛先デバイス14に移動することが可能な、任意のタイプの媒体またはデバイスを備え得る。一例では、コンピュータ可読媒体16は、ソースデバイス12が符号化ビデオデータを宛先デバイス14にリアルタイムで直接送信するのを可能にするための通信媒体を備え得る。符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス14に送信され得る。通信媒体は、無線周波(RF)スペクトルあるいは1つもしくは複数の物理伝送線路など、任意のワイヤレスまたはワイヤード通信媒体を備え得る。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどのパケットベースのネットワークの一部を形成し得る。通信媒体は、ソースデバイス12から宛先デバイス14への通信を容易にするために有用であり得る、ルータ、スイッチ、基地局

40

50

、または任意の他の機器を含み得る。

【 0 0 3 5 】

[0054]いくつかの例では、符号化データは、出力インターフェース 22 からストレージデバイスへ出力され得る。同様に、符号化データは、ストレージデバイスから入力インターフェースによってアクセスされ得る。ストレージデバイスは、ハードドライブ、Blue-ray ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性もしくは不揮発性のメモリ、または符号化ビデオデータを記憶するための任意の他の適切なデジタル記憶媒体など、様々な分散されたまたはローカルにアクセスされるデータ記憶媒体のいずれをも含み得る。さらなる例では、ストレージデバイスは、ソースデバイス 12 によって生成された符号化ビデオを記憶することができるファイルサーバまたは別の中間ストレージデバイスに対応し得る。宛先デバイス 14 は、ストリーミングまたはダウンロードを介して、ストレージデバイスから記憶されたビデオデータにアクセスすることができる。ファイルサーバは、符号化ビデオデータを記憶でき、符号化ビデオデータを宛先デバイス 14 に送信できる、任意のタイプのサーバとすることができる。例示的なファイルサーバは、（たとえば、ウェブサイト用の）ウェブサーバ、FTPサーバ、ネットワークアタッチドストレージ（NAS）デバイス、またはローカルディスクドライブを含む。宛先デバイス 14 は、インターネット接続を含む任意の標準データ接続を介して、符号化ビデオデータにアクセスすることができる。これは、ファイルサーバ上に記憶された符号化ビデオデータにアクセスするのに適した、ワイヤレスチャネル（たとえば、Wi-Fi（登録商標）接続）、ワイヤード接続（たとえば、DSL、ケーブルモデムなど）、またはその両方の組合せを含み得る。ストレージデバイスからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはその組合せとすることができる。

【 0 0 3 6 】

[0055]本開示の技法は、ワイヤレス応用またはワイヤレス設定に必ずしも限定されない。本技法は、無線テレビジョンブロードキャスト、ケーブルテレビジョン送信、衛星テレビジョン送信、HTTP上の動的適応ストリーミング（DASH：dynamic adaptive streaming over HTTP）などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されたデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の応用例など、様々なマルチメディア応用のいずれかをサポートするビデオコーディングに適用され得る。いくつかの例では、システム 10 は、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、および／もしくはビデオ電話などの応用をサポートするために一方向または両方向のビデオ送信をサポートするように構成され得る。

【 0 0 3 7 】

[0056]図 1 の例では、ソースデバイス 12 は、ビデオソース 18 と、深度推定ユニット 19 と、ビデオ符号化器 20 と、出力インターフェース 22 とを含む。宛先デバイス 14 は、入力インターフェース 28 と、ビデオ復号器 30 と、深度画像ベースのレンダリング（DIBR：depth image based rendering）ユニット 31 と、ディスプレイデバイス 32 とを含む。他の例では、ソースデバイスおよび宛先デバイスは、他の構成要素または構成を含んでもよい。たとえば、ソースデバイス 12 は、外部カメラなどの外部のビデオソース 18 からビデオデータを受信し得る。同様に、宛先デバイス 14 は、統合されたディスプレイデバイスを含まず、外部のディスプレイデバイスとインターフェースしてもよい。

【 0 0 3 8 】

[0057]図 1 の例示されたシステム 10 は、一例にすぎない。本開示の技法は、任意のデジタルビデオ符号化および／または復号デバイスによって実行され得る。概して、本開示の技法はビデオ符号化デバイスによって実行されるが、本技法は、一般に「コーデック」と呼ばれるビデオ符号化器／復号器によっても実行され得る。その上、本開示の技法は、ビデオプリプロセッサによっても実行され得る。ソースデバイス 12 および宛先デバイス 14 は、ソースデバイス 12 が、宛先デバイス 14 に送信するためのコード化ビデオデータを生成するような、コーディングデバイスの例にすぎない。いくつかの例では、デバイ

ス 1 2、1 4 は、デバイス 1 2、1 4 の各々がビデオ符号化構成要素とビデオ復号構成要素とを含むように実質的に対称的に動作し得る。したがって、システム 1 0 は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、もしくはビデオ電話のためのビデオデバイス 1 2 とビデオデバイス 1 4 の間の一方向または双方向のビデオ送信をサポートし得る。

【 0 0 3 9 】

[0058] ソースデバイス 1 2 のビデオソース 1 8 は、ビデオカメラなどのビデオキャプチャデバイス、前にキャプチャされたビデオを包含する (containing) ビデオアーカイブ、および / またはビデオコンテンツプロバイダからビデオを受信するためのビデオフィードインターフェースを含み得る。さらなる代替として、ビデオソース 1 8 は、ソースビデオとしてのコンピュータグラフィックススペースのデータ、またはライブビデオとアーカイブビデオとコンピュータ生成ビデオとの組合せを生成することができる。場合によっては、ビデオソース 1 8 がビデオカメラである場合、ソースデバイス 1 2 および宛先デバイス 1 4 は、いわゆるカメラフォンまたはビデオフォンを形成し得る。しかしながら、上記で述べたように、本開示で説明する技法は、一般にビデオコーディングに適用可能であり、ワイヤレスおよび / またはワイヤード用途に適用され得る。各々の場合において、キャプチャされたビデオ、以前にキャプチャされたビデオ、またはコンピュータ生成ビデオは、ビデオ符号化器 2 0 によって符号化され得る。次いで、符号化ビデオ情報は、出力インターフェース 2 2 によってコンピュータ可読媒体 1 6 に出力され得る。

【 0 0 4 0 】

[0059] ビデオソース 1 8 は、ビデオ符号化器 2 0 にビデオデータの複数のビューを提供し得る。たとえば、ビデオソース 1 8 は、各々が、撮影されている特定のシーンに対して一意の水平位置を有するカメラのアレイに対応し得る。代替的に、ビデオソース 1 8 は、たとえばコンピュータグラフィックスを使用して異なる水平カメラの視点からビデオデータを生成することができる。深度推定ユニット 1 9 は、テクスチャ画像内のピクセルに対応する深度ピクセルのための値を決定するように構成され得る。たとえば、深度推定ユニット 1 9 は、音響航法 / 測距 (SONAR : Sound Navigation and Ranging) ユニット、光検出 / 測距 (LIDAR : Light Detection and Ranging) ユニット、またはシーンのビデオデータを記録しながら実質的に同時に深度値を直接決定することが可能な他のユニットを表し得る。

【 0 0 4 1 】

[0060] 追加または代替として、深度推定ユニット 1 9 は、異なる水平カメラ視点から実質的に同時にキャプチャされた 2 つ以上の画像を比較することによって間接的に深度値を計算するように構成され得る。画像内の実質的に同様のピクセル値の間の水平視差を計算することによって、深度推定ユニット 1 9 は、シーン内の様々なオブジェクトの深度を概算することができる。深度推定ユニット 1 9 は、いくつかの例では、ビデオソース 1 8 と機能的に統合され得る。たとえば、ビデオソース 1 8 がコンピュータグラフィックス画像を生成するとき、深度推定ユニット 1 9 は、たとえば、ピクセルの z 座標と、テクスチャ画像をレンダリングするために使用されたオブジェクトの z 座標とを使用して、グラフィカルオブジェクトのための実際の深度マップを提供することができる。

【 0 0 4 2 】

[0061] コンピュータ可読媒体 1 6 は、ワイヤレスブロードキャストもしくはワイヤードネットワーク送信などの一時媒体、または、ハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray (登録商標) ディスク、もしくは他のコンピュータ可読媒体などの記憶媒体 (すなわち、非一時的記憶媒体) を含み得る。いくつかの例では、ネットワークサーバ (図示せず) は、たとえば、ネットワーク送信を介して、ソースデバイス 1 2 から符号化ビデオデータを受信し、宛先デバイス 1 4 に符号化ビデオデータを提供することができる。同様に、ディスクスタンピング設備など、媒体製造設備のコンピューティングデバイスは、ソースデバイス 1 2 から符号化ビデオデータを受信し、その符号化ビデオデータを包含するディスクを生成することができる。し

たがって、様々な例では、コンピュータ可読媒体 16 は、様々な形態の 1 つまたは複数のコンピュータ可読媒体を含むと理解され得る。

【0043】

[0062]宛先デバイス 14 の入力インターフェース 28 は、コンピュータ可読媒体 16 から情報を受信する。コンピュータ可読媒体 16 の情報は、ビデオ符号化器 20 によって定義され、またビデオ復号器 30 によって使用される、ブロックおよび他のコード化ユニット、たとえば、GOP の特性ならびに / または処理を記述するシンタックス要素を含む、シンタックス情報を含み得る。ディスプレイデバイス 32 は、ユーザに復号ビデオデータを表示し、陰極線管 (CRT)、液晶ディスプレイ (LCD)、プラズマディスプレイ、有機発光ダイオード (OLED) ディスプレイ、または別のタイプのディスプレイデバイスなど、様々なディスプレイデバイスのいずれかを備え得る。いくつかの例では、ディスプレイデバイス 32 は、たとえば、閲覧者のための 3D 視覚影響を生成するために、同時にまたは実質的に同時に 2 つ以上のビューを表示することが可能なデバイスを備え得る。

【0044】

[0063]宛先デバイス 14 の DIBR ユニット 31 は、ビデオ復号器 30 から受信された復号ビューのテクスチャおよび深度情報を使用して、合成されたビューをレンダリングすることができる。たとえば、DIBR ユニット 31 は、対応する深度マップ中のピクセルの値に応じて、テクスチャ画像のピクセルデータののための水平視差を決定し得る。DIBR ユニット 31 は、次いで、決定された水平視差によって、テクスチャ画像中のピクセルを左または右にオフセットすることによって、合成された画像を生成し得る。このようにして、ディスプレイデバイス 32 は、任意の組合せにおける、復号ビューおよび / または合成ビューに対応し得る、1 つもしくは複数のビューを表示することができる。ビデオ復号器 30 は、ビューを適切に合成するために、深度範囲とカメラパラメータとを使用することができる DIBR ユニット 31 に深度範囲およびカメラパラメータのための精度値を提供することができる。

【0045】

[0064]図 1 には示されていないが、いくつかの態様では、ビデオ符号化器 20 およびビデオ復号器 30 は各々、オーディオ符号化器およびオーディオ復号器と統合され得、共通のデータストリームもしくは別個のデータストリーム中のオーディオとビデオの両方の符号化を処理するために、適切な MUX - DEMUX ユニット、または他のハードウェアおよびソフトウェアを含み得る。適用可能な場合、MUX - DEMUX ユニットは、ITU H.223 マルチプレクサプロトコル、またはユーザデータグラムプロトコル (UDP) などの他のプロトコルに準拠し得る。

【0046】

[0065]ビデオ符号化器 20 およびビデオ復号器 30 は、現在の開発中の高効率ビデオコーディング (HEVC) 規格などの、ビデオコーディング規格に従って動作し得、HEVC テストモデル (HM) に準拠し得る。代替的に、ビデオ符号化器 20 およびビデオ復号器 30 は、代替的に MPEG-4、Part 10、アドバンストビデオコーディング (AVC) と呼ばれる ITU-T H.264 規格など、他のプロプライエタリ規格もしくは業界規格、または ITU-T H.264 / AVC の MVC 拡張など、そのような規格の拡張に従って動作し得る。

【0047】

[0066]ビデオ符号化器 20 およびビデオ復号器 30 は各々、1 つもしくは複数のマイクロプロセッサ、デジタル信号プロセッサ (DSP)、特定用途向け集積回路 (ASIC)、フィールドプログラマブルゲートアレイ (FPGA)、ディスクリート論理回路、ソフトウェア、ハードウェア、ファームウェアなど、様々な好適な符号化器回路のいずれか、またはそれらの任意の組合せとして実装され得る。本技法がソフトウェアに部分的に実装されるとき、デバイスは、ソフトウェアに対する命令を適切な非一時的コンピュータ可読媒体に記憶し、本開示の技法を実行するために、1 つまたは複数のプロセッサを使用して命令をハードウェアで実行することができる。ビデオ符号化器 20 およびビデオ復号器 3



0の各々は、1つもしくは複数の符号化器または復号器の中に含まれ得、そのいずれかが、それぞれのデバイスにおいて複合符号化器／復号器（コーデック）の一部として統合され得る。ビデオ符号化器20および／またはビデオ復号器30を含むデバイスは、集積回路、マイクロプロセッサ、および／またはセルラー電話機などのワイヤレス通信デバイスを備えることができる。

#### 【0048】

[0067]最初に、HEVCの例示的なコーディング技法について説明する。上記で参照したHEVC WD10は、たとえば、ITU-T H264/AVCに従う既存のデバイスに対して、ビデオコーディングデバイスのいくつかの追加の能力を推定する。たとえば、H.264は9個のイントラ予測符号化モードを提供するが、HEVC WD10は33個もの角度イントラ予測符号化モードに加えてDCモードと平面モードとを提供することができる。

10

#### 【0049】

[0068]HEVC WD10および他のビデオコーディング規格では、ビデオシーケンスは、一般に、一連のピクチャを含む。ピクチャは「フレーム」と呼ばれる場合もある。ピクチャは、 $S_L$ 、 $S_{Cb}$ 、および $S_{Cr}$ と示される3つのサンプルアレイを含み得る。 $S_L$ は、ルーマサンプルの2次元アレイ（すなわち、ブロック）である。 $S_{Cb}$ は、Cbクロミナンスサンプルの2次元アレイである。 $S_{Cr}$ は、Crクロミナンスサンプルの2次元アレイである。クロミナンスサンプルは、本明細書では「クロマ」サンプルと呼ばれる場合もある。他の例では、ピクチャは、モノクロームであり得、ルーマサンプルのアレイのみを含み得る。

20

#### 【0050】

[0069]HEVC WD10は、ビデオフレームまたはピクチャが、ルーマサンプルとクロマサンプルの両方を含むコーディングツリーユニット（CTU）または最大コーディングユニット（LCU）のシーケンスに分割され得ることを記載している。CTUは、H.264規格のマクロブロックと同様の目的を有する。スライスは、コーディング順序でいくつかの連続するCTUを含む。ビデオフレームまたはピクチャは、1つまたは複数のスライスに区分される場合がある。各CTUは、4分木に従って、コーディングユニット（CU）にスプリットされ得る。たとえば、4分木のルートノードとしてのCTUは、4つの子ノードにスプリットされ得、各子ノードは、次に親ノードとなり得、別の4つの子ノードにスプリットされ得る。4分木のリーフノードとしての、最終的な、スプリットされていない子ノードは、コーディングノード、すなわち、コード化ビデオブロックを備える。コード化ビットストリームに関連付けられるシンタックスデータは、CTUがスプリットされ得る最大回数を定義し得、コーディングノードの最小サイズをも定義し得る。

30

#### 【0051】

[0070]CUは、コーディングノードと、コーディングノードに関連付けられた予測ユニット（PU）および変換ユニット（TU）とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状が正方形でなければならない。CUのサイズは、 $8 \times 8$ ピクセルから、 $64 \times 64$ ピクセル以上の最大値を有するツリーブロックのサイズまでに及び得る。各CUは、1つまたは複数のPUと1つまたは複数のTUとを包含し得る。CUに関連付けられたシンタックスデータは、たとえば、1つまたは複数のPUへのCUの区分を記述することができる。区分モードは、CUがスキップモード符号化もしくは直接モード符号化されるのか、イントラ予測モード符号化されるのか、またはインター予測モード符号化されるのかの間で異なる可能性がある。PUは、形状が非正方形に区分され得る。CUに関連付けられたシンタックスデータは、たとえば、4分木に従う1つまたは複数のTUへのCUの区分を記述することもできる。TUは、形状が方形または非方形であり得る。

40

#### 【0052】

[0071]HEVC WD10は、異なるCUに対して異なり得る、TUに従う変換を可能にする。TUは、一般に、区分されたCTUまたはLCUのために定義された所与のCU

50

内のPUのサイズに基づいてサイズ決定されるが、これは常にそうであるとは限らない。TUは、一般に、PU以下のサイズである。いくつかの例では、CUに対応する残差サンプルは、「残差4分木」(RQT)として知られている4分木構造を使用して、より小さいユニットに副分割され得る。RQTのリーフノードはTUと呼ばれる場合がある。TUに関連付けられたピクセル差分値は、量子化され得る変換係数を生成するために変換され得る。

#### 【0053】

[0072]概して、PUは、予測プロセスに関するデータを含む。たとえば、PUがイントラモード符号化されるとき、PUは、PUのためのイントラ予測モードを記述するデータを含み得る。別の例として、PUがインターモード符号化されるとき、PUは、PUのための動きベクトルを定義するデータを含み得る。PUのための動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度(たとえば、1/4ピクセル精度もしくは1/8ピクセル精度)、動きベクトルが指す参照ピクチャ、および/または動きベクトルのための参照ピクチャリスト(たとえば、リスト0もしくはリスト1)を記述し得る。

#### 【0054】

[0073]一般に、TUは、変換処理および量子化処理のために使用される。1つまたは複数のPUを有する所与のCUは、1つまたは複数のTUを含む場合もある。予測に続いて、ビデオ符号化器20は、PUに対応する残差値を計算することができる。残差値は、エントロピーコーディングのためのシリアル化変換係数(serialized transform coefficient)を生成するために、TUを使用して変換係数に変換され、量子化され、スキャンされ得るピクセル差分値を備える。本開示では、一般に、CUのコーディングノードを指すために「ビデオブロック」という用語を使用する。いくつかの特定の場合において、本開示は、コーディングノードとPUおよびTUとを含む、LCUまたはCUと呼ばれる場合もあるCTUと呼ばれる「ビデオブロック」という用語を使用する場合もある。

#### 【0055】

[0074]ビデオシーケンスは、一般に、一連のビデオフレームまたはビデオピクチャを含む。ピクチャグループ(GOP)は、一般に、ビデオピクチャのうちの一連の1つまたは複数を含む。GOPは、GOPに含まれるいくつかのピクチャを記述するシンタックスデータを、GOPのヘッダ中、ピクチャのうちの一つもしくは複数のヘッダ中、または他の場所に含み得る。ピクチャの各スライス、それぞれのスライスのための符号化モードを記述するスライスシンタックスデータを含む場合がある。ビデオ符号化器20は、一般に、ビデオデータを符号化するために、個々のビデオスライス内のビデオブロックに作用する。ビデオブロックは、CU内のコーディングノードに対応し得る。ビデオブロックは、固定されたサイズまたは変化するサイズを有することができ、規定されたコーディング規格に従ってサイズが異なり得る。

#### 【0056】

[0075]一例として、HEVC WD10は、様々なPUサイズでの予測をサポートする。特定のCUのサイズが $2N \times 2N$ であると仮定すると、HEVC WD10は、 $2N \times 2N$ または $N \times N$ のPUサイズでのイントラ予測、および $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称のPUサイズでのインター予測をサポートする。HEVC WD10は、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ のPUサイズでのインター予測のための非対称区分をもサポートする。非対称区分では、CUの一方の方向は区分されず、他方の方向は、25%と75%とに区分される。CUのうちで25%区分に対応する部分は、「n」とそれに続く「Up(上)」、「Down(下)」、「Left(左)」、または「Right(右)」の表示とによって示される。したがって、たとえば、「 $2N \times nU$ 」は、上部で $2N \times 0.5N$  PU、および下部で $2N \times 1.5N$  PUに水平に区分される $2N \times 2N$  CUを指す。

#### 【0057】

[0076]本開示では、「 $N \times N (N \times N)$ 」および「 $N \times N (N \text{ by } N)$ 」は、垂直

10

20

30

40

50

寸法および水平寸法の観点からビデオブロックのピクセル寸法、たとえば、 $16 \times 16$  ( $16 \times 16$ ) ピクセルまたは  $16 \times 16$  ( $16 \text{ by } 16$ ) ピクセルを指すために互換的に使用され得る。一般に、 $16 \times 16$  ブロックは、垂直方向に  $16$  ピクセルを有し ( $y = 16$ )、水平方向に  $16$  ピクセルを有する ( $x = 16$ )。同様に、 $N \times N$  ブロックは、一般に、垂直方向に  $N$  ピクセル、水平方向に  $N$  ピクセルを有し、 $N$  は、非負の整数値を表す。ブロック中のピクセルは、行および列に配列され得る。さらに、ブロックは、必ずしも、水平方向において垂直方向と同じ数のピクセルを有するとは限らない。たとえば、ブロックは  $N \times M$  ピクセルを備え得、この場合、 $M$  は必ずしも  $N$  に等しいとは限らない。

【0058】

[0077]  $CU$  の  $PU$  を使用したイントラ予測コーディングまたはインター予測コーディングに続いて、ビデオ符号化器 20 は、 $CU$  の  $TU$  のための残差データを計算することができる。 $PU$  は、(ピクセル領域とも呼ばれる) 空間領域においてピクセルデータを備える場合があり、 $TU$  は、変換、たとえば、残差ビデオデータに対する離散コサイン変換 ( $DCT$ )、整数変換、ウェーブレット変換、または概念的に同様の変換の適用を受けて、変換領域において係数を備える場合がある。残差データは、符号化されていないピクチャのピクセルと  $PU$  に対応する予測値との間のピクセル差分に対応し得る。ビデオ符号化器 20 は、 $CU$  のための残差データを含む  $TU$  を形成し、次いで、 $CU$  のための変換係数を生成するために  $TU$  を変換することができる。

【0059】

[0078] 変換係数を生成するためのすべての変換に続いて、ビデオ符号化器 20 は、変換係数の量子化を実行することができる。量子化は、一般に、係数を表すために使用されるデータの量をできるだけ低減するために、変換係数が量子化され、さらなる圧縮を実現するプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。たとえば、 $n$  ビットの値は、量子化の間に  $m$  ビットの値に端数を丸められてよく、ここで、 $n$  は  $m$  よりも大きい。

【0060】

[0079] いくつかの例では、ビデオ符号化器 20 は、エントロピー符号化され得るシリアル化ベクトルを生成するために、量子化変換係数をスキャンするためにあらかじめ定義されたスキャン順序を利用し得る。他の例では、ビデオ符号化器 20 は適応スキャンを実行し得る。1次元のベクトルを形成するために、量子化変換係数をスキャンした後、ビデオ符号化器 20 は、たとえば、コンテキスト適応型可変長コーディング ( $CAVLC$ : context-adaptive variable length coding)、コンテキスト適応型バイナリ算術コーディング ( $CABAC$ : context-adaptive binary arithmetic coding)、シンタックスベースコンテキスト適応型バイナリ算術コーディング ( $SBAAC$ : syntax-based context-adaptive binary arithmetic coding)、確率間隔区分エントロピー ( $PIPE$ : Probability Interval Partitioning Entropy) コーディングまたは別のエントロピー符号化の方法に従って、1次元のベクトルをエントロピー符号化することができる。ビデオ符号化器 20 は、ビデオデータを復号する際のビデオ復号器 30 による使用のために、符号化ビデオデータに関連付けられたシンタックス要素をエントロピー符号化することもできる。

【0061】

[0080]  $CABAC$  を実行するために、ビデオ符号化器 20 は、送信されるべきシンボルにコンテキストモデル内のコンテキストを割り当てることができる。コンテキストは、たとえば、シンボルの隣接値が非ゼロであるか否かに関するものとすることができる。 $CAVLC$  を実行するために、ビデオ符号化器 20 は、送信されるべきシンボルの可変長コードを選択することができる。 $VLC$  における符号語は、相対的により短いコードがよりありそうなシンボルに対応し、より長いコードがよりありそうにないシンボルに対応するように構築され得る。この形で、 $VLC$  の使用は、たとえば、送信されるべきシンボルごとに等しい長さの符号語を使用することに対するビット節約を達成することができる。確率決定は、シンボルに割り当てられたコンテキストに基づき得る。

【0062】

[0081]ビデオ符号化器20は、コード化ピクチャおよび関連付けられたデータの表現を形成するビットのシーケンスを含むビットストリームを出力することができる。ビットストリームは、一連のネットワークアブストラクションレイヤ(NAL: network abstraction layer)ユニットを備え得る。NALユニットの各々は、NALユニットヘッダを含み、ローバイトシーケンスペイロード(RBSP: raw byte sequence payload)をカプセル化する。NALユニットヘッダは、NALユニットタイプコードを示すシンタックス要素を含み得る。NALユニットのNALユニットヘッダによって規定されるNALユニットタイプコードは、NALユニットのタイプを示す。RBSPは、NALユニット内にカプセル化された整数個のバイトを包含するシンタックス構造であり得る。いくつかの事例では、RBSPは0ビットを含む。

10

#### 【0063】

[0082]様々なタイプのNALユニットは、様々なタイプのRBSPをカプセル化することができる。たとえば、第1のタイプのNALユニットはピクチャパラメータセット(PPS)のためのRBSPをカプセル化し得、第2のタイプのNALユニットはコード化スライスのためのRBSPをカプセル化し得、第3のタイプのNALユニットは補助強化情報(SEI: supplemental enhancement information)のためのRBSPをカプセル化し得、以下同様である。PPSは、0以上のコード化ピクチャ全体に適用されるシンタックス要素を包含し得るシンタックス構造である。ビデオコーディングデータのためのRBSPをカプセル化するNALユニットは(パラメータセットおよびSEIメッセージのためのRBSPとは対照的に)、ビデオコーディングレイヤ(VCL)NALユニットと呼ばれる場合がある。コード化スライスをカプセル化するNALユニットは、本明細書ではコード化スライスNALユニットと呼ばれる場合がある。コード化スライスのためのRBSPは、スライスヘッダとスライスデータとを含み得る。

20

#### 【0064】

[0083]ビデオ復号器30は、ビットストリームを受信することができる。加えて、ビデオ復号器30は、ビットストリームからシンタックス要素を復号するために、ビットストリームを解析することができる。ビデオ復号器30は、ビットストリームから復号されたシンタックス要素に少なくとも部分的に基づいて、ビデオデータのピクチャを再構築することができる。ビデオデータを再構築するための処理は、概して、ビデオ符号化器20によって実行される処理の逆であり得る。たとえば、ビデオ復号器30は、現在のCUのPUのための予測ブロックを決定するために、PUの動きベクトルを使用することができる。ビデオ復号器30は、PUのための予測ブロックを生成するために、PUの1つまたは複数の動きベクトルを使用することができる。

30

#### 【0065】

[0084]さらに、ビデオ復号器30は、現在のCUのTUに関連付けられた係数ブロックを逆量子化することができる。ビデオ復号器30は、現在のCUのTUに関連付けられた変換ブロックを再構築するために、係数ブロックに対して逆変換を実行することができる。ビデオ復号器30は、現在のCUのPUのための予測ブロックの予測サンプルを、現在のCUのTUの変換ブロックの対応する残差サンプルに加算することによって、現在のCUのコーディングブロックを再構築することができる。ピクチャの各CUのコーディングブロックを再構築することによって、ビデオ復号器30は、ピクチャを再構築することができる。ビデオ復号器30は、出力のためにおよび/または他のピクチャを復号する際に使用するために、復号ピクチャを復号ピクチャバッファに記憶することができる。

40

#### 【0066】

[0085]MV-HEVCおよび3D-HEVCでは、ビデオ符号化器20は、一連のネットワークアブストラクションレイヤ(NAL)ユニットを備えるマルチレイヤビットストリームを生成することができる。ビットストリームの異なるNALユニットが、ビットストリームの異なるレイヤに関連付けられ得る。レイヤは、同じレイヤ識別子を有するビデオコーディングレイヤ(VCL)NALユニットおよび関連付けられた非VCL NALユニットのセットとして定義され得る。レイヤは、マルチビュービデオコーディングにお

50

けるビューと等価であり得る。マルチビュービデオコーディングでは、レイヤは、異なる時間インスタンスを伴う同じレイヤのすべてのビューコンポーネントを包含し得る。各ビューコンポーネントは、特定の時間インスタンスにおける特定のビューに属するビデオシーンのコード化ピクチャであり得る。3D-HEVCでは、レイヤは、特定のビューのすべてのコード化深度ピクチャ、または特定のビューのコード化テクスチャピクチャのいずれかを包含し得る。同様に、スケーラブルビデオコーディングのコンテキストにおいて、レイヤは、一般に、他のレイヤの中のコード化ピクチャと異なるビデオ特性を有するコード化ピクチャに対応する。そのようなビデオ特性は、一般に、空間解像度と品質レベル（信号対雑音比）とを含む。HEVCおよびその拡張では、時間スケーラビリティ（temporal scalability）は、特定の時間レベル（temporal level）を伴うピクチャのグループをサブレイヤと定義することによって、1つのレイヤ内で達成され得る。

10

【0067】

[0086]ビットストリームの各それぞれのレイヤについて、より低いレイヤの中のデータは、任意のより高いレイヤの中のデータと無関係に復号され得る。スケーラブルビデオコーディングでは、たとえば、ベースレイヤの中のデータは、エンハンスメントレイヤの中のデータと無関係に復号され得る。NALユニットは、単一のレイヤのデータをカプセル化するだけである。ビデオ復号器30が、あるビュー中のピクチャを任意の他のレイヤのデータを参照せずに復号することができる場合、そのビューは「ベースレイヤ」と呼ばれる場合がある。ベースレイヤは、HEVCベース仕様に準拠し得る。したがって、ビットストリームの残りの最高レイヤのデータをカプセル化するNALユニットは、ビットストリームの残りのレイヤの中のデータの復号可能性に影響を及ぼすことなく、ビットストリームから除去され得る。MV-HEVCおよび3D-HEVCでは、より高いレイヤは、さらなるビューコンポーネントを含み得る。

20

【0068】

[0087]次に、ビデオコーディングのいくつかの一般的な特徴をより詳細に論じる。最初に、動き情報について論じる。各ブロックに対して、動き情報のセットが利用可能であり得る。動き情報のセットは、前方予測方向および後方予測方向のための動き情報を含む。ここで、前方予測方向および後方予測方向は、双方向予測モードの2つの予測方向であり、「前方」および「後方」という用語は、必ずしも幾何学的な意味を有するとは限らず、代わりに、これらの用語は現在のピクチャの参照ピクチャリスト0（RefPicList0）および参照ピクチャリスト1（RefPicList1）に対応する。1つの参照ピクチャリストのみがピクチャまたはスライスに利用可能であるとき、RefPicList0のみが利用可能であり得、スライスの各ブロックの動き情報は常に前方である。

30

【0069】

[0088]各予測方向に対して、動き情報は参照インデックスと動きベクトルとを含む。場合によっては、簡単のために、動きベクトル自体は、それが関連付けられた参照インデックスを有すると仮定されるように呼ばれる場合がある。参照インデックスは、現在の参照ピクチャリスト（RefPicList0またはRefPicList1）中の参照ピクチャを識別するために使用される。動きベクトルは水平成分と垂直成分とを有する。

【0070】

40

[0089]ビデオコーディング規格において、ピクチャの表示順序を識別するためにピクチャ順序カウンタ（POC：picture order count）が広く使用されている。1つのコード化ビデオシーケンス内の2つのピクチャが同じPOC値を有する場合が存在するが、一般に、コード化ビデオシーケンス内でそのようなことは生じない。複数のコード化ビデオシーケンスがビットストリーム中に存在するとき、同じPOC値を有するピクチャは、復号順序の点で、互いにより近傍であり得る。ピクチャのPOC値は、一般に、参照ピクチャリスト構築、HEVCにおけるような参照ピクチャセットの導出、および動きベクトルスケールングのために使用される。

【0071】

[0090]次に、H.264/AVCにおけるいくつかの特定の技法について論じる。最初

50

に、A V Cにおけるマクロブロック構造について論じる。H . 2 6 4 / A V Cでは、各インターマクロブロック ( M B ) は、1つの $16 \times 16$  M B区分と、2つの $16 \times 8$  M B区分と、2つの $8 \times 16$  M B区分と、4つの $8 \times 8$  M B区分の、4つの異なる方法で区分され得る。1つのM B中の異なるM B区分は、各方向に異なる参照インデックス値 ( R e f P i c L i s t 0またはR e f P i c L i s t 1 ) を有し得る。M Bが4つの $8 \times 8$  M B区分に区分されないとき、M Bは各M B区分について各方向に1つの動きベクトルしか有さない。M Bが、4つの $8 \times 8$  M B区分に区分されるとき、各 $8 \times 8$  M B区分は、各々が各方向に異なる動きベクトルを有し得るサブブロックにさらに区分され得る。

【 0 0 7 2 】

[0091]  $8 \times 8$  M B区分から、サブブロック、すなわち、1つの $8 \times 8$  サブブロック、2つの $8 \times 4$  サブブロック、2つの $4 \times 8$  サブブロック、または4つの $4 \times 4$  サブブロックを得るための、4つの異なる方法がある。各サブブロックは、各方向に異なる動きベクトルを有し得る。したがって、動きベクトルは、サブブロックレベル以上のレベルに存在する。

【 0 0 7 3 】

[0092] 次に、A V Cにおける時間的直接モードについて説明する。A V Cでは、時間的直接モードは、Bスライス中のスキップモードもしくは直接モードのためのM BレベルまたはM B区分レベルのいずれかで有効にされ得る。各M B区分では、動きベクトルを導出するために、現在のブロックのR e f P i c L i s t 1 [ 0 ] 中の現在のM B区分とコロケートされたブロックの動きベクトルが使用される。コロケートブロック中の各動きベクトルはP O C距離に基づいてスケーリングされる。

【 0 0 7 4 】

[0093] 次に、H E V Cにおけるいくつかの特定の技法について論じる。H E V Cでは、スライス中の最大コーディングユニット ( C U ) はコーディングツリーブロック ( C T B ) と呼ばれる。C T Bは4分木を含み、4分木のノードはC Uである。技術的に、 $8 \times 8$  C T Bサイズがサポートされ得るが、C T Bのサイズは、H E V Cメインプロファイルにおいて $16 \times 16$  から $64 \times 64$  に及ぶ。C Uのサイズは、C T Bと同じ大きいサイズから $8 \times 8$  の小さいサイズに及ぶ。各C Uは1つのモードでコーディングされる。C Uがインターコーディングされるとき、C Uは、2つもしくは4つの予測ユニット ( P U ) にさらに区分され得るか、または、さらなる区分が適用されないとき、1つのP Uだけになる。C Uがインターコーディングされるとき、C Uの各P Uのために、1つのセットの動き情報が存在する。加えて、動き情報の関連付けられたセットを導出するために、C Uの各P Uは一意的なインター予測モードでコーディングされる。

【 0 0 7 5 】

[0094] 図2は、C UのP Uを定義するための8つの区分モード3 3 A ~ 3 3 H (「区分モード3 3」) を示す概念図である。P Uは、C Uを区分することによって定義される予測が適用される領域である。図2に示すように、ピクチャ中の実際のオブジェクトの境界に一致するように、C Uの区分を円滑にするために、P Uは形の点で方形になると限定されない。区分モードに応じて、各C Uは、1つ、2つ、または4つのP Uを含む。いくつかの例では、2つのP Uが1つのC U中に存在するとき、P A R T \_ 2 N x N 3 3 B区分モードおよびP A R T \_ N x 2 N 3 3 C区分モードにあるように、P Uは半分のサイズの矩形であり得る。他の例では、2つのP Uが1つのC U中に存在するとき、P Uは、P A R T \_ 2 N x n U 3 3 E区分モード、P A R T \_ 2 N x n D 3 3 F区分モード、P A R T \_ n L x 2 N 3 3 G区分モード、およびP A R T \_ n R x 2 N 3 3 H区分モードにおけるように、C Uのサイズの $1/4$ または $3/4$ のサイズを有する2つの方形であり得る。P A R T \_ 2 N x 2 N 3 3 A区分モードおよびP A R T \_ N x N 3 3 D区分モードがイントラコード化C Uのために使用される。P A R T \_ N x N 3 3 D区分モードは、対応するC Uサイズが最小C Uサイズに等しいときだけ可能にされる。

【 0 0 7 6 】

[0095] 図3は、C U内の変換ツリー構造3 4を示す概念図である。各C Uは、4分木で

ある、1つの変換ツリーに対応し、そのリーフは変換ユニット(TU)である。変換ツリー構造34中のTUの各々は、CUの4分木区分によって定義される、変換プロセスおよび量子化プロセスが適用される方形領域である。図3に示すように、TUは形の点で常に方形である。TUは、32×32サンプルの大きさから4×4サンプルに至るまでであり得る。最大4分木深度、すなわち、変換ツリー構造34がリーフノードを形成するためにどの程度までスプリットされ得るかは、調整可能であり、スライスヘッダシンタックス中で規定される。インターCUの場合、TUは、PUよりも大きい場合があり、すなわち、TUはPU境界を包含し得る。しかしながら、イントラCUの場合、TUはPU境界を超えられない。

【0077】

10

[0096] 1に等しいシンタックス要素 `rq t__r o o t__c b f` は、現在のCUのための `t r a n s f o r m__t r e e` シンタックス構造が存在することを規定する。0に等しいシンタックス要素 `rq t__r o o t__c b f` は、現在のCUのための `t r a n s f o r m__t r e e` シンタックス構造が存在しないことを規定する。シンタックス要素 `rq t__r o o t__c b f` が存在しないとき、その値は1に等しいと推論される。シンタックス要素 `rq t__r o o t__c b f` が0に等しいとき、変換ツリーは1つのノードのみを包含し、つまり、その変換ツリーはさらにスプリットされず、`s p l i t__t r a n s f o r m__f l a g` は0に等しい。変換ツリー内部のノードの場合、そのノードが1に等しい `s p l i t__t r a n s f o r m__f l a g` を有する場合、そのノードは4つのノードにさらにスプリットされ、変換ツリーのリーフは0に等しい `s p l i t__t r a n s f o r m__f l a g` を有する。

20

【0078】

[0097] 簡単のために、TUすなわち変換ツリーが変換を有さないブロックに対応する場合、それは、依然として、TUすなわち変換ユニットと見なされるが、これは、変換が適用されない場合ですら、変換自体の階層は依然として存在するためである。一般に、変換がスキップされるブロックは変換ユニットに対応する。

【0079】

[0098] 1に等しいTUのコード化ブロックフラグ(`c b f`)は、TUが1つまたは複数の非ゼロ変換係数レベルを包含すること、すなわち、0に等しくないことを規定する。0に等しいTUのCBFは、TUのすべての変換係数レベルが0に等しいことを規定する。CBFは、TUの各成分に対して設定され、すなわち、CBFは、ルミナンス(Y)成分ならびにクロミナンス(CbおよびCr)成分の各々に対して設定される。

30

【0080】

[0099] HEVC規格では、現在の予測ユニット(PU)のための動きを予測するための、マージモードおよび高度動きベクトル予測(AMVP)モードと名付けられた、2つのインター予測モードが存在する。スキップモードは、マージモードの特殊事例と見なされる。AMVPモードまたはマージモードのいずれかで、複数の動きベクトル予測子を含む動きベクトル(MV)候補リストが維持される。MV候補リストから候補のうちの1つを選択することによって、動きベクトル、ならびに、マージモードにおいて関連付けられた参照インデックスが現在のPUのために生成される。

40

【0081】

[0100] マージモードでは、MV候補リストは5つまでの候補を含む。マージ候補は、動き情報、たとえば、参照ピクチャリスト(リスト0およびリスト1)ならびに関連付けられた参照インデックスの両方に対応する動きベクトルのセットを包含し得る。マージ候補が現在のPUのためのマージインデックスによって識別される場合、現在のPUの予測のために動きベクトルおよび関連付けられた参照ピクチャが使用される。

【0082】

[0101] AMVPモードでは、MV候補リストは2つのみの候補を含む。AMVP候補は、両方の参照ピクチャリスト(リスト0およびリスト1)に対応する動きベクトルを包含する。AMVP候補が現在のPUのためのAMVPインデックスによって識別される場合

50

、A M V P 候補は動きベクトルのみを含むため、参照インデックスは、A M V P インデックスとともに、リスト 0 またはリスト 1 のいずれかからの各潜在的な予測方向のために明示的にシグナリングされる必要がある。A M V P モードでは、予測動きベクトルは、動きベクトル差分をシグナリングし、動きベクトルスケールを実行することによって、さらに改良され得る。

#### 【 0 0 8 3 】

[0102] 上記で説明したように、マージ候補は、動き情報の完全セットに対して含むが、A M V P 候補は、特定の予測方向のための 1 つの動きベクトルおよび明示的にシグナリングされた参照インデックスのみを含む。両方のモードのための候補は、現在の P U の同じ空間的隣接ブロックおよび時間的隣接ブロック (temporal neighboring blocks) から同様に導出される。

10

#### 【 0 0 8 4 】

[0103] 図 4 A および図 4 B は、それぞれ、マージモードおよび A M V P モードのための空間的隣接候補を示す概念図である。空間的 M V 候補は、現在の P U ( P U 0 ) のための、図 4 A および図 4 B に示す隣接ブロックから導出されるが、ブロックから候補を生成するための方法はマージモードおよび A M V P モードに対して異なる。

#### 【 0 0 8 5 】

[0104] マージモードでは、図 4 A に示す、番号付けされた順序で、P U 0 9 0 の隣接ブロックから 4 つまでの空間的 M V 候補が導出され得る。図 4 A に示す例では、この順序は、次の通り、すなわち、左ブロック ( 0 )、上ブロック ( 1 )、右上ブロック ( 2 )、左下ブロック ( 3 )、および左上ブロック ( 4 ) である。

20

#### 【 0 0 8 6 】

[0105] A M V P モードでは、図 4 B に示す、番号付けされた順序で、P U 0 9 2 の隣接ブロックから 2 つまでの空間的 M V 候補が導出され得る。図 4 B に示すように、P U 0 9 2 の隣接ブロックは 2 つのグループ、すなわち、左下ブロック ( 0 ) と左ブロック ( 1 ) とを含む左グループ、および右上ブロック ( 2 ) と、上ブロック ( 3 ) と、左上ブロック ( 4 ) とを含む上グループに分割される。各グループに対して、P U 0 9 2 のための明示的なシグナリングされた参照インデックスと同じ参照ピクチャを指す隣接ブロックのための動きベクトルは、そのグループの最終的な M V 候補を形成するように選ばれている最高優先順位を有する。いくつかの例では、隣接ブロックのうちのいずれも、明示的にシグナリングされた参照ピクチャを指す動きベクトルを有さない可能性がある。そのような M V 候補が発見され得ない場合、そのグループの最終的な M V 候補を形成するために、第 1 の利用可能な動きベクトルがスケールされることになる。この場合、いずれの時間的な距離差分も補償されることになる。

30

#### 【 0 0 8 7 】

[0106] 図 5 A は、現在の P U ( P U 0 ) 9 4 のための時間的動きベクトル予測 ( T M V P ) 候補を示す概念図である。T M V P 候補は、有効にされ、利用可能である場合、空間的 M V 候補の後に M V 候補リストに追加される。T M V P 候補のための動きベクトル導出のプロセスは、マージモードと A M V P モードの両方に対して同じである。しかしながら、マージモードでは、T M V P 候補のためのターゲット参照インデックスは常に 0 に設定される。

40

#### 【 0 0 8 8 】

[0107] T M V P 候補導出のためのプライマリブロックロケーションは、図 5 A でブロック T 9 7 として示される、P U 0 9 4 のコロケート P U ( P U 1 ) 9 6 の外部の右下ブロックである。右下ブロック T 9 7 は、空間的隣接候補を生成するために使用される上ブロックおよび左ブロックに対するバイアスを補償するように選択される。しかしながら、右下ブロック T が P U 0 9 4 の現在の L C U 9 8 の外部 (すなわち、位置 9 9 ) に位置するか、または右下ブロック T 9 7 のための動き情報が利用可能でない場合、T M V P 候補導出のために使用されるブロックは、P U 0 9 4 の中央ブロック T 9 5 に置換される。

50



## 【 0 0 8 9 】

[0108]図 5 B は、現在のピクチャ 1 0 2 中の現在の P 1 0 0 のための T M V P モードに対する M V スケーリングを示す概念図である。T M V P 候補のための動きベクトルは、スライスレベルで示された、コロケートピクチャ 1 0 6 のコロケート P U 1 0 4 から導出される。上記で説明した、A V C 中の時間的 direct モードと同様に、T M V P 候補の動きベクトルは、図 5 B に示すように、コロケートテンポラル距離（すなわち、コロケートピクチャ 1 0 6 とコロケート参照ピクチャ 1 0 7 との間の距離）と（すなわち、現在のピクチャ 1 0 2 と現在の参照ピクチャ 1 0 3 との間の）現在のテンポラル差分との間の差分を補償するために実行される、スケーリングを使用して、現在の P U 1 0 0 のために生成され得る。

10

## 【 0 0 9 0 】

[0109]次に、H E V C における動きベクトル予測の他の態様について論じる。動きベクトルスケージングでは、動きベクトルの値は提示時間におけるピクチャの距離に比例すると仮定される。動きベクトルは、2 つのピクチャ、すなわち、参照ピクチャと（すなわち、包含ピクチャ（containing picture））動きベクトルを包含するピクチャとを関連付ける。別の動きベクトルを予測するためにある動きベクトルが利用されるとき、包含ピクチャと参照ピクチャの距離は、それらのピクチャのピクチャ順序カウンタ（P O C）値に基づいて計算される。予測されるべき動きベクトルに対して、その包含ピクチャと関連付けられた参照ピクチャの両方は異なり得る。この場合、（P O C に基づいて）新しい距離が計算される。動きベクトルはこれらの 2 つの P O C 距離に基づいてスケージングされる。空間的隣接候補の場合、2 つの動きベクトルに対する包含ピクチャは同じであるが、参照ピクチャは異なる。H E V C では、動きベクトルスケージングは、空間的隣接候補および時間的隣接候補のための T M V P と A M V P の両方に適用される。

20

## 【 0 0 9 1 】

[0110]人工的動きベクトル候補生成では、動きベクトル候補リストが完全でない（すなわち、指定された数の候補未満が利用可能である）場合、そのリストが指定された数の候補を含むまで、人工的動きベクトル候補が生成されて、リストの最後に挿入され得る。マージモードでは、2 つのタイプの人工的 M V 候補が存在する。すなわち、B スライスに対してのみ導出される結合候補、および、第 1 のタイプが十分な人工的候補を提供しない場合、A M V P に対してのみ使用されるゼロ候補である。候補リスト中にすでにあり、必要な動き情報を有する候補の対の各々に対して、リスト 0 中のピクチャを指す第 1 の候補の動きベクトルと、リスト 1 中のピクチャを指す第 2 の候補の動きベクトルの結合に基づいて、双方向結合動きベクトル候補が導出される。

30

## 【 0 0 9 2 】

[0111]候補挿入のための刈り込みプロセスで、異なるブロックからの候補は偶然同じである場合があり、これはマージ / A M V P 候補リストの効率を低減する。刈り込みプロセスはこの問題に対処するために適用され得る。刈り込みプロセスは、ある状況において同一候補を挿入するのを避けるために、現在の候補リスト中のある候補を他の候補と比較する。複雑さを低減するために、各潜在的な候補をすべての他の既存の候補と比較する代わりに、限定された数の刈り込みプロセスのみが適用され得る。

40

## 【 0 0 9 3 】

[0112]次に、H E V C におけるデブロッキングフィルタプロセスについて説明する。デブロッキングフィルタプロセスは、復号プロセスと同じ順序でビデオブロックの各 C U のために実行される。第 1 に、垂直エッジがフィルタリングされ（すなわち、水平フィルタリング）、次いで、水平エッジがフィルタリングされる（すなわち、垂直フィルタリング）。デブロッキングフィルタリングは、両方ともルーマ成分およびクロマ成分に関する、フィルタリングされるべきであると決定された C U 内のブロック境界に適用される。いくつかの例では、複雑さを低減するために、8 × 8 ブロック境界はフィルタリングされるが、4 × 4 ブロック境界は処理されない。2 つの種類の境界、すなわち、T U 境界および P U 境界がデブロッキングフィルタプロセスに関与する。C U 境界は必然的に T U 境界およ

50

びPU境界でもあるため、CU境界も考慮される。

【0094】

[0113]図6は、デブロッキングフィルタプロセスの例示的な処理フロー108を示すフローチャートである。境界は3つのフィルタリング状態値、すなわち、フィルタリングなし、弱いフィルタリング、および強いフィルタリングを有し得る。各フィルタリング決定は、境界強度Bと、しきい値  $t_c$  および  $t_c'$  に基づく。

【0095】

[0114]図7は、デブロッキングフィルタプロセスのための境界強度(B)計算110の例示的な動作を示すフローチャートである。境界強度(B)は、フィルタリングプロセスが境界のためにどの程度強く必要とされ得るかを反映する。たとえば、2のB値は強いフィルタリングを示し、1のB値は弱いフィルタリングを示し、0のB値はデブロッキングフィルタリングがないことを示す。

【0096】

[0115]PおよびQをフィルタリングに関与するブロックと定義し、この場合、Pは、デブロッキングされるべき境界の左(垂直エッジの場合)または上(水平エッジの場合)に対して位置するブロックを表し、Qは、デブロッキングされるべき境界の右(垂直エッジの場合)または上(水平エッジの場合)に対して位置するブロックを表す。図7に示した動作は、イントラコーディングモード、非ゼロ変換係数の存在、参照ピクチャ、動きベクトルの数、および動きベクトル差分に基づいて、B値がどのように計算されるかを示す。

【0097】

[0116]図8は、デブロッキングフィルタプロセスのためのコーディングツリーユニット(CTU)境界113におけるグリッド112中のピクセル間のB計算のために再使用または参照される情報を示す概念図である。Bは4×4ブロックベースで計算されるが、Bは8×8グリッドに再度マッピングされる。4×4グリッド中のラインからなる8つのピクセルに対応するBの2つの値の最大値が、8×8グリッド中の境界のためのBとして選択される。CTU境界113において、ラインバッファメモリ要件を削減するために、図8に示すように、左側にまたは上側に(4×4グリッド上の)すべての第2のブロックの情報が再使用される。

【0098】

[0117]フィルタオン/オフ決定、強いフィルタおよび弱いフィルタの選択、ならびに弱いフィルタリングプロセスにしきい値  $t_c$  および  $t_c'$  が関与する。これらのしきい値は、下の表1に示すように、ルーマ量子化パラメータQの値から導出される。

【0099】

【表1】

表1:入力Qからのしきい値変数の導出

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
B'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
t <sub>c</sub> '	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
B'	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
t <sub>c</sub> '	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
Q	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53			
B'	38	40	42	44	46	48	50	52	54	56	58	60	62	64	-	-			
t <sub>c</sub> '	5	5	6	6	7	8	9	10	11	13	14	16	18	20	22	24			

【0100】

変数  $t_c$  は次のように  $t_c'$  から導出される。

$$t_c = t_c' * (1 \ll (\text{BitDepth}_Y - 8))$$

変数  $t_c$  は次のように  $t_c'$  から導出される。

$$t_c = t_c' * (1 \ll (\text{BitDepth}_Y - 8))$$

## 【 0 1 0 1 】

[0118]図 9 は、境界 1 1 4 におけるデブロッキングフィルタプロセスのためのオン / オフフィルタ決定および強い / 弱いフィルタ選択に関与する P ブロック 1 1 6 ならびに Q ブロック 1 1 7 中のピクセルを示す概念図である。フィルタオン / オフ決定は、計算上の複雑さを低減するために、( 図 9 で、黒の輪郭線のボックスを使用して示した ) ユニットとして各々グループ化された、境界 1 1 4 を超える 4 本のラインのピクセルを使用して行われる。フィルタが 4 本のラインの第 1 のセット 1 1 8 のためにオンであるか、またはオフであるかを決定するために、4 本のラインの第 1 のセット 1 1 8 中の 2 つのボックスの各々に含まれた 6 つのピクセルが使用される。フィルタが 4 本のラインの第 2 のセット 1 1 9 のためにオンであるか、またはオフであるかを決定するために、4 本のラインの第 2 のセット 1 1 9 中の 2 つのボックスの各々に含まれた 6 つのピクセルが使用される。

10

## 【 0 1 0 2 】

[0119]オン / オフフィルタ決定のために、以下の変数が定義される。

$$dp0 = | p_{2,0} - 2 * p_{1,0} + p_{0,0} |$$

$$dp3 = | p_{2,3} - 2 * p_{1,3} + p_{0,3} |$$

$$dq0 = | q_{2,0} - 2 * q_{1,0} + q_{0,0} |$$

$$dq3 = | q_{2,3} - 2 * q_{1,3} + q_{0,3} |$$

$dp0 + dq0 + dp3 + dq3 <$  である場合、4 本のラインの第 1 のセット 1 1 8 のためのフィルタリングがオンにされ、強い / 弱いフィルタ選択プロセスが適用される。この条件が満たされない場合、4 本のラインの第 1 のセット 1 1 8 のために何のフィルタリングも行われない。

20

## 【 0 1 0 3 】

[0120]加えて、この条件が満たされる場合、変数  $dE$ 、 $dEp1$ 、および  $dEp2$  は次のように、すなわち、

$dE$  が 1 に等しいように設定される。

$dp0 + dp3 < ( \quad + ( \quad >> 1 ) ) >> 3$  である場合、変数  $dEP1$  は、1 に等しいように設定される

$dq0 + dq3 < ( \quad + ( \quad >> 1 ) ) >> 3$  である場合、変数  $dEq1$  は、1 に等しいように設定される

フィルタオン / オフ決定は、4 本のラインの第 2 のセット 1 1 9 のために上記で説明したのと同様の方法で行われる。

30

## 【 0 1 0 4 】

[0121]フィルタリングがオンにされる場合、強いフィルタリングと弱いフィルタリングとの間で決定が行われる。関与するピクセルは、図 9 に示したように、フィルタオン / オフ決定のために使用されたピクセルと同じである。以下の 2 つの条件セットが満たされた場合、4 本のラインの第 1 のセット 1 1 8 のフィルタリングのために強いフィルタが使用される。さもなければ、弱いフィルタが使用される。

(1)  $2 * (dp0 + dq0) < ( \quad >> 2 )$ ,  $| p_{3,0} - p_{0,0} | + | q_{0,0} - q_{3,0} | < ( \quad >> 3 )$  および  $| p_{0,0} - q_{0,0} | < ( 5 * t_c + 1 ) >> 1$

(2)  $2 * (dp3 + dq3) < ( \quad >> 2 )$ ,  $| p_{3,3} - p_{0,3} | + | q_{0,3} - q_{3,3} | < ( \quad >> 3 )$  および  $| p_{0,3} - q_{0,3} | < ( 5 * t_c + 1 ) >> 1$

40

4 本のラインの第 2 のセット 1 1 9 のために強いフィルタリングを選択するかまたは弱いフィルタリングを選択するかに関する決定は類似の方法で行われる。

## 【 0 1 0 5 】

[0122]強いフィルタリングの場合、フィルタリングされたピクセル値は以下の方程式によって取得される。3 つのピクセルは、P ブロック 1 1 6 および Q ブロック 1 1 7 の各々のための入力として 4 つのピクセルを使用して修正されることに留意されたい。

$$P_0' = ( p_2 + 2 * p_1 + 2 * p_0 + 2 * q_0 + q_1 + 4 ) >> 3$$

$$q_0' = ( p_1 + 2 * p_0 + 2 * q_0 + 2 * q_1 + q_2 + 4 ) >> 3$$

$$p_1' = ( p_2 + p_1 + p_0 + q_0 + 2 ) >> 2$$

50

$$q_1' = (p_0 + q_0 + q_1 + q_2 + 2) \gg 2$$

$$p_2' = (2 * p_3 + 3 * p_2 + p_1 + p_0 + q_0 + 4) \gg 3$$

$$q_2' = (p_0 + q_0 + q_1 + 3 * q_2 + 2 * q_3 + 4) \gg 3$$

## 【 0 1 0 6 】

[0123]弱いフィルタリングの場合、フィルタリングされたピクセル値は以下の方程式によって取得される。2つのピクセルは、Pブロック116およびQブロック117の各々のための入力として3つのピクセルを使用して修正されることに留意されたい。

デルタ ( ) は次のように定義される。

$$= (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) \gg 4$$

a b s ( ) が  $t_c * 10$  未満であるとき、

$$= \text{Clip3}(-t_c, t_c, )$$

$$p_0' = \text{Clip1}_Y(p_0 + )$$

$$q_0' = \text{Clip1}_Y(q_0 - )$$

d E p 1 が 1 に等しい場合、

$$p = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, ((p_2 + p_0 + 1) \gg 1) - p_1 + ) \gg 1$$

$$p_1' = \text{Clip1}_Y(p_1 + p)$$

d E q 1 が 1 に等しい場合、

$$q = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, ((q_2 + q_0 + 1) \gg 1) - q_1 - ) \gg 1$$

$$q_1' = \text{Clip1}_Y(q_1 + q)$$

## 【 0 1 0 7 】

[0124]上記で説明したデブロッキング決定はルーマ成分をフィルタリングするためである。クロマフィルタリングのための境界強度Bはルーマから継承される。B > 1である場合、クロマフィルタリングが実行される。1つのフィルタのみが適用され得るため、クロマのために何のフィルタ選択プロセスも実行されない。フィルタリングされたサンプル値  $p_0'$  および  $q_0'$  は、次のように導出される。

$$= \text{Clip3}(-t_c, t_c, ((q_0 - p_0) \ll 2) + p_1 - q_1 + 4) \gg 3$$

$$p_0' = \text{Clip1}_C(p_0 + )$$

$$q_0' = \text{Clip1}_C(q_0 - )$$

## 【 0 1 0 8 】

[0125]次に、3D-HEVCにおけるいくつかの特定の技法について論じる。HEVC規格の3Dビデオ拡張(3D-HEVC)はJCT-3Vによって開発中である。3D-HEVC拡張は、マルチビュービデオに加えて深度フォーマットのコーディングをサポートする。

## 【 0 1 0 9 】

[0126]図10は、V3がベースビューを示し、非ベースビュー(すなわち、V1またはV5)中のピクチャが同じ時間インスタンスのベースビュー中のピクチャから予測される、3つのビューの事例に対するマルチビュー予測構造120を示す概念図である。図10に示すマルチビュー予測構造120を使用するビュー間サンプル予測は、HEVCのマルチビュー拡張(MV-HEVC)によってサポートされる。MV-HEVCと3D-HEVCは両方とも、ベースビューまたはテクスチャビュー(すなわち、V3)がHEVC復号器によって復号可能であるような形でHEVCに対して互換性がある。

## 【 0 1 1 0 】

[0127]MV-HEVCでは、非ベースビュー(すなわち、V1またはV5)中の現在のピクチャは、これらのピクチャのすべてを現在のピクチャのための参照ピクチャリスト中に含めることによって、同じタイムインスタンスの同じビュー中のピクチャと参照ビュー中のピクチャの両方によって予測され得る。現在のピクチャの参照ピクチャリストは、したがって、時間的参照ピクチャとビュー間参照ピクチャの両方を包含する。時間的参照ピクチャに対応する参照インデックスに関連付けられた動きベクトルは、時間的動きベクトル

10

20

30

40

50

ルとして示される。ビュー間参照ピクチャに対応する参照インデックスに関連付けられた動きベクトルは、視差動きベクトルとして示される。

【 0 1 1 1 】

[0128] 3D - HEVC 拡張は、ビュー間サンプル予測が 3D - HEVC によってやはりサポートされるように、MV - HEVC のすべての特徴をサポートする。加えて、3D - HEVC 拡張は、MV - HEVC 拡張によってサポートされない、より高度なテクスチャオンリーコーディングツール (texture-only coding tools) および深度関連 / 依存コーディングツール (depth related/dependent coding tools) をサポートする。テクスチャオンリーコーディングツールは、同じオブジェクトに属するビュー間の対応するブロックの識別を必要とし得る。視差ベクトル導出は、したがって、3D - HEVC における基本的な技術である。

10

【 0 1 1 2 】

[0129] マルチビューコーディング技術の 1 つの主な態様は、視差ベクトルの正確で効率的な導出によって、異なるビューの対応するブロックを識別することである。3D - HEVC では、隣接ブロックベースの視差ベクトル (NB DV) 導出は、HEVC における AMVP モードおよびマージモードに類似した方法で設計される。しかしながら、視差ベクトルは隣接ブロックだけから導出され、その結果、視差ベクトルのさらなる改良またはスケールアップのために追加のビットは必要とされない。ビュー間サンプル予測が有効にされるとき、ビュー間参照ピクチャに対応する動きベクトル、すなわち、視差動きベクトルは、通常の動きベクトル、すなわち、時間的動きベクトルとともに、現在のブロックのための動きフィールド中にすでに存在する。NB DV 導出の基本的な考えは、空間的隣接ブロックおよび時間的隣接ブロックのみを確認することによって、動きフィールド中の視差動きベクトルを使用することである。

20

【 0 1 1 3 】

[0130] 図 11 は、現在のブロック 122 のための NB DV 導出のためにアクセスされる空間的隣接ブロックベースがマージモードおよび AMVP モードのためにアクセスされる空間的隣接ブロックと同じである、NB DV 導出のための空間的隣接ブロックを示す概念図である。NB DV のためにアクセスされる空間的隣接ブロック A0、A1、B0、B1、および B2 は、図 4A および図 4B、ならびに図 11 に示した、AMVP モードおよびマージモードのためにアクセスされる空間的隣接ブロックと同じである。

30

【 0 1 1 4 】

[0131] しかしながら、いずれの関連付けられた視差動きベクトルを有する現在のブロック 122 の空間的隣接ブロックが存在しない可能性が高い。この場合、時間的隣接ブロックも確認される。現在のブロック 122 の時間的隣接ブロックは、TMVP モードで利用されるような、コロケートピクチャ、加えて、視差動き補償を使用するより良好な機会を有し得る別のピクチャ、たとえば、ランダムアクセスピクチャ、または最低 temporal Id を有するピクチャの中のブロックである。図 5A に示した TMVP モードと同様の各時間的ピクチャに対して、中央ブロックおよび右下ブロックが確認される。現在のブロック 122 のすべての空間的隣接ブロックおよび時間的隣接ブロックが順に確認される。視差動きベクトルが識別されると、識別された視差動きベクトルと同じになるように現在のブロック 122 の視差ベクトルが導出され、NB DV 導出プロセスは終了する。現在のブロック 122 の視差ベクトルは、たとえば、ビュー間動き予測およびビュー間残差予測におけるように、参照ビューのピクチャ中のビュー間参照ブロックを識別するために使用される。

40

【 0 1 1 5 】

[0132] 3D - HEVC では、隣接ブロックのためにさらに使用されるように、NB DV 導出の結果として、いくつかの導出された視差ベクトルを記憶することが可能であり得る。そのような導出された視差ベクトルは、暗黙的視差ベクトル (IDV : implicit disparity vectors) と名付けられる。たとえば、NB DV 導出が隣接ブロックからのいずれの視差動きベクトルも識別しない場合、現在のブロック 122 のための最終的な視差ベクトル

50

ルを導出するために使用される空間的隣接ブロックのいずれかの利用可能なIDVを識別するために、空間的隣接ブロックが再度確認され得る。最近、3D-HEVCは、より少ない空間的隣接ブロックおよび時間的隣接ブロックを確認することによって、NBDV方法の簡略化も含めた。

#### 【0116】

[0133] 3D-HEVCにおけるビュー合成予測(VSP)は、2つの主要な技術的モジュール、すなわち、深度指向性NBDV(Do-NBDV: Depth-Oriented NBDV)および後方ビュー合成予測(BVSP)によって実現される。

#### 【0117】

[0134] Do-NBDV導出では、NBDV方式によって生成された視差ベクトルは、コード化深度マップ中の情報を使用してさらに改良され得る。このようにして、視差ベクトルの精度は、ベースビュー深度マップ中でコーディングされた情報を利用することによって向上され得る。改良ステップは次のように説明される。

1. 同じアクセスユニットの前にコーディングされた(ベースビューなど)参照深度ビューピクチャ中の現在のPUのためにNBDV方式によって導出された視差ベクトルに基づいて、対応する深度ブロックを位置特定し、対応する深度ブロックのサイズは現在のPUのサイズと同じである。

2. 4つのコーナーピクセル深度値の最大値に基づいて、対応する深度ブロックから現在のPUのための視差ベクトルを計算する。最大値は、計算された視差ベクトルの水平成分に等しいように設定されるが、視差ベクトルの垂直成分は0に設定される。

#### 【0118】

[0135] NBDV方式からの視差ベクトルは、次いで、Do-NBDV方式から新しく導出された視差ベクトルによって置換され、現在のPUのためのビュー間動き予測のために、(Do-NBDVからの)改良された視差ベクトルが使用される。しかしながら、(NBDVからの)改良されていない視差ベクトルは、現在のPUのためのビュー間残差予測のために使用され得る。加えて、改良された視差ベクトルは、BVSPモードでコーディングされる場合、現在のPUの1つの動きベクトルとして記憶され得る。

#### 【0119】

[0136] BVSPモードは、その開始動きベクトルを視差動きベクトルに設定させた特殊マージモード候補として実現される。そのような候補はBVSP候補と呼ばれる。現在のPUの空間的隣接ブロックがBVSPモードでコーディングされるとき、現在のPUのための追加のBVSP候補を定義するために、隣接ブロックからの関連付けられた視差ベクトルおよび参照ビューインデックスが使用される。そのような空間的隣接ブロックから導出された空間的マージモード候補は、1に等しいBVSPフラグを用いてタグ付けされる。

#### 【0120】

[0137] 図12は、現在のピクチャ125中の現在のPU126の異なるサブPUのために異なる動きベクトルを割り当てるBVSPモード124を示す概念図である。本開示では、3D-HEVCにおける現在のBVSPモードは、サブPU BVSPモードと呼ばれる場合がある。BVSPモードでは、現在のPU126の空間的隣接ブロックまたは時間的隣接ブロックから現在のPU126のための第1の動き情報が決定される。上記で説明したように、第1の動き情報は、マージモード候補リスト中のBVSP候補として選択され得る。第1の動き情報は、視差動きベクトルと、ビュー間参照ピクチャ128を識別する、関連付けられた参照ビューインデックスとを含む。BVSPモードの第1のステップは、上記で説明したNBDV導出と実質的に同様であり得る。

#### 【0121】

[0138] 現在のPU126は、サブ領域またはサブPUにさらに区分される。たとえば、そのサイズが $N \times M$ によって示されるPUは、 $K \times L$ (式中、 $K$ および $L$ は8もしくは4であり得るが、両方とも4であり得ない)に等しいサイズを有する、いくつかのサブ領域またはサブPUにさらに区分され得る。各サブ領域またはサブPUに対して、第1の視差

ベクトルに関連付けられた参照ビューインデックスによって識別されたビュー間参照ピクチャ 128 に関連付けられた参照深度ビューピクチャ 129 中の深度ブロック 130 のうちの対応する 1 つの深度ブロックにアクセスすることによって導出された別個の視差動きベクトルを含む第 2 の動き情報が決定される。より具体的には、現在の PU 126 の各サブ PU に対して、深度ブロック 130 のうちの対応する 1 つの深度ブロックの 4 つのコーナーピクセルの最大値が選択され、サブ PU のための視差動きベクトルの水平成分に変換され、視差動きベクトルはゼロに等しい垂直成分を有する。BVS P モードの第 2 のステップは、上記で説明した Do - NBDV 導出と実質的に同様であり得る。

#### 【0122】

[0139] 現在の PU 126 の各サブ領域またはサブ PU がその動き情報を予測させた後、各サブ PU のための第 2 の動き情報に基づいて、ビュー間参照ピクチャ 128 から現在の PU 126 の各サブ領域またはサブ PU を予測するために、HEVC の動き補償エンジンが使用され得る。BVS P モードでは、現在の PU 126 の各サブ PU を予測するために動き補償を実行した後、現在の PU 126 のために選択された第 1 の動き情報内に含まれた第 1 の視差動きベクトルだけが現在の PU 126 のために記憶され、サブ PU のための第 2 の動き情報内に含まれた別個の視差ベクトルは廃棄される。

#### 【0123】

[0140] BVS P モードで、各 PU は、2 つ以上のセットの動き情報を有し得、すなわち、PU 内部のサブ PU は異なる動きベクトルを有し得る。これは、各 PU が 1 つのセットの動き情報のみを有する HEVC とは異なる。一般に、BVS P モードでは、PU 内部のサブ PU は異なる動きベクトルを有し得るが、サブ PU のための参照インデックス値は同じである。BVS P モードの複雑さの増大は、主に電力消費の増大にあり、最悪の複雑さは、サブ PU のサイズ、ならびに、双予測状態が、HEVC において有効にされる動き補償ブロックのサイズよりも決して小さくない限り、HEVC 動き補償と同様である。

#### 【0124】

[0141] 3D - HEVC では、関連付けられたテクスチャビューの後、深度ビューがコーディングされる。深度ビュー中の PU がコーディングされる時、PU のコロケート領域内のテクスチャビューの動き情報は、現在の PU のためのマージ候補を作成し得る。この場合、マージ候補は、フルセットの動き情報を包含する動きパラメータ継承 (MPI) 候補と呼ばれる場合がある。

#### 【0125】

[0142] JCT3V - E0184、Jicheng An ら、3D - CE3 . h 関連 : Sub - PU level inter - view motion prediction」、ITU - T SG16 WP3 と ISO / IEC JTC1 / SC29 / WG11 との 3D ビデオコーディング拡張開発に関する共同研究部会、第 5 回会議、ウィーン、オーストリア、2013 年 7 月 27 日 ~ 8 月 2 日では、ビュー間マージ候補、すなわち、ビュー間参照ピクチャ中の参照ブロックから導出された候補に対するサブ PU レベルのビュー間動き予測方法の使用が説明されている。

#### 【0126】

[0143] 図 13 は、現在のビュー (V1) 132 中の現在の PU 134 のためのサブ PU レベルのビュー間動き予測 131 を示す概念図である。ビュー間動き予測モードが有効にされるとき、現在の PU 134 は、参照ビュー (V0) 136 中の視差ベクトルによって識別された現在の PU と同じサイズを有する参照領域に対応し得る。場合によっては、参照エリアは、現在の PU 134 のための 1 つのセットの動き情報のみの生成のために必要とされるよりもより豊富な動き情報を有し得る。したがって、現在の PU 134 が複数のサブ PU に区分され、視差ベクトル (DV) が使用される、現在の PU 134 のサブ PU の各々のための参照ビュー (V0) 136 中の参照ブロック 138 を識別する、サブ PU レベルのビュー間動き予測 (SPIVMP : sub-PU level inter-view motion prediction) 方法が提案される。SPIVMP モードでは、現在の PU 134 のサブ PU の各々は、現在の PU 134 が複数のセットの動き情報を包含し得るように、参照ブロック 138

10

20

30

40

50

から複写されたフルセットの動き情報を有する。S P I V M Pモードはまた、上記で説明したサブP U B V S Pモードと同様の特殊マージモード候補としてシグナリングされ得る。

【0127】

[0144]Ying ChenおよびLi Zhangに対する、「ADVANCED MOTION PREDICTION FOR 3D VIDEO CODING」と題する、2013年7月24日に出願した米国仮出願第61/858,089号では、MPI候補がサブP Uレベルのビュー間動き予測と同様の方法でも拡張され得ることが提案されている。たとえば、現在の深度P Uが、複数のP Uを包含するコロケート領域を有する場合、現在の深度P Uは、各々が異なるセットの動き情報を有し得るサブP Uに分離され得る。この方法はサブP U MPIと呼ばれる場合がある。

10

【0128】

[0145]上記で説明した、サブP U B V S Pと、サブP Uビュー間動き予測と、サブP U MPIとを含む、3D-HEVCにおけるサブP U設計は、いくつかの問題を経験する可能性がある。

【0129】

[0146]第1に、上記で説明したように、サブP U B V S Pモードでは、現在のP Uのために選択された視差動きベクトルによって識別されたビュー間参照ピクチャの対応する深度ブロックに基づいて導出された別個のサブP U動き情報に基づいて現在のP UのサブP Uの各々を予測するために、動き補償が実行される。しかしながら、サブP Uの各々を予測するために動き補償を実行した後、現在のP Uに対して、各参照ピクチャリストに対応する視差動きベクトルのみが記憶される。この場合、後続P Uを予測するために現在のP Uが使用されるとき、現在のP UのサブP Uは、サブP U B V S Pモードが動きベクトル予測の精度の改善に与える影響がほとんどないように、同じ動きベクトルを有すると見なされる。

20

【0130】

[0147]第2に、現在のサブP U設計は、レイヤ間予測が有効にされるときのみ有効にされる。しかしながら、サブP U設計は、シングルレイヤ予測における動き補償使用の精度を改善するために適用可能であり得る。加えて、現在のサブP U設計は、候補動きベクトルの何のさらなる改良も可能にされないマージインター予測モードにのみ適用可能である。したがって、現在のサブP U設計は、動き改良が可能にされる動き予測モード、たとえば、HEVCベースのコーデックのためのAMVPモードのために有効にされ得ない。

30

【0131】

[0148]第3に、HEVCにおいて使用されるデブロッキングフィルタは、一般に、3D-HEVCを含めて、HEVC拡張において変更されない状態に保たれる。しかしながら、HEVCでは、1つのP U内部の同じT U中のブロックは1つの単一ピクチャのブロック全体として動き補償されると仮定されるため、HEVCにおける現在のデブロッキングフィルタ設計は、サブP Uの境界をフィルタリングすることができない。この場合、何のブロッキングアーティファクトも、P U内に存在する、またはP Uから除去されると予想されない。3D-HEVC拡張におけるデブロッキングフィルタ設計を変更せずに、サブP U境界をデブロッキングするために、本開示は、デブロッキングフィルタプロセスが適用される前に、サブP Uをデブロッキングフレンドリーな構造に変換することについて説明する。たとえば、デブロッキングフィルタリングプロセスの前に、サブP U設計を利用する各P Uは、1つまたは複数のコーディングツリーに変換されることが可能であり、この場合、各サブP Uは、それぞれのコーディングツリーのノードであるC U中のP Uになり得る。この例では、コーディングツリー構造はHEVCシンタックス設計に対して互換性がある。

40

【0132】

[0149]しかしながら、上記の解決策でいくつかの追加の問題が生じる場合がある。第1の例として、予測ユニットA(P U A)が通常のインター予測を用いてコーディングさ

50



れ、PU Aと同じCU内の別の予測ユニット(PU B)がサブPUを用いてコーディングされる場合、これらのPUは、2つのCUに変換されるニーズになり、フィルタリングは、サブ区分されていないPU A内部で生じ得る。第2の例として、1つのTUがサブPUを有する少なくとも1つのPUを含むCU全体のために適用されるとき、そのCUは複数のCUに変換され得、同じTUを有する複数のCUはHEVCによってサポートされない。

【0133】

[0150]本開示は、上記で説明した問題に対処するためのいくつかの技法について説明する。

【0134】

[0151]上記で説明した第1の問題に関して、3D-HEVCにおけるサブPU BVSPモードで、現在のPUのために選択された視差動きベクトルによって識別されたビュー間参照ピクチャの対応する深度ブロックに基づいて導出された別個のサブPU動き情報に基づいて、現在のPUのサブPUの各々を予測するために、動き補償を実行した後、各参照ピクチャリストに対応する視差動きベクトルのみが現在のPUのために記憶される。PUの動き補償が現在のPUの複数のサブPUのために導出された別個の動き情報に含まれた複数の動きベクトルに基づく場合ですら、単一の視差動きベクトルが各参照ピクチャリストのために記憶される。

【0135】

[0152]本開示の技法によれば、サブPU BVSPモードを使用して予測される各PUに対して、ビデオコーディングデバイス、すなわち、ビデオ符号化器20および/またはビデオ復号器30は、動き補償が実行された後ですら、現在のPUのサブPUの各々のために導出された別個の動き情報を保存、たとえば、記憶または維持する。一例として、ビデオ符号化器20および/またはビデオ復号器30は、サブPUの各々のために導出された視差動きベクトルを、現在のPUのための視差動きベクトルに関連付けられた参照ビューインデックスによって識別されたビュー間参照ピクチャを含む参照ピクチャリストとともに、復号ピクチャバッファなどのメモリに記憶することができる。

【0136】

[0153]現在のPUのために記憶された追加のより豊富な動き情報は、次いで、現在のPUが隣接ブロックである後続PUを予測するために使用され得る。たとえば、ビデオ符号化器20および/またはビデオ復号器30は、マージモード候補リスト中のサブPU BVSP候補として、現在のPUのサブPUのうちの少なくとも1つのために記憶された動き情報を含む後続PUを予測するための動き情報のマージモード候補リストを生成することができる。一例では、後続PUがサブPU BVSPモードでコーディングされる場合、ビデオ符号化器20および/またはビデオ復号器30は、後続PUのサブPUの各々のために別個の動き情報を導出する必要がない。代わりに、ビデオ符号化器20および/またはビデオ復号器30は、代わりに、後続PUのサブPUを予測するための動き情報のマージモード候補リストからサブPU BVSP候補として別個の動き情報を選択することができる。サブPU BVSPモードでPUを予測して、PUのサブPUの各々のために導出された動き情報を記憶する動作は、図17に関して下でより詳細に説明される。

【0137】

[0154]上記で説明した第2の問題に関して、本開示は、動きベクトル改良が可能にされ得るシングルレイヤコーディングにおけるPUのサブPUを予測するための高度TMVPモードを実行するための技法について説明する。ビデオ符号化器20および/またはビデオ復号器30は、PUのサブPUの各々のための異なる動きベクトルおよび参照インデックスを含む、PUのための動き情報を導出するために、少なくとも2つのステージでPUのための動きベクトルを決定することを含む、高度TMVPモードを実行するように構成され得る。

【0138】

[0155]図14Aおよび図14Bは、シングルレイヤコーディングにおけるPU中のサブ

10

20

30

40

50

P Uを予測するための高度T M V Pモードを示す概念図である。図14Aに示すように、高度T M V Pモードの第1のステージ170は、現在のピクチャ中の現在のP U 173のための参照ピクチャ中の対応するブロック174を識別する動きベクトルを決定し、高度T M V Pモードの第2のステージ172は、参照ピクチャ中の対応するブロック174のサブP Uから複数のセットの動き情報を抽出し、動き情報のそれらのセットの各々を現在のピクチャ中の現在のP U 173のサブP Uのうちの1つに割り当てる。現在のP U 173の各サブP Uは、したがって、別個に動き補償される。第1のステージ170の動きベクトルは、現在のP U 173の空間的隣接ブロックおよび時間的隣接ブロックから導出され得る。一例では、第1のステージの動きベクトルは、すべての他のマージモード候補のうち、H E V CにおけるT M V P候補と同様の候補を含み得るか、または含み得ないマージモード候補として選択され得る。別の例では、第1のステージの動きベクトルは、すべての他のA M V P候補のうち、A M V Pモード候補として選択され、改良され得る。この例では、サブP Uが各予測方向のために利用可能でない動きベクトルに遭遇するとき、代表的な動きベクトルが使用され得る。

#### 【0139】

[0156]シングルレイヤコーディングおよびサブP U T M V Pに適用可能な、ビデオ符号化器20またはビデオ復号器30は、M V候補リストのための動き予測インデックスを用いてシグナリングされるP UまたはC Uのための動き改良データ、たとえば、動きベクトル差分を決定することができる。一例では、各予測方向に対して、単一の動きベクトル差分が決定され得、サブP UまたはP Uのすべての動きベクトルに適用可能である。別の例では、各予測方向に対して、別個の動きベクトル差分がサブP UまたはP Uの各々のために決定され得る。加えて、各水平成分または垂直成分に対して、動きベクトル差分値は変換され得、結果として生じる、変換係数は、ビデオコーデックにおけるピクセル残差と同様に、さらに量子化または切り捨てられ、コーディングされ得る。

#### 【0140】

[0157]別の例では、H E V Cと同様に、サブP U動きベクトルのための動き改良データは、サブP U M V候補がA M V Pモード候補リストに追加され、マージモード候補リストに追加されないとき、ビデオ符号化器20からビデオ復号器30に送信され得る。1つの代替的な例では、サブP U M V候補は、サブP UまたはP Uのすべての動きベクトルに関連付けられた参照インデックス値が同じであるときのみ適用され得る。別の代替的な例では、サブP U M V候補は常に適用され得、サブP Uのすべての動きベクトルに関連付けられた参照インデックス値は明示的に送信される。加えて、動きベクトル差分値の量子化または変換が適用される場合、動きベクトルは1つの固定参照ピクチャに対してスケールリングされ得る。動きベクトル差分が収集された後、これらの差分はスケールリングされた動きベクトルに追加される。その後、サブP UまたはP Uの異なる参照インデックス値によって識別されたそれらのそれぞれの異なる参照ピクチャに対して新しい動きベクトルがスケールリングし戻される。

#### 【0141】

[0158]以下のセクションは、サブP UまたはP Uの動きベクトルに関連付けられた参照インデックス値が異なる場合の、上記で説明した高度T M V Pモードのための例示的な実装詳細を提供する。第1のステージの動きベクトルの識別についてまず論じる。第1のステージの動きベクトルは、時間的動きベクトルを包含する、現在のP Uの空間的隣接ブロックから変換される。空間的隣接ブロックは、典型的な動きベクトル予測のために使用されるブロック、たとえば、A M V PまたはH E V Cのためのマージで使用するブロックに属する。

#### 【0142】

[0159]空間的隣接ブロックが、異なる参照インデックス値に関連付けられた候補動きベクトルを有するとき、そこからサブP Uレベルの動き情報を決定するための参照ピクチャを識別するために、どの参照インデックスが使用されるかを決定するために、以下の決定プロセスのうちの1つが適用される。

1. 最小参照インデックス値を有するブロックが選ばれる。「第1のステージのベクトル」になるべき時間的動きベクトルを戻すために、その中で、より早期にアクセスされた参照インデックスが選ばれる。これらのブロックは、ブロックの相対的な空間的ロケーションに基づいて、所与の順序でアクセスされると仮定される。

2. 最小参照インデックス値を有するブロックが選ばれる。(2つ以上の場合)これらのブロックの動きベクトルは、「第1のステージのベクトル」になるように平均化され、代替では、中間動作が使用され得る。

3. 隣接ブロックの参照インデックスの中で、最高頻度の参照インデックスが選ばれる。その参照インデックスを有するブロックの中で、「第1のステージのベクトル」になるとして、第1にアクセスされた動きベクトルが選ばれるか、または動きベクトルの平均(たとえば、中間)が使用される。代替的に、第1のステージの動きベクトルを取得するために、他の数学機能が適用され得る。

10

4. 現在のピクチャに最も近接するPOC距離を有する参照インデックスが選ばれる。その参照インデックスを有するブロックの中で、「第1のステージのベクトル」になるとして、第1にアクセスされた動きベクトルが選ばれるか、または動きベクトルの平均(たとえば、中間または他の数学機能)が使用される。

5. 上記で説明した方法と同様の方法を用いて、プライマリ参照インデックスがスライスヘッダ中でシグナリングされ、「第1のステージのベクトル」を作成するために、そのプライマリ参照インデックスに等しい参照インデックスを有するブロックが選ばれる。何のブロックもプライマリ参照インデックスに等しい参照インデックスを有さないとき、参照インデックスを選ぶための上記で説明した方法が使用され得る。

20

6. TMVPのために使用されるピクチャは、一般に、AVCおよびHEVCにおけるように、各スライスのために固定されるため、一次参照インデックスは、TMVPを示す参照インデックスと同じであり得る。

7. 上記の参照インデックスは、RefPicList0またはRefPicList1のいずれかに対するインデックスであり得る。代替的には、参照ピクチャリストユニオン(RefPicListU)は、RefPicList0およびRefPicList1によって生成され、参照インデックスは、RefPicListUに対するインデックスであり得る。RefPicListUによって識別されるいずれのピクチャもRefPicList0もしくはRefPicList1のいずれか、または両方に属し、RefPicListUに属するが、RefPicList0もしくはRefPicList1に属さないピクチャは存在しないことに留意されたい。RefPicListUは、2つの同一ピクチャを有さない。代替的に、および追加として、RefPicListUは、短期ピクチャと同じレイヤ内の時間的参照ピクチャまたは短期ピクチャと標示された時間的参照ピクチャのみを包含し得る。

30

8. 代替的に、参照インデックスおよび「第1のステージのベクトル」は、マージモード候補から選択され得る。

a. 一例では、1つの相対的なブロックの位置、たとえば、左の隣接ブロックから導出された1つの空間的マージモード候補から参照インデックスと「第1のステージのベクトル」とが選択される。

40

b. 代替的に、マージモード候補リスト中の第1の利用可能な候補から参照インデックスと「第1のステージのベクトル」とが選択され得る。

c. さらに、選択されたマージ候補が双予測を使用するとき、動きベクトルおよび参照インデックスは、マージされたセットの動き情報うちの1つから選択され得る。

【0143】

[0160]参照インデックスおよび「第1のステージのベクトル」が識別されるとき、サブPU動き情報を決定するために使用される参照ピクチャ、ならびに、現在のPUに対応する参照ピクチャ中の領域が識別される。1つの事例では、たとえば、参照インデックスは導出されて、TMVPのために使用されるピクチャは明示的にシグナリングされているため、参照インデックスは、TMVPのために使用されることになるピクチャと異なる参照

50

ピクチャを示し得る。この場合、参照インデックスは、TMVPのために使用されるピクチャを識別するために変更され得、動きベクトルは、POC距離に基づいて、TMVPのために使用されるピクチャに対してスケーリングされ得る。

#### 【0144】

[0161]上記の説明で、「第1のステージのベクトル」の識別は、空間的隣接ブロックのみを使用する。代替的に、「第1のステージのベクトル」を識別するために時間的近接ブロックが使用され得、この場合、時間的隣接ブロックの位置はNB DVにおいて使用されるブロックと同様である。そのような位置は、各々が識別された参照ピクチャ内に位置している、現在のPUの中央位置または現在のPUの右下位置を含む。

#### 【0145】

[0162]次に、TMVPのためのサブPU動きの生成について論じる。現在のPUの各サブPUは、「第1のステージのベクトル」によって識別された参照ピクチャの対応するブロック内の対応するサブPUを位置特定することができる。上記で説明したサブPUレベルのビュー間動き予測と同様に、参照ピクチャの対応するサブPU内の動き情報から導出された動き情報のセットが、現在のPU中のサブPUの各々のために生成される。対応するサブPUの動き情報が利用可能でない場合、代替として、「第1のステージのベクトル」が現在のPUのサブPUのために使用され得る。各サブPUは各予測方向のために異なる参照インデックスを有し得るため、各参照ピクチャリストのためのターゲット参照ピクチャに対して動きベクトルをスケーリングする、いくつかの技法が提案される。

#### 【0146】

[0163]図14Bに示すように、動きソース参照ピクチャ182の対応するサブPU185内部の動きベクトル186によって識別された参照ピクチャ183はPicOriであり、対応するサブPU185を包含する参照ピクチャ（すなわち、動きソースピクチャ）182はPicTであり、現在のピクチャ180はPicCurであり、ターゲット参照ピクチャ184はPicTargである。議論のために、動きベクトル186はMVであり、スケーリングされた動きベクトルはMV'（すなわち、現在のピクチャ180中のPU181のサブPUを予測するための動きベクトル予測子188であると仮定される。高度TMVPモードの第1のステージで識別された時間的動きベクトル187がそこからサブPU動き情報が導出されたピクチャでない参照ピクチャに関連付けられるとき、POC距離に基づく動きベクトルのスケーリングがやはり可能であり得ることに留意されたい。

1. 動きベクトルを現在のピクチャの固定参照インデックスに対してスケーリングする。 $MV' = MV * (POC(PicTarg) - POC(PicCur)) / (POC(PicOri) - POC(PicT))$ 、 $POC()$ 関数は所与のピクチャのPOC値を戻す。上記の乗算演算および導出演算はHEVC TMVPにおけるのと同様の方法で簡略化され得ることに留意されたい。

a. 固定参照インデックスは、スライス全体に対して同じであり得、たとえば、固定参照インデックスは、HEVCにおけるように、0に等しく設定され得る。

b. 固定参照インデックスは、空間的隣接ブロックから導出され得る。

2. 動きベクトルを、常に、PicOriである、対応するサブPUの参照ピクチャに対してスケーリングする。 $MV' = MV * (POC(PicOri) - POC(PicCur)) / (POC(PicOri) - POC(PicT))$ 。

3. 動きベクトルを、常に、PicTである、コロケートピクチャに対してスケーリングする。 $MV' = MV * (POC(PicT) - POC(PicCur)) / (POC(PicOri) - POC(PicT))$ 。

#### 【0147】

[0164]上記に示したように、各参照ピクチャリストからのターゲット参照ピクチャは、参照ピクチャリスト中で0に等しい参照インデックスを有する参照ピクチャに設定される。別の例では、各参照ピクチャリストからのターゲット参照ピクチャは、上記で説明したように、「第1のステージのベクトル」とともに識別された同じ参照ピクチャに設定され

る。サブPUレベルの動きベクトル予測のための高度TMVPの動作は、図19に関して下でより詳細に説明される。

【0148】

[0165]上記で説明した第3の問題に関して、本開示は、複数のサブPUを有する、少なくとも1つのPUを含むビデオブロックのCUに適用されるデブロッキングフィルタプロセスに関する技法について説明する。本開示の技法は、HEVCデブロッキングフィルタが3D-HEVCブロックのために引き続き使用され得るように、サブPUをデブロッキングフレンドリーな構造に変換することによって、サブPU境界がデブロッキングされるのを可能にする。このようにして、デブロッキングフィルタを変更されない状態に保ちながら、サブPU境界に沿ってブロックアーティファクトのフィルタリングを可能にするように、デブロッキングフィルタに対する入力を変更することが可能である。本開示では、少なくとも部分的にサブPUを用いてコーディングされたCUは、高度CUと呼ばれ、コーディングされた高度CU内のサブPUを用いてコーディングされたPUは、高度PUと呼ばれる。高度CU中の他のPUは、もしあれば、通常のPUと呼ばれる。

10

【0149】

[0166]本開示において、サブPU境界に沿ってアーティファクトを除去するために、デブロッキングフィルタに対するサブPU用のエッジを導入するための、いくつかの技法について説明する。本開示で説明するサブPU変換プロセスは、デブロッキングフィルタリングプロセスの直前に、すなわち、参照ピクチャとして記憶するためにビデオブロックを再構築した後であるが、再構築されたビデオブロックをビデオ符号化器20および/またはビデオ復号器30の復号ピクチャバッファに実際に記憶する前に、3D-HEVCに導入され得る。

20

【0150】

[0167]第1の例では、デブロッキングフィルタを、サブPUを有するPUを含むCUに適用する直前に、ビデオ符号化器20および/またはビデオ復号器30は、サブPUの各々がコーディングツリーのノードであるコーディングユニット中のPUになることができるように、そのサブPU設計を利用する各PUを1つまたは複数のコーディングツリーに変換するように構成され得る。この例では、サブPU境界は、デブロッキングフィルタのために、人工的PU境界に変換される。コーディングツリー構造は、好ましくは、HEVCシンタックス設計に対して互換性がある。

30

【0151】

[0168]他の例では、サブPUを有するPUを含むビデオブロックのCUにデブロッキングフィルタを適用するのに先立って、ビデオ符号化器20および/またはビデオ復号器30は、代わりに、サブPU境界において人工的PU境界または人工的TU境界を作成するために、CUを変換するように構成され得る。一例では、ビデオ符号化器20および/またはビデオ復号器30は、PUを変換ツリー階層に関連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーを変換するように構成され得る。この例では、サブPU境界は、デブロッキングフィルタのために、人工的TU境界に変換される。別の例では、ビデオ符号化器20および/またはビデオ復号器30は、PUをCUに関連付け、サブPUの各々をPUに関連付けるために、CUをコーディングツリーに変換するように構成され得る。この例では、サブPU境界は、デブロッキングフィルタのために、人工的PU境界に変換される。コーディングツリー構造は、好ましくは、HEVCシンタックス設計に対して互換性がある。

40

【0152】

[0169]上記の例のうちのいずれかにおいて、サブPUをデブロッキングフレンドリーな構造に変換した後、人工的PU境界と人工的TU境界とを含めて、CUの2つの隣接するPU間のPU境界および/またはCUの2つの隣接するTU間のTU境界にデブロッキングフィルタが適用され得る。

【0153】

[0170]サブPU境界が人工的TU境界に変換されるように、PUを変換ツリー階層に

50

連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーが変換される、上記で説明した例では、以下の決定プロセスのうちの1つまたは複数が適用され得る。

1. 変換ツリーが、1に等しい`split_transform_flag`を有するルートを有するとき、高度CUの通常のPU（もしあれば）に対応するノードは変更されない。

2. 変換ツリーが、1に等しい`split_transform_flag`を有するルートを有するとき、変換ツリー階層を次のように導入するために、高度PU中のノードは変換される。

a. 現在のノードに対して、`split_transform_flag`が0であり、TUサイズがサブPUサイズよりも大きい場合、`split_transform_flag`を1に設定し、4つの子ノードの各々のために、以下が適用される。

i. ノードの`cbf_luma`、`cbf_cb`、および`cbf_cr`が親ノードの`cbf_luma`、`cbf_cb`、および`cbf_cr`と同じになり、`split_transform_flag`が0に等しくなるように設定する。

ii. 現在のノードを子ノードに設定し、ステップaに進む。

b. 現在のノードに対して、`split_transform_flag`が1である場合、4つの子ノードの各々に対して、以下が適用される。現在のノードを子ノードに設定し、ステップaに進む。

3. 代替的に、変換ツリー階層がPUのために導入され、`cbf_luma`、`cbf_cb`、および`cbf_cr`が新しくスプリットされたノードに設定されるとき、ノードの`cbf_luma`、`cbf_cb`、および`cbf_cr`は0に設定される。

4. 代替的に、変換ツリー階層がPUのために導入され、`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）が新しくスプリットされたノードに設定されるとき、`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）が0に等しくない場合、ノードの`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）は任意の非ゼロの正の整数値（たとえば1）に設定され、さもなければ、0に設定される。

5. 変換ツリーが0に等しい`split_transform_flag`を有するルートを有する場合、通常のPU（もしあれば）は、次のステップを用いて順に、変換ユニットにスプリットされる。

a. CUの`split_transform_flag`がまず1に設定される。

b. CUの区分が $N \times N$ である場合、通常のPUは1つのノードに対応する。CUの区分が $2N \times N$ または $N \times 2N$ である場合、通常のPUは2つのノードに対応する。

c. 通常のPU内の上記のノードの各々は、0に設定された`split_transform_flag`と、高度CU全体をカバーする元の変換ユニットの`cbf_luma`、`cbf_cb`、および`cbf_cr`に等しくなるように設定された`cbf_luma`、`cbf_cb`、および`cbf_cr`とを有するように設定される。

d. CUの高度PUの各々に対して、PUの方形領域をカバーする1つまたは2つのノードがまず生成され、各ノードに対して、上記のステップ2、3、または4と同様の変換ツリー階層が導入される。

6. 上記のステップ5の代替として、通常のPU内の上記のノードの各々に対して、`cbf_luma`、`cbf_cb`、および`cbf_cr`は0に等しく設定される。

7. 上記のステップ4の代替として、通常のPU内の上記のノードの各々に対して、`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）が0に等しくない場合、その`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）は任意の非ゼロの正の整数値（たとえば、1）に等しく設定され、さもなければ、0に設定される。

【0154】

[0171]サブPU境界が人工的PU境界に変換されるように、PUをCUに関連付け、サブPUの各々をPUに関連付けるために、CUがコーディングツリーに変換される、上記で説明した例では、以下の決定プロセスのうちの1つまたは複数が適用され得る。

1. 変換後、現在の高度 C U は、コーディングツリーのルート、すなわち、少なくとも 4 つのコーディングユニットを包含する 4 分木である変換ルートになる。

2. 高度 C U 中のいずれの通常の P U に対して、その P U は 1 つまたは複数のコーディングユニットに設定される。

3. 加えて、高度 C U 中の通常の P U から変換された変換コーディングユニットの各成分の C B F 値は 0 になるようにさらに設定される。このようにして、P U が 2 つの C U に変換されている場合ですら、1 つの通常のインター予測 P U 内のフィルタリングは回避される。

a. 任意の高度 P U に対して、現在の高度 C U の C B F 値が非ゼロである場合、変換ルートよりも 1 レベル下にある変換コーディングツリーまたは変換コーディングユニットの C B F 値は非ゼロになるように設定される。

10

b. さらに、高度 P U 内部のいずれの変換 C U もしくは変換 P U または変換コーディングツリーに対して、それが、本来、C B F 値をシグナリングさせていない場合、それは、4 分木中のその親ノードの C B F 値と同じになるように、または、4 分木中の親ノードの C B F 値と同じゼロ状態もしくは非ゼロ状態に設定される。

4. 代替的に、高度 C U に対して、1 つの一意の変換ユニットが C U 全体のために適用され得ず、つまり、通常の P U が高度 C U 内部にある場合、その P U は、C U の別の P U によって共有されない変換ユニットを包含しなければならない。

a. 代替または追加として、各高度 P U に対して、変換ツリーのリーフノードは、サブ P U よりも 1 レベル高くまたは深く分散され、つまり、変換 C U 内の 4 つのサブ P U の各々は一意の変換ユニットをシグナリングさせる。

20

5. より具体的には、 $2L \times 2L$  サイズの C U が、サブ P U 設計を利用する P U を包含する場合、C U 中のすべての P U のために以下の変換が実行される。1 に等しいスプリットフラグが現在の高度 C U のために設定され、以下が適用され、現在のノードは C U になるように設定される。

a. ラスタスキャン順序で現在のノードの  $1/4$  方形エリアの各々に対して、以下が適用される。

i. この  $1/4$  エリアを子ノードとして設定する。

ii. 子ノードがサブ P U を用いてコーディングされない場合、子ノードは、 $2N \times 2N$  区分を有する (0 に等しいスプリットフラグを有する) コーディングユニットになるように設定される。

30

1. 代替または追加として、子ノードの各成分の C B F 値は 0 になるように設定される。

2. 代替または追加として、コーディングユニットの T U スプリットフラグは 0 になるように設定される。

iii. さもなければ、子ノードがサブ P U でコーディングされ、2 つ以上のサブ P U を包含する場合、スプリットフラグは、子ノードのために 1 に設定され (したがって、コーディングツリーと見なされ)、以下が適用される。

1. 代替または追加として、子ノードのための C B F 値が存在しない場合、子ノードは、より高いレベルの現在のノード (この子ノードの親ノード) の C B F 値に等しくなるように設定される。

40

2. 子ノードを現在のノードとして設定し、ステップ a に進む。

iv. さもなければ、子ノードが 1 つのサブ P U だけを包含する場合、子ノードはコーディングツリーのリーフ、したがって、(0 に等しいスプリットフラグを有する) コーディングユニットになるように設定される。

1. 子ノードの区分モードはサブ P U の区分モードになるように設定される。

a. サブ P U の区分モードが  $2N \times 2N$  である場合、サブ P U は 1 つの  $2N \times 2N$  ブロックを包含し、コーディングユニットの区分モードは  $2N \times 2N$  になるように設定される。

50

b. サブPUの区分モードが $N \times 2N$ である場合、サブPUは2つの $N \times 2N$ ブロックを包含し、コーディングユニットの区分モードは $N \times 2N$ になるように設定される。

c. サブPUの区分モードが $2N \times N$ である場合、サブPUは2つの $2N \times N$ ブロックを包含し、コーディングユニットの区分モードは $2N \times N$ になるように設定される。

2. 代替または追加として、子ノードのためのCBF値が存在しない場合、子ノードは、より高いレベルの現在のノード（この子ノードの親ノード）のCBF値に等しくなるように設定される。

3. 代替または追加として、コーディングユニットのTUSプリットフラグは0になるように設定される。

10

【0155】

[0172]サブPU境界が人工的TU境界に変換されるように、PUを変換ツリー階層に関連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーが変換される、上記で説明した例に関する別の代替として、以下の決定プロセスのうちの1つまたは複数が各サブPUの少なくとも一部のために適用され得る。

1. 変換ツリーが1に等しいsplit\_transform\_flagを有するルートを有するとき、高度CUの通常のPU（または、通常のPUの一部）のみを包含する通常のノードは変更されない。

2. 変換ツリーが1に等しいsplit\_transform\_flagを有するルートを有するとき、変換ツリー階層を次のように導入するために、任意のサブPU（または、サブPUの一部）を包含する高度ノードは変換される。

20

a. 現在のノードに対して、現在のノードが任意のサブPU（または、サブPUの一部）を包含し、split\_transform\_flagが0であり、TUサイズがサブPUサイズよりも大きい場合、split\_transform\_flagを1に設定し、4つの子ノードの各々のために、以下が適用される。

i. ノードのcbf\_luma、cbf\_cb、およびcbf\_crが親ノードのcbf\_luma、cbf\_cb、およびcbf\_crと同じになり、split\_transform\_flagが0に等しくなるように設定する。

ii. 現在のノードを子ノードに設定し、ステップaに進む。

30

b. 現在のノードに対して、split\_transform\_flagが1である場合、4つの子ノードの各々のために、以下が適用される。現在のノードを子ノードに設定し、ステップaに進む。

3. 代替的に、変換ツリー階層がノードのために導入され、cbf\_luma、cbf\_cb、およびcbf\_crが新しくスプリットされたノードに設定されるとき、新しくスプリットされたノードのcbf\_luma、cbf\_cb、およびcbf\_crは0に設定される。

4. 代替的に、変換ツリー階層がノードのために導入され、cbf\_luma、cbf\_cb、およびcbf\_crが新しくスプリットされたノードに設定されるとき、cbf\_luma（または、cbf\_cb、もしくはcbf\_cr）が0に等しくない場合、新しくスプリットされたノードのcbf\_luma（または、cbf\_cb、もしくはcbf\_cr）は任意の非ゼロの正の整数値（たとえば、1）に設定され、さもなければ、0に設定される。

40

5. 変換ツリーが0に等しいsplit\_transform\_flagを有するルートを有するとき、通常のPU（または、通常のPUの一部）のみを包含する通常のノードは、次のステップの順で、変換ユニットにスプリットされなければならない。

a. CUのsplit\_transform\_flagがまず1に設定される。

b. CUの区分が $N \times N$ である場合、通常のPUは1つの通常のノードに対応する。CUの区分が $2N \times N$ または $N \times 2N$ である場合、通常のPUは2つの通常のノードに対応する。CUの区分が $2N \times nN$ 、 $2N \times nD$ 、 $nL \times 2N$ 、または $nR \times 2N$ である

50



場合、通常のPUは0または2つの通常のノードに対応する。

c. 上記の通常のノードの各々は、`split_transform_flag`を0に等しく設定させ、`cbf_luma`、`cbf_cb`、および`cbf_cr`を、高度CU全体をカバーする元の変換ユニットの`cbf_luma`、`cbf_cb`、および`cbf_cr`に等しくなるように設定させるように設定される。

d. CUのいずれのサブPU（または、サブPUの一部）を包含する高度ノードの各々に対して、上記のステップ2、3、または4と同様の変換ツリー階層が導入される。

6. 上記のステップ5の代替として、上記の通常ノードの各々に対して、`cbf_luma`、`cbf_cb`、および`cbf_cr`は0に等しく設定される。

10

7. 上記のステップ4の代替として、上記の通常ノードの各々に対して、`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）が0に等しくない場合、その`cbf_luma`（または、`cbf_cb`、もしくは`cbf_cr`）は任意の非ゼロの正の整数値（たとえば、1）に等しく設定され、さもなければ、0に設定される。

[0173]いくつかの例示的な実装詳細が下で提供される。下のテキストは本開示で説明する技法のうちのいくつかの実装形態のための、3D-HEVC WD1に対して提案される修正を示す。

復号プロセス

H.8.1 一般的な復号プロセス

20

3. 従属節8.4、8.5、8.6、8.7、および8.8におけるプロセスは、すべてのシンタックス構造レイヤにおいてシンタックス要素を使用した復号プロセスを規定する。スライスへのピクチャの分割、スライス区分へのスライスの分割、およびコーディングツリーユニットへのスライス区分の分割はそれぞれ、ピクチャの区分を形成するように、ピクチャのコード化スライスがピクチャのコーディングツリーユニットごとにスライス区分データを包含するものとするのがビットストリームコンフォーマンスの要件である。

H.8.5.3.3.2 参照ピクチャ選択プロセス

このプロセスへの入力参照インデックス`refIdxLX`である。

30

このプロセスの出力は、ルーマサンプル`refPicLXL`の1つの2次元アレイと、クロマサンプル`refPicLXcb`および`refPicLXcr`の2つの2次元アレイとからなる参照ピクチャである。

出力された参照ピクチャ`RefPicListX[refIdxLX]`は、ルーマサンプル`refPicLXL`の1つの`pic_width_in_luma_samples`アレイと、クロマサンプル`refPicLXcb`および`refPicLXcr`の2つの`PicWidthInSamplesC`×`PicHeightInSamplesC`アレイとからなる。

参照ピクチャサンプルアレイ`refPicLXL`、`refPicLXcb`、および`refPicLXcr`は、前に復号されたピクチャのための従属節8.7および従属節8.8によって導出された復号サンプルアレイ`SL`、`Scb`、ならびに`Scr`に対応する。

40

H.8.7 変換ツリー修正プロセス

H.8.7.1 一般

このプロセスに対する入力、コーディングツリーのスプリットフラグアレイ`split_cu_flag`、コーディングユニットの区分モードアレイ`PartMode`、変換ツリーのスプリットフラグ`split_transform_flag`、コーディングユニットのサブ予測ユニットフラグアレイ`subPuFlag`、サブ予測ユニットサイズ`subPuSize`である。

このプロセスの出力は、変換ツリーの修正されたスプリットフラグアレイ`split_transform_flag`である。

50

各コーディングユニットに対して、そのユニットがインター予測モードでコーディングされ、そのユニットがサブ予測ユニット設計を利用する予測ユニットを包含する場合、変換ツリーのスプリットフラグは、サブ予測ユニット境界が変換ユニット境界になるように修正される。

- 従属節 8 . 7 . 2 で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション (  $x_{Cb}$  ,  $y_{Cb}$  )、( 0 , 0 ) に等しく設定されたルーマロケーション (  $x_{B0}$  ,  $y_{B0}$  )、(  $1 < \log_2 \text{MaxCbSize}$  ) に等しく設定されたルーマコーディングブロックサイズ  $n_{CbS}$ 、0 に等しく設定されたコーディングツリー深度  $cuDepth$ 、コーディングツリーのスプリットフラグアレイ  $spl\_cu\_flag$ 、コーディングユニットの区分モードアレイ  $PartMode$ 、変換ツリーのスプリットフラグアレイ  $spl\_transform\_flag$ 、サブ予測ユニットフラグアレイ  $subPuFlag$ 、およびサブ予測ユニットサイズ  $subPuSize$  と、出力として、変換ツリーの修正されたスプリットフラグアレイ  $spl\_transform\_flag$  とで呼び出される。

10

H . 8 . 7 . 2 . サブ予測ユニットを包含するコーディングユニットの変換ツリー修正プロセス

このプロセスへの入力には以下の通りである。

- 現在のピクチャの左上ルーマサンプルに対する現在のルーマコーディングブロックの左上サンプルを規定するルーマロケーション (  $x_{Cb}$  ,  $y_{Cb}$  )、  
 - 現在のルーマコーディングブロックの左上サンプルに対する現在のルーマブロックの左上サンプルを規定するルーマロケーション (  $x_{B0}$  ,  $y_{B0}$  )、  
 - ルーマコーディングブロックサイズ  $n_{CbS}$ 、  
 - コーディングツリー深度を規定する変数  $cuDepth$ 、  
 - コーディングツリーのスプリットフラグアレイ  $spl\_cu\_flag$ 、  
 - 変換ツリーのスプリットフラグアレイ  $spl\_transform\_flag$ 、

20

- サブ予測ユニットフラグアレイ  $subPuFlag$ 、  
 - サブ予測ユニットサイズ  $subPuSize$ 、

このプロセスの出力は修正済みである。

- 変換ツリーのスプリットフラグアレイ  $spl\_transform\_flag$ 、

30

$spl\_cu\_flag[x_{Cb} + x_{B0}][y_{Cb} + y_{B0}][cuDepth]$  の値に応じて、以下が適用される。

-  $spl\_cu\_flag[x_{Cb} + x_{B0}][y_{Cb} + y_{B0}][cuDepth]$  が 1 に等しい場合、以下の順序のステップが適用される。

1 . 変数  $x_{B1}$  および  $y_{B1}$  は次のように導出される。

【 0 1 5 6 】

- 変数  $x_{B1}$  は、 $x_{B0} + (n_{CbS} > 1)$  に等しく設定される。

- 変数  $y_{B1}$  は、 $y_{B0} + (n_{CbS} > 1)$  に等しく設定される。

2 . この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション (  $x_{Cb}$  ,  $y_{Cb}$  )、ルーマロケーション (  $x_{B0}$  ,  $y_{B0}$  )、(  $n_{CbS} > 1$  ) に等しく設定されたルーマコーディングブロックサイズ  $n_{CbS}$ 、 $cuDepth + 1$  に等しく設定されたコーディングツリー深度  $cuDepth$ 、コーディングツリーのスプリットフラグアレイ  $spl\_cu\_flag$ 、コーディングユニットの区分モードアレイ  $PartMode$ 、変換ツリーのスプリットフラグアレイ  $spl\_transform\_flag$ 、サブ予測ユニットフラグアレイ  $subPuFlag$ 、サブ予測ユニットサイズ  $subPuSize$  と、出力として、変換ツリーの修正されたスプリットフラグアレイ  $spl\_transform\_flag$  とで呼び出される。

40

3 . この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション (  $x_{Cb}$  ,  $y_{Cb}$  )、ルーマロケーション (  $x_{B1}$  ,  $y_{B0}$  )、(  $n_{CbS} >$

50

1) に等しく設定されたルーマコーディングブロックサイズ  $nCbS$ 、 $cuDepth + 1$  に等しく設定されたコーディングツリー深度  $cuDepth$ 、コーディングツリーのスプリットフラグアレイ  $split\_cu\_flag$ 、コーディングユニットの区分モードアレイ  $PartMode$ 、変換ツリーのスプリットフラグアレイ  $split\_transform\_flag$ 、サブ予測ユニットフラグアレイ  $subPuFlag$ 、サブ予測ユニットサイズ  $subPuSize$  と、出力として、変換ツリーの修正されたスプリットフラグアレイ  $split\_transform\_flag$  とで呼び出される。

4. この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション  $(xCb, yCb)$ 、ルーマロケーション  $(xB0, yB1)$ 、 $(nCbS > 1)$  に等しく設定されたルーマコーディングブロックサイズ  $nCbS$ 、 $cuDepth + 1$  に等しく設定されたコーディングツリー深度  $cuDepth$ 、コーディングツリーのスプリットフラグアレイ  $split\_cu\_flag$ 、コーディングユニットの区分モードアレイ  $PartMode$ 、変換ツリーのスプリットフラグアレイ  $split\_transform\_flag$ 、サブ予測ユニットフラグアレイ  $subPuFlag$ 、サブ予測ユニットサイズ  $subPuSize$  と、出力として、変換ツリーの修正されたスプリットフラグアレイ  $split\_transform\_flag$  とで呼び出される。

5. この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション  $(xCb, yCb)$ 、ルーマロケーション  $(xB1, yB1)$ 、 $(nCbS > 1)$  に等しく設定されたルーマコーディングブロックサイズ  $nCbS$ 、 $cuDepth + 1$  に等しく設定されたコーディングツリー深度  $cuDepth$ 、コーディングツリーのスプリットフラグアレイ  $split\_cu\_flag$ 、コーディングユニットの区分モードアレイ  $PartMode$ 、変換ツリーのスプリットフラグアレイ  $split\_transform\_flag$ 、サブ予測ユニットフラグアレイ  $subPuFlag$ 、サブ予測ユニットサイズ  $subPuSize$  と、出力として、変換ツリーの修正されたスプリットフラグアレイ  $split\_transform\_flag$  とで呼び出される。

- さもなければ  $(split\_cu\_flag[xCb + xB0][yCb + yB0][cuDepth])$  が 0 に等しい)、 $nCbS$  が  $subPuSize$  より大きい場合、以下の順序のステップが適用される。

1. 変数  $xB1$  および  $yB1$  は次のように導出される。

- 変数  $xB1$  は、 $xB0 + (nCbS > 1)$  に等しく設定される。

- 変数  $yB1$  は、 $yB0 + (nCbS > 1)$  に等しく設定される。

2. 変数  $subPuDeblockingFlag$  を以下の順序のステップによって導出する。

-  $subPuDeblockingFlag$  が 0 に等しく設定される。

-  $subPuFlag[xCb + xB0][yCb + yB0]$  が 1 に等しい場合、 $subPuDeblockingFlag$  は 1 に等しく設定される。

-  $subPuFlag[xCb + xB1][yCb + yB0]$  が 1 に等しい場合、 $subPuDeblockingFlag$  は 1 に等しく設定される。

-  $subPuFlag[xCb + xB0][yCb + yB1]$  が 1 に等しい場合、 $subPuDeblockingFlag$  は 1 に等しく設定される。

-  $subPuFlag[xCb + xB1][yCb + yB1]$  が 1 に等しい場合、 $subPuDeblockingFlag$  は 1 に等しく設定される。

-  $PartMode[xCb + xB0][yCb + yB0]$  が  $PART\_nL \times 2N$  に等しい場合、または  $PartMode[xCb + xB0][yCb + yB0]$  が  $PART\_nR \times 2N$  に等しい場合、または  $PartMode[xCb + xB0][yCb + yB0]$  が  $PART\_2N \times nU$  に等しい場合、または  $PartMode[xCb + xB0][yCb + yB0]$  が  $PART\_2N \times nD$  に等しい場合、 $subPuDeblockingFlag$  は 0 に等しく設定される。

3.  $subPuDeblockingFlag$  が 1 に等しい場合、次の順序ステップが適用される。

- `split_transform_flag[xCb + xB0][yCb + yB0][cuDepth]` が 0 に等しい場合、`split_transform_flag[xCb + xB0][yCb + yB0][cuDepth]` を 1 に等しくなるように設定する。

- 従属節 8.7.3 で規定された変換ツリーの修正プロセスは、入力として、ルーマロケーション(`xCb`, `yCb`)、ルーマロケーション(`xB0`, `yB0`)、コーディングツリー深度`cuDepth`、ブロックサイズ`nCbS`、変換ツリーのスプリットフラグアレイ`split_transform_flag`、予測ユニットのサブ予測ユニットフラグアレイ`subPuFlag`、予測ユニットのサブ予想ユニットサイズアレイ`subPuSize`と、出力として、コーディングツリーの修正されたスプリットフラグアレイ`split_transform_flag`とで呼び出される。

10

H.8.7.3 ルーマコーディングブロックの変換ツリー修正プロセス

このプロセスへの入力は以下の通りである。

- 現在のピクチャの左上ルーマサンプルに対する現在のルーマ予測ブロックの左上サンプルを規定するルーマロケーション(`xCb`, `yCb`)、
- 現在のルーマコーディングブロックの左上サンプルに対する現在のルーマブロックの左上サンプルを規定するルーマロケーション(`xB0`, `yB0`)、
- ブロックサイズを規定する変数`nCbS`、
- 変換ツリー深度を規定する変数`trafoDepth`、
- 変換ツリーのスプリットフラグアレイ`split_transform_flag`、

20

- サブ予測ユニットフラグアレイ`subPuFlag`、
- サブ予測ユニットサイズ`subPuSize`、

このプロセスの出力は修正済みである。

- 変換ツリーのスプリットフラグアレイ`split_transform_flag`、

`nCbS` が `subPuSize` よりも大きい場合、以下の順序のステップが適用される。

- 変数`xB1`および`yB1`は次のように導出される。
  - 変数`xB1`は、`xB0 + (nCbS >> 1)` に等しく設定される。
  - 変数`yB1`は、`yB0 + (nCbS >> 1)` に等しく設定される。
- `xB0`、`xB1`の`x`に対して
  - `yB0`、`yB1`の`y`に対して、
    - `subPuFlag[xCb + x][yCb + y]` が 1 に等しい場合、
    - `split_transform_flag[xCb + x][yCb + y][trafoDepth + 1]` が 0 に等しい場合、
    - `split_transform_flag[xCb + x][yCb + y][trafoDepth + 1]` を 1 に等しくなるように設定する、
    - この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション(`xCb`, `yCb`)、ルーマロケーション(`xCb + x`, `yCb + y`)、変換ツリー深度`trafoDepth`は`trafoDepth + 1`に等しく設定される、ブロックサイズ`nCbS`は`(nCbS >> 1)`に等しく設定される、変換ツリーのスプリットフラグアレイ`split_transform_flag`、サブ予測ユニットフラグアレイ`subPuFlag`、サブ予測ユニットサイズ`subPuSize`と、出力として、コーディングツリーの修正されたスプリットフラグアレイ`split_transform_flag`とで呼び出される。

30

40

- さもなければ(`split_transform_flag[xCb + x][yCb + y][trafoDepth + 1]` が 1 に等しい場合)、
  - この従属節で規定される変換ツリーの修正プロセスは、入力として、ルーマロケーション(`xCb`, `yCb`)、ルーマロケーション(`xCb + x`, `yCb +`

50

y)、変換ツリー深度  $\text{trafoDepth}$  は  $\text{trafoDepth} + 1$  に等しく設定される、ブロックサイズ  $\text{nCbS}$  は  $(\text{nCbS} > 1)$  に等しく設定される、変換ツリーのスプリットフラグ  $\text{split\_transform\_flag}$ 、サブ予測ユニットフラグ  $\text{subPufFlag}$ 、サブ予測ユニットサイズ  $\text{subPuSize}$  と、出力として、コーディングツリーの修正されたスプリットフラグ  $\text{split\_transform\_flag}$  とで呼び出される。

H. 8. 8 インループフィルタプロセス

...

【0157】

[0174] 図15は、本開示の技法を実装するように構成され得るビデオ符号化器20の一例を示すブロック図である。ビデオ符号化器20は、ビデオスライス、たとえば、テクスチャ画像と深度マップの両方のスライス内のビデオブロックのイントラコーディングおよびインターコーディング（ビュー間コーディングを含む）を実行し得る。テクスチャ情報は、概して、ルミナンス（輝度または強度）情報とクロミナンス（色、たとえば、赤い色相および青い色相）情報とを含む。いくつかの例では、ビデオ符号化器20は、ルミナンススライスに対するコーディングモードを決定し、クロミナンス情報を符号化するために（たとえば、区分情報、イントラ予測モード選択、動きベクトルなどを再使用することによって）ルミナンス情報をコーディングすることからの予測情報を再使用することができる。イントラコーディングは、所与のビデオフレームまたはピクチャ内のビデオの空間冗長性を低減または除去するために空間的予測に依拠する。インターコーディングは、ビデオシーケンスの隣接するフレームまたはピクチャ内のビデオの時間的冗長性を低減または除去するために時間的予測に依拠する。イントラモード（Iモード）は、いくつかの空間ベースのコーディングモードのいずれかを指す場合がある。単一方向予測（Pモード）または双予測（Bモード）などのインターモードは、いくつかの時間ベースのコーディングモードのいずれかを指す場合がある。

【0158】

[0175] 図15に示すように、ビデオ符号化器20は、符号化されるべきビデオフレーム（たとえば、テクスチャ画像または深度マップ）内の現在のビデオブロック（すなわち、ルミナンスブロック、クロミナンスブロック、または深度ブロックなどのビデオデータのブロック）を受信する。図15の例では、ビデオ符号化器20は、モード選択ユニット40と、ビデオデータメモリ41と、復号ピクチャバッファ64と、加算器50と、変換処理ユニット52と、量子化ユニット54と、フィルタユニット63と、エンロピー符号化ユニット56とを含む。フィルタユニット63は、本開示で説明するように、デブロッキングフィルタプロセスを適用することができる。モード選択ユニット40は、今度は、動き補償ユニット44と、動き推定ユニット42と、イントラ予測処理ユニット46と、区分ユニット48とを含む。ビデオブロックの再構築のために、ビデオ符号化器20はまた、逆量子化ユニット58と、逆変換処理ユニット60と、加算器62とを含む。フィルタユニット63は、再構築されたビデオからブロックネスアーティファクトを除去するためにブロック境界をフィルタリングするためのデブロッキングフィルタおよび/またはSAOフィルタを含み得る。追加のフィルタ（インループまたはポストループ）も、デブロッキングフィルタに加えて使用され得る。そのようなフィルタは、簡潔さのために図示されていないが、望まれる場合には、加算器50の出力を（インループフィルタとして）フィルタリングすることができる。

【0159】

[0176] ビデオデータメモリ41は、ビデオ符号化器20の構成要素によって符号化されるべきビデオデータを記憶することができる。ビデオデータメモリ41に記憶されるビデオデータは、たとえば、ビデオソース18から取得され得る。復号ピクチャバッファ64は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオ符号化器20によってビデオデータを符号化する際に使用するための参照ビデオデータを記憶する参照ピクチャメモリであり得る。ビデオデータメモリ41および復号ピクチャバ

10

20

30

40

50

ッファ 64 は、同期 DRAM ( SDRAM )、磁気抵抗 RAM ( MRAM )、抵抗 RAM ( RRAM (登録商標) )、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ ( DRAM ) のような様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ 41 および復号ピクチャバッファ 64 は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ 41 は、ビデオ符号化器 20 の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

#### 【 0160 】

[0177] 符号化プロセス中に、ビデオ符号化器 20 は、コーディングされるべきビデオフレームまたはスライスを受信する。フレームまたはスライスは、複数のビデオブロックに分割され得る。動き推定ユニット 42 および動き補償ユニット 44 は、時間的予測を提供するために、1つまたは複数の参照フレーム中の1つまたは複数のブロックに対して、受信されたビデオブロックのインター予測コーディングを実行する。イントラ予測処理ユニット 46 は、代替的に、空間的予測を行うために、コーディングされるべきブロックと同じフレームまたはスライス中の1つもしくは複数の隣接ブロックに対して、受信されたビデオブロックのイントラ予測コーディングを実行し得る。ビデオ符号化器 20 は、たとえば、ビデオデータの各ブロックに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することができる。

#### 【 0161 】

[0178] さらに、区分ユニット 48 は、前のコーディングパス内の前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分することができる。たとえば、区分ユニット 48 は、最初に、フレームまたはスライスを LCU に区分し、レートひずみ分析 (たとえば、レートひずみ最適化) に基づいて、LCU の各々をサブCU に区分することができる。モード選択ユニット 40 は、サブCU への LCU の区分を示す 4 分木データ構造をさらに生成することができる。四分木の葉ノード CU は、1つまたは複数の PU と 1つまたは複数の TU とを含み得る。

#### 【 0162 】

[0179] モード選択ユニット 40 は、たとえば、誤差結果に基づいて、コーディングモードのうちの1つ、すなわち、イントラまたはインターを選択し、結果のイントラコード化ブロックまたはインターコード化ブロックを、残差ブロックデータを生成するために加算器 50 に、参照フレームとしての使用のために符号化されたブロックを再構築するために加算器 62 に供給することができる。モード選択ユニット 40 はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのようなシンタックス情報などのシンタックス要素をエントロピー符号化ユニット 56 に提供する。

#### 【 0163 】

[0180] 動き推定ユニット 42 および動き補償ユニット 44 は、高度に統合され得るが、概念上の目的から別々に図示されている。動き推定ユニット 42 によって実行される動き推定は、ビデオブロックのため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、たとえば、現在のフレーム (もしくは、他のコード化ユニット) 内でコーディングされている現在のブロックに対する参照フレーム (もしくは、他のコード化ユニット) 内の予測ブロックに対する現在のビデオフレームまたは現在のピクチャ内のビデオブロックの PU の変位を示すことができる。

#### 【 0164 】

[0181] 予測ブロックは、絶対差の合計 ( SAD : sum of absolute difference )、二乗差の合計 ( SSD : sum of square difference )、または他の差分基準によって決定され得るピクセル差に関する、コーディングされるべきブロックとよく一致することがわかったブロックである。いくつかの例では、ビデオ符号化器 20 は、復号ピクチャバッファ 64 に記憶された参照ピクチャのサブ整数ピクセル位置のための値を計算し得る。たとえば、ビデオ符号化器 20 は、参照ピクチャの 1 / 4 ピクセル位置、1 / 8 ピクセル位置、または他の分数ピクセル位置の値を補間することができる。したがって、動き推定ユニット

42は、フルピクセル位置および分数ピクセル位置に対する動き検索を実行し、分数ピクセル精度を有する動きベクトルを出力することができる。

【0165】

[0182]動き推定ユニット42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って、インターコード化スライス中のビデオブロックのためのPUの動きベクトルを計算する。参照ピクチャは、第1の参照ピクチャリスト(リスト0)または第2の参照ピクチャリスト(リスト1)から選択され得、それらの各々は、復号ピクチャバッファ64に記憶された1つまたは複数の参照ピクチャを識別する。参照ピクチャリストは、本開示の技法を使用して構築され得る。動き推定ユニット42は、計算された動きベクトルを、エントロピー符号化ユニット56および動き補償ユニット44に送る。

10

【0166】

[0183]動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって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または生成することに関与し得る。この場合も、いくつかの例では、動き推定ユニット42と動き補償ユニット44とは機能的に統合され得る。現在のビデオブロックのPUのための動きベクトルを受信するとすぐ、動き補償ユニット44は、参照ピクチャリストのうちの1つにおいて動きベクトルが指す予測ブロックを位置特定することができる。加算器50は、下で論じるように、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42は、ルーマ成分に対して動き推定を実行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方のために、ルーマ成分に基づいて計算された動きベクトルを使用する。このように、動き補償ユニット44は、動き推定ユニット42がクロマ成分のための動き探索を実行する必要がないように、クロマ成分をコーディングするためにルーマ成分に対して決定された動き情報を再使用することができる。モード選択ユニット40はまた、ビデオスライスのビデオブロックを復号する際にビデオ復号器30が使用するためのビデオブロックとビデオスライスとに関連するシンタックス要素を生成することができる。

20

【0167】

[0184]イントラ予測処理ユニット46は、上記で説明したように、動き推定ユニット42と動き補償ユニット44とによって実行されるインター予測の代替として、現在のブロックをイントラ予測することができる。特に、イントラ予測処理ユニット46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ことができる。いくつかの例では、イントラ予測処理ユニット46は、たとえば、別個の符号化パスの間に、様々なイントラ予測モードを使用して現在のブロックを符号化することができ、イントラ予測処理ユニット46(または、いくつかの例では、モード選択ユニット40)は、テストされたモードから使用するのに適切なイントラ予測モードを選択することができる。

30

【0168】

[0185]たとえば、イントラ予測処理ユニット46は、様々なテストされたイントラ予測モードにレートひずみ分析を使用してレートひずみ値を計算し、テストされたモードの中で最良のレートひずみ特性を有するイントラ予測モードを選択することができる。レートひずみ分析は、一般に、符号化されたブロックと、符号化されたブロックを生成するために符号化された元の符号化されていないブロックとの間のひずみ(または、誤差)の量、ならびに符号化されたブロックを作成するのに使用されたビットレート(すなわち、ビットの個数)を決定する。イントラ予測処理ユニット46は、どのイントラ予測モードがブロックのための最良のレートひずみ値を示すのかを決定するために、様々な符号化されたブロックのひずみおよびレートから比を算出することができる。

40

【0169】

[0186]ブロックのためのイントラ予測モードを選択した後、イントラ予測処理ユニット46は、エントロピー符号化ユニット56にブロックのために選択されたイントラ予測モードを示す情報を提供することができる。エントロピー符号化ユニット56は、選択され

50

たイントラ予測モードを示す情報を符号化することができる。ビデオ符号化器 20 は、複数のイントラ予測モードインデックステーブルおよび（符号語マッピングテーブルとも呼ばれる）複数の修正されたイントラ予測モードインデックステーブルと、様々なブロックのための符号化コンテキストの定義と、コンテキストの各々について使用すべき、最確イントラ予測モード、イントラ予測モードインデックステーブル、および修正されたイントラ予測モードインデックステーブルの表示とを含み得る構成データを、送信されるビットストリーム中に含めることができる。

#### 【0170】

[0187]ビデオ符号化器 20 は、モード選択ユニット 40 からの予測データを、コーディングされている元のビデオブロックから減算することによって、残差ビデオブロックを形成する。加算器 50 は、この減算演算を実行する 1 つまたは複数の構成要素を表す。変換処理ユニット 52 は、離散コサイン変換（DCT）または概念的には類似する変換のような変換を残差ブロックに適用し、残差変換係数の値を備えるビデオブロックを生成する。変換処理ユニット 52 は、概念的には DCT に類似する他の変換を実行することができる。ウェーブレット変換、整数変換、サブバンド変換または他のタイプ変換も使用され得る。いずれの場合でも、変換処理ユニット 52 は、変換を残差ブロックに適用し、残差変換係数のブロックを生成する。

#### 【0171】

[0188]変換は、ピクセル値領域からの残差情報を、周波数領域のような変換領域に変換することができる。変換処理ユニット 52 は、得られた変換係数を量子化ユニット 54 へ送ることができる。量子化ユニット 54 は、ビットレートをさらに低減するために、変換係数を量子化する。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって修正され得る。いくつかの例では、量子化ユニット 54 は、次いで、量子化された変換係数を含む行列のスキャンを実行することができる。代替的に、エントロピー符号化ユニット 56 がスキャンを実行することができる。

#### 【0172】

[0189]量子化に続いて、エントロピー符号化ユニット 56 は、量子化された変換係数をエントロピーコーディングする。たとえば、エントロピー符号化ユニット 56 は、コンテキスト適応可変長コーディング（CAVLC）、コンテキスト適応 2 進算術コーディング（CABAC）、シンタックススペースコンテキスト適応 2 進算術コーディング（SBAC）、確率区間区分エントロピー（PIPE）コーディング、または別のエントロピーコーディング技法を実行することができる。コンテキストベースのエントロピーコーディングの場合、コンテキストは、隣接ブロックに基づき得る。エントロピー符号化ユニット 56 によるエントロピーコーディングに続いて、符号化されたビットストリームは、別のデバイス（たとえば、ビデオ復号器 30）に送信され、または後の送信もしくは取出のためにアーカイブされ得る。

#### 【0173】

[0190]逆量子化ユニット 58 および逆変換処理ユニット 60 は、たとえば、参照ブロックとして後で使用するために、ピクセル領域中で残差ブロックを再構築するために、それぞれ、逆量子化および逆変換を適用する。動き補償ユニット 44 は、残差ブロックを復号ピクチャバッファ 64 に記憶された参照ピクチャのうちの 1 つの予測ブロックに加算することによって参照ブロックを計算することができる。動き補償ユニット 44 は、動き推定での使用のためにサブ整数ピクセル値を計算するために、再構築された残差ブロックに 1 つまたは複数の補間フィルタを適用することにもできる。加算器 62 は、復号ピクチャバッファ 64 に記憶するための再構築されたビデオブロックを生成するために、再構築された残差ブロックを動き補償ユニット 44 によって生成された動き補償された予測ブロックに加算する。再構築されたビデオブロックは、後続ビデオフレーム中のブロックをインターコーディングするための参照ブロックとして、動き推定ユニット 42 と動き補償ユニット 44 とによって使用され得る。



## 【 0 1 7 4 】

[0191]ビデオ符号化器 20 は、対応するクロミナンス成分がなくとも、ルミナンス成分をコーディングするためのコーディング技法に実質的に似るように深度マップを符号化することができる。たとえば、イントラ予測処理ユニット 46 は、深度マップのブロックをイントラ予測することができるが、動き推定ユニット 42 および動き補償ユニット 44 は、深度マップのブロックをインター予測し得る。しかしながら、上記で論じたように、深度マップのインター予測の間、動き補償ユニット 44 は、深度範囲と、それらの深度範囲のための精度値との差に基づいて、参照深度マップの値をスケーリング（すなわち、調整する）ことができる。たとえば、現在の深度マップ中および参照深度マップ中の異なる最大深度値が同じ実世界深度に対応する場合、ビデオ符号化器 20 は、予測のために、現在の深度マップ中の最大深度値に等しくなるように、参照深度マップの最大深度値をスケーリングすることができる。追加または代替として、ビデオ符号化器 20 は、たとえば、ビュー間予測と実質的に同様の技法を使用して、ビュー合成予測のためのビュー合成ピクチャを生成するために、更新された深度範囲値と精度値とを使用することができる。

10

## 【 0 1 7 5 】

[0192]ビデオ符号化器 20 は、本開示で説明する技法のいずれかを、単独でまたは任意の組合せで実行するように構成され得るビデオ符号化器の一例を表す。たとえば、ビデオ符号化器 20 は、3D-HEVCにおけるビデオコーディングのためのサブPUレベルの動き予測のための技法を実行するように構成され得る。

## 【 0 1 7 6 】

[0193]一例では、ビデオ符号化器 20 は、2つ以上のサブPUを含むPUを予測するためにサブPU BVS Pモードを実行するように構成され得る。サブPU BVS Pモードでは、ビデオ符号化器 20 の動き補償ユニット 44 は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、現在のPUのための第1の動き情報を決定する。動き補償ユニット 44 は、次いで、現在のPUを2つ以上のサブPUに区分し、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、サブPUの各々のための第2の動き情報を決定する。動き補償ユニット 44 は、サブPUの各々のための第2の動き情報に基づいて、PUのサブPUの各々を予測するために、動き補償を実行する。本開示の技法によれば、サブPU BVS Pモードを使用して予測される各PUに対して、ビデオ符号化器 20 は、動き補償が実行された後ですら、現在のPUのサブPUの各々のために導出された第2の動き情報を記憶する。第2の動き情報は、復号ピクチャバッファ 64 に記憶され得る。現在のPUのために記憶された追加の動き情報は、次いで、現在のPUが隣接ブロックである後続PUを予測するために使用され得る。

20

30

## 【 0 1 7 7 】

[0194]別の例では、ビデオ符号化器 20 は、動きベクトル改良が可能にされ得るシングルレイヤコーディングにおけるPUのサブPUを予測するための高度TMVPモードを実行するように構成され得る。高度TMVPモードでは、ビデオ符号化器 20 の動き補償ユニット 44 は、現在のPUに対応する参照ピクチャのブロックを識別する、現在のPUのための第1のステージの動きベクトルを決定する。動き補償ユニット 44 は、次いで、現在のPUを2つ以上のサブPUに区分し、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブPUの各々のための第2のステージの動き情報を決定し、この場合、サブPUの各々のための第2のステージの動き情報は、少なくとも1つの動きベクトルと、関連付けられた参照インデックスとを含む。動き補償ユニット 44 は、サブPUの各々のための第2のステージの動き情報に基づいてサブPUの各々を別個に予測するために、動き補償を実行する。いくつかの例では、動き補償ユニット 44 は、サブPUの各々のための第2のステージの動き情報の少なくとも1つの動きベクトルを改良するために、動きベクトル差分を決定することができる。

40

## 【 0 1 7 8 】

[0195]別の例では、ビデオ符号化器 20 は、複数のサブPUを有する、少なくとも1つ

50

のPUを含むビデオブロックのCUに適用されたデブロッキングフィルタプロセスに関する技法を実行するように構成され得る。本開示の技法によれば、サブPUを有するPUを含むビデオブロックのCUにデブロッキングフィルタを適用するのに先立って、ビデオ符号化器20のフィルタユニット63は、サブPU境界において人工的PU境界または人工的TU境界を作成するために、CUを変換する。一例では、フィルタユニット63は、サブPU境界が人工的TU境界に変換されるように、PUを変換ツリー階層に関連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーを変換する。別の例では、フィルタユニット63は、サブPU境界が人工的PU境界に変換されるように、PUをCUに関連付け、サブPUの各々をPUに関連付けるために、CUをコーディングツリーに変換する。次いで、フィルタユニット63は、人工的PU境界と人工的TU境界とを含めて、CUの2つの隣接するPU間のPU境界および/またはCUの2つの隣接するTU間のTU境界にデブロッキングフィルタを適用する。

10

#### 【0179】

[0196]図16は、本開示の技法を実装し得るビデオ復号器30の一例を示すブロック図である。図16の例では、ビデオ復号器30は、エン트로ピー復号ユニット70と、ビデオデータメモリ71と、動き補償ユニット72と、イントラ予測処理ユニット74と、逆量子化ユニット76と、逆変換処理ユニット78と、復号ピクチャバッファ82と、フィルタユニット83と、加算器80とを含む。ビデオ復号器30は、いくつかの例では、ビデオ符号化器20(図15)に関して説明された符号化パスとは全般に逆の復号パスを実行することができる。動き補償ユニット72は、エン트로ピー復号ユニット70から受信された動きベクトルに基づいて予測データを生成することができ、イントラ予測処理ユニット74は、エン트로ピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて予測データを生成することができる。

20

#### 【0180】

[0197]ビデオデータメモリ71は、ビデオ復号器30の構成要素によって復号されるべき、符号化ビデオビットストリームなどのビデオデータを記憶することができる。ビデオデータメモリ71に記憶されるビデオデータは、たとえば、コンピュータ可読媒体16から、たとえば、カメラなどのローカルビデオソースから、ビデオデータのワイヤードもしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって取得され得る。ビデオデータメモリ71は、符号化ビデオビットストリームからの符号化ビデオデータを記憶するコード化ピクチャバッファ(CPB)を形成することができる。復号ピクチャバッファ82は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオ復号器30によってビデオデータを復号する際に使用するための参照ビデオデータを記憶する参照ピクチャメモリであり得る。ビデオデータメモリ71および復号ピクチャバッファ82は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM)、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)のような様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ71および復号ピクチャバッファ82は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ71は、ビデオ復号器30の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

30

40

#### 【0181】

[0198]復号プロセスの間、ビデオ復号器30は、符号化ビデオスライスのビデオブロックおよび関連付けられたシンタックス要素を表す符号化ビットストリームをビデオ符号化器20から受信する。ビデオ復号器30のエン트로ピー復号ユニット70は、量子化された係数と、動きベクトルまたはイントラ予測モードインジケータと、他のシンタックス要素とを生成するために、ビットストリームをエン트로ピー復号する。エン트로ピー復号ユニット70は、動きベクトルと他のシンタックス要素とを動き補償ユニット72に転送する。ビデオ復号器30は、ビデオスライスレベルおよび/またはビデオブロックレベルでシンタックス要素を受け取ることができる。

50

## 【 0 1 8 2 】

[0199]ビデオスライスがイントラコード化（I）スライスとしてコーディングされるとき、イントラ予測処理ユニット74は、シグナリングされたイントラ予測モードと、現在のフレームまたは現在のピクチャの、前に復号されたブロックからのデータとに基づいて、現在のビデオスライスのビデオブロックのための予測データを生成することができる。ビデオフレームがインターコード化（すなわち、BまたはP）スライスとしてコーディングされるとき、動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルと他のシンタックス要素とに基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリストのうちの1つの中の参照ピクチャのうちの1つから生成され得る。ビデオ復号器30は、復号ピクチャバッファ82に記憶された参照ピクチャに基づいて、本開示の技法を使用して、参照フレームリスト、すなわち、リスト0およびリスト1を構築することができる。動き補償ユニット72は、動きベクトルと他のシンタックス要素とを解析することによって現在のビデオスライスのビデオブロックのための予測情報を決定し、復号されている現在のビデオブロックの予測ブロックを生成するために、その予測情報を使用する。たとえば、動き補償ユニット72は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード（たとえば、イントラ予測またはインター予測）と、インター予測スライスタイプ（たとえば、BスライスまたはPスライス）と、スライスのための参照ピクチャリストのうちの1つまたは複数についての構築情報と、スライスの各インター符号化ビデオブロックのための動きベクトルと、スライスの各インターコード化ビデオブロックのためのインター予測ステータスと、現在のビデオスライス中のビデオブロックを復号するための他の情報とを決定するために、受信されたシンタックス要素のうちのいくつかを使用する。

10

20

## 【 0 1 8 3 】

[0200]動き補償ユニット72はまた、補間フィルタに基づいて補間を実行することができる。動き補償ユニット72は、参照ブロックのサブ整数ピクセルのために補間された値を計算するために、ビデオブロックの符号化の間にビデオ符号化器20によって使用された補間フィルタを使用することができる。この場合、動き補償ユニット72は、受信されたシンタックス要素からビデオ符号化器20によって使用された補間フィルタを決定し、予測ブロックを生成するために補間フィルタを使用することができる。

30

## 【 0 1 8 4 】

[0201]逆量子化ユニット76は、ビットストリーム内で提供され、エントロピー復号ユニット70によって復号された、量子化変換係数を逆量子化（inverse quantize）、すなわち、逆量子化（de-quantize）する。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するための、ビデオスライス中の各ビデオブロックのためにビデオ復号器30によって計算される量子化パラメータ $QP_V$ の使用を含み得る。逆変換処理ユニット78は、ピクセル領域において残差ブロックを生成するために、逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換プロセスを変換係数に適用する。

40

## 【 0 1 8 5 】

[0202]動き補償ユニット72が、動きベクトルおよび他のシンタックス要素に基づいて現在のビデオブロックのための予測ブロックを生成した後、ビデオ復号器30は、逆変換処理ユニット78からの残差ブロックを動き補償ユニット72によって生成された対応する予測ブロックと加算することによって、復号ビデオブロックを形成する。加算器90は、この加算演算を実行する1つまたは複数の構成要素を表す。フィルタユニット63は、デブロッキングフィルタプロセスを適用することができる。フィルタユニット63は、再構築されたビデオからブロックネスアーティファクトを除去するためにブロック境界をフィルタリングするためのデブロッキングフィルタおよび/またはSAOフィルタを含み得る。追加のフィルタ（インループまたはポストループ）も、デブロッキングフィルタに加えて使用され得る。そのようなフィルタは、簡潔さのために示されていないが、望まれる

50

場合には、加算器 80 の出力を（インループフィルタとして）フィルタリングすることができる。所与のフレームまたはピクチャ中の復号ビデオブロックは、次いで、後続の動き補償のために使用される参照ピクチャを記憶する復号ピクチャバッファ 82 に記憶される。復号ピクチャバッファ 82 はまた、図 1 のディスプレイデバイス 32 などのディスプレイデバイス上での後の表示のために、復号ビデオを記憶する。

【0186】

[0203]ビデオ復号器 30 は、対応するクロミナンス成分がなくとも、ルミナンス成分を復号するための復号技法に実質的に似るように深度マップを復号することができる。たとえば、イントラ予測処理ユニット 74 は、深度マップのブロックをイントラ予測することができるが、動き補償ユニット 72 は、深度マップのブロックをインター予測し得る。しかしながら、上記で論じたように、深度マップのインター予測の間、動き補償ユニット 72 は、深度範囲と、それらの深度範囲のための精度値との差に基づいて、参照深度マップの値をスケーリング（すなわち、調整する）ことができる。たとえば、現在の深度マップおよび参照深度マップ中の異なる最大深度値が同じ実世界深度に対応する場合、ビデオ復号器 30 は、予測のために、現在の深度マップ中の最大深度値に等しくなるように、参照深度マップの最大深度値をスケーリングすることができる。追加または代替として、ビデオ復号器 30 は、たとえば、ビュー間予測に実質的に類似した技法を使用して、ビュー合成予測のためのビュー合成ピクチャを生成するために、更新された深度範囲値と精度値とを使用することができる。

【0187】

[0204]ビデオ復号器 30 は、本開示で説明する技法のいずれかを、単独でまたは任意の組合せで実行するように構成され得るビデオ復号器の一例を表す。たとえば、ビデオ復号器 30 は、3D-HEVC におけるビデオコーディングのためのサブPUレベルの動き予測のための技法を実行するように構成され得る。

【0188】

[0205]一例では、ビデオ復号器 30 は、2つ以上のサブPUを含むPUを予測するためにサブPU BVS Pモードを実行するように構成され得る。サブPU BVS Pモードでは、ビデオ復号器 30 の動き補償ユニット 72 は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む現在のPUのための第1の動き情報を決定する。動き補償ユニット 72 は、次いで、現在のPUを2つ以上のサブPUに区分し、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、サブPUの各々のための第2の動き情報を決定する。動き補償ユニット 72 は、サブPUの各々のための第2の動き情報に基づいてPUのサブPUの各々を予測するために、動き補償を実行する。本開示の技法によれば、サブPU BVS Pモードを使用して予測される各PUに対して、ビデオ復号器 30 は、動き補償が実行された後ですら、現在のPUのサブPUの各々のために導出された第2の動き情報を記憶する。第2の動き情報は、復号ピクチャバッファ 82 に記憶され得る。現在のPUのために記憶された追加の動き情報は、次いで、現在のPUが隣接ブロックである後続PUを予測するために使用され得る。

【0189】

[0206]別の例では、ビデオ復号器 30 は、動きベクトル改良が可能にされ得るシングルレイヤコーディングにおけるPUのサブPUを予測するための高度TMVPモードを実行するように構成され得る。高度TMVPモードでは、ビデオ復号器 30 の動き補償ユニット 72 は、現在のPUに対応する参照ピクチャのブロックを識別する、現在のPUのための第1のステージの動きベクトルを決定する。動き補償ユニット 72 は、次いで、現在のPUを2つ以上のサブPUに区分し、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブPUの各々のための第2のステージの動き情報を決定し、この場合、サブPUの各々のための第2のステージの動き情報は、少なくとも1つの動きベクトルと、関連付けられた参照インデックスとを含む。動き補償ユニット 72 は、サブPUの各々のための第2のステージの動き情報に基づいてサブPUの各々を別個に予測

するために、動き補償を実行する。いくつかの例では、動き補償ユニット72は、サブPUの各々のための第2のステージの動き情報の少なくとも1つの動きベクトルを改良するために、動きベクトル差分を決定することができる。

【0190】

[0207]別の例では、ビデオ復号器30は、複数のサブPUを有する、少なくとも1つのPUを含むビデオブロックのCUに適用されるデブロッキングフィルタプロセスに関する技法を実行するように構成され得る。本開示の技法によれば、サブPUを有するPUを含むビデオブロックのCUにデブロッキングフィルタを適用するのに先立って、ビデオ復号器30のフィルタユニット83は、サブPU境界において人工的PU境界または人工的TU境界を作成するために、CUを変換する。一例では、フィルタユニット83は、サブPU境界が人工的TU境界に変換されるように、PUを変換ツリー階層に関連付け、サブPUの各々をTUに関連付けるために、CUの変換ツリーを変換する。別の例では、フィルタユニット83は、サブPU境界が人工的PU境界に変換されるように、PUをCUに関連付け、サブPUの各々をPUに関連付けるために、CUをコーディングツリーに変換する。次いで、フィルタユニット83は、人工的PU境界と人工的TU境界とを含めて、CUの2つの隣接するPU間のPU境界および/またはCUの2つの隣接するTU間のTU境界にデブロッキングフィルタを適用する。

【0191】

[0208]図17は、サブPU BVS Pモードを使用して、現在のPUを予測し、決定されたサブPU動き情報記憶を記憶する例示的な動作を示すフローチャートである。本開示では、示した演算について、図16のビデオ復号器30を参照しながら説明する。他の例では、示した演算は、図15のビデオ符号化器20、または3D-HEVC規格に従って動作する任意の他の符号化デバイスもしくは復号デバイスによって実行され得る。

【0192】

[0209]ビデオ復号器30は、符号化スライスのビデオブロックと、関連付けられたシンタックス要素とを表す符号化ビデオビットストリームを受信する。ビデオ復号器30のエントロピー復号ユニット70は、量子化された変換係数と、動き情報および予測モードインジケータと、他のシンタックス要素とを生成するために、ビットストリームを復号する。エントロピー復号ユニット70は、復号されるべきビデオブロックの残差ブロックを再構築するために、復号された量子化変換係数を逆量子化ユニット76および逆変換処理ユニット78に送る。エントロピー復号ユニット70は、復号動き情報とインター予測モードインジケータとを動き補償ユニット72に送る。

【0193】

[0210]動き補償ユニット72は、マージモードまたはAMVPインター予測モードのうちの示されたモードに従って復号されるべきビデオブロックの各CUの各PUを予測する。たとえば、マージモードでは、動き補償ユニット72は、動き情報、すなわち、現在のPUの空間的隣接ブロックおよび時間的隣接ブロックの動きベクトル、ならびに、関連付けられた参照インデックスを含む動き情報のマージモード候補リストを生成することができる。この場合、復号動き情報は、現在のPUを予測するために、マージモード候補リスト中の動き情報のセットのうちの1つを示すマージインデックスを含み得る。BVS Pモードでは、マージモード候補リストは、視差動きベクトルと関連付けられた参照ビューインデックスとを含む動き情報を有する特殊BVS P候補を含み、深度情報は動き情報を改良するために使用される。

【0194】

[0211]本開示の技法によれば、動き補償ユニット72は、BVS Pモードに従って、PUの隣接ブロックから、現在のPUのための第1の動き情報を決定し、この場合、第1の動き情報は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む(140)。上記で説明したように、動き補償ユニット72は、マージモード候補リストからのBVS P候補として、現在のPUのための第1の動き情報を選択することができる。場合によっては、現在のPUのための

第1の動き情報は、第1の参照ピクチャリストおよび第2の参照ピクチャリストの各々に対応する視差動きベクトルを含み得る。

【0195】

[0212]動き補償ユニット72は、次いで、現在のPUを2つ以上のサブPUに区分する(142)。動き補償ユニット72は、サブPUの各々のための第2の動き情報を決定する、この場合、第2の動き情報は、サブPUの各々に対応するビュー間参照ピクチャの少なくとも1つの視差動きベクトル関連深度ブロックを含む(144)。たとえば、動き補償ユニット72は、サブPUの各々に対応するビュー間参照ピクチャの深度ブロックのための4つのコーナーピクセルの最大値を選択し、最大値を、サブPUの各々のための視差動きベクトルの水平成分に変換することができる。サブPUの各々のための視差動きベクトルの垂直成分はゼロに等しい。場合によっては、サブPUの各々のための第2の動き情報は、第1の参照ピクチャリストおよび第2の参照ピクチャリストの各々に対応する視差動きベクトルを含み得る。

10

【0196】

[0213]動き補償ユニット72は、第2の動き情報に基づいて、ビュー間参照ピクチャから現在のPUのサブPUの各々を予測するために、動き補償を実行する(146)。動き補償を実行した後、ビデオ復号器30は、後続のPUを予測するために使用されることになる、現在のPUのサブPUの各々のための第2の動き情報をメモリ、たとえば、ビデオ復号器30の復号ピクチャバッファ82に記憶する(148)。たとえば、ビデオ復号器30は、サブPUの各々のために導出された視差動きベクトルを、PUのための第1の動き情報の参照ビューインデックスによって識別されたビュー間参照ピクチャを含む参照ピクチャリストに関連付けられた復号ピクチャバッファ82に記憶することができる。動き補償の間、PUのサブPUの各々のための予測ブロックを生成した後、ビデオ復号器30は、対応する残差ブロックの再構築されたバージョンと、サブPUの各々のための予測ブロックとに基づいて、ビデオブロックの再構築されたバージョンを生成する。

20

【0197】

[0214]従来、3D-HEVCのBVSPモードでは、サブPUの各々を予測するために動き補償を実行した後、各参照ピクチャリストに対応する単一の視差動きベクトルのみが現在のPUのために記憶される。PUの動き補償がPUの複数のサブPUのための複数の動きベクトルに基づく場合ですら、各参照ピクチャリストのための単一の視差動きベクトルが記憶される。この場合、後続PUを予測するために現在のPUが使用されるとき、現在のPUのサブPUは、サブPU BVSPモードが動きベクトル予測の精度の改善に与える影響がほとんどないように、同じ動きベクトルを有すると見なされる。

30

【0198】

[0215]本開示の技法によれば、サブPU BVSPモードでコーディングされる各PUに対して、ビデオ復号器30は、動き補償が実行された後ですら、現在のPUのサブPUの各々のために導出された第2の動き情報を記憶する。現在のPUのために記憶された追加の動き情報は、次いで、現在のPUが隣接ブロックである後続PUを予測するために使用され得る。たとえば、動き補償ユニット72は、マージモード候補リスト中のサブPU BVSP候補として、復号ピクチャバッファ82に記憶されたPUのサブPUのうちの少なくとも1つのための第2の動き情報を含む、後続PUを予測するための動き情報のマージモード候補リストを生成することができる。一例では、後続PUがサブPU BVSPモードでコーディングされる場合、動き補償ユニット72は、後続PUのサブPUの各々のための第2の動き情報を導出する必要がない。代わりに、動き補償ユニット72は、代わりに、後続PUのサブPUを予測するための動き情報のマージモード候補リストからサブPU BVSP候補として第2の動き情報を選択することができる。

40

【0199】

[0216]図18は、CU中のサブPU境界を含めて、TU境界およびPU境界をフィルタリングするために、ビデオブロックの各CUにデブロッキングフィルタを適用する例示的な動作を示すフローチャートである。ビデオブロックを再構築した後、および参照ピクチャ

50

ャのブロックとしてビデオブロックを復号ピクチャバッファに記憶する前に、デブロッキングフィルタが適用される。本開示では、示した演算について、図 16 のビデオ復号器 30 を参照しながら説明する。他の例では、示した演算は、図 15 のビデオ符号化器 20、またはサブ P U 設計およびデブロッキングフィルタを使用する任意の他の符号化デバイスもしくは復号デバイスのビデオブロック再構築ループで実行され得る。

#### 【 0 2 0 0 】

[0217]ビデオ復号器 30 は、対応する残差ブロックの再構築されたバージョンと予測ブロックとに基づいて、ビデオブロックの再構築バージョンを生成する ( 1 5 0 )。ビデオブロックは少なくとも 1 つの C U を含み、C U は、2 つ以上のサブ P U に区分された、少なくとも 1 つの P U を含み得る。図 17 に関して上記で説明したように、ビデオ復号器 30 の動き補償ユニット 72 は、P U のサブ P U の各々のためのビデオ圧縮の間、予測ブロックを生成することができる。H E V C 規格のためのデブロッキングフィルタは、P U 内で、すなわち、サブ P U 境界に沿って、フィルタするように設計されていないが、これは、H E V C ブロックでは、動き補償は P U 全体のために同じであると仮定されるためである。本開示は、H E V C デブロッキングフィルタが 3 D - H E V C ブロックのために引き続き使用され得るように、サブ P U を有する P U をデブロッキングフレンドリーな構造に変換するための技法について説明する。

#### 【 0 2 0 1 】

[0218]2 つ以上のサブ P U を有する P U を含むビデオブロックの C U にデブロッキングフィルタを適用するのに先立って、ビデオ復号器 30 のフィルタユニット 83 は、P U の 2 つの隣接するサブ P U 間のサブ P U 境界において人工的 P U 境界または人工的 T U 境界を作成するために、C U を変換する ( 1 5 2 )。一例では、フィルタユニット 83 は、サブ P U 境界が人工的 T U 境界に変換されるように、P U を変換ツリー階層に関連付け、サブ P U の各々を T U に関連付けるために、C U の変換ツリーを変換する。別の例では、フィルタユニット 83 は、サブ P U 境界が人工的 P U 境界に変換されるように、P U を C U に関連付け、サブ P U の各々を P U に関連付けるために、C U をコーディングツリーに変換する。

#### 【 0 2 0 2 】

[0219]次いで、フィルタユニット 83 は、人工的 P U 境界と人工的 T U 境界とを含めて、C U の 2 つの隣接する P U 間の P U 境界および / または C U の 2 つの隣接する T U 間の T U 境界にデブロッキングフィルタを適用する ( 1 5 4 )。ビデオブロックの再構築されたバージョンの C U の各々をフィルタリングした後、フィルタユニット 83 は、参照ピクチャのブロックとしてビデオブロックを復号ピクチャバッファ 82 に記憶する ( 1 5 6 )。

#### 【 0 2 0 3 】

[0220]図 19、シングルレイヤコーディングにおける P U のサブ P U を予測するための高度 T M V P モードの例示的な動作を示すフローチャート。本開示では、示した演算について、図 16 のビデオ復号器 30 を参照しながら説明する。他の例では、示した演算は、図 15 のビデオ符号化器 20、またはサブ P U 設計を使用する任意の他の符号化デバイスもしくは復号デバイスによって実行され得る。

#### 【 0 2 0 4 】

[0221]ビデオ復号器 30 は、符号化ビデオスライスのビデオブロックと、関連付けられたシンタックス要素とを表す符号化ビデオビットストリームを受信する。ビデオ復号器 30 のエンтроピー復号ユニット 70 は、量子化された変換係数と、動き情報および予測モードインジケータと、他のシンタックス要素とを生成するために、ビットストリームを復号する。エンтроピー復号ユニット 70 は、復号されるべきビデオブロックの残差ブロックを再構築するために、復号された量子化変換係数を逆量子化ユニット 76 および逆変換処理ユニット 78 に送る。エンтроピー復号ユニット 70 は、復号動き情報とインター予測モードインジケータとを動き補償ユニット 72 に送る。

#### 【 0 2 0 5 】

[0222]動き補償ユニット72は、マージモードまたはAMVPインター予測モードのうちの示されたモードに従って復号されるべきビデオブロックの各CUの各PUを予測する。たとえば、マージモードでは、動き補償ユニット72は、動き情報、すなわち、現在のPUの空間的隣接ブロックおよび時間的隣接ブロックの動きベクトル、ならびに、関連付けられた参照インデックスを含む動き情報のマージモード候補リストを生成することができる。この場合、復号動き情報は、現在のPUを予測するために、マージモード候補リスト中の動き情報のセットのうちの1つを示すマージインデックスを含み得る。別の例では、AMVPモードでは、動き補償ユニット72は、現在のPUの空間的隣接ブロックおよび時間的隣接ブロックの動きベクトルのみを含むAMVPモード候補リストを生成することができる。この場合、復号動き情報は、AMVPモード候補リスト中の動きベクトルのうちの1つを示すAMVPインデックスを含み得、現在のPUを予測するために選択された動きベクトルを改良するために、明示的にシグナリングされた参照インデックスおよび任意の動きベクトル差分をやはり含み得る。

10

#### 【0206】

[0223]従来、サブPU設計は、マージインター予測モードを使用したレイヤ間予測またはビュー間予測のためにのみ有効にされる。本開示は、動きベクトル改良が可能にされ得るシングルレイヤコーディングにおけるPUのサブPUを予測するための高度TMVPモードについて説明する。一例では、現在のPUを予測するための高度TMVPモードの性能を示すために、動き補償ユニット72は、PUのためのマージモード候補リスト中に高度TMVP候補を生成することができ、この場合、高度TMVP候補の選択は、PUを予測するための高度TMVPモードの性能を示す。

20

#### 【0207】

[0224]本開示の技法によれば、動き補償ユニット72は、現在のPUに対応する参照ピクチャのブロックを識別するPUの隣接ブロックから、現在のPUのための第1のステージの動きベクトルを決定する(160)。一例として、動き補償ユニット72は、現在のPUの空間的隣接ブロックおよび/または時間的隣接ブロックから第1のステージの動きベクトルを導出することができる。別の例では、動き補償ユニット72は、現在のPUのためのマージモード候補リストから第1のステージの動きベクトルを選択することができる。別の例では、第1のステージの動きベクトルは、一定の値またはあらかじめ定義された値になるように設定され得る。

30

#### 【0208】

[0225]動き補償ユニット72は、次いで、現在のPUを2つ以上のサブPUに区分する(162)。動き補償ユニット72は、第1のステージの動きベクトルによって識別された参照ピクチャのブロックからサブPUの各々のための第2のステージの動き情報を決定し、この場合、サブPUの各々のための第2のステージの動き情報は、少なくとも1つの動きベクトルと、関連付けられた参照インデックスとを含む(164)。場合によっては、サブPUの各々のための第2のステージの動き情報は、第1の参照ピクチャリストおよび第2の参照ピクチャリストの各々に対応する動きベクトルを含み得る。サブPUのうちの1つのための第2のステージの動き情報の動きベクトルのうちの少なくとも1つが利用可能でない場合、動き補償ユニット72は、サブPUのうちの1つのために、第1のステージの動きベクトルなど、代表的な動きベクトルを使用することができる。

40

#### 【0209】

[0226]いくつかの例では、単にマージモードで動作する代わりに、AMVPモードと同様の動きベクトル改良が有効にされ得る。たとえば、動き補償ユニット72は、高度TMVPモードを使用して予測される現在のPUのための動き改良データを決定することができる。1つの事例では、動き補償ユニット72は、サブPUのうちの1つまたは複数のための第2のステージの動き情報に適用可能な動きベクトル差分を決定することができる。上記で説明したように、動きベクトル差分は、符号化ビデオビットストリームに含まれた動き情報中でシグナリングされ得る。この例では、現在のPUを予測するための動き改良データを使用する高度TMVPモードの性能を示すために、動き補償ユニット72は、P

50



UのためのAMVPモード候補リスト中に高度TMVP候補を生成することができ、この場合、高度TMVP候補の選択は、PUを予測するための動き改良データを使用する高度TMVPモードの性能を示す。

【0210】

[0227]動き補償ユニット72は、サブPUの各々のための第2のステージの動き情報に基づいてサブPUの各々を別個に予測するために、動き補償を実行する(166)。動き補償の間、PUのサブPUの各々のための予測ブロックを生成した後、ビデオ復号器30は、対応する残差ブロックの再構築されたバージョンと、サブPUの各々のための予測ブロックとに基づいて、ビデオブロックの再構築されたバージョンを生成する。

【0211】

[0228]1つまたは複数の例では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は1つもしくは複数の命令またはコードとしてコンピュータ可読媒体上に記憶されるか、あるいはコンピュータ可読媒体を介して送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含み得る。このようにして、コンピュータ可読媒体は、一般に、(1)非一時的である有形のコンピュータ可読記憶媒体、または(2)信号もしくはキャリア波のような通信媒体に対応し得る。データ記憶媒体は、本開示で説明する技法の実装のための命令、コードおよび/あるいはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【0212】

[0229]限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM(登録商標)、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージ、もしくは他の磁気ストレージデバイス、フラッシュメモリ、または、命令もしくはデータ構造の形態の所望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る任意の他の媒体を備え得る。また、任意の接続が、コンピュータ可読媒体と適切に呼ばれる。たとえば、命令が、ウェブサイト、サーバ、または他の遠隔ソースから、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、マイクロ波などのワイヤレス技術は、媒体の定義に含まれる。ただし、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的媒体を含まないが、代わりに、非一時的な有形記憶媒体を対象とすることを理解されたい。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)、およびBlu-rayディスク(disc)を含み、ここで、ディスク(disk)は、通常、データを磁氣的に再生し、ディスク(disc)は、データをレーザーで光学的に再生する。上の組合せも、コンピュータ可読媒体の範囲の中に含まれるべきである。

【0213】

[0230]命令は、1つもしくは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、あるいは他の同等の集積回路またはディスクリート論理回路などの1つもしくは複数のプロセッサによって実行され得る。したがって、「プロセッサ」という用語は、本明細書で使用される時に、前述の構造のいずれかまたは本明細書で説明した技法の実装に適切な任意の他の構造を指すことができる。加えて、いくつかの態様では、本明細書で説明した機能は、符号化および復号のために構成されるか、または複合コーデック

10

20

30

40

50

に組み込まれる、専用のハードウェアモジュールおよび/もしくはソフトウェアモジュール内で提供され得る。また、本技法は、1つもしくは複数の回路または論理要素で十分に実装され得る。

【0214】

[0231]本開示の技法は、ワイヤレスハンドセット、集積回路(IC)もしくはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置で実装され得る。様々な構成要素、モジュール、またはユニットは、開示した技術を実行するように構成されたデバイスの機能的態様を強調するように本開示において説明されているが、異なるハードウェアユニットによる実現を必ずしも必要としない。そうではなく、上記で説明したように、様々なユニットは、コーデックハードウェアユニット中で組み合わせられるか、または上記で説明した1つもしくは複数のプロセッサを含む、適切なソフトウェアおよび/またはファームウェアとともに相互動作可能なハードウェアユニットの集合によって提供され得る。

10

【0215】

[0232]様々な例が、説明された。これらおよび他の例は、以下の特許請求の範囲に含まれる。

以下に、出願当初の特許請求の範囲に記載された発明を付記する。

【C1】

ビデオデータを処理する方法であって、

後方ビュー合成予測(BVSP)モードに従って、予測ユニット(PU)の隣接ブロックからビデオブロックのコーディングユニット(CU)の前記PUのための第1の動き情報を決定すること、  
ここにおいて、前記第1の動き情報は、少なくとも1つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

20

前記PUを2つ以上のサブPUに区分することと、

前記サブPUの各々のための第2の動き情報を決定すること、  
ここにおいて、前記第2の動き情報は、前記サブPUの各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも1つの視差動きベクトルを含む、と、

前記第2の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブPUの各々を予測するために、動き補償を実行することと、

30

後続PUを予測するために使用されることになる、前記PUの前記サブPUの各々のための前記第2の動き情報をメモリに記憶することと

を備える、方法。

【C2】

後続PUを予測するための動き情報のマージモード候補リストを生成することをさらに備え、  
前記PUは、前記後続PUの隣接ブロックであり、前記マージモード候補リストを生成することは、  
前記マージモード候補リスト中のサブPU BVSP候補として、前記メモリに記憶された前記PUの前記サブPUのうちの少なくとも1つのための前記第2の動き情報を含めることを備える、

C1に記載の方法。

40

【C3】

前記PUのための前記第1の動き情報を決定することは、前記PUを予測するための動き情報のマージモード候補リストからBVSP候補として前記第1の動き情報を選択することを備える、

C1に記載の方法。

【C4】

前記サブPUの各々のための前記第2の動き情報を決定することは、前記サブPUの各々に対応する前記ビュー間参照ピクチャの前記深度ブロックのための4つのコーナーピクセルの最大値に等しい水平成分と、ゼロに等しい垂直成分とを有するように、前記サブPUの各々のための前記少なくとも1つの視差動きベクトルを導出することを備える、

50

C 1 に記載の方法。

[ C 5 ]

前記サブ P U の各々のための前記第 2 の動き情報を決定することは、サブ P U を予測するための動き情報のマージモード候補リストからサブ P U B V S P 候補として前記第 2 の動き情報を選択することを備える、

C 1 に記載の方法。

[ C 6 ]

前記メモリは、復号ピクチャバッファを備え、前記第 2 の動き情報を記憶することは、前記 P U のための前記第 1 の動き情報の前記参照ビューインデックスによって識別された前記ビュー間参照ピクチャを含む参照ピクチャリストとともに、前記サブ P U の各々のための前記視差動きベクトルを前記復号ピクチャバッファに記憶することを備える、

C 1 に記載の方法。

[ C 7 ]

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記ビデオブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、残差ブロックを生成することと、

前記残差ブロックとビデオビットストリーム中の前記 P U のための少なくとも前記第 1 の動き情報のインジケータとを符号化することと

をさらに備える、C 1 に記載の方法。

[ C 8 ]

残差ブロックと受信されたビデオビットストリームからの前記 P U のための少なくとも前記第 1 の動き情報のインジケータとを復号することと、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記残差ブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、前記ビデオブロックの再構築されたバージョンを生成することと

をさらに備える、C 1 に記載の方法。

[ C 9 ]

前記ビデオブロックを復号ピクチャバッファに記憶するのに先立って、前記ビデオブロックの再構築されたバージョンの各 C U にデブロッキングフィルタを適用することをさらに備え、前記 2 つ以上のサブ P U を有する前記 P U を含む前記ビデオブロックの前記 C U に前記デブロッキングフィルタを適用することは、

前記 C U に前記デブロッキングフィルタを適用するのに先立って、前記 P U 中の 2 つの隣接するサブ P U 間のサブ P U 境界において人工的 P U 境界または人工的変換ユニット ( T U ) 境界を作成することと、

前記人工的 P U 境界または前記人工的 T U 境界を含む、前記 C U の 2 つの隣接する P U 間の P U 境界または前記 C U の 2 つの隣接する T U 間の T U 境界のうちの少なくとも 1 つに前記デブロッキングフィルタを適用することと

を備える、C 1 に記載の方法。

[ C 1 0 ]

前記人工的 P U 境界または前記人工的 T U 境界を作成することは、前記 P U を変換ツリー階層に関連付け、前記サブ P U の各々を T U に関連付けるために、前記 C U の変換ツリーを変換することを備え、前記サブ P U 境界は、人工的 T U 境界を備える、

C 9 に記載の方法。

[ C 1 1 ]

前記人工的 P U 境界または前記人工的 T U 境界を作成することは、前記 P U を変換ツリー階層に関連付け、前記サブ P U の各々の少なくとも一部を T U に関連付けるために、前記 C U の変換ツリーを変換することであり、ここにおいて、前記サブ P U 境界が人工的 T U を備える、を備える、C 9 に記載の方法。

[ C 1 2 ]

前記人工的 P U 境界または前記人工的 T U 境界を作成することは、前記 P U を C U に関連付け、前記サブ P U の各々を P U に関連付けるために、前記 C U をコーディングツリーに変換することを備え、前記サブ P U 境界は、人工的 P U 境界を備える、

C 9 に記載の方法。

[ C 1 3 ]

ビデオ処理デバイスであって、

ビデオデータを記憶するように構成されたメモリと、

前記メモリと通信する 1 つまたは複数のプロセッサと

を備え、前記 1 つまたは複数のプロセッサは、

後方ビュー合成予測 ( B V S P ) モードに従って、予測ユニット ( P U ) の隣接ブロックからビデオブロックのコーディングユニット ( C U ) の前記 P U のための第 1 の動き情報を決定すること、前記第 1 の動き情報は、少なくとも 1 つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記 P U を 2 つ以上のサブ P U に区分することと、

前記サブ P U の各々のための第 2 の動き情報を決定すること、ここにおいて、前記第 2 の動き情報は、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも 1 つの視差動きベクトルを含む、と、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブ P U の各々を予測するために、動き補償を実行することと

を行うように構成され、

前記メモリは、後続 P U を予測するために使用されることになる、前記 P U の前記サブ P U の各々のための前記第 2 の動き情報を記憶することを行うように構成される、

ビデオ処理デバイス。

[ C 1 4 ]

前記 1 つまたは複数のプロセッサは、後続 P U を予測するための動き情報のマージモード候補リストを生成することを行うように構成され、前記 P U は、前記後続 P U の隣接ブロックであり、前記マージモード候補リストは、前記マージモード候補リスト中のサブ P U B V S P 候補として、前記メモリに記憶された前記 P U の前記サブ P U のうちの少なくとも 1 つのための前記第 2 の動き情報を含む、

C 1 3 に記載のデバイス。

[ C 1 5 ]

前記 1 つまたは複数のプロセッサは、前記 P U を予測するための動き情報のマージモード候補リストから B V S P 候補として前記第 1 の動き情報を選択することを行うように構成される、

C 1 3 に記載のデバイス。

[ C 1 6 ]

前記 1 つまたは複数のプロセッサは、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの前記深度ブロックのための 4 つのコーナーピクセルの最大値に等しい水平成分と、ゼロに等しい垂直成分とを有するように、前記サブ P U の各々のための前記第 2 の動き情報に含まれた前記少なくとも 1 つの視差動きベクトルを導出することを行うように構成される、

C 1 3 に記載のデバイス。

[ C 1 7 ]

前記 1 つまたは複数のプロセッサは、サブ P U を予測するための動き情報のマージモード候補リストからサブ P U B V S P 候補として前記サブ P U の各々のための前記第 2 の動き情報を選択することを行うように構成される、

C 1 3 に記載のデバイス。

[ C 1 8 ]

前記メモリは、復号ピクチャバッファを備え、前記 1 つまたは複数のプロセッサは、前

10

20

30

40

50

記 P U のための前記第 1 の動き情報の前記参照ビューインデックスによって識別された前記ビュー間参照ピクチャを含む参照ピクチャリストとともに、前記サブ P U の各々のための前記第 2 の動き情報に含まれた前記視差動きベクトルを前記復号ピクチャバッファに記憶することを行うように構成される、

C 1 3 に記載のデバイス。

[ C 1 9 ]

前記ビデオ処理デバイスは、ビデオ符号化デバイスを備え、前記 1 つまたは複数のプロセッサは、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記ビデオブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、残差ブロックを生成することと、

前記残差ブロックとビデオビットストリーム中の前記 P U のための少なくとも前記第 1 の動き情報のインジケータとを符号化することと

を行うように構成される、C 1 3 に記載のデバイス。

[ C 2 0 ]

前記ビデオ処理デバイスは、ビデオ復号デバイスを備え、前記 1 つまたは複数のプロセッサは、

残差ブロックと受信されたビデオビットストリームからの前記 P U のための少なくとも前記第 1 の動き情報のインジケータとを復号することと、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記 P U の前記サブ P U の各々のための予測ブロックを生成することと、

前記残差ブロックと前記サブ P U の各々のための前記予測ブロックとに基づいて、前記ビデオブロックの再構築されたバージョンを生成することと

を行うように構成される、C 1 3 に記載のデバイス。

[ C 2 1 ]

前記 1 つまたは複数のプロセッサは、前記ビデオブロックを復号ピクチャバッファに記憶するのに先立って、前記ビデオブロックの再構築されたバージョンの各 C U にデブロッキングフィルタを適用することを行うように構成され、前記 2 つ以上のサブ P U を有する前記 P U を含む前記ビデオブロックの前記 C U に前記デブロッキングフィルタを適用するために、前記 1 つまたは複数のプロセッサは、

前記 C U に前記デブロッキングフィルタを適用するのに先立って、前記 P U 中の 2 つの隣接するサブ P U 間のサブ P U 境界において人工的 P U 境界または人工的変換ユニット ( T U ) 境界を作成することと、

前記人工的 P U 境界または前記人工的 T U 境界を含む、前記 C U の 2 つの隣接する P U 間の P U 境界または前記 C U の 2 つの隣接する T U 間の T U 境界のうちの少なくとも 1 つに前記デブロッキングフィルタを適用することと

を行うように構成される、C 1 3 に記載のデバイス。

[ C 2 2 ]

前記 1 つまたは複数のプロセッサは、前記 P U を変換ツリー階層に関連付け、前記サブ P U の各々を T U に関連付けるために、前記 C U の変換ツリーを変換することを行うように構成され、前記サブ P U 境界は、人工的 T U 境界を備える、

C 2 1 に記載のデバイス。

[ C 2 3 ]

前記 1 つまたは複数のプロセッサは、前記 P U を変換ツリー階層に関連付け、前記サブ P U の各々の少なくとも一部を T U に関連付けるために、前記 C U の変換ツリーを変換することを行うように構成され、前記サブ P U 境界は、人工的 T U を備える、

C 2 1 に記載のデバイス。

[ C 2 4 ]

前記 1 つまたは複数のプロセッサは、前記 P U を C U に関連付け、前記サブ P U の各々

10

20

30

40

50

を P U に関連付けるために、前記 C U をコーディングツリーに変換することを行うように構成され、前記サブ P U 境界は、人工的 P U 境界を備える、

C 2 1 に記載のデバイス。

[ C 2 5 ]

後方ビュー合成予測 ( B V S P ) モードに従って、予測ユニット ( P U ) の隣接ブロックからビデオブロックのコーディングユニット ( C U ) の前記 P U のための第 1 の動き情報を決定するための手段、ここにおいて、前記第 1 の動き情報は、少なくとも 1 つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、と、

前記 P U を 2 つ以上のサブ P U に区分するための手段と、

前記サブ P U の各々のための第 2 の動き情報を決定するための手段、ここにおいて、前記第 2 の動き情報が、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも 1 つの視差動きベクトルを含む、と、

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブ P U の各々を予測するために、動き補償を実行するための手段と、

後続 P U を予測するために使用されることになる、前記 P U の前記サブ P U の各々のための前記第 2 の動き情報をメモリに記憶するための手段と

を備える、ビデオ処理デバイス。

[ C 2 6 ]

ビデオデータを処理するための命令を記憶したコンピュータ可読記憶媒体であって、前記命令は、実行されると、1 つまたは複数のプロセッサに、

後方ビュー合成予測 ( B V S P ) モードに従って、予測ユニット ( P U ) の隣接ブロックからビデオブロックのコーディングユニット ( C U ) の前記 P U のための第 1 の動き情報を決定すること、ここにおいて、前記第 1 の動き情報は、少なくとも 1 つの視差動きベクトルと、ビュー間参照ピクチャを識別する、関連付けられた参照ビューインデックスとを含む、

前記 P U を 2 つ以上のサブ P U に区分することと、

前記サブ P U の各々のための第 2 の動き情報を決定すること、ここにおいて、前記第 2 の動き情報は、前記サブ P U の各々に対応する前記ビュー間参照ピクチャの深度ブロックに関連付けられた、少なくとも 1 つの視差動きベクトルを含む、と

前記第 2 の動き情報に基づいて、前記ビュー間参照ピクチャから前記サブ P U の各々を予測するために、動き補償を実行することと、

後続 P U を予測するために使用されることになる、前記 P U の前記サブ P U の各々のための前記第 2 の動き情報をメモリに記憶することと

を行わせる、コンピュータ可読記憶媒体。

10

20

30

【図 1】

図 1

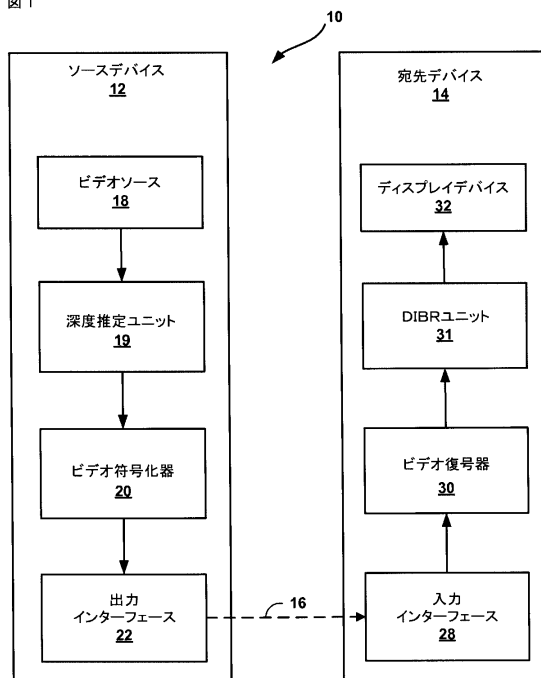


FIG. 1

【図 2】

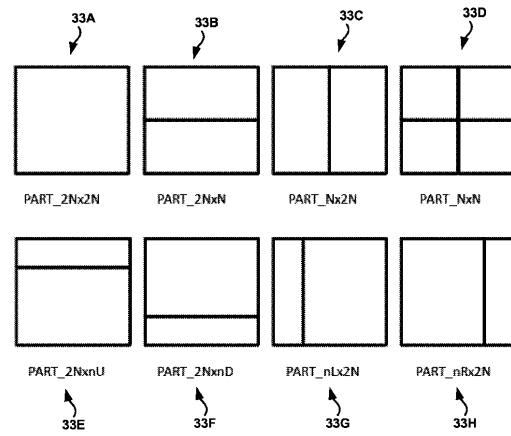


FIG. 2

【図 3】

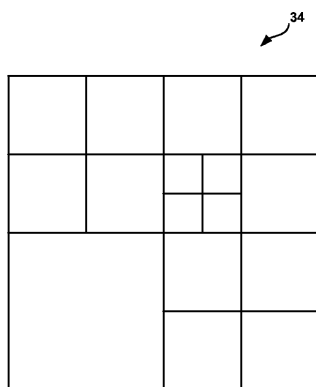


FIG. 3

【図 4 A】

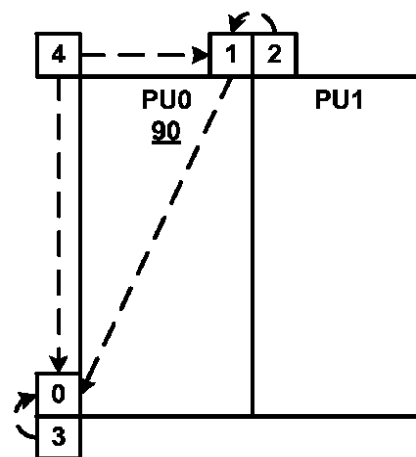


FIG. 4A

【図 4 B】

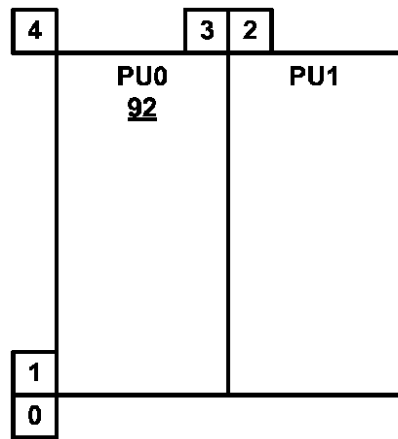


FIG. 4B

【図 5 A】

図 5A

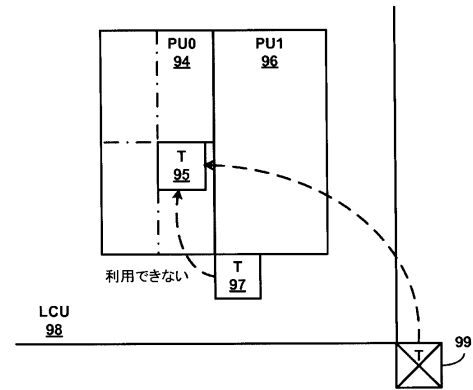


FIG. 5A

【図 5 B】

図 5B

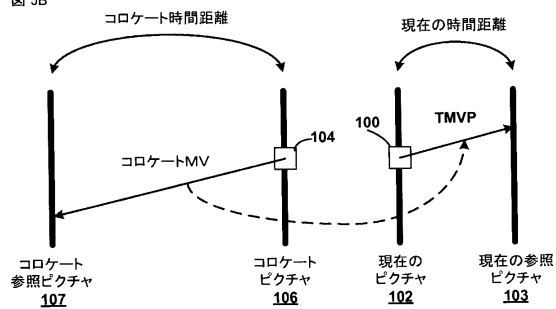


FIG. 5B

【図 6】

図 6

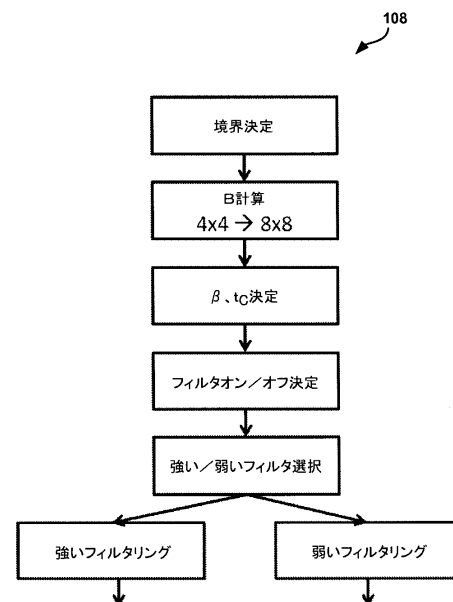


FIG. 6



【図 7】

図 7

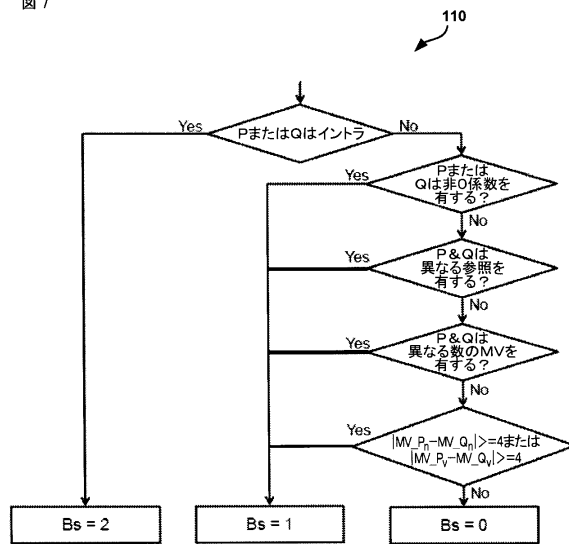


FIG. 7

【図 8】

図 8

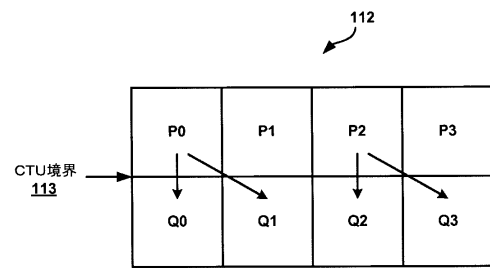


FIG. 8

【図 9】

図 9

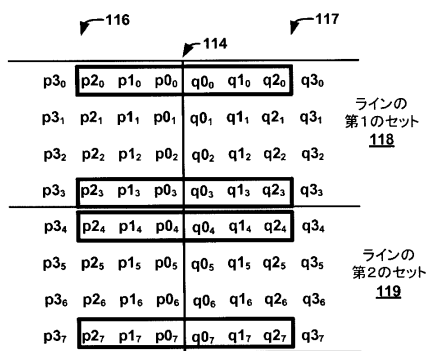


FIG. 9

【図 10】

120

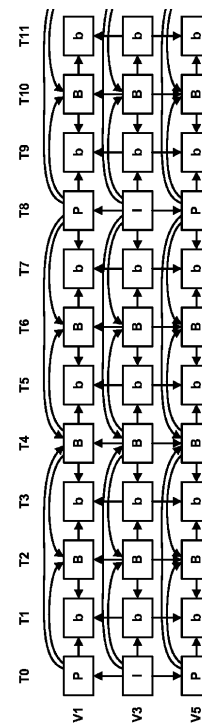


FIG. 10

【図 1 1】

図 11

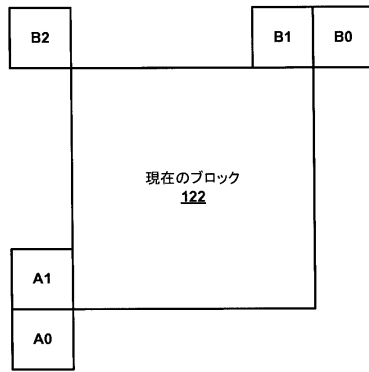


FIG. 11

【図 1 2】

図 12

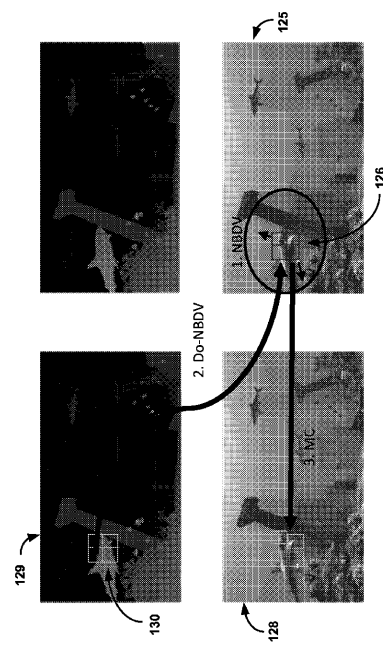


FIG. 12

【図 1 3】

図 13

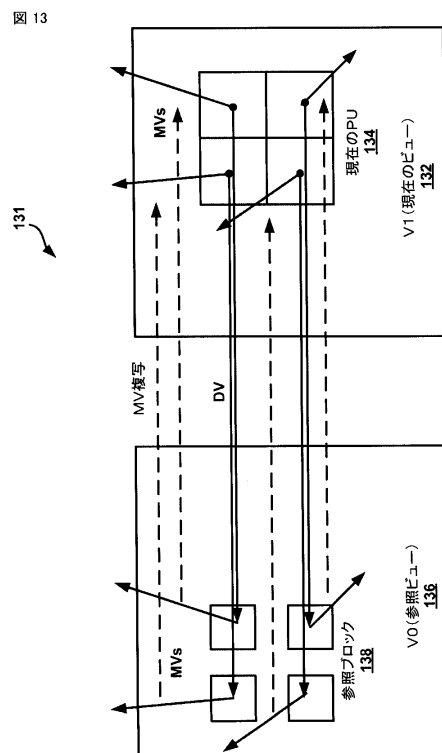


FIG. 13

【図 1 4 A】

図 14A

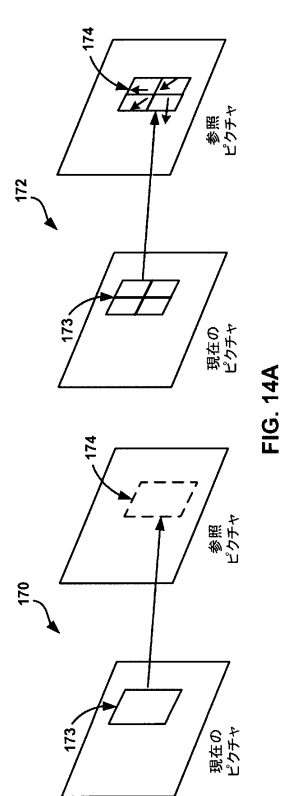


FIG. 14A

【図 14B】

図 14B

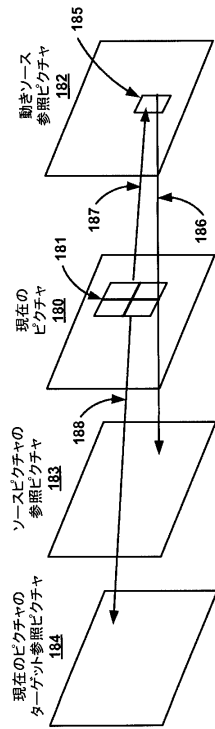


FIG. 14B

【図 15】

図 15

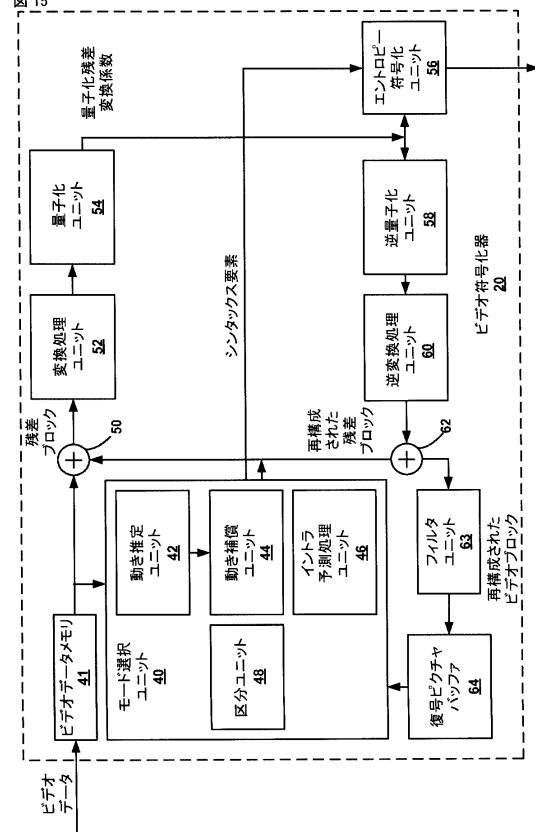


FIG. 15

【図 16】

図 16

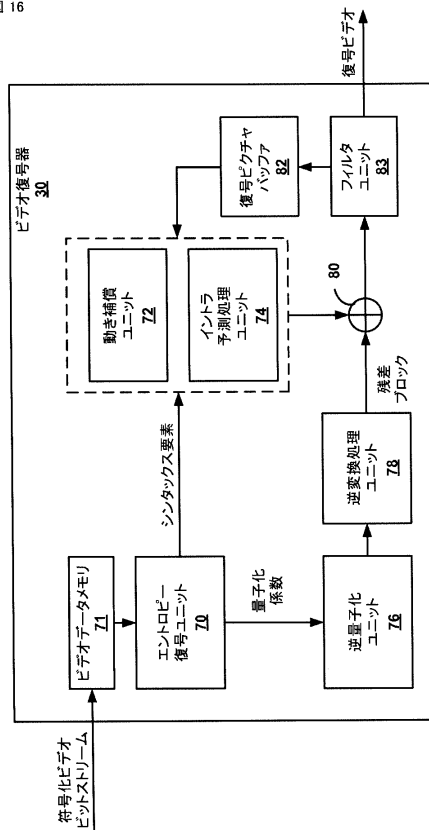


FIG. 16

【図 17】

図 17

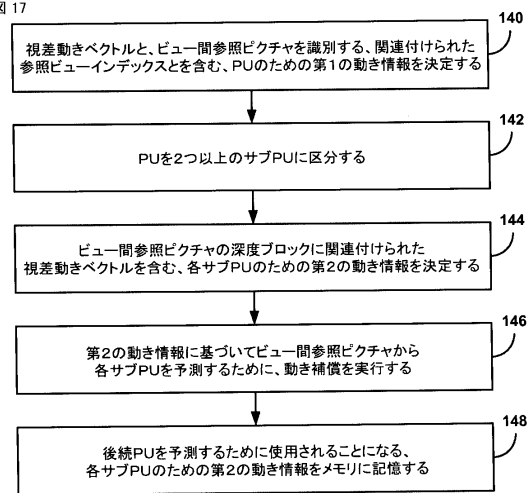


FIG. 17

【図 18】

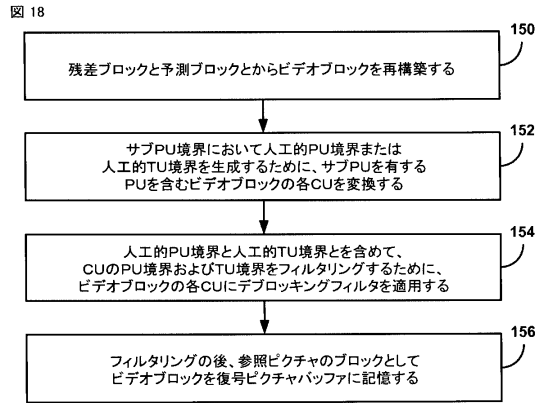


FIG. 18

【図 19】

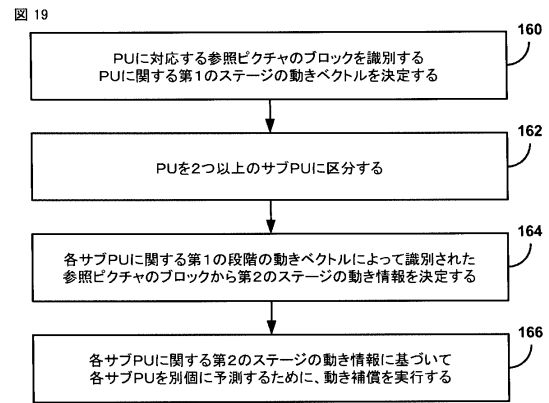


FIG. 19

## フロントページの続き

(51)Int.Cl. F I  
H 0 4 N 19/167 (2014.01) H 0 4 N 19/167

- (31)優先権主張番号 61/892,327  
(32)優先日 平成25年10月17日(2013.10.17)  
(33)優先権主張国 米国(US)  
(31)優先権主張番号 61/898,385  
(32)優先日 平成25年10月31日(2013.10.31)  
(33)優先権主張国 米国(US)  
(31)優先権主張番号 14/496,991  
(32)優先日 平成26年9月25日(2014.9.25)  
(33)優先権主張国 米国(US)

## 早期審査対象出願

- (72)発明者 チェン、イン  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5  
(72)発明者 ファン・デア・オーウェラ、ゲールト  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5  
(72)発明者 ジャン、リ  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5  
(72)発明者 リウ、ホンピン  
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

審査官 牛丸 太希

- (56)参考文献 D.Tian et al, CE1.h: Backward View Synthesis Prediction using Neighbouring Blocks, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 3rd Meeting: Geneva, CH, 17-23 Jan. 2013, 2013年 1月 21日, JCT3V-C0152  
J.An et al, 3D-CE3.h related: Sub-PU level inter-view motion prediction, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC 1/SC29/WG11 5th Meeting: Vienna, AT, 27 July - 2 Aug. 2013, 2013年 7月 27日, JCT3V-E0184  
Gerhard Tech et al., 3D-HEVC Draft Text 1, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 5th Meeting: Vienna, AT, 27 Jul. -2 Aug. 2013, 2013年 9月 11日, JCT3V-E1001-v3, 45 - 60 頁  
H.Liu et al, CE3 related: HEVC compatible de-blocking for sub-PU, Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 6th Meeting: Geneva, CH, 25 Oct. - 1 Nov. 2013, 2013年 10月 18日, JCT3V-F0135

- (58)調査した分野(Int.Cl., DB名)  
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8