

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6416115号
(P6416115)

(45) 発行日 平成30年10月31日(2018.10.31)

(24) 登録日 平成30年10月12日(2018.10.12)

(51) Int.Cl.		F I
HO 4 N 19/52	(2014.01)	HO 4 N 19/52
HO 4 N 19/597	(2014.01)	HO 4 N 19/597
HO 4 N 19/30	(2014.01)	HO 4 N 19/30
HO 4 N 19/70	(2014.01)	HO 4 N 19/70

請求項の数 15 (全 69 頁)

(21) 出願番号	特願2015-551771 (P2015-551771)	(73) 特許権者	595020643
(86) (22) 出願日	平成26年1月3日(2014.1.3)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2016-507969 (P2016-507969A)		QUALCOMM INCORPORATED
(43) 公表日	平成28年3月10日(2016.3.10)		ED
(86) 国際出願番号	PCT/US2014/010161		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開番号	W02014/107565		121-1714、サン・ディエゴ、モア
(87) 国際公開日	平成26年7月10日(2014.7.10)		ハウス・ドライブ 5775
審査請求日	平成28年12月5日(2016.12.5)	(74) 代理人	100108855
(31) 優先権主張番号	61/749, 213		弁理士 蔵田 昌俊
(32) 優先日	平成25年1月4日(2013.1.4)	(74) 代理人	100109830
(33) 優先権主張国	米国 (US)		弁理士 福原 淑弘
(31) 優先権主張番号	61/754, 864	(74) 代理人	100158805
(32) 優先日	平成25年1月21日(2013.1.21)		弁理士 井関 守三
(33) 優先権主張国	米国 (US)	(74) 代理人	100194814
			弁理士 奥村 元宏

最終頁に続く

(54) 【発明の名称】 ビュー間またはレイヤ間参照ピクチャについてのビットストリーム制約および動きベクトル制限

(57) 【特許請求の範囲】

【請求項1】

ビュー間またはレイヤ間ビデオ復号の方法であって、前記方法は、
候補動きベクトル予測子のリストを構成することと、
候補動きベクトル予測子の前記リストから動きベクトル予測子を選択するための、候補動きベクトル予測子の前記リストへのインデックスを含む情報と、現在のブロックについての動きベクトルを前記選択された動きベクトル予測子から導出するための動きベクトル差分に関する情報とを含むビットストリームを受信することと、
ここで、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャにおけるブロックを指し、ここで、前記動きベクトル差分に関する前記情報を用いて導出された前記現在のブロックについての前記動きベクトルが動きベクトル制限に準拠するように前記動きベクトル差分に関する前記情報を決定することによって、前記ビットストリームは制約され、ここで、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、
前記ビットストリーム中の前記動きベクトル差分に関する前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出することと、
前記現在のブロックについての前記導出された動きベクトルによって指される前記ブロックのピクセル値に基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号することを備える方法。

【請求項 2】

前記動きベクトル差分が前記動きベクトル予測子に加算されたときに、得られる動きベクトルが前記動きベクトル制限に準拠するように前記動きベクトル差分を、前記受信されたビットストリームから決定することをさらに備え、

前記動きベクトルを導出することは、前記動きベクトル予測子と前記動きベクトル差分とを加算することを備える、請求項 1 に記載の方法。

【請求項 3】

ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を、前記受信されたビットストリームから決定することと、

前記垂直動きベクトル範囲に基づいて、前記ビュー間またはレイヤ間参照ピクチャと前記現在のブロックを含む現在のピクチャとを並列復号することとをさらに備える、請求項 1 に記載の方法。

10

【請求項 4】

ビュー間またはレイヤ間ビデオ符号化の方法であって、

候補動きベクトル予測子のリストを構成することと、

候補動きベクトル予測子の前記リストから動きベクトル予測子を選択することと、

現在のブロックについての動きベクトルを前記動きベクトル予測子から導出するために使用される動きベクトル差分に関する情報を決定することと、ここにおいて、前記動きベクトル差分に関する前記情報を用いて導出された前記現在のブロックについての前記動きベクトルは、動きベクトル制限に準拠し、ここにおいて、前記現在のブロックについての前記導出された動きベクトルは、ビュー間またはレイヤ間参照ピクチャにおけるブロックを指し、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限し、前記動きベクトル差分に関する前記情報は、候補動きベクトル予測子の前記リストへのインデックスを含む、

20

前記動きベクトル差分に関する前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出することと、

前記現在のブロックについての前記導出された動きベクトルによって指される前記ブロックのピクセル値に基づいて、前記現在のブロックをビュー間またはレイヤ間予測符号化することと、

前記動きベクトルを導出するための前記動きベクトル差分に関する前記決定された情報

30

を出力用に生成することと

を備える方法。

【請求項 5】

前記動きベクトル差分に関する前記情報は、シンタックス要素を含む、請求項 1 または 4 に記載の方法。

【請求項 6】

前記動きベクトル差分に関する情報を決定することは、

前記動きベクトル差分が前記動きベクトル予測子に加算されたときに、得られる動きベクトルが前記動きベクトル制限に準拠するように、前記動きベクトル差分を決定することをさらに備える、請求項 4 に記載の方法。

40

【請求項 7】

前記動きベクトル差分に関する前記情報はフラグの値を含み、前記動きベクトル差分に関する前記情報を決定することは、前記フラグが示す値に基づいて、前記フラグの参照ピクチャリスト中のピクチャについての前記動きベクトル差分が 0 ではないことを決定することをさらに備える、請求項 4 に記載の方法。

【請求項 8】

前記動きベクトルが前記レイヤ間参照ピクチャを指すときに、前記動きベクトル制限は、前記導出された動きベクトルが 0 に等しくなるべきであることを示す、請求項 4 に記載の方法。

【請求項 9】

50

ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を出力用に生成すること

をさらに備える、請求項 4 に記載の方法。

【請求項 10】

ブロックサイズの前記ユニットは、最大コーディングユニットの高さおよび最小コーディングユニットの高さのユニットの 1 つを備える、請求項 9 に記載の方法。

【請求項 11】

前記ビュー間またはレイヤ間参照ピクチャは、第 1 のビュー間またはレイヤ間参照ピクチャを含み、前記方法は、

前記現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、前記第 1 のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを決定することと、

ここにおいて、前記動きベクトルを導出することは、前記現在のブロックについての前記動きベクトルを、前記動きベクトルが任意のビュー間またはレイヤ間参照ピクチャ中の前記利用不可能な参照エリアを指さないように、前記決定された利用不可能なエリアに基づいて導出することを備え、

前記利用不可能な参照エリアを示す情報を出力用に生成することと

をさらに備える、請求項 4 に記載の方法。

【請求項 12】

前記利用不可能な参照エリアを示す情報を出力用に生成することは、ラスト走査順序で前記利用不可能な参照エリアを示す情報を出力用に生成することを備える、請求項 11 に記載の方法。

【請求項 13】

実行されたときに、ビデオデータをコーディングするためのデバイスの 1 つまたは複数のプロセッサに、請求項 1 から 12 のいずれか一項に記載の方法を実行させる、命令を記憶したコンピュータ可読記憶媒体。

【請求項 14】

候補動きベクトル予測子のリストを構成するための手段と、

候補動きベクトル予測子の前記リストから動きベクトル予測子を選択するための手段と

、
現在のブロックについての動きベクトルを前記動きベクトル予測子から導出するために使用される動きベクトル差分に関する情報を決定するための手段と、ここにおいて、前記動きベクトル差分に関する前記情報を用いて導出された前記現在のブロックについての前記動きベクトルは、動きベクトル制限に準拠し、ここにおいて、前記現在のブロックについての前記導出された動きベクトルは、ビュー間またはレイヤ間参照ピクチャにおけるブロックを指し、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限し、前記動きベクトル差分に関する前記情報は、候補動きベクトル予測子の前記リストへのインデックスを含む

、
前記動きベクトル差分に関する前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出するための手段と、

前記現在のブロックについての前記導出された動きベクトルによって指される前記ブロックのピクセル値に基づいて、前記現在のブロックをビュー間またはレイヤ間予測符号化するための手段と、

前記動きベクトルを導出するための前記動きベクトル差分に関する前記決定された情報を出力用に生成するための手段と

を備えるデバイス。

【請求項 15】

候補動きベクトル予測子のリストを構成するための手段と、

候補動きベクトル予測子の前記リストから動きベクトル予測子を選択するための、候補

10

20

30

40

50

動きベクトル予測子の前記リストへのインデックスを含む情報と、現在のブロックについての動きベクトルを前記選択された動きベクトル予測子から導出するための動きベクトル差分に関する情報とを含むビットストリームを受信するための手段と、ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャにおけるブロックを指し、ここにおいて、前記動きベクトル差分に関する前記情報を用いて導出された前記現在のブロックについての前記動きベクトルが動きベクトル制限に準拠するように前記動きベクトル差分に関する前記情報を決定することによって、前記ビットストリームは制約され、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

前記ビットストリーム中の前記動きベクトル差分に関する前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出するための手段と、

前記現在のブロックについての前記導出された動きベクトルによって指される前記ブロックのピクセル値に基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号するための手段と

を備えるデバイス。

【発明の詳細な説明】

【技術分野】

【0001】

[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1月4日に提出された米国仮出願第61/749,213号、2013年1月21日に提出された米国仮出願第61/754,864号、2013年1月22日に提出された米国仮出願第61/755,274号、および2013年1月29日に提出された米国仮出願第61/758,206号の利益を主張する。

【0002】

[0002]本開示は、ビデオコーディングに関し、より詳細には、ビデオデータをコーディングするための技法に関する。

【背景技術】

【0003】

[0003]デジタルビデオ機能は、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)、ラップトップまたはデスクトップコンピュータ、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、ビデオゲームコンソール、携帯電話または衛星無線電話、ビデオ遠隔会議デバイスなどを含む、広範囲にわたるデバイスに組み込まれ得る。デジタルビデオデバイスは、デジタルビデオ情報をより効率的に送信、受信および記憶するための、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)によって定義された規格、現在開発中の高効率ビデオコーディング(HEVC: High Efficiency Video Coding)規格、およびそのような規格の拡張に記載されているビデオ圧縮技法など、ビデオ圧縮技法を実装する。

【0004】

[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予測および/または時間的予測を含む。ブロックベースのビデオコーディングでは、ビデオフレームまたはスライスがブロックに区分され得る。各ブロックはさらに区分され得る。イントラコーディングされた(I)フレームまたはスライス中のブロックは、同じフレームまたはスライス中の隣接ブロック中の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。インターコーディングされた(PまたはB)フレームまたはスライス中のブロックは、同じフレームまたはスライス中の隣接ブロック中の参照サンプルに対する空間的予測、あるいは他の参照フレーム中の参照サンプルに対する時間的予測を使用し得る。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についての予測ブロックが生じる。残差データは、コーディングされるべき元のブロックと予測

10

20

30

40

50

ブロックとの間のピクセル差分を表す。

【0005】

[0005]インターコーディングされたブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトル、およびコーディングされたブロックと予測ブロックとの間の差分を示す残差データに従って符号化される。イントラコーディングされたブロックは、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て残差変換係数をもたらすことができ、その残差変換係数が、次いで量子化され得る。最初は2次元アレイに構成される量子化変換係数は、エントロピーコーディングのための変換係数の1次元ベクトルを生成するために、特定の順序で走査され得る。

10

【発明の概要】

【0006】

[0006]一般に、本開示は、導出された動きベクトルが動きベクトル制限に違反しないことを確実にするビデオコーディング技法について説明する。たとえば、ビデオエンコーダは、ビデオデコーダが動きベクトルを導出する方法を定義するシンタックス要素をビットストリーム中でシグナリングすることができる。本技法は、動きベクトル制限に違反する動きベクトルをビデオデコーダに導出させることになるシンタックス要素値をビットストリームが含まないことを確実にし得る。

【0007】

[0007]本開示はまた、並列復号に有用であり得る復号遅延を示すための技法について説明する。たとえば、本技法は、ビデオデコーダによって並列復号に広く使用され得る復号遅延を示すことができる。復号遅延を示すためのそのような技法は、導出された動きベクトルが動きベクトル制限に違反しないことを確実にする技法とともに使用され得る。ただし、本開示で説明する技法は、そのように限定されない。

20

【0008】

[0008]一例では、本開示は、ビュー間またはレイヤ間ビデオ復号の方法について説明し、本方法は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信することを備え、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約され、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。本方法はまた、ビットストリーム中の情報に基づいて、現在のブロックについての動きベクトルを導出することと、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号することを含む。

30

【0009】

[0009]一例では、本開示は、ビュー間またはレイヤ間ビデオ符号化の方法について説明し、本方法は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するために使用される、導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定することを備え、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。本方法はまた、情報に基づいて、現在のブロックについての動きベクトルを導出することと、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測符号化することと、動きベクトルを導出するための決定された情報を出力用に生成することとを含む。

40

【0010】

[0010]一例では、本開示は、ビデオデコーダを備えるデバイスについて説明し、ビデオデコーダは、1つまたは複数のプロセッサを備え、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信するように構成され、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指

50

し、ここにおいて、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約され、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。ビデオデコーダはまた、ビットストリーム中の情報に基づいて、現在のブロックについての動きベクトルを導出し、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号するように構成される。

【 0 0 1 1 】

[0011]一例では、本開示は、ビデオエンコーダを備えるデバイスについて説明し、ビデオエンコーダは、1つまたは複数のプロセッサを備え、現在のブロックについての動きベクトルを動きベクトル予測子から導出するために使用される、導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定するように構成され、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。ビデオエンコーダはまた、情報に基づいて、現在のブロックについての動きベクトルを導出し、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測符号化し、動きベクトルを導出するための決定された情報を出力用に生成するように構成される。

【 0 0 1 2 】

[0012]一例では、本開示は、命令を記憶したコンピュータ可読記憶媒体について説明し、命令は実行されたときに、ビデオデータを復号するためのデバイスの1つまたは複数のプロセッサに、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信させ、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約され、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。命令はまた、1つまたは複数のプロセッサに、ビットストリーム中の情報に基づいて、現在のブロックについての動きベクトルを導出させ、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号させる。

【 0 0 1 3 】

[0013]一例では、本開示は、デバイスについて説明し、デバイスは、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信するための手段を備え、ここにおいて、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約され、ここにおいて、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。デバイスはまた、ビットストリーム中の情報に基づいて、現在のブロックについての動きベクトルを導出するための手段と、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号するための手段とを含む。

【 0 0 1 4 】

[0014]1つまたは複数の例の詳細が、添付の図面および以下の説明に記載されている。他の特徴、目的、および利点は、その説明および図面から、および特許請求の範囲から明らかになる。

【 図面の簡単な説明 】

【 0 0 1 5 】

【 図 1 】 [0015]本開示で説明する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 】 [0016]例示的なマルチビュービデオコーディング (M V C) 復号順序を示すグラフ図。

10

20

30

40

50

【図3】[0017]マルチビュービデオコーディング用の構造を示す概念図。

【図4】[0018]2つのビューのための並列コーディングプロセスの一例を示すフロー図。

【図5】[0019]本開示で説明する技法を実装し得る例示的なビデオエンコーダを示すブロック図。

【図6】[0020]本開示で説明する技法を実装し得る例示的なビデオデコーダを示すブロック図。

【図7】[0021]本開示で説明する技法によるビュー間またはレイヤ間ビデオ復号の例示的な動作を示すフローチャート。

【図8】[0022]本開示で説明する技法によるビュー間またはレイヤ間ビデオ符号化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】

【0016】

[0023]動きベクトル制限は、動きベクトルのx成分またはy成分のうちの少なくとも1つを値の範囲内に限定することを指す。このようにして、動きベクトル制限は、現在のブロックをインター予測するために使用され得る参照ピクチャのブロックを制限する。たとえば、y成分が一定の範囲内に制限される場合、y成分範囲の外にある参照ピクチャ中の任意のブロックは、現在のブロックをインター予測するために使用され得ない。一般に、動きベクトル制限は、より詳細に説明するように、ビュー間またはレイヤ間参照ピクチャであり得る参照ピクチャのサブ部分を指すように動きベクトルの範囲を限定するために使用され得る。

【0017】

[0024]動きベクトル制限が有用であり得る様々な理由があり得る。一例として、動きベクトル制限は、現在のブロックをインター予測するためにアクセスされる必要がある参照ピクチャのデータの量を減らすことができ、このことは、メモリ帯域幅効率を高める。たとえば、制限範囲外に存在するブロックのビデオデータは、現在のブロックをインター予測するためにアクセスされる必要はないことがある。

【0018】

[0025]別の例として、動きベクトル制限は、並列復号を可能にし得る。たとえば、マルチビュービデオコーディングまたはスケーラブルビデオコーディングでは、従属ビューまたはレイヤ中のピクチャが、参照ビューまたはレイヤ（たとえば、ベースビューまたはレイヤ）中のピクチャによりインター予測（たとえば、ビュー間予測またはレイヤ間予測）され得る。動きベクトル制限がない場合、ビデオデコーダは、従属ビューまたはレイヤのピクチャを再構成する前に、参照ビューまたはレイヤのピクチャ全体を再構成する必要がある。この理由で、従属ビューまたはレイヤのピクチャを再構成する前に参照ビューまたはレイヤのピクチャ全体を再構成する必要を回避するために、動きベクトル制限が望まれ得る。

【0019】

[0026]その上、動きベクトル制限がある場合、ビデオデコーダは、参照ビューまたはレイヤのピクチャと並列に、従属ビューまたはレイヤのピクチャを再構成することが可能であり得る。たとえば、動きベクトル制限が、従属ビューまたはレイヤのピクチャ中の任意のブロックについての動きベクトルのy成分が参照ビューまたはレイヤ（たとえば、ベースビューまたはレイヤ）中で T_h （ T_h は、y成分制限を定義する）を超えることができないと定義すると仮定する。言い換えれば、現在のブロックが、従属ビューまたはレイヤのピクチャ中の（ x, y ）に位置する場合、参照ビューまたはレイヤのピクチャ中の（ $x, y + T_h$ ）以上に位置するブロックのみが、現在のブロックをインター予測するために使用され得る。

【0020】

[0027]この例では、ビデオデコーダが参照ビューまたはレイヤのピクチャを $y + T_h$ ブロックまで再構成した後、ビデオデコーダは、参照ビューまたはレイヤのピクチャと並列に従属ビューまたはレイヤのピクチャを再構成することができる。この理由は、参照ビ

10

20

30

40

50

ユーまたはレイヤのピクチャ中の $y + Th$ ブロックの後のいかなるブロックも、「 y 」の y 成分値を有する従属ビューまたはレイヤのピクチャ中のブロックをインター予測するために使用され得ないことである。この意味で、動きベクトル制限は、並列復号遅延（すなわち、ビデオデコーダが従属ビューまたはレイヤのピクチャの再構成を開始し得る前に再構成されなければならない参照ビューまたはレイヤのピクチャの行の数）を定義することができる。いくつかの例では、参照ビューまたはレイヤのピクチャを再構成する際に使用されるフィルタ処理のタイプに基づいて、ビデオデコーダは、 $y + Th$ 行の後の行中のいくつかのブロックが再構成されるまで、従属ビューまたはレイヤのピクチャの再構成を開始することが可能ではないことがある。

【 0 0 2 1 】

[0028]しかしながら、動きベクトル制限を使用することに伴う問題があり得る。たとえば、多くの場合、ビデオエンコーダは、現在のブロックの実際の動きベクトルの情報（たとえば、シンタックス要素）をシグナリングしないことがある。代わりに、ビデオエンコーダは、ビデオデコーダが動きベクトルを導出する際の導出元となる情報（たとえば、シンタックス要素）をシグナリングすることがある。一例として、ビデオエンコーダは、現在のブロックの動きベクトルの情報をシグナリングするのではなく、マージ/スキップモードまたは高度動きベクトル予測（AMVP：advanced motion vector prediction）モードの一部として、候補動きベクトル予測子のリストへのインデックスをシグナリングし得る。ビデオデコーダは、受信されたインデックスに基づいて、候補動きベクトル予測子のリストから動きベクトル予測子を選択し、動きベクトル予測子に基づいて、現在のブロックについての動きベクトルを導出することができる。これらの例のうちのいくつかでは、動きベクトルを導出する際に、動きベクトル制限に違反する動きベクトルをビデオデコーダが導出することが考えられ得る。

【 0 0 2 2 】

[0029]たとえば、マージ/スキップモードでは、ビデオデコーダは、動きベクトル予測子の動きベクトルを採用する（すなわち、現在のブロックの動きベクトルを動きベクトル予測子に等しくセットする）。現在のブロックの動きベクトルが動きベクトル予測子に等しくセットされた場合に、それ、動きベクトルが現在のブロックの動きベクトル制限に準拠する保証がないことがある。

【 0 0 2 3 】

[0030]別の例として、AMVPモードでは、ビデオデコーダは、選択された動きベクトル予測子とベストマッチ探索から決定された動きベクトルとの間の動きベクトル差分（MVD）を受信する。ビデオデコーダは、現在のブロックについての動きベクトルを導出するために、選択された動きベクトル予測子にMVDを加算する。しかしながら、いくつかの例では、動きベクトル予測子にMVDを加算することによって、得られる導出された動きベクトルが動きベクトル制限に違反することが考えられ得る。

【 0 0 2 4 】

[0031]以下でより詳細に説明する時間的動きベクトル予測子（TMVP）の場合など、動きベクトル制限を使用するとき存在する他のそのような問題があり得る。以下でより詳細に説明するように、動きベクトル制限を使用するとき、特定の参照ピクチャリストを指す動きベクトル予測子についてのすべてのMVDが0であることを示すMVDフラグに伴う問題もあり得る。

【 0 0 2 5 】

[0032]ビデオデコーダが動きベクトルを導出するために使用するビットストリーム中の情報が、導出された動きベクトルが動きベクトル制限に違反しないことを確実にすることを、本開示で説明する技法は確実にし得る。一例として、マージ/スキップモードの場合、本技法は、現在のブロックについての動きベクトルとして採用された場合に、導出された動きベクトルに動きベクトル制限に違反させることになる動きベクトル予測子についての候補動きベクトル予測子のリストへのインデックスをビットストリームが含めないことを確実にし得る。別の例として、AMVPモードの場合、本技法は、動きベクトル予測子

10

20

30

40

50

に加算されたときに、得られる導出された動きベクトルに動きベクトル制限に準拠させるMVDをビットストリームが含むことを確実にし得る。

【0026】

[0033]並列復号の一部として動きベクトル制限を使用することに伴う問題または懸念もあり得る。いくつかの技法は、2つのマクロブロック行の固定値に並列復号遅延をセットすることができる。しかしながら、いくつかの最近のビデオコーディング規格は、マクロブロックに依存せず、代わりに、ツリーブロックの階層構造または最大コーディングユニット(LCU)もしくはコーディングツリーユニット(CTU)を使用する。各ツリーブロックは、4分木に従ってコーディングユニット(CU)に分割され得る。たとえば、4分木のルートノードとしてのツリーブロックは、4つの子ノードに分割され得、各子ノードは、次に、親ノードとなり、別の4つの子ノードに分割され得る。4分木のリーフノードとしての、最終的な、分割されていない子ノードは、コーディングノード(すなわち、コーディングされたビデオブロック)を備える。したがって、「マクロブロック」を使用する圧縮技法からの「行」の概念は、マクロブロックの代わりにツリーブロックを使用する技法にはうまく当てはまらないことがある。

10

【0027】

[0034]実際、そのような階層構造を利用するビデオコーディング規格には、マクロブロック行からブロック行への単純な拡張がないことがある。言い換えれば、ツリーブロック階層構造を使用するビデオコーディング技法には、2つのマクロブロック行の並列復号遅延の単純な方法がないことがある。

20

【0028】

[0035]動きベクトル制限をいくつかのマクロブロック行に固定することの限界を克服するために、いくつかの技法は、たとえば、LCU行のユニットにおける動きベクトル制限をシグナリングする。しかしながら、ビデオデコーダは、並列復号をいつ開始すべきか(たとえば、並列復号遅延)を決定するためにビデオデコーダが利用することができる値にLCU行のユニットを変換する必要が依然としてあり得る。

【0029】

[0036]一例として、ビデオデコーダは、LCUサイズに基づいて、並列復号をいつ開始すべきかを決定するためにビデオデコーダが利用することができる値にLCU行を変換する必要がある。しかしながら、あらゆるビデオデコーダが、並列復号のためにビデオデコーダが利用することができる値にLCU行を変換し得る統一的方法がないことがあり、結果的に、異なるビデオデコーダが異なる量の並列復号遅延を決定することになり得る。異なるビデオデコーダが異なる量の並列復号遅延を決定することは、復号の非効率、および場合によっては復号エラーをもたらし得る。

30

【0030】

[0037]いくつかの例では、本開示で説明する技法は、コーディングユニットサイズで並列復号遅延をシグナリングし得る。この形では、あらゆるビデオデコーダが並列復号遅延を決定することが可能であり得る一貫した方法があり得る。たとえば、ビデオデコーダは、並列復号がいつ開始できるかを決定するために利用され得る値に受信済み並列復号遅延情報をさらに変換する必要があることがある。

40

【0031】

[0038]図1は、本開示で説明する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されたビデオデータを生成するソースデバイス12を含む。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、ビデオゲームコンソール、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

50

デバイス 14 は、ワイヤレス通信に対応し得る。

【0032】

[0039]宛先デバイス 14 は、リンク 16 を介して、復号されるべき符号化されたビデオデータを受信し得る。リンク 16 は、ソースデバイス 12 から宛先デバイス 14 に符号化されたビデオデータを移す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ることができる。一例では、リンク 16 は、ソースデバイス 12 が、符号化されたビデオデータをリアルタイムに宛先デバイス 14 に直接送信できるようにする通信媒体を備えることができる。符号化されたビデオデータは、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、無線周波数 (RF) スペクトルまたは一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のような、パケットベース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 への通信を支援するために有用であり得るルータ、スイッチ、基地局、または任意の他の機器を含み得る。

10

【0033】

[0040]代替的に、符号化されたデータは、出力インターフェース 22 からストレージデバイス 34 に出力され得る。同様に、符号化されたデータは、入力インターフェースによってストレージデバイス 34 からアクセスされ得る。ストレージデバイス 34 は、ハードドライブ、Blu-ray (登録商標) 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メモリ、または符号化されたビデオデータを記憶するための任意の他の好適なデジタル記憶媒体など、様々な分散したまたはローカルでアクセスされるデータ記憶媒体のいずれかを含み得る。さらなる一例では、ストレージデバイス 34 は、ソースデバイス 12 によって生成された符号化されたビデオを保持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 34 から、記憶されたビデオデータにアクセスし得る。ファイルサーバは、符号化されたビデオデータを記憶し、その符号化されたビデオデータを宛先デバイス 14 に送信することが可能な任意のタイプのサーバであり得る。例示的なファイルサーバとしては、(たとえば、ウェブサイト用の) ウェブサーバ、FTPサーバ、ネットワーク接続ストレージ (NAS) デバイス、またはローカルディスクドライブがある。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的なデータ接続を通じて、符号化されたビデオデータにアクセスし得る。これは、ファイルサーバに記憶された符号化されたビデオデータにアクセスするのに適している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)、または両方の組合せを含むことができる。ストレージデバイス 34 からの符号化された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、または両方の組合せであり得る。

20

30

【0034】

[0041]本開示の技法は、必ずしもワイヤレス適用例または設定に限定されずとは限らない。本技法は、オーバージエアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、たとえばインターネットを介したストリーミングビデオ送信、データ記憶媒体に記憶するためのデジタルビデオの符号化、データ記憶媒体に記憶されたデジタルビデオの復号、または他の適用例など、様々なマルチメディア適用例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/またはビデオテレフォニーなどの適用例をサポートするために、一方向または双方向のビデオ送信をサポートするように構成され得る。

40

【0035】

[0042]図 1 の例では、ソースデバイス 12 は、ビデオソース 18 と、ビデオエンコーダ 20 と、出力インターフェース 22 とを含む。場合によっては、出力インターフェース 2

50

2 は変調器 / 復調器 (モデム) および / または送信機を含み得る。ソースデバイス 1 2 において、ビデオソース 1 8 は、たとえばビデオカメラなどのビデオキャプチャデバイス、以前にキャプチャされたビデオを含んでいるビデオアーカイブ、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、および / もしくはソースビデオとしてコンピュータグラフィックスデータを生成するためのコンピュータグラフィックスシステムなどのソース、またはそのようなソースの組合せを含み得る。一例として、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、いわゆるカメラ付き携帯電話またはビデオ電話を形成し得る。ただし、本開示で説明する技法は、概してビデオコーディングに適用可能であり得、ワイヤレスおよび / またはワイヤード適用例に適用され得る。

10

【 0 0 3 6 】

[0043] 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されたビデオは、ビデオエンコーダ 2 0 によって符号化され得る。符号化されたビデオデータは、ソースデバイス 1 2 の出力インターフェース 2 2 を介して宛先デバイス 1 4 に直接送信され得る。符号化されたビデオデータは、さらに (または代替的に)、復号および / または再生のための宛先デバイス 1 4 または他のデバイスによる後のアクセスのためにストレージデバイス 3 4 上に記憶され得る。

【 0 0 3 7 】

[0044] 宛先デバイス 1 4 は、入力インターフェース 2 8 と、ビデオデコーダ 3 0 と、ディスプレイデバイス 3 2 とを含む。場合によっては、入力インターフェース 2 8 は、受信機および / またはモデムを含み得る。宛先デバイス 1 4 の入力インターフェース 2 8 は、リンク 1 6 を介して、符号化されたビデオデータを受信する。リンク 1 6 を介して通信され、またはストレージデバイス 3 4 上に提供された符号化されたビデオデータは、ビデオデータを復号する際にビデオデコーダ 3 0 などのビデオデコーダが使用するための、ビデオエンコーダ 2 0 によって生成された様々なシンタックス要素を含み得る。そのようなシンタックス要素は、通信媒体上で送信され記憶媒体上に記憶される符号化されたビデオデータとともに含まれ得、またはファイルサーバを記憶した。本開示では、ビットストリーム中の「情報」という用語は、シンタックス要素を含むビットストリーム中の任意の情報を一般的に指すために使用される。

20

【 0 0 3 8 】

[0045] ディスプレイデバイス 3 2 は、宛先デバイス 1 4 と一体化されること、またはその外部に存在することがある。いくつかの例では、宛先デバイス 1 4 は、一体型ディスプレイデバイスを含むことができ、また、外部ディスプレイデバイスとインターフェースするように構成され得る。他の例では、宛先デバイス 1 4 はディスプレイデバイスであり得る。一般に、ディスプレイデバイス 3 2 は、復号されたビデオデータをユーザに対して表示し、液晶ディスプレイ (L C D)、プラズマディスプレイ、有機発光ダイオード (O L E D) 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。

30

【 0 0 3 9 】

[0046] ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、代替的に M P E G - 4、P a r t 1 0、アドバンストビデオコーディング (A V C) と呼ばれる I T U - T H . 2 6 4 規格などのビデオ圧縮規格、またはそのような規格の拡張に従って動作し得る。代替的に、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、高効率ビデオコーディング (H E V C) 規格などの他のプロプライエタリ規格または業界規格に従って動作し得、H E V C テストモデル (H M : H E V C T e s t M o d e l) に準拠し得る。ただし、本開示の技法は、いかなる特定のコーディング規格にも限定されない。ビデオ圧縮規格の他の例には、M P E G - 2 および I T U - T H . 2 6 3 がある。

40

【 0 0 4 0 】

[0047] 図 1 には示されていないが、いくつかの態様では、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、それぞれオーディオエンコーダおよびオーディオデコーダと統合

50

されてよく、共通のデータストリームまた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M U X - D E M U X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んでもよい。適用可能な場合、いくつかの例では、M U X - D E M U Xユニットは、I T U H . 2 2 3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(U D P)などの他のプロトコルに準拠し得る。

【 0 0 4 1 】

[0048]ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ(D S P)、特定用途向け集積回路(A S I C)、フィールドプログラマブルゲートアレイ(F P G A)、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組合せなどの様々な適切なエンコーダ回路のいずれかとして実装され得る。本技法が部分的にソフトウェアで実装されるとき、デバイスは、好適な非一時的コンピュータ可読媒体にソフトウェアの命令を記憶し、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ハードウェアでその命令を実行し得る。

10

【 0 0 4 2 】

[0049]ビデオエンコーダ20およびビデオデコーダ30の各々は1つまたは複数のエンコーダまたはデコーダ中に含まれてよく、そのいずれも、それぞれのデバイスにおいて複合エンコーダ/デコーダ(コーデック)の一部として統合されてよい。一例として、デバイスはビデオデコーダ30を含むことができ、デバイスの一例は集積回路(I C)であり、デバイスの別の例はマイクロプロセッサである。別の例として、デバイスはビデオデコーダ30を含むことができ、デバイスの一例はワイヤレス通信デバイスである。ビデオエンコーダ20は同様に、I C、マイクロプロセッサで、またはワイヤレス通信デバイスで形成され得る。

20

【 0 0 4 3 】

[0050] I T U - T Video Coding Experts Group (V C E G) と I S O / I E C Motion Picture Experts Group (M P E G) との Joint Collaboration Team on Video Coding (J C T - V C) によって開発された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(H E V C) がある。H E V C規格化の取り組みは、H E V Cテストモデル(H M) と呼ばれるビデオコーディングデバイスの発展的モデルに基づく。H Mは、たとえば、I T U - T H . 2 6 4 / A V C 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仮定する。たとえば、H . 2 6 4は9つのイントラ予測符号化モードを提供するが、H Mは33個ものイントラ予測符号化モードを提供し得る。

30

【 0 0 4 4 】

[0051]これ以降H E V C W D 9 と呼ばれる、H E V Cの最新のワーキングドラフト(W D) が、http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zipから入手可能である。「H E V C Working Draft 10」または「W D 10」と呼ばれるH E V C規格の別の最近のドラフトは、文書J C T V C - L 1 0 0 3 v 3 4、Brossら、「High efficiency video coding (H E V C) text specification draft 10 (for FDIS & Last Call)」、I T U - T S G 1 6 W P 3 と I S O / I E C J T C 1 / S C 2 9 / W G 1 1 との Joint Collaborative Team on Video Coding (J C T - V C)、第12回会合：ジュネーブ、スイス、2013年1月14～23日に記載されており、この文書は、http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zipからダウンロード可能である。H E V C規格のさらに別のドラフトは、本明細書で「W D 10改訂」と呼ばれ、Brossら、「Editors' p

40

50

roposed corrections to HEVC version 1」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのJoint Collaborative Team on Video Coding (JCT-VC)、第13回会合、仁川、韓国、2013年4月に記載されており、この文書は、http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zipから入手可能である。HEVC WD9、WD10、およびWD10改訂の内容全体が参照により本明細書に組み込まれる。

【0045】

[0052]概して、HMのワーキングモデルは、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一連のツリーブロックまたは最大コーディングユニット(LCU)に分割され得ることを記載している。ツリーブロックは、H.264規格のマクロブロックと同様の目的を有し得る。スライスは、コーディング順序でいくつかの連続するツリーブロックを含む。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得る。各ツリーブロックは、4分木に従ってコーディングユニット(CU)に分割され得る。たとえば、4分木のルートノードとしてのツリーブロックは、4つの子ノードに分割され得、各子ノードは、次に、親ノードとなり、別の4つの子ノードに分割され得る。4分木のリーフノードとしての、最終的な、分割されていない子ノードは、コーディングノード(すなわち、コーディングされたビデオブロック)を備える。コーディングされたビットストリームに関連するシンタックスデータは、ツリーブロックが分割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。

【0046】

[0053]CUは、コーディングノードと、コーディングノードに関連する予測ユニット(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、8×8ピクセルから最大64×64以上のピクセルをもつ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とを含み得る。CUに関連するシンタックスデータは、たとえば、CUを1つまたは複数のPUに区分することを記述し得る。区分モードは、CUが、スキップモード符号化もしくはダイレクトモード符号化されるか、イントラ予測モード符号化されるか、またはインター予測モード符号化されるかによって異なり得る。PUは、形状が非正方形になるように区分され得る。CUに関連するシンタックスデータは、たとえば、4分木に従って、CUを1つまたは複数のTUに区分することも記述し得る。TUは、形状が正方形または非正方形であり得る。

【0047】

[0054]HEVC規格は、CUごとに異なり得るTUに従う変換を可能にする。TUは、一般に、区分されたLCUについて定義された所与のCU内のPUのサイズに基づいてサイズ決定されるが、常にそうであるとは限らない。TUは、通常、PUと同じサイズであるか、またはPUよりも小さい。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: residual quad tree)として知られる4分木構造を使用して、より小さいユニットに再分割され得る。RQTのリーフノードは変換ユニット(TU)と呼ばれることがある。TUに関連するピクセル差分値は、変換係数を生成するように変換されてよく、その変換係数は量子化され得る。

【0048】

[0055]一般に、PUは、予測プロセスに係するデータを含む。たとえば、PUがイントラモード符号化(イントラ予測)される時、PUは、PUについて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(インター予測)される時、PUは、PUについての動きベクトルを定義するデータを含み得る。PUについて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

トルの参照ピクチャリスト（たとえば、リスト0（L0）もしくはリスト1（L1））を記述し得る。リスト0およびリスト1は、それぞれRefPicList0およびRefPicList1と呼ばれることがある。

【0049】

[0056]概して、TUは、変換プロセスと量子化プロセスと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つまたは複数の変換ユニット（TU）を含む場合もある。予測の後に、ビデオエンコーダ20は、PUに対応する残差値を計算し得る。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、走査され得るピクセル差分値を備える。本開示は、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とTUとを含む、ツリーブロック、すなわち、LCUまたはCUを指すために「ビデオブロック」という用語を使用する場合もある。

10

【0050】

[0057]たとえば、現在開発中の高効率ビデオコーディング（HEVC）規格に従うビデオコーディングでは、（ピクチャとも呼ばれる）ビデオフレームがコーディングユニット（CU）と予測ユニット（PU）と変換ユニット（TU）とに区分され得る。CUは、概して、ビデオ圧縮のために様々なコーディングツールが適用される基本ユニットとして働く画像領域を指す。CUは、一般に正方形の形状を有し、たとえば、ITU-T H.264などの他のビデオコーディング規格の下でのいわゆる「マクロブロック」と同様であると見なされ得る。

20

【0051】

[0058]より良いコーディング効率を達成するために、CUは、それが含んでいるビデオデータに応じて可変サイズを有し得る。すなわち、CUは、より小さいブロックまたはサブCUに区分または「分割」され得、その各々はCUと呼ばれることもある。さらに、サブCUに分割されない各CUは、それぞれ、CUの予測および変換のために1つまたは複数のPUとTUとにさらに区分され得る。

【0052】

[0059]PUは、H.264などの他のビデオコーディング規格の下でのいわゆるブロックのパーティションと同様であると見なされ得る。PUは、「残差」係数を生成するためにブロックについての予測が実行されるベースである。CUの残差係数は、CUのビデオデータと、CUの1つまたは複数のPUを使用して決定されたCUについての予測データとの間の差を表す。詳細には、1つまたは複数のPUは、CUが予測のためにどのように区分されるかを指定し、CUの各パーティション内に含まれているビデオデータを予測するためにどの予測モードが使用されるかを指定する。

30

【0053】

[0060]CUの1つまたは複数のTUは、CUのための残差変換係数のブロックを生成するために、ブロックにどの変換が適用されるかに基づいて、CUの残差係数のブロックのパーティションを指定する。1つまたは複数のTUはまた、適用される変換のタイプに関連し得る。変換は、残差係数をピクセル領域または空間領域から周波数領域などの変換領域に変換する。さらに、1つまたは複数のTUは、量子化残差変換係数のブロックを生成するために残差変換係数の得られたブロックにどの量子化が適用されるかに基づいてパラメータを指定し得る。残差変換係数は、場合によっては、係数を表すために使用されるデータの量を低減するために量子化され得る。

40

【0054】

[0061]CUは、一般に、Yとして示される1つのルミナンス成分とUおよびVとして示される2つのクロミナンス成分とを含む。言い換えれば、サブCUにさらに分割されない所与のCUは、Y成分とU成分とV成分とを含み得、その各々は、前に説明したように、CUの予測および変換のために1つまたは複数のPUとTUとにさらに区分され得る。たとえば、ビデオサンプリングフォーマットに応じて、サンプルの数で表されるU成分およ

50

びV成分のサイズは、Y成分のサイズと同じであるかまたはそれとは異なり得る。したがって、予測、変換、および量子化に関して上記で説明した技法は、所与のCUのY成分、U成分およびV成分の各々について実行され得る。

【0055】

[0062]CUを符号化するために、CUの1つまたは複数のPUに基づいて、CUのための1つまたは複数の予測子が最初に導出される。予測子は、CUについての予測データを含んでいる参照ブロックであり、前に説明したように、CUのための対応するPUに基づいて導出される。たとえば、PUは、予測データが決定される際の対象となるCUのパーティションと、予測データを決定するために使用される予測モードとを示す。予測子は、イントラ(I)予測(すなわち、空間的予測)モードまたはインター(PまたはB)予測(すなわち、時間的予測)モードのいずれかを通して導出され得る。したがって、いくつかのCUは、同じフレーム中の隣接参照ブロックまたはCUに対する空間的予測を使用してイントラコーディング(I)され得るが、他のCUは、他のフレーム中の参照ブロックまたはCUに対してインターコーディング(PまたはB)され得る。

10

【0056】

[0063]CUの1つまたは複数のPUに基づいて1つまたは複数の予測子を識別するときに、1つまたは複数のPUに対応するCUの元のビデオデータと1つまたは複数の予測子中に含まれているCUについての予測データとの間の差が計算される。予測残差とも呼ばれるこの差は、残差係数を備え、前に説明したように、1つまたは複数のPUと1つまたは複数の予測子とによって指定されたCUの部分間のピクセル差分を指す。残差係数は、概して、1つまたは複数のPU \circ CUに対応する2次元(2D)アレイに構成される。

20

【0057】

[0064]さらなる圧縮を達成するために、予測残差は、概して、たとえば、離散コサイン変換(DCT)、整数変換、カルーネンレーベ(Karhunen-Loeve)(K-L)変換、または別の変換を使用して変換される。変換は、同じく前に説明したように、空間領域中の予測残差、すなわち、残差係数を変換領域、たとえば、周波数領域中の残差変換係数に変換する。変換係数はまた、概して、CUの1つまたは複数のTUに対応する2Dアレイに構成される。さらなる圧縮のために、残差変換係数は、同じく前に説明したように、場合によっては、係数を表すために使用されるデータの量を低減するために量子化され得る。

【0058】

30

[0065]またさらなる圧縮を達成するために、エントロピーコーダは、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(CBAC)、確率間隔区分エントロピーコーディング(PIPE: Probability Interval Partitioning Entropy Coding)、または別のエントロピーコーディング方法を使用して、得られた残差変換係数を後で符号化する。エントロピーコーディングは、他のCUと比較して、係数によって表される、CUのビデオデータに固有の統計的冗長性を低減または除去することによって、このさらなる圧縮を達成し得る。

【0059】

[0066]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つまたは複数を用意する。GOPは、GOP中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、または他の場所を含み得る。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み得る。ビデオエンコーダ20は、通常、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対して動作する。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、固定サイズまたは可変サイズを有し得、指定のコーディング規格に応じてサイズが異なり得る。

40

【0060】

50

[0067]一例として、HM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、HMは、 $2N \times 2N$ または $N \times N$ のPUサイズでのイントラ予測をサポートし、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称的なPUサイズでのインター予測をサポートする。HMはまた、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、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サポートする。非対称区分では、CUの一方向は区分されないが、他の方向は25%と75%とに区分される。25%の区分に対応するCUの部分は、「n」とその後ろに付く「Up」、「Down」、「Left」、または「Right」という表示によって示される。したがって、たとえば、「 $2N \times nU$ 」は、上部の $2N \times 0.5N$ PUと下部の $2N \times 1.5N$ PUとで水平方向に区分された $2N \times 2N$ CUを指す。

10

【0061】

[0068]本開示では、「 $N \times N (NxN)$ 」および「 $N \times N (N \text{ by } N)$ 」は、垂直寸法および水平寸法に関する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。一般的に、 16×16 ブロックは、垂直方向に16個のピクセルを有し($y = 16$)、水平方向に16個のピクセルを有する($x = 16$)。同様に、 $N \times N$ ブロックは、概して、垂直方向にN個のピクセルを有し、水平方向にN個のピクセルを有し、ここでNは非負整数値を表す。ブロック中のピクセルは行と列とに構成され得る。その上、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必要はない。たとえば、ブロックは $N \times M$ ピクセルを備えてよく、この場合に、Mは必ずしもNに等しいとは限らない。

20

【0062】

[0069]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の後に、ビデオエンコーダ20は、CUのTUのための残差データを計算し得る。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得、TUは、変換、たとえば、残差ビデオデータへの離散コサイン変換(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後に、変換領域において係数を備え得る。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。ビデオエンコーダ20は、CUのための残差データを含むTUを形成し、次いで、CU用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。

30

【0063】

[0070]変換係数を生成するための任意の変換の後に、ビデオエンコーダ20は、変換係数の量子化を実施し得る。量子化は、概して、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、さらなる圧縮を行うプロセスを指す。量子化プロセスは、係数の一部または全部に関連するビット深度を低減し得る。たとえば、量子化中にnビット値がmビット値に切り捨てられてよく、ここでnはmよりも大きい。

【0064】

[0071]いくつかの例では、ビデオエンコーダ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、量子化変換係数を走査するためにあらかじめ定義された走査順序を利用し得る。他の例では、ビデオエンコーダ20は適応走査を実行し得る。量子化変換係数を走査して1次元ベクトルを形成した後に、ビデオエンコーダ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)、コンテキスト適応型バイナリ算術コーディング(CABAC)、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(SBAC: syntax-based context-adaptive binary arithmetic coding)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピー符号化方法に従って1次元ベクトルをエントロピー符号化し得る。ビデオエンコーダ20はまた、ビデオデータを復号する際にビデオデコーダ30が使用するための符号化されたビデオデータ(たとえば、符号化されたビデオデータに関連する情報)に関連するシンタ

40

50

ックス要素をエントロピー符号化し得る。

【0065】

[0072] C A B A C を実行するために、ビデオエンコーダ 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得る。コンテキストは、たとえば、シンボルの隣接値が非 0 であるか否かに関係し得る。C A V L C を実行するために、ビデオエンコーダ 20 は、送信されるべきシンボルの可変長コードを選択し得る。V L C におけるコードワードは、比較的短いコードが優勢シンボルに対応し、より長いコードが劣勢シンボルに対応するように構成され得る。このようにして、V L C を使用すると、たとえば、送信されるべき各シンボルのために等長コードワードを使用するよりも、ビット節約を実現し得る。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くことができる。

10

【0066】

[0073] ビデオエンコーダ 20 およびビデオデコーダ 30 は、本開示で説明する技法を実装するように構成され得る。ビデオエンコーダ 20 およびビデオデコーダ 30 は、一般にビデオコードと呼ばれることがある。本開示の例では、ビデオコードは、動きベクトル範囲および行 / 列遅延のうちの少なくとも 1 つを決定することができ、決定に基づいてブロックをインター予測することができる。いくつかの例では、ビデオコードは、1 つまたは複数の制約に基づいて、異なるビュー中のピクチャおよび異なるレイヤ中のピクチャのうちの少なくとも 1 つにアクセスすることができ、アクセスされたピクチャに基づいて、ブロックをビュー間またはレイヤ間予測することができる。

20

【0067】

[0074] たとえば、上記は H E V C 規格に関する技法について説明した。本開示は、高効率ビデオコーディング (H E V C) 規格と、S H V C と呼ばれることがあるスケーラブルビデオコーディング (S V C) ならびに M V - H E V C および 3 D V - H E V C と呼ばれることがあるマルチビュービデオコーディング (M V C 、 3 D V) のような H E V C の拡張とに関係し得る。ただし、本開示で説明する技法は、任意の特定の規格に限定されると見なされるべきではなく、任意の規格または任意の拡張の説明は、単に例示のために提供されている。

【0068】

[0075] ビデオコーディング規格は、I T U - T H . 2 6 1 、 I S O / I E C M P E G - 1 V i s u a l 、 I T U - T H . 2 6 2 または I S O / I E C M P E G - 2 V i s u a l 、 I T U - T H . 2 6 3 、 I S O / I E C M P E G - 4 V i s u a l 、 およびそのスケーラブルビデオコーディング (S V C) 拡張とマルチビュービデオコーディング (M V C) 拡張とを含む (I S O / I E C M P E G - 4 A V C としても知られる) I T U - T H . 2 6 4 を含む。マルチビュービデオコーディング (M V C) は H . 2 6 4 / A V C の拡張である。M V C 拡張の最近の公的に入手可能な共同ドラフトは、「Advanced video coding for generic audio visual services」、I T U - T 勧告 H . 2 6 4 、 2 0 1 0 年 3 月に記載されている。M V C 拡張のさらに最近の公的に入手可能な共同ドラフトは、「Advanced video coding for generic audiovisual services」、I T U - T 勧告 H . 2 6 4 、 2 0 1 1 年 6 月に記載されている。M V C 拡張の現在の共同ドラフトは、2 0 1 2 年 1 月時点で承認されている。

30

40

【0069】

[0076] 本開示では、「マルチビュービデオコーディング」という句は、完全に書かれたときには、一般的に、マルチビューが存在するビデオコーディングを指す。頭文字 M V C は、より具体的に、H . 2 6 4 / A V C の M V C 拡張または H E V C の M V C 拡張 (H E V C ベースの M V C または M V - H E V C) を指すために使用され得る。他のマルチビュービデオコーディング技法が、H E V C の 3 D V 拡張 (H E V C ベースの 3 D V または 3 D V - H E V C) において説明されている。

【0070】

50

[0077]本開示で説明する技法は、マルチビュービデオコーディングならびにスケーラブルビデオコーディングに適用可能である。たとえば、H E V C規格が最近確定されており、H E V C規格の拡張ならびにH E V Cに基づく規格についての議論が進行中である。マルチビュービデオコーディングの場合、H E V C規格の拡張は、M V - H E V C（深度のないマルチビュー）と3 D - H E V C（深度のあるマルチビュー）とを含む。また、3 D V規格はH E V C規格に基づく。スケーラブルビデオコーディングの場合、H E V C規格の拡張はS H V Cを含む。本開示で説明する技法は、これらの規格のすべてに適用可能であり、概して、規格に基づくかどうかにかかわらず、ビデオコーディングのための任意の技法に適用可能である。

【 0 0 7 1 】

[0078]マルチビュービデオコーディングでは、複数のピクチャをそれぞれ含む複数の異なるビューが存在する。ビューのうちの一つはベースビューと呼ばれ、他のビューは従属ビューと呼ばれる。ビデオエンコーダ20またはビデオデコーダ30は、ベースビュー中のピクチャを、従属ビュー中のピクチャをインター予測するために利用することがあるが、ビデオエンコーダ20またはビデオデコーダ30は、従属ビュー中のピクチャを、ベースビュー中のピクチャをインター予測するために利用することはない。ビデオエンコーダ20またはビデオデコーダ30は、従属ビュー中のピクチャを、別の従属ビュー中のピクチャをインター予測するために利用することがある。言い換えれば、ビデオエンコーダ20またはビデオデコーダ30は、ベースビュー中のピクチャを別のビュー中のピクチャによりインター予測することはないが、従属ビュー中のピクチャを他のビュー（すなわち、

【 0 0 7 2 】

[0079]ビデオエンコーダ20またはビデオデコーダ30がベースビュー中のピクチャをベースビュー中の別のピクチャによりインター予測することがあり、同様に、ビデオエンコーダ20またはビデオデコーダ30が従属ビュー中のピクチャを同じ従属ビュー中の別のピクチャによりインター予測することがあることを理解されたい。これらの場合、インター予測に使用されるピクチャは、インター予測されているピクチャと同じビュー中にあるので、インター予測に使用されるピクチャは、ビュー間参照ピクチャではない。

【 0 0 7 3 】

[0080]スケーラブルビューコーディングでは、複数のレイヤが存在する。レイヤのうちの一つはベースレイヤと呼ばれ、他のレイヤはエンハンスメントレイヤと呼ばれる。マルチビュービデオコーディングと同様に、ビデオエンコーダ20またはビデオデコーダ30は、ベースレイヤ中のピクチャを、エンハンスメントレイヤ中のピクチャをインター予測するために利用することがあるが、ビデオエンコーダ20またはビデオデコーダ30は、エンハンスメントレイヤ中のピクチャを、ベースレイヤ中のピクチャをインター予測するために利用することはない。ビデオエンコーダ20またはビデオデコーダ30は、インター予測されているピクチャがインター予測に使用されるピクチャよりも上位のレイヤにある限り、エンハンスメントレイヤ中のピクチャを、別のエンハンスメントレイヤ中のピクチャをインターピクチャするために利用することがある。言い換えれば、スケーラブルビデオコーディングは、上位レイヤからのピクチャが下位レイヤからのピクチャによりインター予測され得るが、下位レイヤからのピクチャが上位レイヤからのピクチャによりインター予測され得ない階層コーディング手法と見なされ得る。しかしながら、場合によっては、下位レイヤからのピクチャは上位レイヤからのピクチャによりインター予測され得る。ベースレイヤは最下位レイヤであり、したがって、上位レイヤ中のピクチャがベースレイヤ中のピクチャをインター予測するために使用されることはない。ピクチャが別のレイヤ中のピクチャによりインター予測されるとき、他のレイヤ中のピクチャは、レイヤ間参照ピクチャと呼ばれる。

【 0 0 7 4 】

[0081]マルチビュービデオコーディングと同様に、ビデオエンコーダ20またはビデオデコーダ30がベースレイヤ中のピクチャをベースレイヤ中の別のピクチャによりインター予測することがあり、同様に、ビデオエンコーダ20またはビデオデコーダ30がエンハンスメントレイヤ中のピクチャを同じエンハンスメントレイヤ中の別のピクチャによりインター予測することがあることを理解されたい。これらの場合、インター予測に使用されるピクチャは、インター予測されているピクチャと同じレイヤ中にあるので、インター予測に使用されるピクチャは、レイヤ間参照ピクチャではない。

【0075】

[0082]図2は、例示的なマルチビュービデオコーディング(MVC)復号順序を示すグラフ図である。図2は、MVCビットストリーム構造の例を示す。たとえば、典型的なMVC復号順序(すなわち、ビットストリーム順序)が図2に示されている。復号順序の構成は時間優先コーディングと呼ばれる。各アクセスユニットは、1つの出力時間インスタンスのためのすべてのビューのコーディングされたピクチャを含むように定義される。図2に示す復号順序は出力順序または表示順序と同じではないことがあることを理解されたい。

10

【0076】

[0083]図2では、S0~S7はそれぞれ、マルチビュービデオの異なるビューを指す。T0~T8はそれぞれ、1つの出力時間インスタンスを表す。アクセスユニットは、1つの出力時間インスタンスについてのすべてのビューのコーディングされたピクチャを含み得る。たとえば、第1のアクセスユニットは、時間インスタンスT0についてのビューS0~S7(すなわち、ピクチャ0~7)のすべてを含み、第2のアクセスユニットは、時間インスタンスT1についてのビューS0~S7(すなわち、ピクチャ8~15)のすべてを含み、以下同様である。

20

【0077】

[0084]図2では、ビューの各々はピクチャのセットを含む。たとえば、ビューS0はピクチャ0、8、16、24、32、40、48、56、および64のセットを含み、ビューS1はピクチャ1、9、17、25、33、41、49、57、および65のセットを含み、以下同様である。いくつかの例では、ビューの各ピクチャは2つのコンポーネントを含み、一方のコンポーネントはテクスチャビューコンポーネントと呼ばれ、他方のコンポーネントは深度ビューコンポーネントと呼ばれる。ビュー内のテクスチャビューコンポーネントおよび深度ビューコンポーネントは、互いに対応すると見なされ得る。たとえば、ビューのテクスチャビューコンポーネントは、そのビューの深度ビューコンポーネントに対応すると見なされ得、その逆も同様である(すなわち、深度ビューコンポーネントはセット中のそのテクスチャビューコンポーネントに対応し、その逆も同様である)。本開示で使用する、テクスチャビューコンポーネントと、対応する深度ビューコンポーネントとは、単一のアクセスユニットの同じビューの一部であると見なされ得る。

30

【0078】

[0085]テクスチャビューコンポーネントは、表示される実際の画像コンテンツを含む。たとえば、テクスチャビューコンポーネントは、ルーマ(Y)成分と、クロマ(CbおよびCr)成分とを含み得る。深度ビューコンポーネントは、その対応するテクスチャビューコンポーネント中のピクセルの相対深度を示し得る。一例として、深度ビューコンポーネントは、ルーマ値のみを含むグレースケール画像と同様であり得る。言い換えれば、深度ビューコンポーネントは、任意の画像コンテンツを伝達するのではなく、テクスチャビューコンポーネント中のピクセルの相対深度の測定値を提供することができる。

40

【0079】

[0086]たとえば、深度ビューコンポーネント中のピクセルについてのピクセル値は、0~255の範囲にあり得る(たとえば、8ビット値)。いくつかの例では、より小さいピクセル値は、ピクセルが閲覧者の観点からより近いことを示すことができ、より大きいピクセル値は、ピクセルが閲覧者の観点からより遠いことを示すことができる。また、この8ビット値はグレースケールにマッピングし得る。たとえば、より小さいピクセル値は明

50

るいピクセルにマッピングし、ここで0値は純粹に白いピクセルにマッピングする一方、より大きいピクセル値は暗いピクセルにマッピングし、ここで255の値は黒いピクセルにマッピングし、中間にあるピクセル値は異なるグレーレベルにマッピングする。したがって、ピクセルの深度を示すピクセル値は、ルーマ成分のみ（たとえば、グレースケールにマッピングする8ビット値）を必要とし得る。

【0080】

[0087]深度を識別するためにルーマ値（たとえば、強度値）のみを使用する深度ビューコンポーネントは、例示のために提供され、限定するものと見なされるべきではない。いくつかの例では、テクスチャビューコンポーネント中のピクセルの相対深度を示すために任意の技法が利用され得る。

10

【0081】

[0088]あらゆる例において深度ビューコンポーネントが必要であるとは限らない。たとえば、マルチビュービデオコーディングのいくつかの例では、深度ビューコンポーネントは存在しない。これらの例では、テクスチャビューコンポーネントとピクチャとは同義語であり得る。

【0082】

[0089]マルチビュービデオコーディングによれば、テクスチャビューコンポーネントは、同じビュー中のテクスチャビューコンポーネントから、または1つもしくは複数の異なるビュー中のテクスチャビューコンポーネントからインター予測される。テクスチャビューコンポーネントは、「ビデオブロック」と呼ばれ、H.264コンテキストでは一般に「マクロブロック」と呼ばれる、ビデオデータのブロック中でコーディングされ得る。HEVC規格など、他のビデオコーディング規格は、ビデオブロックをツリーブロックまたはコーディングユニット(CU)と呼ぶ場合がある。

20

【0083】

[0090]図2は、時間優先コーディングの一例を示しているが、いくつかの例では、図4に関してより詳細に説明するように、ピクチャが並列に符号化または復号されることが考えられ得る。たとえば、ビューS1の第1のピクチャ（たとえば、テクスチャビューコンポーネント、および場合によっては深度ビューコンポーネント）は、ビューS0の第1のピクチャによりインター予測され得る。たとえば、ビューS1の第1のピクチャ中のブロックは、ビューS0の第1のピクチャ中のブロックを予測ブロックとして使用することができ、ここでビューS1の第1のピクチャ中のブロックについての動きベクトルは、ビューS0の第1のピクチャ中の予測ブロックを指す。この意味で、ビューS0中の第1のピクチャは、インター予測の目的上、ビューS1中の第1のピクチャのための参照ピクチャであり得る。

30

【0084】

[0091]場合によっては、ビューS1の第1のピクチャ中のブロックをビューS0の第1のピクチャ中のブロックによりインター予測するために、ビデオデコーダ30は、ビューS1の第1のピクチャの再構成を開始する前に（すなわち、ビューS1の第1のピクチャを復号する前に）、ビューS0の第1のピクチャを完全に再構成する（すなわち、ビューS0の第1のピクチャを完全に復号する）ことがある。並列復号では、ビデオデコーダ30は、ビューS0の第1のピクチャの復号と並列にビューS1の第1のピクチャの復号を開始することがある。

40

【0085】

[0092]一般に、並列復号は、動きベクトル制限の概念に基づく。動きベクトル制限は、ブロックについての動きベクトルのx成分またはy成分のうち少なくとも1つ（および場合によっては両方）に対する制限を指す。一例として、動きベクトル制限は、動きベクトル成分の範囲を定義し得る。

【0086】

[0093]動きベクトル成分の定義された範囲の外にある参照ピクチャ中の任意のブロックは、現在のブロックをインター予測するために使用され得ない。一例として、動きベクトル

50

ル制限は、 y 成分しきい値 (Th) を定義し得る。この例では、現在のピクチャ中の任意のブロックについての動きベクトルの y 成分は、 Th よりも大きくなり得ない。これは、現在のピクチャ中のブロックに対して、垂直距離が Th を超える参照ピクチャ中の任意のブロックは、現在のピクチャ中のブロックをインター予測するために使用され得ないことを意味する。

【0087】

[0094]より詳細に説明するように、本開示で説明する技法は、ブロックについての導出された動きベクトルが動きベクトル制限に準拠することを確実にすることができる。たとえば、ビデオエンコーダ20は、ビデオデコーダ30が現在のブロックについての動きベクトルを導出する方法を定義する情報を含むビットストリームを出力用に生成することが
10
できる。本技法は、ビデオエンコーダ20がビットストリーム中に出力用に生成することができる情報を、ビデオデコーダ30がビットストリーム中の情報に基づいて現在のブロックについての動きベクトルを導出するときに、動きベクトル制限に違反する動きベクトルをビデオデコーダ30が導出しないように、制約することができる。言い換えれば、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約される。

【0088】

[0095]その上、より詳細に説明するように、ビデオエンコーダ20によって出力用に生成されたビットストリームは、受信された値をさらに変換する必要なしにピクチャの並列復号をいつ開始すべきかを決定する目的でビデオデコーダ30が利用することができるユニ
20
ットにおける並列復号遅延を指定する情報を含むことができる。たとえば、ビデオデコーダ30は、HEVCビデオコーディング技法の階層構造が必ずしも行に対応するとは限らないので、並列復号遅延が行のユニットにある場合に、ピクチャの並列復号をいつ開始すべきかを決定することが可能ではないことがある。より詳細に説明するように、ビデオエンコーダ20は、コーディングユニットサイズ(たとえば、LCUサイズ、最小コーディングユニット(SCU)サイズ)または固定サイズに基づいて、並列復号遅延を示す情報をシグナリングすることができる。

【0089】

[0096]図3は、マルチビュービデオコーディング用の構造を示す概念図である。図3は、マルチビュービデオコーディング構造を示す。たとえば、図3に、マルチビュービデオ
30
コーディング用の典型的なマルチビュービデオコーディング予測(各ビュー内のピクチャ間予測とビュー間予測の両方を含む)構造が示され、ここでは矢印によって予測が示され、矢印の終点のオブジェクトは、予測参照のために矢印の始点のオブジェクトを使用する。マルチビュービデオコーディングでは、H.264/AVC動き補償のシンタックスを使用するが、異なるビュー中のピクチャが参照ピクチャとして使用されることを可能にする視差動き補償によって、ビュー間予測がサポートされる。

【0090】

[0097]2つのビューのコーディングは、マルチビュービデオコーディングによってもサ
40
ポートされる可能性があり、マルチビュービデオコーディングの利点の1つは、ビデオエンコーダ20が3Dビデオ入力として3つ以上のビューをとらえることができ、ビデオデコーダ30がそのようなマルチビュー表現を復号することができることである。したがって、ビデオデコーダ30のようなビデオデコーダをもつ任意のレンダラは、3つ以上のビューをもつ3Dビデオコンテンツを予想し得る。

【0091】

[0098]マルチビュービデオコーディングでは、同じアクセスユニット中の(すなわち、
同じ時間インスタンスをもつ)ピクチャ間でビュー間予測が可能になる。非ベースビューのうちの一つ中のピクチャをコーディングするとき、ピクチャが異なるビュー中にあるが同じ時間インスタンスをもつ場合、そのピクチャは参照ピクチャリストに追加され得る。ビュー間予測参照ピクチャは、任意のインター予測参照ピクチャと同様に、参照ピクチャ
50
リストの任意の位置に置かれ得る。ビュー間予測参照ピクチャが動き補償に使用されると

き、対応する動きベクトルは「視差動きベクトル」と呼ばれる。

【0092】

[0099]図3では、ビューS0はベースビューと見なされ得、他のビューS1～S7は従属ビューと見なされ得る。たとえば、ビューS0は、ビューS0中のピクチャが別のビュー中のピクチャによりビュー間予測されないので、ベースビューであり得る。しかしながら、ビューS0中のピクチャは、ビューS0中の他のピクチャによりインター予測される。他のビューは、これらの他のビュー中の少なくとも1つのピクチャがビュー間予測されるので、従属ビューであり得る。たとえば、ビューS1中の第1のピクチャは、ビューS0中の第1のピクチャおよびビューS2中の第1のピクチャによりビュー間予測される。従属ビュー中のいくつかのピクチャが同じビュー中の他のピクチャによりインター予測され得ることを理解されたい。

10

【0093】

[0100]たとえば、「インター予測」という用語は、現在のピクチャ中のブロックが参照ピクチャ中のブロック（たとえば、参照ピクチャ中の予測ブロック）に基づいて予測される場合を指す。現在のピクチャ中のブロックについての動きベクトルは、参照ピクチャ中のブロックを指す。マルチビュービデオコーディングでは、参照ピクチャは、現在のピクチャと同じビュー中にあるピクチャまたは別のビュー中にあるピクチャであり得る。参照ピクチャが別のビュー中にある場合、参照ピクチャと現在のピクチャとは同じアクセスユニット中にある。たとえば、前述のように、アクセスユニットは、同じ時間インスタンスについての異なるビューのピクチャを含み、別のビュー中の参照ピクチャは、現在のピクチャと同じアクセスユニット中にあることが必要とされ得る。

20

【0094】

[0101]参照ピクチャが現在のピクチャとは別のビュー中にあるとき、他のビュー中の参照ピクチャを指す動きベクトルは、視差動きベクトルと呼ばれる。また、参照ピクチャが現在のピクチャとは別のビュー中にあるとき、現在のピクチャ中のブロックは、ビュー間予測されると呼ばれ得る。言い換えれば、ビュー間予測では、現在のピクチャ中のブロックが、現在のピクチャを含むビュー以外のビュー中にある参照ピクチャ中のブロックを指す視差動きベクトルによりビュー間予測される。

【0095】

[0102]インター予測（すなわち、ビュー間予測または同じビュー中のピクチャによるインター予測）を実行するために、ビデオエンコーダ20およびビデオデコーダ30は、1つまたは2つの参照ピクチャリスト（RefPicList0およびRefPicList1と呼ばれる）を構成することができる。ビデオエンコーダ20およびビデオデコーダ30は、参照ピクチャリスト中で識別されたピクチャをインター予測目的に利用することができる。本開示で説明する技法では、ビュー間参照ピクチャ（すなわち、符号化または復号されている現在のピクチャのビュー以外のビュー中にある参照ピクチャ）が、参照ピクチャリスト内の任意のロケーションに挿入され得る。ビデオエンコーダ20およびビデオデコーダ30が参照ピクチャリストを構成した（すなわち、RefPicList0とRefPicList1とを構成した）後、参照ピクチャリストへの参照インデックスが、参照ピクチャリスト中に含まれる任意の参照ピクチャを識別することができる。

30

40

【0096】

[0103]たとえば、双予測ピクチャ（Bピクチャ）の場合、ビデオエンコーダ20は、双予測ピクチャを符号化するために使用された参照ピクチャを識別するRefPicList0へのインデックスとRefPicList1へのインデックスとをシグナリングすることができる。ビデオデコーダ30は、現在のピクチャを復号する（たとえば、再構成する）ために必要とした参照ピクチャ（RefPicList0からの1つおよびRefPicList1からの1つ）を決定するためにインデックスを利用することができる。単方向予測ピクチャ（Pピクチャ）の場合、ビデオエンコーダ20は、単方向予測ピクチャを符号化するために使用された参照ピクチャを識別するRefPicList0もしくはRefPicList1のうちの1つへのインデックス、またはRefPicList0

50

へのインデックスのみ（その場合、ビデオデコーダ30はRefPicList1を再構成することはない）をシグナリングすることができる。ビデオデコーダ30は、現在のピクチャを復号する（たとえば、再構成する）ために必要とされる参照ピクチャを決定するためにインデックスを利用することができる。

【0097】

[0104]一般に、Bピクチャ（双予測ピクチャ）の第1または第2の参照ピクチャリストについての参照ピクチャリスト構成は、2つのステップ、すなわち参照ピクチャリスト初期化と、参照ピクチャリスト並べ替え（修正）を含む。参照ピクチャリスト初期化は、参照ピクチャメモリ（復号ピクチャバッファ（DPB）としても知られる）中の参照ピクチャを、POC（ピクチャの表示順で整列されるピクチャ順序カウンタ）値の順序に基づいてリストに入れる明示的機構である。参照ピクチャリスト並べ替え機構は、参照ピクチャリスト初期化中にリストに入れられたピクチャの位置を任意の新しい位置に修正すること、または参照ピクチャメモリ中の任意の参照ピクチャを、そのピクチャが初期化リストに属さなくても、任意の位置に入れることができる。参照ピクチャリスト並べ替え（修正）後のいくつかのピクチャは、リスト中のはるかに離れた位置に入れられる場合がある。ただし、ピクチャの位置が、リストのアクティブ参照ピクチャの数を超える場合、ピクチャは、最終参照ピクチャリストのエントリとは見なされない。アクティブ参照ピクチャの数は、各リスト用のスライスヘッダに入れてシグナリングされ得る。

10

【0098】

[0105]図4は、2つのビューのための並列コーディングプロセスの一例を示すフロー図である。図4は、補足エンハンスメント情報（SEI）の並列コーディング（たとえば、復号）情報を示す。H.264/AVCのMVC拡張では、並列復号をサポートするSEIメッセージがある。ビューコンポーネントごとに、1つのマクロブロックを復号するとき、現在のビューと参照ビューとの2つの間に少なくとも2つのマクロブロック（MB）行遅延があることが考慮される。追加MB行のユニットにおける追加遅延が、各ビューおよびその参照ビューの各々についてSEIメッセージでシグナリングされる。このSEIメッセージの使用の一例が図4に示されている。

20

【0099】

[0106]図4は、第2のビュー（ビュー1）のピクチャ42の復号と並列な第1のビュー（ビュー0）のピクチャ40の復号を示す。この例では、ピクチャ40は、ピクチャ42に対する参照ピクチャである（すなわち、ピクチャ40のブロックは、ピクチャ42のブロックをインター予測、特にビュー間予測するために使用される）。また、図4は、ピクチャ40がプロセッサP0によって復号されていることと、ピクチャ42がプロセッサP1によって復号されていることを示す。プロセッサP0およびP1への言及は、ピクチャ40とピクチャ42とを復号するために実行した処理アルゴリズムを示すことである。プロセッサP0およびP1は、ビデオデコーダ30の一部であり得る。言い換えれば、図4は、ピクチャ40とピクチャ42とを並列に復号するビデオデコーダ30の一例を示している。

30

【0100】

[0107]時間44において、ビデオデコーダ30は、復号プロセスを開始するが、ピクチャ42については、ビデオデコーダ30がピクチャ42の第1の行の復号を開始する前に遅延がある。たとえば、図4では、並列復号遅延は、マクロブロックの2行である。前述のように、いくつかの技法は、少なくとも、2行の復号遅延を必要とし得る。ビデオエンコーダ20は、SEIメッセージ中で正確な復号遅延をビデオデコーダ30にシグナリングすることができる。

40

【0101】

[0108]時間46において、ビデオデコーダ30は、ピクチャ40のマクロブロックの第1の行を再構成するために、ピクチャ40のマクロブロックの第1の行を復号する。時間48において、ビデオデコーダ30は、ピクチャ40のマクロブロックの再構成された第1の行を、ビデオデコーダ30の復号ピクチャバッファ（DPB）に記憶する。時間50

50

において、ビデオデコーダ30は、ピクチャ40のマクロブロックの第2の行を再構成するために、第2の行を復号する。時間52において、ビデオデコーダ30は、ピクチャ40のマクロブロックの再構成された第2の行を、ビデオデコーダ30のDPBに記憶する。また、時間52において、ビデオデコーダ30は、ピクチャ42の並列復号が開始できることを決定する。たとえば、第1の処理アルゴリズム(プロセッサP0)は、第2の処理アルゴリズム(プロセッサP1)に、(たとえば、プロセッサP1に制御信号「ビュー1通知」を送ることによって)ピクチャ42の並列復号が開始できることを通知する。

【0102】

[0109]時間54において、ビデオデコーダ30は、ピクチャ40の第3の行を再構成するために、ピクチャ40の第3の行を復号し、並列に(たとえば、同時に)、ピクチャ42の第1の行を再構成するために、ピクチャ42の第1の行を復号する。時間56において、ビデオデコーダ30は、ピクチャ40の再構成された第3の行をDPBに記憶する。図示されていないが、ビデオデコーダ30は、ピクチャ42の再構成された第1の行をDPBに記憶することもできる。

【0103】

[0110]また、時間56において、プロセッサP0はプロセッサP1に、ピクチャ42の次の行の復号が開始できることを通知する。時間58において、ビデオデコーダ30は、ピクチャ40の第4の行を再構成するために、ピクチャ40の第4の行を復号し、並列に(たとえば、同時に)、ピクチャ42の第2の行を再構成するために、ピクチャ42の第2の行を復号する。ビデオデコーダ30は、ピクチャ40とピクチャ42とを再構成するために、これらのステップを繰り返す。このようにして、ビデオデコーダ30は、ビューからの1つのピクチャ(たとえば、ビュー0のピクチャ40)が別のビューからの別のピクチャ(たとえば、ビュー1のピクチャ42)のための参照ピクチャである、異なるビューからの2つのピクチャの並列復号を実施することができる。

【0104】

[0111]いくつかの例では、動きベクトル制限は並列復号を促進する。図4に示す例では、ピクチャ42中のブロックについての動きベクトルは、ピクチャ42中のブロックについての動きベクトルのy成分が1行の距離よりも大きくなり得ないように制限され得る。一例として、図4では、ビデオデコーダ30は、ピクチャ40の第3の行の復号と並列にピクチャ42の第1の行の復号を開始する(すなわち、ビデオデコーダ30は、ピクチャ42の第1の行を復号する前にピクチャ40の2行が再構成されるまで待機しなければならない)。この例では、ピクチャ42の第1の行中のブロックについての動きベクトルが、ピクチャ40の第2の行の後にあるブロックを指している場合、ビデオデコーダ30がピクチャ40のその行をまだ復号していないことがあるので、復号エラーがあり得る。動きベクトル制限がある場合、ビデオエンコーダ20は、ブロックのy成分が1行の距離よりも大きいピクチャ42のブロックについての動きベクトル値をビデオデコーダ30にシグナリングすることはない。

【0105】

[0112]図4に示す例では、動きベクトル制限はピクチャ42についての動きベクトルのy成分についての1行であり得る。このようにして、ビデオデコーダ30がピクチャ40の第2の行を再構成した後、ビデオデコーダ30は、ピクチャ42中のブロックが、垂直距離がブロックから1行を超える参照ピクチャ中の任意のブロックを指さないの、ピクチャ42の復号を開始することが可能であり得る。したがって、動きベクトル制限は、並列復号遅延を示すものと見なされ得る。以下でより詳細に説明するように、いくつかの例では、ビデオデコーダ30によって実行されるフィルタ処理に基づいて、ビデオデコーダ30は、現在のピクチャ(たとえば、ピクチャ42)の再構成を開始する前に、参照ピクチャ(たとえば、ピクチャ40)の追加の行を再構成する必要があり得る。

【0106】

[0113]図4の上記の例では、ビデオエンコーダ20およびビデオデコーダ30は、H.264/AVCのMVC拡張に従って機能し得る。しかしながら、動きベクトル制限およ

10

20

30

40

50

び並列復号遅延に関係する技法が、H E V Cベースのビデオコーディング技法に拡張されるとき、いくつかの問題があり得る。たとえば、異なるビューに参照ピクチャがあるときに動きベクトル制限を実施することに伴ういくつかの問題があることがあり、H E V Cコーディング構造により並列復号遅延を示すための技法を利用することに伴ういくつかの問題があり得る。さらに、上記の例はマルチビュービデオコーディングについて説明されているが、本開示で説明する技法は、H E V Cのスケラブル拡張（たとえば、S H V C）を含むスケラブルビデオコーディングにも拡張され得る。

【 0 1 0 7 】

[0114]スケラブルビデオコーディング（S V C）では、多重レイヤがあり得る。最下位レベルにあるレイヤはベースレイヤ（B L）としてのみ働き、最上位レベルにあるレイヤはエンハンスメントレイヤ（E L）としてのみ働き得る。中間にあるすべてのレイヤは、E LとB Lの両方として働き得る。たとえば、中間にあるレイヤは、その下のレイヤのためのE Lであり、同時にその上のレイヤのためのB Lとしてのものであり得る。説明を簡単にするために、本開示は、現在の技法を示す際に、B LおよびE Lという2つのレイヤがあることを仮定する。本開示で説明する技法は多重レイヤを伴う場合にも適用可能であることに留意されたい。

10

【 0 1 0 8 】

[0115]たとえば、上記の例は、従属ビューからのピクチャの現在のブロックが別のビュー（たとえば、ベースビュー）中の参照ピクチャによりインター予測されるビュー間予測の概念について説明した。いくつかの例では、レイヤ間予測において、エンハンスメントレイヤからのピクチャの現在のブロックが、別のレイヤ（たとえば、ベースレイヤ）中の参照ピクチャによりインター予測される。本開示で説明する技法では、動きベクトル制限および並列復号がスケラブルビデオコーディング技法にも利用され得る。たとえば、レイヤ間ピクチャ（たとえば、ベースレイヤピクチャ）を指すエンハンスメントレイヤピクチャのブロックについての動きベクトル制限があり得る。また、2つのレイヤを並列復号する場合で、図4に関して上述した方法と同様の方法でインター予測目的（たとえば、レイヤ間予測目的）のために一方のレイヤが他方のレイヤを使用することが考えられ得る。

20

【 0 1 0 9 】

[0116]前述のように、マルチビュービデオコーディングの場合、異なるH E V Cベースのマルチビュービデオコーディング拡張に関する進行中の議論がある。1つの例は、H E V Cの3 D V拡張であり、別の例は、H E V CのM V拡張である。たとえば、M P E Gは、H E V Cに基づいて3 D VおよびM Vの規格を開発しており、3 D VおよびM Vの規格に対して、規格化の取り組みの一部は、H E V Cに基づくマルチビュービデオコーデックの規格化も含む。H E V Cベースの3 D VおよびM Vでは、異なるビューからの再構成されたビューコンポーネントに基づくビュー間予測が有効にされる。

30

【 0 1 1 0 】

[0117]H E V Cベースの3 D VとH E V CベースのM Vとの間の差異の1つは、ハイレベルシンタックスのみ（H L Sのみ）という要件である。たとえば、A V Cでは、マルチビュー拡張が、拡張が「H L Sのみ」という要件を実際に満たすように行われた。「H L Sのみ」という要件は、A V C中のマクロブロックレベルにおけるモジュールが再設計される必要がなく、マルチビュービデオコーディング（M V C）に完全に再使用され得ないように、M V Cにおいてハイレベルシンタックス（H L S）の変更しか存在しないことを保証する。

40

【 0 1 1 1 】

[0118]このH L Sのみ要件は、M V - H E V Cに拡張されるが、3 D V - H E V Cには拡張されない。たとえば、M V - H E V C規格の1つの要件は、M V - H E V CがH E V C規格に対して、H L Sのみ要件を満たすことであり得る。3 D V - H E V C規格は、H L Sのみ変更を必要としないことがある。3 D V - H E V CはH L Sのみ変更を必要としないが、H E V C規格に対してH L Sのみ変更が行われた場合、結果は、3 D V - H E V Cに依然として準拠し得る。言い換えれば、「H L Sのみ」という要件は、H E V CのM

50

V C / 3 D V 拡張に対して、また、マルチフレーム復号が受け入れ可能であると考えられる場合には、H E V C のスケーラブルビデオコーディング (S V C) 拡張 (S H V C) に対しても満たされ得ることが可能である。

【 0 1 1 2 】

[0119]ビュー間予測を可能にするために、最低限の、および場合によっては、必ずしも必要ではないが、H L S の変更が、以下のピクチャ識別目的のためのものであり、ここで参照ピクチャリストの構成および標識は、特定のビュー中のピクチャを識別することが可能である必要がある。たとえば、マルチビュービデオコーディングの場合、参照ピクチャリスト (R e f P i c L i s t 0 および R e f P i c L i s t 1) の一方または両方は、ビュー間ピクチャ (すなわち、符号化または復号されているピクチャと同じビュー中にな

10

【 0 1 1 3 】

[0120]H L S の変更が、H . 2 6 4 / M V C における「H L S のみ」という要件を満たすには十分ではない場合、ローレベルコーディングモジュールが、H E V C のローレベルコーディングモジュールの構成目的とされていない機能を実施する必要がある (たとえば、無動きに関するスケーリングを処理する) 状況に決して遭遇することがないように、他の制約、仮定が行われる。そのような制約、修正、および仮定は以下の通りである。(1) コロケートされたピクチャがビュー間 (のみ) 参照ピクチャである場合、時間的ダイレクトモードを無効にする、(2) ビュー間 (のみ) 参照ピクチャを、短期の、空間的ダイ

20

【 0 1 1 4 】

[0121]H L S のみ H E V C 拡張の1つの例は、M V - H E V C であり、M V - H E V C の最新のワーキングドラフトは、J C T 3 V - B 1 0 0 4、「M V - H E V C D r a f t T e x t 2」、G . T e c h、K . W e g n e r、Y . C h e n、M . H a n n u k s e l a であり、これは http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/2_Shanghai/wg11/JCT3V-B1004-v1.zip で入手可能であり、その内容全体が参照により本明細書に組み込まれる。

【 0 1 1 5 】

[0122]本開示は、N a k a g a m i らによる「M V - H E V C : V e r t i c a l l e n g t h r e s t r i c t i o n o f i n t e r - v i e w v e c t o r f o r H E V C s i m p l e 3 D e x t e n s i o n」と題する J C T 3 V - B 0 0 3 7 も参照し、これは http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=225 から入手可能であり、その内容全体が参照により組み込まれる。J C T 3 V - B 0 0 3 7 では、メモリ帯域幅要件を緩和し、デコーダをより並列に使いやすくするために、M V - H E V C における動きベクトルのポインティングを、垂直方向のベースビュー参照フレームに制限することが示唆されている。この制限は、次のような S P S においてシグナリングされるフラグによって示されることが示唆されている。

30

40

【 0 1 1 6 】

(p r o f i l e _ i d c = 「マルチビューサポートプロファイル」) の場合

【数 1】

```

{
    interview_vector_v_restrict_flag                u(1)
    (interview__vector__v__restrict__flag==1)の場合 {
        max_interview_vector_v_length_div4        u(4)
    }
}

```

10

【0117】

【0123】動きベクトル制限は並列復号に関して上述されているが、動きベクトル制限は、他の追加の、独立した理由により有用であり得る。言い換えれば、いくつかの例では、ビデオデコーダ30が並列復号を実施するように構成されない場合でも、動きベクトル制限は有用であるか、または望ましいことがある。

【0118】

【0124】一例として、動きベクトル制限は、メモリ帯域幅の低減（たとえば、メモリアクセスおよび/またはアクセスされる必要のあるデータの量を低減すること）に有用であり得る。たとえば、動きベクトル制限がある場合、現在のピクチャを符号化または復号するために、ビデオエンコーダ20またはビデオデコーダ30が参照ピクチャからアクセスする必要があるデータの量は低減され得る。メモリ帯域幅は、予測ピクセルごとの参照ピクチャからのアクセスされるピクセルの数に基づいて推定され得る。たとえば、参照ピクチャ中でアクセスされるピクセルの数の低減により、メモリ帯域幅はより低くなる。多くの使用事例では、メモリ帯域幅はボトルネックと見なされることがあり、最悪ケースに備えたメモリ帯域幅の低減は有利と見なされ得る。たとえば、本技法は、他の技法と比べて、メモリアクセスを単純化、低減、または除去することができる。

20

【0119】

【0125】たとえば、別のビュー中のピクチャを指す動きベクトルの垂直成分が制約される時、参照ビューピクチャのピクセルにアクセスすることによって必要とされるメモリ帯域幅は低減され得る。一例として、垂直成分（すなわち、動きベクトルのy成分）がより小さい値に制限される場合、補間目的のためにブロックの外の垂直方向における任意のピクセルにアクセスする必要はないことがある。

30

【0120】

【0126】たとえば、ブロックについての動きベクトルは、参照ピクチャ中の整数ピクセルを指さず、代わりに、サブ整数ピクセル（ペルと呼ばれる）を指すことがある。ビデオエンコーダ20およびビデオデコーダ30は、これらのサブ整数ピクセルまたはペルを、参照ピクチャ中の隣接整数ピクセルのピクセル値を補間することによって生成する。たとえば、ビデオエンコーダ20およびビデオデコーダ30は、サブ整数ピクセルを補間するためにDPBから、動きベクトルによって指されるブロックの外に及ぶピクセルについてのピクセル値を含む、参照ピクチャについてのピクセル値にアクセスする。動きベクトル制限がある場合、ビデオエンコーダ20およびビデオデコーダ30は、補間のために動きベクトルが制限されるエリアの外に存在するピクセルについてのピクセル値にアクセスする必要はないことがあり、このことは、ビデオエンコーダ20およびビデオデコーダ30がそれぞれのDPBから取り出す必要があるデータの量を低減し、それによりメモリ帯域幅低減を促進する。

40

【0121】

【0127】それでも、動きベクトルが動きベクトル制限に違反しないことを確実にするために、追加の態様が考慮される必要があり得る。たとえば、ビデオエンコーダ20は、ビデオ

50

オデコーダ30がブロックについての動きベクトルを決定するために使用するブロックについての動きベクトルのx成分およびy成分をシグナリングしないことがある。代わりに、ビデオエンコーダ20は、ビデオデコーダ30が動きベクトルを導出するために使用する情報をシグナリングすることがある。

【0122】

[0128]たとえば、ビデオデコーダ30は、マージノスキップモードおよび高度動きベクトル予測(A MVP)モードの一部として、動きベクトル予測子に基づいて、現在のブロックについての動きベクトルを導出することができる。マージノスキップモードおよびA MVPモードでは、ビデオエンコーダ20は、現在のブロックについての動きベクトルをシグナリングしないことがあるが、代わりに候補動きベクトル予測子のリストへのインデックスをシグナリングし得る。ビデオデコーダ30は、現在のブロックについての動きベクトルを、候補動きベクトル予測子のリストへのインデックスによって識別された動きベクトル予測子から導出することができる。

10

【0123】

[0129]たとえば、マージノスキップモードおよびA MVPモードでは、ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、候補動きベクトル予測子のリストを構成する。ビデオエンコーダ20およびビデオデコーダ30は、候補動きベクトル予測子のそれぞれのリストを構成するために、リスト中で識別された候補動きベクトル予測子がビデオエンコーダ20とビデオデコーダ30の両方にとって同じになるように、実質的に同様の技法を実施することができる。

20

【0124】

[0130]マージノスキップモードおよびA MVPモードでは、ビデオエンコーダ20およびビデオデコーダ30は、隣接ブロックの動き情報に基づいて候補動きベクトル予測子のリストを構成する。隣接ブロックの例としては、空間的隣接ブロックおよび時間的隣接ブロックがある。空間的隣接ブロックは、現在のブロックに隣接し、現在のブロックと同じピクチャ中にあるブロックを指す。時間的隣接ブロックは、現在のブロックに隣接するが、別のピクチャ中にあるブロックを指す。現在のブロックについての動きベクトル予測子であり得る空間的隣接ブロックについての動きベクトルは、空間的動きベクトル予測子(S MVP)と呼ばれる。現在のブロックについての動きベクトル予測子であり得る時間的隣接ブロックについての動きベクトルは、時間的動きベクトル予測子(T MVP)と呼ばれる。

30

【0125】

[0131]マージノスキップモードまたはA MVPモードの両方では、ビデオエンコーダ20は、ビデオデコーダ30が受信する候補動きベクトル予測子のリストへのインデックスをシグナリングする。次いでビデオデコーダ30は、候補動きベクトル予測子のリストへのシグナリングされたインデックスから動き情報(たとえば、参照ピクチャおよび1つまたは複数の動きベクトル)を識別し、識別された動き情報に基づいて、現在のブロックについての動きベクトルを決定する。

【0126】

[0132]たとえば、マージノスキップモードでは、ビデオデコーダ30は、候補動きベクトル予測子のリストへのインデックスを受信し、シグナリングされたインデックスに基づいて、候補動きベクトル予測子のリストに記憶された動き情報を識別する。ビデオデコーダ30は、識別された動き情報から参照ピクチャリストと、参照インデックスと、動きベクトルとを決定する。次いでビデオデコーダ30は、識別された動き情報からの決定された参照ピクチャリストと、参照インデックスと、動きベクトルとを、現在のブロックについての動き情報として採用する。言い換えれば、現在のブロックは、候補動きベクトル予測子のリストへのインデックスによって識別されたブロックの動き情報を継承する。このようにして、ビデオデコーダ30は、動きベクトル予測子に基づいて現在のブロックについての動きベクトルを導出する(たとえば、ビデオデコーダ30は、現在のブロックについての動きベクトルを動きベクトル予測子に等しくセットする)。

40

50

【 0 1 2 7 】

[0133]いくつかの例では、マージノスキップモードにおいて、候補動きベクトル予測子のリストへのインデックスによって識別されたブロックが時間的隣接ブロックを指す場合、ビデオデコーダ30は、この動きベクトル候補に関連する動き情報を採用することはない。ビデオデコーダ30は、参照ピクチャリストへの参照インデックスを決定する（たとえば、参照ピクチャリストの一方または両方へのインデックス0を選択する）ために、任意の既知の技法または開発されることになる任意の技法を利用することができる。本開示では、時間的隣接ブロックについての動きベクトルが動きベクトル予測子として使用されるときに、参照ピクチャリストへの参照インデックスとしてデフォルトインデックス（たとえば、インデックス0）が使用されるときなど、参照ピクチャリストに対するいくつかの制約があり得る。

10

【 0 1 2 8 】

[0134]このようにして、ビデオデコーダ30は、マージノスキップモードにおいて現在のブロックについての動き情報を決定することができる。マージモードでは、ビデオデコーダ30はまた、現在のブロックと決定された動き情報によって指されるブロックとの間の残差データを受信することができ、ビデオデコーダ30は、現在のブロックのピクセル値を決定するために、残差データを利用することができる。スキップモードでは、ビデオデコーダ30は、現在のブロックと決定された動き情報によって指されるブロックとの間の残差データを受信することはない。この例では、ビデオデコーダ30は、残差データが0であると仮定する（すなわち、現在のブロックのピクセル値を、決定された情報によって指されるブロックのピクセル値に等しくなるようにセットする）ことができる。

20

【 0 1 2 9 】

[0135]AMVPモードはマージノスキップモードと同様であり得るが、ビデオデコーダ30は、候補動きベクトル予測子のリストへのインデックスを受信することに加えて、ビデオエンコーダ20からの参照ピクチャリストについての参照インデックス値と動きベクトル差分を受信することもできる。動きベクトル差分(MVD)は、候補動きベクトル予測子のリストへのインデックスによって識別されたブロックの動きベクトルと別の動きベクトル（たとえば、ビデオエンコーダ20によってベストマッチ探索に基づいて発見された動きベクトル）との間の動きベクトルの差であり得る。

30

【 0 1 3 0 】

[0136]たとえば、ビデオデコーダ30は、候補動きベクトル予測子のリストへのインデックスによって識別されたブロックの動きベクトルを決定することができる。ビデオデコーダ30は、現在のブロックについての動きベクトルを決定するために、候補動きベクトル予測子のリストへのインデックスによって識別されたブロックの動きベクトルの値を、シグナリングされた動きベクトル差分により加算または減算することができる。加えて、ビデオデコーダ30は、参照ピクチャリストを示すシグナリングされた情報およびその参照ピクチャリストへのシグナリングされたインデックスに基づいて、決定された動きベクトルが指す参照ピクチャを決定することができる。このようにして、ビデオデコーダ30は、AMVPモードにおいて現在のブロックについての動き情報を決定することができる。

40

【 0 1 3 1 】

[0137]前述のように、マージノスキップモードおよびAMVPモードの場合、ビデオエンコーダ20およびビデオデコーダ30は、候補動きベクトル予測子のリストを構成する。以下では、候補動きベクトル予測子のリストを構成する例示的な方法について説明する。いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、時間的隣接ブロックの動き情報が候補動きベクトル予測子のリスト中に含まれるべきである場合に、ピクチャ順序カウント(POC)値に基づいて時間的隣接ブロックの動きベクトル情報をスケールリングすることができる。

【 0 1 3 2 】

[0138]以下では、どの時間的動きベクトル予測子が候補動きベクトル予測子のリスト中

50

に含むべきかをビデオエンコーダ20およびビデオデコーダ30が決定し得る方法に関するいくつかの説明を行う。時間的動きベクトル予測子(TMV P)を得るために、ビデオエンコーダ20およびビデオデコーダ30は最初に、コロケートされたピクチャを識別することができる。現在のピクチャがBスライス(たとえば、双予測スライス)である場合、ビデオエンコーダ20は、コロケートされたピクチャがどの参照ピクチャリストからのものであるか(たとえば、コロケートされたピクチャがRefPicList0からのものであるか、それともRefPicList1からのものであるか)を示すために、スライスヘッダ中でcollocated_from_l0_flagをシグナリングすることができる。ビデオデコーダ30は、コロケートされたピクチャがどの参照ピクチャからのものであるかを、スライスヘッダ中でシグナリングされたcollocated_from_l0_flagに基づいて決定することができる。

10

【0133】

[0139]参照ピクチャリストが識別された後、ビデオエンコーダ20は、スライスヘッダ中のcollocated_ref_id xシンタックス要素が、識別された参照ピクチャリスト中のピクチャを識別することになることをシグナリングすることができる。collocated_ref_id xシンタックス要素は、ビデオエンコーダ20がビットストリーム中でシグナリングし得る情報の一例である。ビデオデコーダ30は、collocated_ref_id xシンタックス要素に基づいて、識別された参照ピクチャリスト中でピクチャを識別ことができ、ここでこのcollocated_ref_id xシンタックス要素は、ビデオデコーダ30が受信するビットストリーム中の情報の一例である。

20

【0134】

[0140]ビデオエンコーダ20およびビデオデコーダ30は、コロケートされたピクチャをチェックすることによって、コロケートされた予測ユニット(PU)を識別することができる。コロケートされたPUは、コロケートされた領域の右下PU(たとえば、BRブロック)、またはいわゆる「中央3」もしくはCRブロック(すなわち、コロケートされた領域の中央の右下PU)のいずれかであり得る。

【0135】

[0141]ルーマロケーション(xP, yP)が、現在のピクチャの左上ルーマサンプルに対して現在のルーマ予測ブロックの左上サンプルを指定し、変数nPbWおよびnPbHが、それぞれ、ルーマ予測ブロックの幅および高さを指定すると仮定すると、BRブロックおよび中央3またはCRブロックは次のように定義され得る。

30

【0136】

- コロケートされた領域の右下PU :

【数2】

$$xPRb = xP + nPbW$$

$$yPRb = yP + nPbH$$

40

【0137】

- (yP >> Log2CtbSizeY)が(yPRb >> Log2CtbSizeY)に等しく、xPRbがpic_width_in_luma_samples未満である場合、以下が当てはまる。

【0138】

- コロケートされたPUは、colPicによって指定されたコロケートされたピクチャ内の((xPRb >> 4) << 4, (yPRb >> 4) << 4)によって与えられた修正ロケーションをカバーするルーマ予測ブロックを指定する。

【0139】

50

- 中央 3 :
【数 3】

$$\begin{aligned} xPCtr &= xP + (nPbW \gg 1) \\ yPCtr &= yP + (nPbH \gg 1) \end{aligned}$$

【0140】

- 変数であるコロケートされたPUは、colPic内の $((xPCtr \gg 4) < 4, (yPCtr \gg 4) < 4)$ によって与えられた修正ロケーションをカバーするルーマ予測ブロックを指定する。

10

【0141】

[0142]いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30が、AMVPまたはマージノスキップモードで動き候補を生成するために上記のプロセスによって動きベクトルを識別したとき、ビデオエンコーダ20およびビデオデコーダ30は、(POC値によって反映された)時間的ロケーションに基づいて動きベクトルをスケールすることができる。HEVCでは、シーケンスパラメータセット(SPS)は、フラグ`sp_s__temporal__mvp__enable__flag`を含み、スライスヘッダは、`sp_s__temporal__mvp__enable__flag`が1に等しいとき、フラグ`pic__temporal__mvp__enable__flag`を含む。ある特定のピクチャに対して`pic__temporal__mvp__enable__flag`と`temporal__id`の両方が0に等しいとき、復号順序においてその特定のピクチャの前にあるピクチャからの動きベクトルは、特定のピクチャ、または復号順序で特定のピクチャの後にあるピクチャの復号において、時間的動きベクトル予測子として使用されない。

20

【0142】

[0143]動きベクトル制限を使用するとき、マージノスキップモードまたはAMVPモードのいずれかにおいて動きベクトル予測子を使用する導出された動きベクトルが動きベクトル制限に準拠することを確実にすることに伴ういくつかの問題があり得る。たとえば、マージノスキップモードにおいて、現在のブロックについての動きベクトルが動きベクトル予測子に等しくセットされた場合に、現在のブロックについての動きベクトルが動きベクトル制限に違反することが考えられ得る。別の例として、AMVPモードにおいて、動きベクトル予測子が動きベクトル差分により加算または減算された場合に、得られる導出された動きベクトルが動きベクトル制限に違反することが考えられ得る。以下でより詳細に説明する動きベクトル制限を使用することに伴う他の問題があり得る。

30

【0143】

[0144]さらに、並列復号技法がHEVCベースのマルチビュービデオコーディング技法に拡張されるとき、いくつかの問題があり得る。たとえば、前述のように、H.264/AVCのMVC拡張の場合、ビデオエンコーダ20は、SEIメッセージ中で並列復号遅延情報をシグナリングし、並列復号される遅延は少なくとも2行である。しかしながら、HEVCベースの技法のコーディングユニットの階層構造が予定され、MV-HEVC、3DV-HEVC、またはSHVCには、マクロブロック行からブロック行への単純な拡張はない。

40

【0144】

[0145]また、JCT3V-B0037において説明されている技法では、ビデオエンコーダは、垂直視差動きベクトル範囲(すなわち、y成分のための動きベクトル制限)をシグナリングし得る。しかしながら、LCU行の遅延を決定することは、ビデオデコーダにとって単純ではないことがある。ビデオデコーダは、一例として、LCUサイズに基づいて、LCU行遅延を計算する必要がある。言い換えれば、LCU行の値は、参照ピクチャとしてビュー間またはレイヤ間ピクチャを使用するピクチャの並列復号をいつ開始す

50

べきかを決定するためにビデオデコード30が利用することができる値ではないことがあり、LCUサイズに基づく値にLCU行の値を追加的に変換することは、ビデオデコード30にとって、ピクチャの並列復号をいつ開始すべきかを決定するために必要とされ得る。しかしながら、LCU行に関して動きベクトル制限がシグナリングされる例には、ピクチャの並列復号をいつ開始すべきかを決定する統一的方法はないことがある。したがって、異なるビデオデコードは異なる時間に並列復号を開始することがあり、これは復号エラーをもたらす可能性がある。

【0145】

[0146]以下では、本開示による技法について説明する。たとえば、上記の問題および関連する問題、たとえば、動きベクトル制限に違反する動きベクトルを導出することに対処するためのマージノスキップモードおよびAMVPモードについての様々な技法が提案した。本開示はまた、並列復号をいつ開始すべきかを決定するためにLCU行に依存することの上記限界に対処する、並列復号を改良するための技法について説明する。

10

【0146】

[0147]特に、本開示は最初に、動きベクトル制限が利用されるときメモリ帯域幅低減に関係する技法について説明する。次いで本開示は、並列復号のサポートに関係する技法について説明する。動きベクトル制限が利用されるときメモリ帯域幅低減についての本開示で説明する技法は、並列復号についての本開示で説明する技法とは別個であり得ることを理解されたい。言い換えれば、ビデオデコード30が並列復号用に構成されていない場合でも、ビデオデコード30は依然として、本開示で説明するメモリ帯域幅低減技法を利用することができ、その逆も同様である。

20

【0147】

[0148]前述のように、動きベクトル制限の1つの潜在的利益は、メモリ帯域幅低減である。たとえば、動きベクトル制限の利益は、参照ピクチャ(すなわち、現在のピクチャのブロックをインター予測するためにブロックが使用されるピクチャ)がビュー間参照ピクチャまたはレイヤ間参照ピクチャである例において実現され得るが、本開示で説明する技法はそのように限定されない。たとえば、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分(たとえば、ピクチャ全体よりも小さい部分)を指すように動きベクトルの範囲を制限する。垂直視差動きベクトルが制約されるとき、ピクチャのピクセルにアクセスするために必要とされるメモリ帯域幅が低減され得る。言い換えれば、別のビュー中のピクチャを指す(すなわち、ビュー間参照ピクチャを指す)動きベクトルの範囲y成分が制限される場合、ビュー間参照ピクチャを記憶しているDPBから取り出される必要のあるデータの量が低減され得る。たとえば、範囲垂直成分が比較的小さい値に制限される場合、ビデオエンコーダ20およびビデオデコード30は、補間目的のために動きベクトルによって指されるブロックのDPB出力に記憶されたビュー間参照ピクチャのピクセルのピクセル値にアクセスする必要はないことがある。同じことは、ビュー間参照ピクチャではなくレイヤ間参照ピクチャが使用される場合にも当てはまり得る。

30

【0148】

[0149]説明しやすいように、動きベクトル制限によるメモリ帯域幅低減についての本開示で説明する技法は、参照ピクチャがビュー間またはレイヤ間参照ピクチャである例に関して説明される。さらに、いくつかの例では、HEVCに基づくスケーラブルビデオコーディング(SHVC)の場合、レイヤ間参照ピクチャを指す動きベクトルの一方または両方の成分(水平および垂直)は、常に0に制限され得る。いくつかの例では、HEVCに基づくマルチビュービデオコーディング(MV-HEVCまたは3DV-HEVC)の場合、ビュー間参照ピクチャを指す動きベクトルのための動きベクトル制限は選択可能であってよく、ビュー間参照ピクチャを指す動きベクトルのための制限範囲は選択可能であってよい。

40

【0149】

[0150]前述のように、ビュー間またはレイヤ間参照ピクチャに関する動きベクトル制限を使用する問題のうちの1つは、動きベクトル予測子から導出された動きベクトルが動き

50

ベクトル制限に違反し得ることである。本開示で説明する技法によれば、ビデオエンコーダ20が出力用に生成し、ビデオデコーダ30が(すなわち、リンク16を介して、またはストレージデバイス34を介して)受信するビットストリームは、現在のブロックについての導出された動きベクトルが動きベクトル制限に違反しないように制約され得る。言い換えれば、導出された動きベクトルが動きベクトル制限に準拠することを情報が確実にするように、ビットストリームは制約される。

【0150】

[0151]たとえば、本開示で説明する技法では、ビットストリームは、動きベクトルに動きベクトル制限を満たさないようにさせることになる、ビデオデコーダ30によるレイヤ間またはビュー間参照ピクチャでの動き補償に使用されることになる、動きベクトルまたは動きベクトルを形成する(すなわち、導出する)ために使用される他のシンタックスを含まないものとする。言い換えれば、ビットストリームは、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含み、ここで動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここで動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。たとえば、動きベクトル成分(すなわち、水平成分すなわちx成分および/または垂直成分すなわちy成分)の一方または両方がビデオデコーダ30による動き補償のために0に制限される場合、ビットストリームは、動きベクトルの水平成分および/または垂直成分を非0にさせる、動きベクトルまたは動きベクトルを導出するためのシンタックスを含むことはない。

【0151】

[0152]この例では、動きベクトルの範囲が0に制限され、このことは、導出された動きベクトルが、動きベクトル制限に準拠するように0の値を有しなければならないことを意味する。たとえば、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルを制限し、この例では、ビュー間またはレイヤ間参照ピクチャのサブ部分は、0の動きベクトルとなるように指されるものである。言い換えれば、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャ中の、現在のブロックが現在のピクチャに位置するのと同じ位置に位置するブロックのみを指すように動きベクトルを制限する。

【0152】

[0153]ビデオデコーダ30は、マージ/スキップモードまたはAMVPモードにおいて動きベクトル予測子から動きベクトルを導出することができる。本開示で説明する技法では、参照ピクチャがビュー間またはレイヤ間参照ピクチャであるときに、マージ/スキップモードまたはAMVPモードにおいて導出された動きベクトルが動きベクトル制限に違反するような形で、ビデオデコーダ30に動きベクトル予測子から動きベクトルを導出させる情報(たとえば、シンタックス要素)をビットストリームが含むことはない。

【0153】

[0154]たとえば、マージ/スキップモードでは、動き補償に使用されることになるマージ候補を識別するための候補動きベクトル予測子のリストへのマージインデックスを、ビデオエンコーダ20は出力用の生成されたビットストリーム中に含め、ビデオデコーダ30はビットストリームから受信する。候補動きベクトル予測子のリストへのインデックスにより、ビデオデコーダ30は、参照ピクチャリストと、参照ピクチャリストへのインデックスと、動きベクトルとを含む動き情報を決定することができる。

【0154】

[0155]マージ/スキップモードでは、ビデオデコーダ30は、選択された動きベクトル予測子の動き情報を採用する(すなわち、動きベクトルを導出するために動きベクトル予測子に等しく動きベクトルをセットする)。この例では、ビデオエンコーダ20は、候補動きベクトル予測子のリストへのインデックスが、ビュー間またはレイヤ間参照ピクチャについての参照ピクチャリストへのインデックスを識別するかどうかを決定することができる。候補動きベクトル予測子のリストへのインデックスが、ビュー間またはレイヤ間参

照ピクチャを指す動き情報を識別する場合、ビデオエンコーダ20は、候補動きベクトル予測子のリストへのインデックスによって識別された動きベクトル（すなわち、動きベクトル予測子）が動きベクトル制限に違反しているか、それとも動きベクトル制限に準拠しているかを決定することができる。動きベクトル予測子が動きベクトル制限に違反しているとビデオエンコーダ20が決定した場合、ビデオエンコーダ20は、動きベクトル制限に違反する導出された動きベクトルをもたらすことになる動きベクトル予測子をビデオデコーダ30に選択させることになる候補動きベクトル予測子のリストへのインデックスを含めることを回避すること（すなわち、含めないこと）ができる。

【0155】

[0156]このようにして、マージ/スキップモードの場合、ビデオデコーダ30が受信するビットストリームおよびビデオエンコーダ20が出力用に生成するビットストリームは、動きベクトル制限に違反する現在のブロックについての動きベクトルをビデオデコーダ30に導出させる情報を含むことはない。動きベクトル制限は、動きベクトルがビュー間またはレイヤ間参照ピクチャを指すときのためのものであり得る。たとえば、候補動きベクトル予測子のリストへのインデックスによって識別される動き情報の動きベクトルは、マージモードにおいて制限を満たさなければならない。一例として、制限は、SHVCの場合のように、水平成分と垂直成分の両方が0であるというものであり得る。

【0156】

[0157]言い換えれば、ビデオエンコーダ20は、値が動きベクトル制限に準拠する、動きベクトルを導出するために使用される、動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを決定することができる。これは、ビデオ復号30が動きベクトルを導出するために動きベクトル予測子に等しく動きベクトルをセットすることになるからである。したがって、動きベクトルが値を採用する動きベクトル予測子は、動きベクトル制限に準拠することが必要とされる。ビデオエンコーダ20は、候補動きベクトル予測子のリストへのインデックスを出力用に生成することができ、ビデオデコーダ30が受信するビットストリーム中の情報は、候補動きベクトル予測子のリストへのインデックスを含む。

【0157】

[0158]前述のように、AMVPは、マージ/スキップモードとは若干異なり得る。たとえば、マージ/スキップモードと同様に、ビデオエンコーダ20は、ビデオデコーダ30が動きベクトル予測子を選択する際の選択元となる候補動きベクトル予測子のリストへのインデックスをシグナリングし得る。マージ/スキップモードとは異なり、AMVPモードでは、ビデオデコーダ30は、現在のブロックについての動きベクトルとして、動きベクトル予測子を採用しない。代わりに、ビデオエンコーダ20は、ビットストリーム中で動きベクトル差分もシグナリングし、ビデオデコーダ30は、現在のブロックについての動きベクトルを導出するために、動きベクトル差分により動きベクトル予測子を加算または減算する。また、マージ/スキップモードとは異なり、AMVPモードでは、ビデオデコーダ30は、動きベクトル予測子の参照ピクチャリストおよび参照ピクチャリストインデックスを、参照ピクチャリストおよび参照ピクチャリストインデックスとして採用しない。代わりに、ビデオエンコーダ20は、参照ピクチャリストを識別し、参照ピクチャリストへのインデックスを識別する情報をシグナリングする。この情報から、ビデオデコーダ30は、現在のブロックについての動き情報を決定する。

【0158】

[0159]AMVPモードでは、ビデオエンコーダ20は、ビデオエンコーダ20によって識別される参照ピクチャリストへの、ビデオエンコーダ20がシグナリングすべき参照ピクチャリストインデックスが、ビュー間またはレイヤ間参照ピクチャを識別するかどうかを決定することができる。参照ピクチャリストインデックスがビュー間またはレイヤ間参照ピクチャを識別する場合、ビデオエンコーダ20は、動きベクトル差分が動きベクトル予測子に加算された、または動きベクトル予測子から減算されたときに、導出された動きベクトルが動きベクトル制限に準拠するような動きベクトル差分をシグナリングし得る。

【 0 1 5 9 】

[0160] 言い換えれば、ビデオエンコーダ 20 は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを決定することができ、動きベクトル差分が動きベクトル予測子に加算されたときに、得られる動きベクトルが動きベクトル制限に準拠するように、動きベクトル差分を決定することができる。

【 0 1 6 0 】

[0161] ビデオエンコーダ 20 は、候補動きベクトル予測子のリストへのインデックスを出力用に生成することができ、ビデオデコーダ 30 が受信するビットストリーム中の情報は、候補動きベクトル予測子のリストへのインデックスを含む。ビデオエンコーダ 20 はまた、動きベクトル差分を出力用に生成することができ、ビデオデコーダ 30 が受信するビットストリーム中の情報は、動きベクトル差分を含む。

10

【 0 1 6 1 】

[0162] ビデオデコーダ 30 は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを、受信されたビットストリームから決定することができ、値が、動きベクトル差分が動きベクトル予測子に加算されたときに、得られる動きベクトルが動きベクトル制限に準拠するようなものである動きベクトル差分を、受信されたビットストリームから決定することができる。ビデオデコーダ 30 は、動きベクトル予測子と動きベクトル差分とを加算することによって、動きベクトルを導出することができる。

20

【 0 1 6 2 】

[0163] ここでも、AMVPモードでは、候補動きベクトル予測子のリストへのインデックスが、動きベクトル予測子を選択するために使用され、参照ピクチャリストを識別する情報および参照ピクチャリストへの参照ピクチャリストインデックスを識別する情報も、参照ピクチャを選択するためにシグナリングされる。本開示で説明する技法では、ビデオエンコーダ 20 は、選択された参照ピクチャがビュー間またはレイヤ間参照ピクチャであるかどうかを決定することができる。選択された参照ピクチャがビュー間またはレイヤ間参照ピクチャである場合、ビデオエンコーダ 20 は、候補動きベクトル予測子のリストへのインデックスを介して選択された動きベクトル予測子に加算されたとき、または動きベクトル予測子から減算されたときに、動きベクトル制限に準拠する動きベクトルをビデオデコーダ 30 に導出させる動きベクトル差分を決定することができる。

30

【 0 1 6 3 】

[0164] 一例として、レイヤ間参照ピクチャを指すブロックについての動きベクトルが 0 である（すなわち、 x 成分と y 成分の両方が 0 である）ことを動きベクトル制限が要求すると仮定する。また、ビデオエンコーダ 20 が、候補動きベクトル予測子のリストへのインデックスを決定しており、ビデオデコーダ 30 が、候補動きベクトル予測子のリストへのインデックスから動きベクトル予測子（MVP）を識別したと仮定する。この例では、AMVPモードにおいて、ビデオエンコーダ 20 は、動きベクトル差分（MVD）が MVP に加算されたときに、得られる動きベクトルが 0 に等しくなる（すなわち、MVD の x 成分が MVP の x 成分の負であり、MVD の y 成分が MVP の y 成分の負である）ような MVD をシグナリングすることができる。

40

【 0 1 6 4 】

[0165] このようにして、AMVP の場合、ビデオデコーダ 30 が受信するビットストリームおよびビデオエンコーダ 20 が出力用に生成するビットストリームは、動きベクトル制限に違反する現在のブロックについての動きベクトルをビデオデコーダ 30 に導出させる情報を含むことはない。動きベクトル制限は、動きベクトルがビュー間またはレイヤ間参照ピクチャを指すときのためのものであり得る。たとえば、動きベクトル差分は、導出された動きベクトルが動きベクトル制限に準拠するように動きベクトル予測子を補償することができる。言い換えれば、本開示で説明する技法では、合わせると動きベクトル制限を満たさない MVP および MVD がビットストリーム中に存在すべきではなく、ここでビ

50

ットストリーム中の候補動きベクトル予測子のリストへのインデックスはMVPを識別し、ここでMVDはビットストリーム中に含まれる。

【0165】

[0166] マージノスキップモードおよびAMVPで動きベクトル制限を使用するとき、対処される必要があるいくつかの特別の場合があり得る。マージノスキップモードでは、TMVPに関するさらなる制約が必要とされることがあり、AMVPモードでは、ビットストリーム中に含まれるフラグに関するさらなる制約が必要とされることがある。次に、各々についてより詳しく説明する。

【0166】

[0167] 前述のように、ビデオエンコーダ20およびビデオデコーダ30は、空間的隣接ブロックと時間的隣接ブロックとを含む隣接ブロックの動き情報に基づいて、候補動きベクトル予測子のリストを構成する。時間的隣接ブロックの動きベクトルは、時間的動きベクトル予測子(TMVP)と呼ばれる。空間的隣接ブロック(すなわち、SMVP)からの動きベクトル予測子の場合、空間的隣接ブロックが現在のブロックと同じピクチャ中にあるので、それらが指す参照ピクチャがピクチャの参照ピクチャリスト中で識別されるという保証がある。また、空間的隣接ブロックについての動きベクトルがビュー間またはレイヤ間参照ピクチャを指す場合、空間的隣接ブロックについての動きベクトルが動きベクトル制限に適合するという保証がある。

【0167】

[0168] しかしながら、時間的隣接ブロックの動きベクトルの場合、動きベクトルが現在のピクチャの参照ピクチャリストのうちの1つにおける参照ピクチャを指すという保証はないことがある。さらに、時間的隣接ブロックについての動きベクトルが動きベクトル制限に違反することがある。

【0168】

[0169] AMVPモードでは、参照ピクチャリストおよび参照インデックスがシグナリングされるので、TMVPが選択された場合でも、導出された動きベクトルは依然として、現在のピクチャの参照ピクチャリスト中で利用可能なピクチャを参照することになる。また、AMVPモードでは、TMVPが動きベクトル制限に違反する場合、動きベクトル差分は、導出された動きベクトルが動きベクトル制限に準拠するように、TMVPを補償することができる。

【0169】

[0170] マージノスキップモードでは、そのような制御がない(たとえば、TMVPが動きベクトル制限に違反する場合にTMVPを補償する方法がない)ことがある。したがって、マージノスキップモードでは、TMVP(たとえば、時間的マージ候補)を選択することは、ビュー間またはレイヤ間参照ピクチャではない長期参照ピクチャを指す動きベクトルを時間的隣接ブロック(すなわち、コロケートされたブロック)が含み得るので、外部、動きベクトル範囲という動きベクトルをもたらず(すなわち、視差動きベクトルまたは別のレイヤを指す動きベクトルが動きベクトル制限に違反する)ことがある。

【0170】

[0171] たとえば、マージノスキップモードでは、TMVPが選択された場合、ビデオエンコーダ20およびビデオデコーダ30は、参照ピクチャリストと参照ピクチャリストインデックスとを決定する必要がある。マージノスキップモードでは、ビデオエンコーダ20およびビデオデコーダ30は、TMVPの参照ピクチャリストと参照ピクチャリストインデックスとを使用するのではなく、デフォルト参照ピクチャリストとデフォルトインデックス(たとえば、HEVCにおけるインデックス0)とを利用することができる。TMVPに使用されるデフォルトインデックスは、デフォルトインデックスが必ずしもインデックス0に限定される必要はないので、TMVP参照インデックスと呼ばれることがある。

【0171】

[0172] TMVPが動きベクトル制限に準拠することを確実にする1つの技法は、動きベ

10

20

30

40

50

クトルが動きベクトル範囲内に収まるように、ビデオエンコーダ20およびビデオデコーダ30が動きベクトルクリッピングを実施することである。しかしながら、そのようなTMVPクリッピングは、上述のハイレベルシンタックスのみ(HLSのみ)変更では許容されないピクチャのブロックレベルでのビデオコーディング変更を必要とすることになる。言い換えれば、TMVPクリッピングは、3DV-HEVCではうまく機能するが、MV-HEVCまたはSHVCではうまく機能しないことがある。

【0172】

[0173]ブロックレベルの変更を回避し、HLSのみ要件に準拠するために、TMVPが動きベクトル制限の違反を引き起こさないように、参照ピクチャリストおよび/またはどのピクチャがインター予測に使用され得るかに対する追加の制約があり得る。一例として、ビデオエンコーダ20およびビデオデコーダ30は、ビュー間参照ピクチャではない通常の時間的長期参照ピクチャを、ビュー間予測を使用する任意のビュー(すなわち、任意のピクチャ)に使用することはない。同じことは、レイヤ間参照ピクチャでのレイヤ間予測を任意のレイヤに使用する任意のピクチャにも当てはまり得る。別の例として、ビデオエンコーダ20およびビデオデコーダ30がTMVP中に別のピクチャのためにコロケートされたピクチャ(たとえば、時間的ピクチャ)として使用する任意のピクチャは、長期参照(時間的)ピクチャを、その参照ピクチャリストのうちのいずれの中にも有することができない。

【0173】

[0174]別の例として、ビデオエンコーダ20およびビデオデコーダ30は、ビュー間またはレイヤ間参照ピクチャ(たとえば、ベースビューまたはベースレイヤからの参照ピクチャ)を、TMVP参照インデックスに等しい参照インデックスを有する参照ピクチャリスト中に含めることはない。たとえば、前述のように、TMVPについてのデフォルト参照インデックスはHEVCにおいて0(すなわち、参照ピクチャリスト中の最初のエントリ)である。いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、ビュー間またはレイヤ間参照ピクチャを、参照ピクチャリストの0の参照インデックス中に含めることはない。

【0174】

[0175]また別の例として、ビデオエンコーダ20およびビデオデコーダ30は、候補動きベクトル予測子のリストにTMVPを含めることを回避することができる。たとえば、ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル制限が適用され、参照ピクチャリストのTMVP参照インデックス(たとえば、HEVCにおける0のインデックス)がビュー間またはレイヤ間参照ピクチャ(たとえば、ベースビュー中またはベースレイヤ中のピクチャ)を識別する場合、TMVPを無効にすることができる。

【0175】

[0176]マージ/スキップモードについてのTMVPに対処するための上記技法は、単に例示のために提供されている。これらの技法は、別の技法とともに使用されること、または別個に使用されることがある。

【0176】

[0177]AMVPモードの場合、動きベクトル差分(MVD)制御に関する追加の制約があり得る。前述のように、動きベクトル(たとえば、ベストマッチ探索を使用して発見されたもの)と動きベクトル予測子(MVP)との間の動きベクトル差分(MVD)を、ビデオエンコーダ20はシグナリングし、ビデオデコーダ30は受信する。HEVC規格は、RefPicList1中で識別されたピクチャについてのMVDがシグナリングされず、0であると推測されることを示すmvd_l1_zero_flagシンタックス要素を使用する。言い換えれば、mvd_l1_zero_flagが真(たとえば、デジタルの1)である場合、ビデオデコーダ30は、参照ピクチャがRefPicList1中にある場合に、MVDは0であると決定する。この場合、MVDは0であると推測されるので、導出された動きベクトルはMVPに等しい。

【0177】

[0178]しかしながら、動きベクトル制限が有効化されている場合、動きベクトル制限に違反するMVPの成分を補償するためにMVDを使用する必要があるあり得る。たとえば、垂直MV成分が制限されることになる場合、ビデオデコーダ30がMVPの場合によっては大きい垂直成分をMVDにより補償して、垂直方向における最終的な導出された動きベクトルが動きベクトル制限を満たすようにすることが可能であるべきである。したがって、いくつかの例では、ビデオエンコーダ20は、動きベクトル制限が有効化されているときには常にMVDをシグナリングすることができ、`mvd__l1__zero__flag`は、動きベクトル制限により有効化されることはない。

【0178】

[0179]したがって、いくつかの例では、`mvd__l1__zero__flag`に対する追加の制約が必要とされ得る。たとえば、`mvd__l1__zero__flag`は、ビュー間予測を使用する任意のビューコンポーネント（たとえば、テクスチャビューコンポーネントまたは深度ビューコンポーネント）について、常に0に等しくなるように制約され得る。別の例として、動きベクトル制限が有効化されている場合、ビデオエンコーダ20およびビデオデコーダ30は、`mvd__l1__zero__flag`の値を0に等しいと推測することができ、ビットストリーム中で`mvd__l1__zero__flag`の値を、ビデオエンコーダ20がシグナリングすることはなく、ビデオデコーダ30が受信することはない。別の例として、`mvd__l1__zero__flag`は依然としてシグナリングされるが、無効化された値（たとえば、0）を伴うことがある。

【0179】

[0180]別の例として、本技法は逆を適用し得る。たとえば、`mvd__l1__zero__flag`が真であるべきである場合、動きベクトル制限は有効化されることはない。言い換えれば、ビデオエンコーダ20は、`mvd__l1__zero__flag`が真である場合に、動きベクトル制限が有効化されていることを示す動きベクトル制限フラグをシグナリングすることはなく、`mvd__l1__zero__flag`が偽である場合にのみ、動きベクトル制限が有効化されていることを示す動きベクトル制限フラグをシグナリングすることができる。

【0180】

[0181]ビデオエンコーダ20は、以下の方法で、`mvd__l1__zero__flag`が偽である場合にのみ、動きベクトル制限が有効化されていることを示すフラグをシグナリングする条件を実装し得る。

【0181】

(`profile_idc` = 「マルチビューサポートプロファイル」 && `mvd__l1__zero__flag` = 偽) の場合

【数4】

```

{
    interview_vector_v_restrict_flag                u(1)
    (interview__vector__v__restrict__flag==1)の場合
    {
        max_interview_vector_v_length_div4        u(4)
    }
}

```

【0182】

[0182]別の例として、単純化された技法は、動きベクトル制限が有効化されているとき

に、(AMVPモードにおける)明示的にシグナリングされる参照インデックスがビュー間またはレイヤ間参照ピクチャを指すことはできないというものであり得る。たとえば、AMVPモードでは、ビデオエンコーダ20は、動きベクトル制限が有効化されているかどうかを決定することができる。動きベクトル制限が有効化されている場合、ビデオエンコーダ20は、ビュー間またはレイヤ間参照ピクチャを指さない参照ピクチャリストへの参照インデックスを決定することができる。この例では、ビデオエンコーダ20がビュー間またはレイヤ間参照ピクチャを選択しないことになるので、`mv_d__l1__zero__flag`は0に等しくなるように制約されなくてよい。したがって、導出された動きベクトルはビュー間またはレイヤ間参照ピクチャを指さないことになり、ここで動きベクトル制限は適用可能ではないことがある。

10

【0183】

[0183]すぐ上の例示的な技法は、参照インデックスに対して制約的であると見なされ得る。この例では、ビデオエンコーダ20は、(負の垂直値成分は、参照ビューについてのすでに再構成されたブロックを指し得るので)絶対値または代替的に正の値のみを動きベクトル制限要件と比較することができる。動きベクトルが動きベクトル制限の条件で満たされない場合のみ、ビュー間またはレイヤ間参照ピクチャ(たとえば、ベースビューまたはベースレイヤ)を指す参照インデックスは、使用することが許されない。たとえば、`mv_d__l1__zero__flag`が1に等しく、参照インデックスがビュー間またはレイヤ間参照ピクチャを指す場合の導出された動きベクトルが動きベクトル制限ルールに適合することが考えられ得る。`mv_d__l1__zero__flag`が真である場合でも、導出された動きベクトルが動きベクトル制限ルールに適合する場合には、ビデオエンコーダ20は、ビュー間またはレイヤ間参照ピクチャを指す参照インデックスを選択することができる。

20

【0184】

[0184]前述のように、マルチビュービデオコーディングの場合、動きベクトル制限の使用は、選択可能であってよく、動きベクトル制限が有効化されていることを示すフラグまたは他のシンタックス要素によりビットストリーム中で示され得る。スケーラブルビデオコーディングの場合、動きベクトル制限は、永続的に有効化されることがあり、いくつかの例では、動きベクトル制限要件は、導出された動きベクトルが0ベクトルである(たとえば、動きベクトルがレイヤ間参照ピクチャを指す場合に、導出された動きベクトルの水平(x)成分および垂直(y)成分が両方とも0に等しい)というものであり得る。

30

【0185】

[0185]上述した例示的なMVD制御技法は、スケーラブルビデオコーディング技法(たとえば、SHVC)でも使用され得る。たとえば、動きベクトルが制約的である(たとえば、動きベクトル制限が使用されている)場合、ビデオエンコーダ20は、導出された動きベクトルが制約を満たす(たとえば、 $MV = MVD + MVP = 0$)ようにMVDによる動きベクトル予測子の補償を可能にするために、`mv_d__l1__zero__flag`を0にセットすることができる。言い換えれば、ビデオエンコーダ20は、動きベクトル制限に違反する導出された動きベクトルをもたらすMVPとMVDとの組合せを回避することができる。

40

【0186】

[0186]別の例として、ビデオエンコーダ20は、`mv_d__l1__zero__flag`が有効化されている場合に、導出された動きベクトルに動きベクトル制限に違反させることになる動きベクトル予測子の選択を回避することができる。たとえば、`mv_d__l1__zero__flag`が有効化されている場合、ビデオエンコーダ20は、制約(すなわち、動きベクトル制限)を満たすことができる動きベクトル予測子(たとえば、AMVP候補)のみを選択することができる。

【0187】

[0187]たとえば、1つの例は、レイヤ間参照ピクチャについて、インター予測を形成するために使用されるMVを0に制限することができる。これは、AMVPモードの場合に

50

MV = MVP + MVD が 0 であることを意味する。前述のように、ビデオデコーダ 30 は、ビデオエンコーダ 20 によってシグナリングされ、ビデオデコーダ 30 によって受信された MVP インデックス（すなわち、候補動きベクトル予測子のリストへのインデックス）に基づいて MVP を決定する。参照リスト L1（すなわち、RefPicList1）の場合、参照インデックスがレイヤ間参照ピクチャを指すとき、MV は 0 に等しくなるべきであるので、有効化された `mv_d_l1_zero_flag` により、ビデオエンコーダ 20 は、値が 0 である（すなわち、水平成分および垂直成分が 0 に等しい）動きベクトル予測子を指す候補動きベクトル予測子のリストへのインデックスのみを選択することができる。このようにして、有効化されている `mv_d_l1_zero_flag` から、MVD は 0 に等しいと推測されるので、MVP が 0 に等しいので、導出された動きベクトルは 0 に等しくなる（すなわち、 $MV = MVP + MVD$ 、および $MVP = 0$ 、および $MVD = 0$ 、したがって $MV = 0$ ）。したがって、ビデオエンコーダ 20 は、有効化された `mv_d_l1_zero_flag` と、レイヤ間参照ピクチャについて 0 に制約された MV とにより、非 0 の AMVP 候補（すなわち、MVP は 0 に等しくない）を使用するレイヤ間参照ピクチャを指す参照インデックスについて、MVP インデックス（すなわち、候補動きベクトル予測子のリストへのインデックス）を使用およびシグナリングすることはない。

10

【0188】

[0188]別の例として、MV 制約（すなわち、動きベクトル制限）を使用するレイヤについて `mv_d_l1_zero_flag` が無効化される（0 としてシグナリングされる）こともある。また別の例として、ビデオエンコーダ 20 は、ビュー間またはレイヤ間（たとえば、ベースビュー/レイヤ）についてのみ `mv_d_l1_zero_flag` をシグナリングすることができ、MV 制約のあるエンハンスメントビュー/レイヤについて、`mv_d_l1_zero_flag` シグナリングは省略されてよく、`mv_d_l1_zero_flag` の値は 0 であると推測される。

20

【0189】

[0189]別の例として、ビデオエンコーダ 20 は、参照ピクチャリスト L1（RefPicList）が少なくとも 1 つのビュー間/レイヤ間参照ピクチャを含む場合、MV 制約のあるエンハンスメントビュー/レイヤについて、0 であるか、または 0 であると推測される `mv_d_l1_zero_flag` のみをシグナリングすることができる。また別の例として、ビデオエンコーダ 20 およびビデオデコーダ 30 は、参照ピクチャに従って別様に `mv_d_l1_zero_flag` を扱うことができる。たとえば、参照ピクチャがビュー間/レイヤ間ピクチャである場合、`mv_d_l1_zero_flag` に関係なく、ビデオエンコーダ 20 は常に MVD 値をシグナリングすることができる。

30

【0190】

[0190]同様の特徴は MVC 拡張にも適用され得る。また、MV は、上記の例において 0 以外の他の値に対して制約的であってよく、制約は、垂直 MV 成分、水平 MV 成分、または両方について行われ得る。言い換えれば、動きベクトル制限は、導出された動きベクトルが非 0 値に制限されることを要求し得る。また、制限は、y 成分のみ、x 成分のみに適用可能であること、または x 成分と y 成分の両方に適用可能であることがある。さらに、一方の成分の制限は必ずしも、他方の成分の制限に等しくなる必要はない。

40

【0191】

[0191]一般に、いくつかの例では、ビデオエンコーダ 20 は、参照ピクチャリスト（たとえば、RefPicList1）中のピクチャについての動きベクトル差分が 0 ではないことをフラグの値（たとえば、`mv_d_l1_zero_flag`）が常に示すことを決定することがあり、フラグの値を出力用に生成することがある。ビデオデコーダ 30 の観点から、ビデオデコーダ 30 は、参照ピクチャリスト中で参照ピクチャリスト（たとえば、RefPicList1）中で識別されたピクチャについての動きベクトル差分が 0 であるかどうかを示すフラグ値（たとえば、`mv_d_l1_zero_flag`）を、受信されたビットストリームから決定し、参照ピクチャリスト中のピクチャについての動き

50

ベクトル差分が0ではないことをフラグ値が常に示すことを決定し得る。

【0192】

[0192]いくつかの例では、動きベクトル制限では、ビュー間またはレイヤ間参照ピクチャ（たとえば、ベースビューまたはベースレイヤ参照ピクチャ）の配置は、TMVPに関して上述したビュー間またはレイヤ間参照ピクチャの配置に加えて、メモリ帯域幅低減に影響を与え得る。たとえば、メモリ帯域幅低減にとって重要なケースは、最悪ケースシナリオであり得る。あるいは、ビデオデコーダ30にとって、何らかの他の非最悪ケース条件が考慮される場合でも、最悪ケースの実装は依然として考慮される必要があることがあり、さもなければ、実際の単純化またはメモリ帯域幅低減は達成されないことがある。この目的で、動きベクトル制限（たとえば、マルチビュービデオコーディングの場合の視差動きベクトル範囲またはスケーラブルビデオコーディングの場合のレイヤ間動きベクトル範囲）から恩恵を受けるために、動きベクトル制限が適用されるときに、参照ピクチャに対するいくつかの制約があり得る。

10

【0193】

[0193]たとえば、ビュー間予測を使用しないビューの各ビューコンポーネント（すなわち、テクスチャビューコンポーネントまたは深度ビューコンポーネント）についての異なる時間的参照ピクチャの最大数が $maxNumTempPic0$ であると仮定し、ビュー間予測によりコーディングされた各ビューコンポーネントについての異なる時間的参照ピクチャの最大数が $maxNumTempPic1$ であると仮定する。いくつかの例では、 $maxNumTempPic0$ は $maxNumTempPic1$ 以下となるべきである。

20

【0194】

[0194]そのため、ビュー間またはレイヤ間参照ピクチャ（たとえば、ベースビューまたはベースレイヤ参照ピクチャ）を指す動きベクトルの制限垂直成分による実際のメモリ帯域幅低減を行うためには、そのビュー間またはレイヤ間参照ピクチャは、少なくとも1つの参照リストに存在する必要がある。そうではなく、ビュー間またはレイヤ間参照ピクチャが参照ピクチャリストに挿入されていない場合、たとえば、他の時間的参照ピクチャが代わりに使用され得る場合、ビュー間またはレイヤ間参照ピクチャを指す動きベクトルのための動きベクトル制限は、この例では参照リスト中に参照ピクチャとしてすべての時間的ピクチャを有することになる最悪ケースに対処しない。

30

【0195】

[0195]言い換えれば、動きベクトル制限が有効化されている場合、メモリ帯域幅低減などの動きベクトル制限の利益は、ビュー間またはレイヤ間参照ピクチャが参照ピクチャリストのうちの一つ中に含まれている場合にのみ実現可能であり得る。そうでない場合、参照ピクチャリスト中のすべてのピクチャは時間的ピクチャとなり、動きベクトル制限は適用可能ではなくなる。したがって、いくつかの例では、動きベクトル制限が有効化されている場合、ビデオエンコーダ20およびビデオデコーダ30は、少なくとも一つのビュー間またはレイヤ間参照ピクチャを参照ピクチャリストのうちの一つ中に含めることを必要とするように構成され得る。

【0196】

[0196]参照ピクチャリストに対する1つの例示的な制約として、非ベースビューまたはレイヤ（たとえば、従属ビューまたは従属/エンハンスメントレイヤ）のすべてのビューコンポーネントは、最大で2つの時間的参照ピクチャと1つのビュー間またはレイヤ間参照ピクチャ（たとえば、ベースビューまたはベースレイヤ）とを有する。参照ピクチャリストに対する別の例示的な制約として、非ベースビューまたはレイヤのすべてのビューコンポーネントは、最大で2つの時間的参照ピクチャを有し、（1つのビュー間またはレイヤ間参照ピクチャにより）ビュー間またはレイヤ間予測されるビューコンポーネントもあれば、ビュー間またはレイヤ間予測されないビューコンポーネントもある。

40

【0197】

[0197]前述のように、マルチビューコーディングの場合、動きベクトル制限の使用は選

50

択可能であり得る。いくつかの例では、メモリアクセスを容易にするために、LCUごとにピクセルメモリが構成されているときは特に、視差動きベクトルが0の垂直成分を有するときに（または有するときにのみ）メモリ帯域幅低減の利益があり得る。視差動きベクトル範囲のシグナリングの単純化のために、いくつかの例では、ビデオエンコーダ20は、補足エンハンスメント情報（SEI）メッセージ、シーケンスパラメータセット、ビデオパラメータセット、またはビデオユーザビリティ情報（VUI: video usability information）メッセージ中で、視差動きベクトルのy成分が0に制限されることを示すフラグをシグナリングすることができる。

【0198】

[0198]以下の擬似コードは、視差動きベクトルのy成分が0に制限されることを示すフラグをSEIメッセージ中に含めるための方法を示す。

10

【表1】

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
zero_vertical_disparity_mv_flag	u(1)
(! zero__vertical__disparity__mv__flag) の場合	
pdi_init_delay_ctb_vertical_minus3	ue(v)
pdi_init_delay_ctb_horizontal_minus2	ue(v)
}	

20

【0199】

[0199]1に等しいzero__vertical__disparity__mv__flagは、0に等しい垂直成分を視差動きベクトルが常に有することを示す。0に等しいzero__vertical__disparity__mv__flagは、0に等しくない垂直成分を視差動きベクトルが有し得ることを示す。

30

【0200】

[0200]video__parameter__set__idは、ビュー間従属関係情報を含むビデオパラメータセットを指定する。video__parameter__set__idの値は、並列復号情報SEIメッセージを含むアクセスユニットのコーディングされたピクチャのビューコンポーネントによって参照されるvideo__parameter__set__idの値に等しいものとする。

40

【0201】

[0201]pdi__init__delay__ctb__vertical__minus3およびpdi__init__delay__ctb__horizontal__minus2は、アクティブビデオパラメータセット識別子で指定されるようなビュー間予測を使用するコーディングされたビューコンポーネントによってビュー間参照に使用されないものとする任意の参照ビューコンポーネント中の利用不可能な参照エリアが、現在のSEIメッセージ中に含まれるシンタックス要素video__parameter__set__idに等しいことを指定する。

【0202】

[0202]変数horCtbは次のように導出される。horCtb = pdi__init__

50

`delay__ctb__horizontal__minus2?pd_i__init__delay__ctb__horizontal__minus2+2+(CtbAddrInRS%PicWidthInCtbsY):0`。変数 `verCtb` は次のように導出される。

$$verCtb = CtbAddrInRS / PicWidthInCtbsY + pd_i__init__delay__ctb__vertical__minus3 + 3。$$

【0203】

[0203] `pd_i__init__delay__ctb__vertical__minus3` は、存在しないときには -1 と推測され、または同等に、`verCtb` は、`CtbAddrInRS / PicWidthInCtbsY + 2` と推測される。

【0204】

[0204]このようにして、ビットストリーム制約ルールは、以下の制限セットとして公式化され得る。(たとえば、マージ/スキップまたはAMVPモードから参照インデックスによって識別される)レイヤ間/ビュー間参照ピクチャでの動き補償に使用されるべき動きベクトルに対する制限がある場合、マージ/スキップモードが使用される場合には、動きベクトル制限を満たさない動きベクトルをもたらすことになるマージ/スキップインデックス(すなわち、マージ/スキップモードでの候補動きベクトル予測子のリストへのインデックス)を、またはAMVPモードが使用される場合には、動きベクトル制限を満たさない動きベクトルをもたらすことになるMVPインデックス(すなわち、AMVPモードでの候補動きベクトル予測子のリストへのインデックス)およびMVDを、ビットストリームは含むことはない(たとえば、含まないものとする)。さらに、 $MV = MVP + MVD$ が参照リスト `L1 (RefPicList1)` についてのMV制約(すなわち、動きベクトル制限)を満たすように非0のMVDを有することが必要とされ得るので、ビットストリームは、`mv_d_l1_zero_flag` について(このフラグが無効化されていることを意味して)非0値を含むことはない(たとえば、含まないものとする)。

【0205】

[0205]たとえば、動きベクトルの一方または両方の成分がレイヤ間/ビュー間参照ピクチャについて0になるように制限される場合、マージモードの場合には、非0の動きベクトルもしくは動きベクトル成分をもたらすことになるマージインデックスを、またはAMVPモードの場合には、非0の動きベクトルもしくは動きベクトル成分をもたらすことになるMVPインデックスおよびMVDを、ビットストリームは含むことはない(たとえば、含まないものとする)。さらに、 $MV = MVP + MVD$ が参照リスト `L1 (RefPicList1)` について0に等しくなるように非0のMVDを有することが必要とされ得るので、ビットストリームは、`mv_d_l1_zero_flag` について(このフラグが無効化されていることを意味して)非0値を含むことはない(たとえば、含まないものとする)。

【0206】

[0206]上述の制約の一部または全部は、ビットストリームに存在するシンタックス要素(たとえば、情報)に課せられ得る。たとえば、ビットストリームについての制約ルールは次のように公式化され得る。あらゆる参照ピクチャリスト `RefPicListX` について、`X` は0または1のいずれかであり、マージインデックス(`merge_idx`)、MVPインデックス(`mv_p_lX_flag`)、参照インデックス(`ref_idx_lX`)、MVD(`Mv_dLX`)、および `mv_d_l1_zero_flag` はMVに、動きベクトル制限を満たす動きベクトル成分、たとえば、いずれも0に等しい成分(たとえば、SHVCの場合)を提供するものとする。

【0207】

[0207]言い換えれば、ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、候補動きベクトル予測子のリストを構成することができる。マージ/スキップモードでは、ビデオエンコーダ20は、値が動きベクトル制限に準拠する、動きベクトルを導出するために使用される、動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを決定することができ、候補動きベクトル予測子のリストへのインデックスを

10

20

30

40

50

出力用に生成する。マージ/スキップモードでは、ビデオデコーダ30は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを、受信されたビットストリームから決定することができる。この例では、ビデオデコーダ30は、動きベクトル予測子に等しく動きベクトルをセットすることができ、その上、動きベクトル予測子は、動きベクトル制限に準拠している。

【0208】

[0208] A M V P モードの場合、ビデオエンコーダ20は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを決定することができる。ビデオエンコーダ20は、動きベクトル差分が動きベクトル予測子に加算されたときに、得られる動きベクトルが動きベクトル制限に準拠するように、動きベクトル差分を決定することもできる。ビデオエンコーダ20は、候補動きベクトル予測子のリストへのインデックスと動きベクトル差分とを出力用に生成することもできる。A M V P モードでは、ビデオデコーダ30は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスを、受信されたビットストリームから決定することができ、その値が、動きベクトル差分が動きベクトルに加算されて、得られる動きベクトルが動きベクトル制限に準拠するようなものである動きベクトル差分を、受信されたビットストリームから決定する。ビデオデコーダ30は、動きベクトル予測子と動きベクトル差分とを加算することによって、動きベクトルを導出することができる。

10

【0209】

[0209]説明はすべて、H L Sのみソリューション(ブロックレベルでの変更がない)またはブロックレベルでの変更が許容されるH L Sソリューションとして実現されるH E V C S V C 拡張に有効である。ベースレイヤコーデックは、H E V C、A V C、M P E G - 2 などであり得る。制限は、垂直動きベクトル(M V)成分、水平M V成分、または両方に適用されることがあり、また、参照リストL 0 (R e f P i c L i s t 0)、参照リストL 1 (R e f P i c L i s t 1)、または両方を含み得る。しかしながら、S V Cの場合、レイヤ間参照ピクチャを指すM Vの両方の成分について、0の動きベクトルとする制限は、使用事例のうちの1つである。

20

【0210】

[0210]したがって、いくつかの例では、ビデオエンコーダ20は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための、導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定するように構成され得る。これらの例では、動きベクトルは、ビュー間またはレイヤ間参照ピクチャ(たとえば、現在のブロックを含むピクチャに対する別のビューまたはレイヤ中のピクチャ)を指し得る。また、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャを指す動きベクトルに関連付けられる。

30

【0211】

[0211]ビデオエンコーダ20は、情報に基づいて、現在のブロックについての動きベクトルを導出し、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号することができる。ビデオエンコーダ20はまた、動きベクトルを導出するための決定された情報を出力用に生成することもできる。

40

【0212】

[0212]いくつかの例では、ビデオデコーダ30は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信するように構成され得る。上記と同様に、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャを指す動きベクトルに関連付けられる。本開示で説明する技法では、ビットストリーム中の情報は、導出された動きベクトルが動きベクトル制限に準拠することを確実にする。

【0213】

[0213]本開示で説明する技法では、ビデオデコーダ30は、ビットストリーム中の情報

50

に基づいて、現在のブロックについての動きベクトルを導出することができる。ビデオデコーダ30はまた、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号することができる。

【0214】

[0214]上記では、メモリ帯域幅低減のための動きベクトル制限に関して説明した。以下では、並列復号に関する技法について説明する。メモリ帯域幅低減のための動きベクトル制限に関して説明した技法は必ずしも、並列復号とともに使用される必要はない。言い換えれば、ビデオエンコーダ20およびビデオデコーダ30は、ビデオエンコーダ20およびビデオデコーダ30が並列復号を実施しない場合でも、動きベクトル制限を実施するように構成され得る。

10

【0215】

[0215]前述のように、いくつかの技法は、並列復号遅延を示すためにマクロブロック行に依存している。しかしながら、マクロブロック行に依存することは、HEVCベースのビデオコーディング技法にとって実現可能ではないことがある。また、いくつかの技法は、LCU行における並列復号遅延をシグナリングしたが、そのような値は依然として、並列復号をいつ開始すべきかをビデオデコーダ30が決定できるように、LCUサイズに基づいて変換される必要があった。

【0216】

[0216]たとえば、いくつかの技法は(たとえば、JCT3V-B0037において)動きベクトル制限を直接シグナリングした。動きベクトル制限のこの直接シグナリングは、動きベクトル成分の値の制限をシグナリングすることを意味する。そのような技法は、動きベクトル成分の値の制限をLCU行のユニットに変換することを必要とする可能性が非常に高く、このことは、一例として、LCUサイズに基づいてLCU行遅延が計算されることが依然として必要となるので、問題があり得る。

20

【0217】

[0217]いくつかの例では、本開示で説明する技法によれば、ブロックサイズに基づくユニットにおける動きベクトル制限情報を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。動きベクトル制限情報に基づいて、ビデオデコーダ30は、並列復号をいつ開始すべきかを決定することができる。一例として、ビデオエンコーダ20は、ブロックサイズのユニットにおける動きベクトル制限の垂直動きベクトル範囲を出力用に生成することができ、ビデオデコーダ30は、ブロックサイズのユニットにおける動きベクトル制限の垂直動きベクトル範囲を、受信されたビットストリームから決定することができる。ブロックサイズのユニットは、最大コーディングユニットの高さおよび最小コーディングユニットの高さのユニットの1つを備え得る。ビデオデコーダ30は、垂直動きベクトル範囲に基づいて、ビュー間またはレイヤ間参照ピクチャと現在のブロックを含む現在のピクチャとを並列復号することができる。

30

【0218】

[0218]たとえば、動きベクトル制限は、水平および垂直成分についての値の範囲を定義し得る。一例では、最大コーディングユニット(LCU)の高さまたは幅のユニットにおける垂直動きベクトル範囲または水平動きベクトル範囲(たとえば、垂直もしくは水平視差動きベクトル範囲またはレイヤ間参照ピクチャについての垂直もしくは水平動きベクトル範囲)を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。別の例として、最小コーディングユニット(SCU)のユニットにおける動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。また別の例として、4のユニットにおける動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。

40

【0219】

[0219]いくつかの例では、垂直動きベクトル範囲が0に等しいか、またはほぼ0であることを動きベクトル制限が示す(0に等しいか、または0よりも小さい(たとえば、LC

50

Uの高さよりも小さい)垂直成分を動きベクトルが常に有することを意味する)場合のみ、ビデオエンコーダ20は水平動きベクトル範囲をシグナリングすることができる。いくつかの例では、ビデオエンコーダ20は常に水平動きベクトル範囲をシグナリングすることができる。

【0220】

[0220]いくつかの例では、ビデオエンコーダ20がLCU行遅延を直接シグナリングすること考えられ得る。LCU行遅延は垂直遅延を示す。これらの例では、ビデオエンコーダ20がLCU行遅延をシグナリングするとき、ビデオエンコーダ20はLCU列遅延も直接シグナリングすることができる。LCU列遅延は水平遅延を示す。これらの例では、ビデオデコーダ30は、LCU行(垂直)遅延とLCU列(水平)遅延とをビットストリームから直接受信することができる。一例として、動きベクトル(たとえば、視差動きベクトルまたはレイヤ間参照ピクチャを指す動きベクトル)が通常小さい(たとえば、垂直成分では0および水平成分では-2)と見なされる場合、LCU行遅延は1にされてよく、LCU列遅延も同じくらい小さくされ得る(たとえば、1または2)。

10

【0221】

[0221]上記の例では、ビデオエンコーダ20は、行または列に関する並列復号遅延(たとえば、LCU行遅延およびLCU列遅延)を直接シグナリングすることができる。行または列に関する並列復号遅延を直接シグナリングすることによって、動きベクトル成分に基づく動き制限を行または列に変換することは必要ではなくなることがある。言い換えれば、ビデオエンコーダ20は、LCU行遅延とLCU列遅延とをシグナリングする際に、LCUサイズまたはブロックのサイズをすでに考慮していることがある。

20

【0222】

[0222]上記の例は、動きベクトル制限(たとえば、x成分およびy成分についての動きベクトル範囲またはLCU行もしくは列遅延)のシグナリングについて説明している。いくつかの例では、補足エンハンスメント情報(SEI)メッセージ中で動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。いくつかの例では、ビデオパラメータセット中で動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。いくつかの例では、シーケンスパラメータセット中で動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。いくつかの例では、ビデオユーザビリティ情報(VUI)メッセージ中で動きベクトル制限を、ビデオエンコーダ20はシグナリングすることができ、ビデオデコーダ30は受信することができる。

30

【0223】

[0223]以下では、並列復号情報をシグナリングするための例示的な技法について説明する。たとえば、以下では、並列復号を実施するための例示的な擬似コードと擬似コードのためのセマンティクスとを提供する。これらの技法は、別個に、または連携して実施され得る。また、並列復号情報に関する以下で提供される例は、限定するものと見なされるべきではなく、単に例示のために提供される。さらに、説明しやすいように、本技法はマルチビュービデオコーディングに関して説明されるが、そのような技法はスケーラブルビデオコーディング技法にも拡張され得る。たとえば、説明が「ビュー間参照」などの用語を使用する場合、その用語はレイヤ間参照に置き換えられ得る。言い換えれば、ビュー間参照のための技法は、レイヤ間参照にも拡張され得る。

40

【0224】

[0224]並列復号情報SEIメッセージシンタックスが以下で証明される。

【表 2】

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
pdi_init_delay_ctb_vertical_minus2	ue(v)
pdi_init_delay_ctb_horizontal_minus2	ue(v)
}	

10

【0225】

[0225] video_parameter_set_id は、ビュー間またはレイヤ間従属関係情報を含むビデオパラメータセットを指定する。video_parameter_set_id の値は、並列復号情報 SEI メッセージを含むアクセスユニットのコーディングされたピクチャのビューコンポーネントまたはレイヤコンポーネントによって参照される video_parameter_set_id の値に等しいものとする。

【0226】

[0226] pdi_init_delay_ctb_vertical_minus2 および pdi_init_delay_ctb_horizontal_minus2 は、アクティブビデオパラメータセット識別子で指定されるようなビュー間またはレイヤ間予測を使用するコーディングされたビューコンポーネントによってビュー間またはレイヤ間参照に使用されないものとする任意の参照ビューコンポーネントまたはレイヤピクチャ中の利用不可能な参照エリアが、現在の SEI メッセージ中に含まれるシンタックス要素 video_parameter_set_id に等しいことを指定する。

20

【0227】

[0227] 変数 horCtb は次のように導出される。 $horCtb = pdi_init_delay_ctb_horizontal_minus2 ? pdi_init_delay_ctb_horizontal_minus2 + 2 + (CtbAddrInRS \% PicWidthInCtbsY) : 0$ 。

30

【0228】

[0228] 変数 verCtb は次のように導出される。

【0229】

$verCtb = CtbAddrInRS / PicWidthInCtbsY + pdi_init_delay_ctb_vertical_minus2 + 2$ 。

【0230】

[0229] refCtbAddr の refCtb アドレスは次のように導出される。

【0231】

$refCtbAddr = Min(horCtb, PicWidthInCtbsY) + verCtb * PicWidthInCtbsY$ 。

40

【0232】

[0230] 利用不可能な参照エリアは、refCtbAddr 以上のコーディングツリーブロックアドレスすべてを含む。コーディングされたビューコンポーネントを復号するとき、参照ビューのビューコンポーネントからの利用不可能な参照エリアからのサンプルは、ビュー間またはレイヤ間予測プロセスによって参照されないものとする。

【0233】

[0231] pdi_init_delay_ctb_vertical_minus2 の値は、両端値を含む $0 \sim PicHeightInCtbsY - 2$ の範囲内とする。pdi_init_delay_ctb_horizontal_minus2 の値は、両端値を

50

含む 0 ~ PicWidthInCtbsY - 2 の範囲内とする。

【 0 2 3 4 】

[0232] p d i _ i n i t _ d e l a y _ c t b _ h o r i z o n t a l _ m i n u s 2 が 0 に 等 しい とき、 v e r C t b 未 満 の 垂 直 ア ド レ ス を 有 す る L C U 行 中 の 任 意 の L C U (コーディングツリーブロック) が、ビュー間またはレイヤ間予測に使用され得る。垂直 L C U 行 遅 延 は、2 以 上 で あり 得 る (た と え ば、2 以 上 と す る) 。 水 平 L C U 列 遅 延 は、適 用 可 能 な 場 合、2 以 上 で あり 得 る (た と え ば、2 以 上 と す る) 。

【 0 2 3 5 】

[0233] いくつかの例では、各ビューが同じ L C U (コーディングツリーブロック) サイ ズ を 有 す る と い う 制 約 が 必 要 と さ れ 得 る。ビュが異なる L C U サイ ズ を 有 す る と き、最 大 L C U サイ ズ が 上 記 導 出 に 適 用 さ れ 得 る。そ の よ う な 最 大 L C U サイ ズ は、ビ デ オ パ ラ メー タ セ ャ 拡 張 で シ グ ナ リ ン グ さ れ 得 る。

10

【 0 2 3 6 】

[0234] 現在の L C U を 復 号 す る と き、水 平 視 差 動 き ベ ク ト ル が 非 常 に 小 さ い と 仮 定 す る と、ビ デ オ デ コー ダ 3 0 は、コ ロ ケー ト さ れ た L C U を 必 要 と し 得 る。し か し な が ら、コ ロ ケー ト さ れ た L C U を 利 用 可 能 に す る た め に、コ ロ ケー ト さ れ た L C U の 下、右 お よ び 右 下 の L C U が、少 なく と も デ ブ ロ ッ キ ン グ フ ィ ル タ 処 理 さ れ る 必 要 が あり 得 る。そ れ を 可 能 に す る た め に、上 記 L C U の う ち の 下 お よ び 右 の L C U が (た と え ば、動 き 補 償 お よ び / ま た は イ ン ト ラ 予 測 に よ っ て) 予 測 さ れ る 必 要 が あり 得 る。こ の 場 合、少 なく と も 2 つ の L C U 行 遅 延 お よ び 2 つ の L C U 列 遅 延 が 必 要 と さ れ 得 る。言 い 換 え れ ば、サ ン プ ル 適 応 オ フ セ ャ ッ ト (S A O : S a m p l e A d a p t i v e O f f s e t) フ ィ ル タ 処 理 お よ び デ ブ ロ ッ キ ン グ フ ィ ル タ 処 理 な ど の ル ー プ 内 フ ィ ル タ 処 理 を 可 能 に す る た め に、ビ デ オ デ コー ダ 3 0 は、現 在 の ピ ク チ ャ の 並 列 復 号 を 開 始 す る 前 に、ビュ間またはレイヤ間参照ピクチャの追加の行が再構成されるまで待機する必要がある。

20

【 0 2 3 7 】

[0235] いくつかの例では、ビ デ オ エ ン コー ダ 2 0 は、p d i _ i n i t _ d e l a y _ c t b _ v e r t i c a l _ m i n u s 2 が 2 に 等 しい とき の み、p d i _ i n i t _ d e l a y _ c t b _ h o r i z o n t a l _ m i n u s 2 を シ グ ナ リ ン グ す る こ と が 可 能 だ り 得 る。別 の 例 と し て、ビ デ オ エ ン コー ダ 2 0 は、次 の よ う に 垂 直 L C U 行 遅 延 を シ グ ナ リ ン グ す る だ け で あり 得 る。

30

【 表 3 】

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
pdi_init_delay_ctb_minus3	ue(v)
}	

40

【 0 2 3 8 】

[0236] v e r C t b = C t b A d d r I n R S / P i c W i d t h I n C t b s Y + p d i _ i n i t _ d e l a y _ c t b _ v e r t i c a l _ m i n u s 3 + 3 .

【 0 2 3 9 】

[0237] 利 用 不 可 能 な 参 照 エ リ ア は、座 標 (0 , (v e r C t b * P i c W i d t h I n C t b s Y)) を 左 上 隅 と し、座 標 (P i c W i d t h I n S a m p l e s , P i c H e i g h t I n S a m p l e s) を 右 下 隅 と す る 方 形 エ リ ア で あり 得 る。こ の 場 合、利 用 不 可 能 な 参 照 エ リ ア 中 の サ ン プ ル は、ビュ間またはレイヤ間予測プロセスによって参照されることはない(たとえば、参照されないものとする)。

50

【0240】

[0238]いくつかの例では、ビデオエンコーダ20は、ラスト走査順序でLCU遅延を指定するpdi__init__delay__ctb__rs__addressをシグナリングすることができる。いくつかの例では、ビデオエンコーダ20は、pdi__init__delay__ctb__rs__address__minus__ctbwidthをシグナリングし得る。これらの例では、pdi__init__delay__ctb__rs__address__minus__ctbwidth+PicWidthInCtbsYは、ラスト走査順序でLCU遅延を示す。

【0241】

[0239]いくつかの例では、ビデオエンコーダ20は、各ビューの参照ビューまたはレイヤのすべてについてビューごとに別個にpdi__init__delay__ctb__vertical__minus2とpdi__init__delay__ctb__horizontal__minus2とをシグナリングすることができる。いくつかの例では、ビデオエンコーダ20は、ビュー間またはレイヤ間予測を使用する各ビューまたはレイヤの参照ビューまたはレイヤごとに別個にpdi__init__delay__ctb__vertical__minus2とpdi__init__delay__ctb__horizontal__minus2とをシグナリングすることができる。たとえば、次の通りである。

【表4】

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
ここで(i=1;i≤vps_max_layers_minus1;i++)	
ここで(j=0;j<num_direct_ref_layers[i];j++) {	
pdi_init_delay_ctb_vertical_minus2[i][j]	ue(v)
pdi_init_delay_ctb_horizontal_minus2[i][j]	ue(v)
}	
}	

【0242】

[0240]いくつかの例では、上記のケースすべてについて、pdi__init__delay__ctb__horizontal__minus1はシグナリングされず、0に等しいと推測される。いくつかの例では、video_parameter_set_idは固定長、たとえばu(4)としてシグナリングされ得る。

【0243】

[0241]ピクセル再構成(たとえば、動き補償およびイントラ予測を含む)のみがLCUごとに行われる必要がある一方、デブロッキングはブロックごとに(すなわち、8×8ブロック)ごとに行われてよく、サンプル適応オフセット(SAO)フィルタ処理はラインごとに行われてよいと仮定されるとき、LCU遅延は大幅に減少し得る。たとえば、ビデオエンコーダ20が垂直LCU行遅延のみをシグナリングするとき、最小行遅延は2である。ビデオエンコーダ20が垂直LCU行遅延と水平LCU列遅延の両方をシグナリングするとき、最小行遅延は1であり、最小LCU列遅延も1である。後者の場合、シンタックスは次のように修正され得る。

10

20

30

40

【表 5】

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
pdi_init_delay_ctb_vertical_minus1	ue(v)
pdi_init_delay_ctb_horizontal_minus1	ue(v)
}	

10

【 0 2 4 4 】

[0242]セマンティクスは同様であるが、 $verCtb$ および $horCtb$ の計算は次のように若干修正され得る。変数 $horCtb$ は次のように導出される。 $horCtb = pdi_init_delay_ctb_horizontal_minus1 ? pdi_init_delay_ctb_horizontal_minus1 + 1 + (CtbAddrInRS \% PicWidthInCtbsY) : 0$ 。変数 $verCtb$ は次のように導出される。

【 0 2 4 5 】

$verCtb = CtbAddrInRS / PicWidthInCtbsY + pdi_init_delay_ctb_vertical_minus1 + 1$ 。

20

【 0 2 4 6 】

[0243]以下では、異なるLCUサイズを有する並列復号情報SEIメッセージが以下で説明されることについて説明する。この例は、並列復号情報SEIメッセージについての選択肢のうちの1つと同様であるが、様々なコーディングされたビューコンポーネント中の異なるCTUサイズに対する追加のサポートがある。さらに、例は、SEIメッセージに関して説明されるが、本技法はVUIシンタックス要素に拡張され得る。

【 0 2 4 7 】

[0244]並列復号情報SEIメッセージシンタックスが以下で提供される。

【表 6】

30

parallel_decoding_info(payloadSize) {	記述子
video_parameter_set_id	ue(v)
ここで(i=1;i≤num_views_minus1;i++)	
ここで(j=0;j<num_direct_ref_layers[i];j++)	
pdi_init_delay_ctu_vertical_minus2[i][j]	ue(v)
pdi_init_delay_ctu_horizontal_minus2[i][j]	ue(v)
}	
}	
}	

40

【 0 2 4 8 】

[0245]並列復号情報SEIメッセージは、アクセスユニットに関連付けられ得る。SEIメッセージ中でシグナリングされる情報は、SEIメッセージが関連付けられるアクセ

50

スユニットから始まり、もっぱら、同じタイプのSEIメッセージを含む、復号順序で次のアクセスユニットまで、またはコーディングされたビデオシーケンスの終わりまでのうち、復号順序で早い方までの、すべてのアクセスユニットに適用される。並列復号情報SEIメッセージ中で並列復号情報がシグナリングされるいくつかのビューコンポーネントが、コーディングされたビデオシーケンスに存在しない場合があることを理解されたい。

【0249】

[0246] video__parameter__set__idは、ビュー間またはレイヤ間従属関係情報を含むビデオパラメータセットを指定する。video__parameter__set__idの値は、並列復号情報SEIメッセージを含むアクセスユニットのコーディングされたピクチャのビューコンポーネントによって参照されるvideo__parameter__set__idの値に等しいことがある(たとえば、等しいものとする)。

10

【0250】

[0247]変数refLog2CtbSizeY、refPicWidthInCtbsY[i][j]およびrefPicHeightInCtbsY[i][j]は、コーディングされたビューコンポーネントiのj番目の参照ビューのビューコンポーネントについて、それぞれ、Log2CtbSizeY、PicWidthInCtbsYおよびPicHeightInCtbsYに等しくセットされ得る。

【0251】

[0248]pdi__init__delay__ctu__vertical__minus2[i][j]およびpdi__init__delay__ctu__horizontal__minus2[i][j]は、video__parameter__set__idによって識別されるアクティブビデオパラメータセットで指定されるビュー間またはレイヤ間予測を使用するi番目のコーディングされたビューコンポーネントによってビュー間参照に使用されることのない(たとえば、使用されないものとする)i番目のコーディングされたビューコンポーネントのj番目の参照ビューコンポーネント中の利用不可能な参照エリアを指定する。pdi__unit__delay__ctu__vertical__minus2[i][j]の範囲は、両端値を含む0~refPicHeightInCtbs-3であり得る(たとえば、両端値を含む0~refPicHeightInCtbs-3とする)。pdi__unit__delay__ctu__horizontal__minus2[i][j]の範囲は、両端値を含む0~refPicWidthInCtbs-3であり得る(たとえば、両端値を含む0~refPicWidthInCtbs-3とする)。

20

30

【0252】

[0249]いくつかの例では、ビデオエンコーダ20は、単一のシンタックス要素pdi__init__delay__ctu__minus2[i][j]をシグナリングすることがあり、pdi__init__delay__ctu__vertical__minus2[i][j]およびpdi__init__delay__ctu__horizontal__minus2[i][j]をシグナリングすることはない。これらの例では、ビデオデコーダ30は次のように、pdi__init__delay__ctu__vertical__minus2[i][j]およびpdi__init__delay__ctu__horizontal__minus2[i][j]の値をpdi__init__delay__ctu__minus2[i][j]から導出することができる。pdi__init__delay__ctu__horizontal__minus2[i][j]=pdi__init__delay__ctu__horizontal__minus2[i][j]%refPicWidthInCtbsおよびpdi__init__delay__ctu__vertical__minus2[i][j]=pdi__init__delay__ctu__minus2[i][j]/refPicWidthInCtbs。

40

【0253】

[0250]変数horCtb[i][j]は次のように導出される。horCtb[i][j]=pdi__init__delay__ctu__horizontal__minus2[i][j]?pdi__init__delay__ctu__horizontal__minu

50

$s2[i][j] + 2 : 0$ 。変数 $verCtb[i][j]$ は次のように導出される。
 $verCtb[i][j] = pdi_init_delay_ctu_vertical_minus2[i][j] ? pdi_init_delay_ctu_vertical_minus2[i][j] + 2 : 0$ 。

【0254】

[0251]変数 $ctbAddr$ は次のように導出される。

【0255】

($Log2CtbSizeY < refLog2CtbSizeY$) の場合、
 $ctbAddr = CtbAddrInRS >> ((refLog2CtbSizeY - Log2CtbSizeY) * 2)$

10

それ以外の場合、

$ctbAddr = CtbAddrInRS << ((Log2CtbSizeY - refLog2CtbSizeY) * 2)$

[0252] $refCtb[i][j]$ のアドレスを示す変数 $refCtbAddr[i][j]$ は次のように導出される。
 $refCtbAddr[i][j] = horCtb[i][j] + verCtb[i][j] * refPicWidthInCtbsY[i][j] + ctbAddr$ 。

【0256】

[0253] コーディングされたビューコンポーネント i の j 番目の参照ビュー中の利用不可能な参照エリアは、アドレスが $refCtbAddr[i][j]$ 以上であるすべてのコー
 ディングツリーユニットを含む。 i 番目のコーディングされたビューコンポーネントを
 復号するとき、コーディングされたビューコンポーネント i の j 番目の参照ビューのビ
 ューコンポーネントからの利用不可能な参照エリアからのサンプルは、ビュー間またはレイ
 ヤ間予測プロセスによって参照されることはない(たとえば、参照されないものとする)

20

【0257】

[0254] 図5は、本開示で説明する技法を実装し得る例示的なビデオエンコーダ20を示すブロック図である。ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターコーディングとを実行することができる。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的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(登録商標))は、いくつかの空間ベースの圧縮モードのいずれかを指し得る。単方向予測(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圧縮モードのいずれかを指し得る。ビデオエンコーダ20はまた、ビュー間予測コーディングとレイヤ間予測コーディングとを実行することができ、この場合にビデオエンコーダ20は、他のビューまたはレイヤについてのブロックをインター予測に使用する。

30

【0258】

[0255] 図5の例では、ビデオエンコーダ20は、区分ユニット350と、予測処理ユニット410と、参照ピクチャメモリ640と、加算器500と、変換処理ユニット520と、量子化処理ユニット540と、エントロピー符号化ユニット560とを含む。参照ピクチャメモリ640は、復号ピクチャバッファ(DPB)の一例である。予測処理ユニット410は、動き推定ユニット420と、動き補償ユニット440と、イントラ予測ユニット460とを含む。ビデオブロック再構成のために、ビデオエンコーダ20はまた、逆量子化処理ユニット580と、逆変換処理ユニット600と、加算器620とを含む。再構成されたビデオからブロックキ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処理するデブロッキングフィルタ(図5に図示せず)も含まれ得る。所望される場合、デブロッキングフィルタは、一般に、加算器620の出力をフィルタ処理することになる。デブロッキングフィルタに加えて、(ループ内またはループ後の)追加ループフィ

40

50

ルタも使用され得る。

【0259】

[0256]図5に示すように、ビデオエンコーダ20はビデオデータを受信し、区分ユニット350はデータをビデオブロックに区分する。この区分は、たとえば、LCUおよびCUの4分木構造に応じて、スライス、タイル、または他のより大きいユニットへの区分、ならびにビデオブロック区分をも含み得る。ビデオエンコーダ20は、概して、符号化されるべきビデオスライス内のビデオブロックを符号化する構成要素を示す。スライスは、複数のビデオブロックに（および、場合によっては、タイルと呼ばれるビデオブロックのセットに）分割され得る。予測処理ユニット410は、誤差結果（たとえばコーディングレートおよびひずみのレベル）に基づいて現在のビデオブロックについて、複数のイントラコーディングモードのうちの1つ、または複数のインターコーディングモードのうちの1つなど、複数の可能なコーディングモードのうちの1つを選択することができる。予測処理ユニット410は、得られたイントラコーディングされたブロックまたはインターコーディングされたブロックを、残差ブロックデータを生成するために加算器500に与え、参照ピクチャとして使用するための符号化されたブロックを再構成するために加算器620に与え得る。

10

【0260】

[0257]予測処理ユニット410内のイントラ予測ユニット460は、空間圧縮を行うために、コーディングされるべき現在のブロックと同じフレームまたはスライス中の1つまたは複数の隣接ブロックに対する現在の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。予測処理ユニット410内の動き推定ユニット420および動き補償ユニット440は、時間圧縮を行うために、1つまたは複数の参照ピクチャ中の1つまたは複数の予測ブロックに対する現在の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。

20

【0261】

[0258]動き推定ユニット420は、ビデオシーケンスの所定のパターンに従ってビデオスライスのためのインター予測モードを決定するように構成され得る。動き推定ユニット420と動き補償ユニット440とは、高度に統合され得るが、概念的な目的のために別々に示してある。動き推定ユニット420によって実行される動き推定は、ビデオブロック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、参照ピクチャ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ピクチャ内のビデオブロックのPUの変位を示し得る。動き推定ユニット420はまた、動きベクトル差分(MVD)を決定するために動きベクトル予測子を決定し得る。

30

【0262】

[0259]予測ブロックは、絶対値差分和(SAD)、差分2乗和(SSD)、または他の差分尺度によって決定され得るピクセル差分に関して、コーディングされるべきビデオブロックのPUに厳密に一致することがわかるブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ640に記憶された参照ピクチャのサブ整数ピクセル位置の値を計算し得る。たとえば、ビデオエンコーダ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し得る。したがって、動き推定ユニット420は、フルピクセル位置と分数ピクセル位置とに対する動き探索を実行し、分数ピクセル精度で動きベクトルを出力し得る。

40

【0263】

[0260]動き推定ユニット420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、インターコーディングされたスライスにおけるビデオブロックのPUについての動きベクトルを計算する。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RefPicList0)または第2の参照ピクチャリスト(リスト1またはRefPicList1)から選択されてよく、それらの参照ピクチャリストの各々は、参照ピクチャメモリ640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。動き推定ユニット420は、エントロピー符号化ユニット56と動き補償ユニット440とに計算された動きベクトルを送る。動き推定ユニット420は、計算された動きベクトル

50

ル予測子とMVDとをエントロピー符号化ユニット560に送り得る。

【0264】

[0261]動き補償ユニット440によって実行される動き補償は、動き推定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、場合によってはサブピクセル精度への補間を実行することを伴い得る。現在のビデオブロックのPUについての動きベクトルを受信すると、動き補償ユニット440は、動きベクトルが参照ピクチャリストのうちの1つにおいて指す予測ブロックの位置を特定し得る。ビデオエンコーダ20は、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって残差ビデオブロックを形成する。ピクセル差分値は、ブロックの残差データを形成し、ルーマ差分成分とクロマ差分成分の両方を含み得る。加算器500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。動き補償ユニット440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デコーダ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(たとえば、情報)を生成することができる。

10

【0265】

[0262]イントラ予測ユニット460は、前述のように、動き推定ユニット420と動き補償ユニット440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し得る。特に、イントラ予測ユニット460は、現在のブロックを符号化するために使用すべきイントラ予測モードを決定し得る。いくつかの例では、イントラ予測ユニット460は、たとえば、別個の符号化パス中に、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、イントラ予測ユニット460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。たとえば、イントラ予測ユニット460は、様々なテストされたイントラ予測モードのための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し得る。レートひずみ分析は、概して、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生成するために使用されるビットレート(すなわち、ビット数)を決定する。イントラ予測ユニット460は、どのイントラ予測モードがブロックについて最良のレートひずみ値を呈する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率を計算し得る。

20

30

【0266】

[0263]いずれの場合も、ブロックのイントラ予測モードを選択した後に、イントラ予測ユニット460は、ブロックについての選択されたイントラ予測モードを示す情報をエントロピー符号化ユニット560に提供し得る。エントロピー符号化ユニット560は、本開示の技法に従って、選択されたイントラ予測モードを示す情報を符号化し得る。ビデオエンコーダ20は、送信ビットストリーム中に、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複数の変更されたイントラ予測モードインデックステーブル(コードワードマッピングテーブルとも呼ばれる)と、様々なブロック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、および変更されたイントラ予測モードインデックステーブルの指示とを含み得る構成データを含め得る。

40

【0267】

[0264]予測処理ユニット410が、インター予測またはイントラ予測のいずれかを介して、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、ビデオエンコーダ20は、現在のビデオブロックから予測ブロックを減算することによって残差ビデオブロックを形成する。残差ブロック中の残差ビデオデータは、1つまたは複数のTU中に含まれ、変換処理ユニット520に適用され得る。変換処理ユニット520は、離散コサイン変換(DCT)または概念的に同様の変換などの変換を使用して、残差ビデオデータを残差変換係数に変換する。変換処理ユニット520は、残差ビデオデータをピクセル領域から周波

50

数領域などの変換領域に変換し得る。

【0268】

[0265]変換処理ユニット520は、得られた変換係数を量子化処理ユニット540に送り得る。量子化処理ユニット540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、係数の一部または全部に関連するビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって変更され得る。いくつかの例では、量子化処理ユニット540は、次いで、量子化変換係数を含む行列の走査を実行し得る。代替的に、エントロピー符号化ユニット560が走査を実行し得る。

【0269】

[0266]量子化の後に、エントロピー符号化ユニット560は、量子化変換係数をエントロピー符号化する。たとえば、エントロピー符号化ユニット560は、コンテキスト適応型可変長コーディング(CAVLC)、コンテキスト適応型バイナリ算術コーディング(CABAC)、シンタックススペースコンテキスト適応型バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディングまたは別のエントロピー符号化方法もしくは技法を実行し得る。エントロピー符号化ユニット560によるエントロピー符号化の後に、符号化されたビットストリームは、ビデオデコーダ30に送信されるか、またはビデオデコーダ30が後で送信するかもしくは取り出すためにアーカイブされ得る。エントロピー符号化ユニット560はまた、コーディングされている現在のビデオスライスについての動きベクトルと他のシンタックス要素とをエントロピー符号化することができる。

【0270】

[0267]逆量子化処理ユニット580および逆変換処理ユニット600は、参照ピクチャの参照ブロックとして後で使用するためのピクセル領域において残差ブロックを再構成するために、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440は、参照ピクチャリストのうちの1つ内の参照ピクチャのうちの1つの予測ブロックに残差ブロックを加算することによって参照ブロックを計算し得る。動き補償ユニット440はまた、動き推定において使用するためのサブ整数ピクセル値を計算するために、再構成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し得る。加算器620は、参照ピクチャメモリ640に記憶するための参照ブロックを生成するために、再構成された残差ブロックを動き補償ユニット440によって生成された動き補償予測ブロックに加算する。参照ブロックは、後続のビデオフレームまたはピクチャ中のブロックをインター予測するために、動き推定ユニット420と動き補償ユニット440とによって参照ブロックとして使用され得る。

【0271】

[0268]いくつかの例では、予測処理ユニット410は、本開示で説明する技法を実装するように構成され得る。ただし、本開示で説明する技法は、そのように限定されない。いくつかの例では、ビデオエンコーダ20内の1つまたは複数の他のユニットとともに、予測処理ユニット410は、本開示で説明する技法を実装するように構成され得る。いくつかの例では、処理ユニット(図5に図示せず)は、本開示で説明する技法を実装するように構成され得る。言い換えれば、ビデオエンコーダ20は、1つまたは複数のプロセッサを含むことができ、本開示で説明する例示的な技法を実装するように構成され得る。

【0272】

[0269]たとえば、ビデオエンコーダ20は(たとえば、予測処理ユニット410を介して)、ビデオデコーダ30が受信するビットストリーム中でシグナリングされる情報が、動きベクトル制限に準拠する動きベクトルをビデオデコーダ30に導出させることを確実にし得る。一例として、マージ/スキップモードの場合、ビデオエンコーダ20は、値が動きベクトル制限に準拠する、動きベクトルを導出するために使用される、動きベクトル予測子を識別する候補動きベクトル予測子のリストへのインデックスなどの情報を決定することができる。別の例として、AMVPモードの場合、ビデオエンコーダ20は、動きベクトルを導出するために使用される動きベクトル予測子を識別する候補動きベクトル予

10

20

30

40

50

測子のリストへのインデックスを決定することができる。ビデオエンコーダ20は、動きベクトル差分が動きベクトル予測子に加算されたときに、得られる動きベクトルが動きベクトル制限に準拠するように、動きベクトル差分を決定することもできる。マージ/スキップモードおよびAMVPモードの例では、ビデオエンコーダ20は、候補動きベクトル予測子のリストへのインデックスを出力用に生成することができる。AMVPモードの場合、ビデオエンコーダ20は、動きベクトル差分を出力用に生成することもできる。

【0273】

[0270]いくつかの例では、ビデオエンコーダ20は、参照ピクチャ中のピクチャについての動きベクトル差分が0ではないことをフラグの値(たとえば、`mv_d_l1_zero_flag`)が常に示すことを決定することができる。ビデオエンコーダ20は、ビット

10

【0274】

[0271]並列復号の場合、ビデオエンコーダ20は、ブロックサイズのユニットにおける動きベクトル制限の垂直動きベクトル範囲を出力することができる。たとえば、動きベクトル制限は、垂直動きベクトル範囲および水平動きベクトル範囲のうち少なくとも1つを含むことができる。これらの例では、ブロックサイズのユニットは、垂直ベクトル範囲の最大コーディングユニットの高さおよび最小コーディングユニットの高さのユニットならびに水平ベクトル範囲の最大コーディングユニットの幅および最小コーディングユニットの幅のユニットの1つであり得る。

【0275】

20

[0272]いくつかの例では、ビデオエンコーダ20は、現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、第1のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを決定することができる。これらの例では、ビデオエンコーダ20は、現在のブロックについての動きベクトルを、動きベクトルが任意のビュー間またはレイヤ間参照ピクチャ中の利用不可能な参照エリアを指さないように、決定された利用不可能なエリアに基づいて導出することができる。ビデオエンコーダ20はまた、利用不可能な参照エリアを示す情報を出力用に生成するように構成され得る。一例として、ビデオエンコーダ20は、ラスタ走査順序で利用不可能な参照エリアを示す情報を出力用に生成することができる。

【0276】

30

[0273]図6は、本開示で説明する技法を実装し得る例示的なビデオデコーダ30を示すブロック図である。図6の例では、ビデオデコーダ30は、エンтроピー復号ユニット800と、予測処理ユニット810と、逆量子化処理ユニット860と、逆変換ユニット880と、加算器900と、参照ピクチャメモリ920とを含む。参照ピクチャメモリ920は、復号ピクチャバッファ(DPB)の一例である。予測処理ユニット810は、動き補償ユニット820と、イントラ予測ユニット840とを含む。ビデオデコーダ30は、いくつかの例では、図5のビデオエンコーダ20に関して説明した符号化パスとは概して逆の復号パスを実行し得る。

【0277】

[0274]いくつかの例では、予測処理ユニット810は、本開示で説明する技法を実装するように構成され得る。ただし、本開示で説明する技法は、そのように限定されない。いくつかの例では、予測処理ユニット810は、ビデオデコーダ30内の1つまたは複数の他のユニットとともに、本開示で説明する技法を実装するように構成され得る。いくつかの例では、処理ユニット(図6に図示せず)は、本開示で説明する技法を実装するように構成され得る。

40

【0278】

[0275]復号プロセス中に、ビデオデコーダ30は、ビデオエンコーダ20から、符号化されたビデオスライスビデオブロックと、関連するシンタックス要素とを表す符号化されたビデオビットストリームを受信する。ビデオデコーダ30のエンтроピー復号ユニット800は、量子化係数と、動きベクトルと、他のシンタックス要素とを生成するために

50

ビットストリームをエントロピー復号する。エントロピー復号ユニット800は、動きベクトルと他のシンタックス要素とを予測処理ユニット810に転送する。ビデオデコーダ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信し得る。

【0279】

[0276]ビデオスライスがイントラコーディングされた(I)スライスとしてコーディングされるとき、予測処理ユニット810のイントラ予測ユニット840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し得る。ビデオフレームがインターコーディングされた(すなわち、B、PまたはGPB)スライスとしてコーディングされるとき、予測処理ユニット810の動き補償ユニット820は、エントロピー復号ユニット800から受信された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つ内の参照ピクチャのうちの1つから生成され得る。ビデオデコーダ30は、参照ピクチャメモリ920に記憶された参照ピクチャに基づいて、デフォルトの構成技法または任意の他の技法を使用して、参照フレームリスト、すなわち、リスト0とリスト1とを構成し得る。

【0280】

[0277]動き補償ユニット820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について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ための予測ブロックを生成するために、予測情報を使用する。たとえば、動き補償ユニット820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、Pスライス、またはGPBスライス)と、スライスの参照ピクチャリストのうちの1つまたは複数のための構成情報と、スライスの各インター符号化されたビデオブロックについての動きベクトルと、スライスの各インターコーディングされたビデオブロックについて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いくつかを使用する。

【0281】

[0278]動き補償ユニット820はまた、補間フィルタに基づいて補間を実行し得る。動き補償ユニット820は、参照ブロックのサブ整数ピクセルのための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中にビデオエンコーダ20によって使用された補間フィルタを使用し得る。この場合、動き補償ユニット820は、受信されたシンタックス要素からビデオエンコーダ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、その補間フィルタを使用し得る。

【0282】

[0279]逆量子化処理ユニット860は、ビットストリーム中で与えられ、エントロピー復号ユニット800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するために、同様に、適用されるべき逆量子化の程度を決定するために、ビデオスライス中の各ビデオブロックについてビデオエンコーダ20によって計算される量子化パラメータを使用することを含み得る。逆変換処理ユニット880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。

【0283】

[0280]動き補償ユニット820が、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ブロックのための予測ブロックを生成した後に、ビデオデコーダ30は、逆変換処理ユニット880からの残差ブロックを動き補償ユニット820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、復号されたビデオブロックを形成する

10

20

30

40

50

。加算器900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。所望される場合、ブロックネスアーティファクトを除去するために復号されたブロックをフィルタ処理するデブロッキングフィルタも適用され得る。ピクセル遷移を平滑化するために、または場合によってはビデオ品質を改善するために、(コーディンググループ内またはコーディンググループ後のいずれかの)他のループフィルタも使用され得る。所与のフレームまたはピクチャ中の復号されたビデオブロックは、次いで、その後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参照ピクチャメモリ920に記憶される。参照ピクチャメモリ920はまた、図4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提示のために、復号されたビデオを記憶する。

【0284】

[0281]このようにして、ビデオデコーダ30は、本開示で説明する例示的な技法を実装するように構成されたビデオデコーダの一例である。たとえば、ビデオデコーダ30は、1つまたは複数のプロセッサを含むことができ、本開示で説明する例示的な技法を実装するように構成され得る。

【0285】

[0282]たとえば、ビデオデコーダ30は、動きベクトル制限に準拠することを確実にされる動きベクトル予測子から動きベクトルを導出することができる。たとえば、マージノースキップモードでは、ビデオデコーダ30は、動きベクトル予測子に等しく動きベクトルをセットするので、動きベクトル予測子は、動きベクトル制限に準拠することが必要とされる。AMVPモードでは、ビデオデコーダ30は、値が、動きベクトル差分が動きベクトル予測子に加算されたときに、得られる動きベクトルが動きベクトル制限に準拠するようなものである動きベクトル差分を、ビットストリームから決定する。この例では、ビデオデコーダ30は、動きベクトル予測子と動きベクトル差分とを加算することによって、動きベクトルを導出することができる。

【0286】

[0283]いくつかの例では、ビデオデコーダ30は、ブロックサイズの動きベクトル制限の垂直動きベクトル範囲を、受信されたビットストリームから決定することができる。ビデオデコーダ30は、垂直動きベクトル範囲に基づいて、ビュー間またはレイヤ間参照ピクチャと現在のブロックを含む現在のピクチャとを並列復号することができる。

【0287】

[0284]また、いくつかの例では、ビデオデコーダ30は、現在のブロックによってビュー間またはレイヤ間参照に使用され得ない任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを、受信されたビットストリームから決定することができる。これらの例では、ビデオデコーダ30は、現在のブロックについての動きベクトルを、動きベクトルが任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを指さないように、ビットストリーム中の情報に基づいて導出することができる。一例として、ビデオデコーダ30は、ラスタ走査順序で利用不可能な参照エリアを決定することができる。

【0288】

[0285]図7は、本開示で説明する技法によるビュー間またはレイヤ間ビデオ復号の例示的な動作を示すフローチャートである。例示のために、本技法は、ビデオデコーダ30に関して説明される。たとえば、ビデオデコーダ30は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信する(1000)ことができる。動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し得る。ビットストリームから受信された情報が、導出された動きベクトルが動きベクトル制限に準拠することを確実にし得るように、ビットストリームは制約され得る。また、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。

【0289】

[0286]ビデオデコーダ30は、ビットストリーム中の情報に基づいて、現在のブロック

10

20

30

40

50

についての動きベクトルを導出する(1002)ことができる。ビデオデコーダ30は、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測復号する(1004)ことができる。

【0290】

[0287]図8は、本開示で説明する技法によるビュー間またはレイヤ間ビデオ符号化の例示的な動作を示すフローチャートである。例示のために、本技法は、ビデオデコーダ30に関して説明される。たとえば、ビデオエンコーダ20は、現在のブロックについての動きベクトルを動きベクトル予測子から導出するために使用される、導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定する(1006)ことができる。上記と同様に、動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、動きベクトル制限は、ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように動きベクトルの範囲を制限する。

【0291】

[0288]ビデオエンコーダ20は、情報に基づいて、現在のブロックについての動きベクトルを導出する(1008)ことができる。ビデオエンコーダ20は、導出された動きベクトルに基づいて、現在のブロックをビュー間またはレイヤ間予測符号化する(1010)ことができる。ビデオエンコーダ20は、動きベクトルを導出するための決定された情報を出力用に生成する(1012)ことができる。

【0292】

[0289]1つまたは複数の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組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つまた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、また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てよい。コンピュータ可読媒体は、たとえば、データ記憶媒体などの有形媒体、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に対応する、コンピュータ可読記憶媒体を含むことができる。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コンピュータ可読記憶媒体、または、(2)信号もしくは搬送波などの通信媒体に対応することができる。データ記憶媒体は、本開示で説明した技法を実装するための命令、コードおよび/また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むことができる。

【0293】

[0290]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または他の光ディスクストレージ、磁気ディスクストレージ、または他の磁気ストレージデバイス、フラッシュメモリ、あるいは、命令また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、いかなる接続も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、同軸ケーブル、光ファイバケーブル、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、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバケーブル、ツイストペア、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は、媒体の定義に含まれる。しかしながら、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。本明細書で使用されるディスク(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(di

10

20

30

40

50

sk) は、通常、データを磁気的に再生し、ディスク (disc) は、データをレーザーで光学的に再生する。上記の組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。

【0294】

[0291] 命令は、1つもしくは複数のデジタル信号プロセッサ (DSP) などの1つもしくは複数のプロセッサ、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論理アレイ (FPGA)、または他の等価な集積回路もしくはディスクリット論理回路によって実行され得る。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造、または本明細書で説明した技法の実装に適した任意の他の構造のいずれかを指し得る。さらに、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た専用のハードウェアおよび/またはソフトウェアモジュール内に設けられる場合があるか、または複合コーデックに組み込まれる場合がある。また、本技法は、1つまたは複数の回路または論理要素において完全に実装され得る。

10

【0295】

[0292] 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において実装され得る。本開示では、開示される技法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ために様々な構成要素、モジュール、またはユニットが説明されたが、それらの構成要素、モジュール、またはユニットは、必ずしも異なるハードウェアユニットによって実現する必要があるとは限らない。むしろ、前述のように、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、様々なユニットがコーデックハードウェアユニットにおいて組み合わせられ得るか、または前述のような1つもしくは複数のプロセッサを含む、相互動作可能なハードウェアユニットの集合体によって設けられ得る。

20

【0296】

[0293] 様々な例について説明してきた。これら例および他の例は、以下の特許請求の範囲内にある。

[C1] ビュー間またはレイヤ間ビデオ復号の方法であって、

現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信することと、
ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、
ここにおいて、前記導出された動きベクトルが動きベクトル制限に準拠することを前記情報が確実にするように、前記ビットストリームは制約され、
ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

30

前記ビットストリーム中の前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出することと、

前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号することと

を備える方法。

[C2] 前記情報は、前記ビットストリーム中にシンタックス要素を含む、C1に記載の方法。

40

[C3] 候補動きベクトル予測子のリストを構成することと、

前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを、前記受信されたビットストリームから決定することと

をさらに備え、

前記動きベクトルを導出することは、前記動きベクトル予測子に等しく前記動きベクトルをセットすることを備え、

前記動きベクトル予測子は、前記動きベクトル制限に準拠することが必要とされる、C1に記載の方法。

[C4] 候補動きベクトル予測子のリストを構成することと、

50

前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを、前記受信されたビットストリームから決定することと、

その値が、動きベクトル差分が前記動きベクトル予測子に加算されたときに、前記得られる動きベクトルが前記動きベクトル制限に準拠するようなものである前記動きベクトル差分を、前記受信されたビットストリームから決定することと

をさらに備え、

前記動きベクトルを導出することは、前記動きベクトル予測子と前記動きベクトル差分とを加算することを備える、C 1 に記載の方法。

[C 5] 参照ピクチャリスト中で識別されたピクチャについての動きベクトル差分が 0 であるかどうかを示すフラグ値を、前記受信されたビットストリームから決定することとをさらに備え、前記参照ピクチャリスト中の前記ピクチャについての前記動きベクトル差分が 0 ではないことを前記フラグ値は常に示す、C 1 に記載の方法。

10

[C 6] 前記動きベクトルが前記レイヤ間参照ピクチャを指すときに、前記動きベクトル制限は、前記導出された動きベクトルが 0 に等しくなるべきであることを示す、C 1 に記載の方法。

[C 7] ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を、前記受信されたビットストリームから決定することと、

前記垂直動きベクトル範囲に基づいて、前記ビュー間またはレイヤ間参照ピクチャと前記現在のブロックを含む現在のピクチャとを並列復号することと

20

をさらに備える、C 1 に記載の方法。

[C 8] ブロックサイズの前記ユニットは、最大コーディングユニットの高さおよび最小コーディングユニットの高さのユニットの 1 つを備える、C 7 に記載の方法。

[C 9] 前記ビュー間またはレイヤ間参照ピクチャは、第 1 のビュー間またはレイヤ間参照ピクチャを含み、前記方法は、

前記現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、前記第 1 のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを、前記受信されたビットストリームから決定すること

30

をさらに備え、

前記動きベクトルを導出することは、前記現在のブロックについての前記動きベクトルを、前記動きベクトルが任意のビュー間または任意のレイヤ間参照ピクチャ中の前記利用不可能な参照エリアを指さないように、前記ビットストリーム中の前記情報に基づいて導出することを備える、C 1 に記載の方法。

[C 10] 前記利用不可能な参照エリアを決定することは、ラスト走査順序で前記利用不可能な参照エリアを決定することを備える、C 9 に記載の方法。

[C 11] ビュー間またはレイヤ間ビデオ符号化の方法であって、

現在のブロックについての動きベクトルを動きベクトル予測子から導出するために使用される、前記導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定することと、

40

ここで、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここで、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出することと、

前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測符号化することと、

前記動きベクトルを導出するための前記決定された情報を出力用に生成することとを備える方法。

[C 12] 前記情報は、出力用に生成されるシンタックス要素を含む、C 11 に記載の方法。

50

[C 1 3] 候補動きベクトル予測子のリストを構成すること
をさらに備え、

情報を決定することは、値が前記動きベクトル制限に準拠する、前記動きベクトルを導出するために使用される、前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを決定することを備え

出力用に生成することは、候補動きベクトル予測子の前記リストへの前記インデックスを出力用に生成することを備える、C 1 1 に記載の方法。

[C 1 4] 候補動きベクトル予測子のリストを構成すること
をさらに備え、

情報を決定することは、

前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを決定することと、

動きベクトル差分が前記動きベクトル予測子に加算されたときに、前記得られる動きベクトルが前記動きベクトル制限に準拠するように、前記動きベクトル差分を決定することと
を備え、

出力用に生成することは、候補動きベクトル予測子の前記リストへの前記インデックスと前記動きベクトル差分とを出力用に生成することを備える、C 1 1 に記載の方法。

[C 1 5] 情報を決定することは、参照ピクチャリスト中のピクチャについての動きベクトル差分が 0 ではないことをフラグの値が常に示すことを決定することを備え、出力用に生成することは、前記フラグの前記値を出力用に生成することを備える、C 1 1 に記載の方法。

[C 1 6] 前記動きベクトルが前記レイヤ間参照ピクチャを指すときに、前記動きベクトル制限は、前記導出された動きベクトルが 0 に等しくなるべきであることを示す、C 1 1 に記載の方法。

[C 1 7] ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を出力用に生成すること
をさらに備える、C 1 1 に記載の方法。

[C 1 8] ブロックサイズの前記ユニットは、最大コーディングユニットの高さおよび最小コーディングユニットの高さのユニットの 1 つを備える、C 1 7 に記載の方法。

[C 1 9] 前記ビュー間またはレイヤ間参照ピクチャは、第 1 のビュー間またはレイヤ間参照ピクチャを含み、前記方法は、

前記現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、前記第 1 のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを決定することと、

ここにおいて、前記動きベクトルを導出することは、前記現在のブロックについての前記動きベクトルを、前記動きベクトルが任意のビュー間またはレイヤ間参照ピクチャ中の前記利用不可能な参照エリアを指さないように、前記決定された利用不可能なエリアに基づいて導出することを備え、

前記利用不可能な参照エリアを示す情報を出力用に生成することと
をさらに備える、C 1 1 に記載の方法。

[C 2 0] 前記利用不可能な参照エリアを示す情報を出力用に生成することは、ラスト走査順序で前記利用不可能な参照エリアを示す情報を出力用に生成することを備える、C 1 9 に記載の方法。

[C 2 1] ビデオデコーダを備えるデバイスであって、前記ビデオデコーダは、1 つまたは複数のプロセッサを備え、

現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信し、ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、前記導出された動きベクトルが動きベクトル制限に準拠することを前記情報が確実にするように、前記ビットストリームは制約され

10

20

30

40

50

、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

前記ビットストリーム中の前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出し、

前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号する

ように構成される、デバイス。

[C 2 2] 前記情報は、前記ビットストリーム中にシンタックス要素を含む、C 2 1に記載のデバイス。

[C 2 3] 前記ビデオデコーダは、

候補動きベクトル予測子のリストを構成し、

前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを、前記受信されたビットストリームから決定する

ように構成され、

前記動きベクトルを導出するために、前記ビデオデコーダは、前記動きベクトル予測子に等しく前記動きベクトルをセットするように構成され、

前記動きベクトル予測子は、前記動きベクトル制限に準拠することが必要とされる、C 2 1に記載のデバイス。

[C 2 4] 前記ビデオデコーダは、

候補動きベクトル予測子のリストを構成し、

前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを、前記受信されたビットストリームから決定し、

値が、動きベクトル差分が前記動きベクトル予測子に加算されたときに、前記得られる動きベクトルが前記動きベクトル制限に準拠するようなものである前記動きベクトル差分を、前記受信されたビットストリームから決定する

ように構成され、

前記動きベクトルを導出するために、前記ビデオデコーダは、前記動きベクトル予測子と前記動きベクトル差分とを加算するように構成される、C 2 1に記載のデバイス。

[C 2 5] 前記ビデオデコーダは、

参照ピクチャリスト中で識別されたピクチャについての動きベクトル差分が0であるかどうかを示すフラグ値を、前記受信されたビットストリームから決定するように構成され、前記参照ピクチャリスト中の前記ピクチャについての前記動きベクトル差分が0ではないことを前記フラグ値は常に示す、C 2 1に記載のデバイス。

[C 2 6] 前記ビデオデコーダは、

ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を、前記受信されたビットストリームから決定し、

前記垂直動きベクトル範囲に基づいて、前記ビュー間またはレイヤ間参照ピクチャと前記現在のブロックを含む現在のピクチャとを並列復号する

ように構成される、C 2 1に記載のデバイス。

[C 2 7] 前記ビュー間またはレイヤ間参照ピクチャは、第1のビュー間またはレイヤ間参照ピクチャを含み、前記ビデオデコーダは、

前記現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、前記第1のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを、前記受信されたビットストリームから決定する

ように構成され、

前記動きベクトルを導出するために、前記ビデオデコーダは、前記現在のブロックについての前記動きベクトルを、前記動きベクトルが任意のビュー間または任意のレイヤ間参照ピクチャ中の前記利用不可能な参照エリアを指さないように、前記ビットストリーム中

10

20

30

40

50

の前記情報に基づいて導出するように構成される、C 2 1 に記載のデバイス。

[C 2 8] 前記利用不可能な参照エリアを決定するために、前記ビデオデコーダは、ラスト走査順序で前記利用不可能な参照エリアを決定するように構成される、C 2 7 に記載のデバイス。

[C 2 9] 前記デバイスは、
ワイヤレス通信デバイス、
マイクロプロセッサ、および
集積回路

のうちの1つを備える、C 2 1 に記載のデバイス。

[C 3 0] ビデオエンコーダを備えるデバイスであって、前記ビデオエンコーダは、1つまたは複数のプロセッサを備え、

現在のブロックについての動きベクトルを動きベクトル予測子から導出するために使用される、前記導出された動きベクトルが動きベクトル制限に準拠することを確実にする情報を決定し、ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出し、
前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測符号化し、

前記動きベクトルを導出するための前記決定された情報を出力用に生成するように構成される、デバイス。

[C 3 1] 前記情報は、出力用に生成されるシンタックス要素を含む、C 3 0 に記載のデバイス。

[C 3 2] 前記ビデオエンコーダは、
候補動きベクトル予測子のリストを構成するように構成され、
情報を決定するために、前記ビデオエンコーダは、値が前記動きベクトル制限に準拠する、前記動きベクトルを導出するために使用される、前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを決定するように構成され、

出力用に生成するために、前記ビデオエンコーダは、候補動きベクトル予測子の前記リストへの前記インデックスを出力用に生成するように構成される、C 3 0 に記載のデバイス。

[C 3 3] 前記ビデオエンコーダは、
候補動きベクトル予測子のリストを構成するように構成され、
情報を決定するために、前記ビデオエンコーダは、
前記動きベクトルを導出するために使用される前記動きベクトル予測子を識別する候補動きベクトル予測子の前記リストへのインデックスを決定し、

動きベクトル差分が前記動きベクトル予測子に加算されたときに、前記得られる動きベクトルが前記動きベクトル制限に準拠するように、前記動きベクトル差分を決定するように構成され、

出力用に生成するために、前記ビデオエンコーダは、候補動きベクトル予測子の前記リストへの前記インデックスと前記動きベクトル差分とを出力用に生成するように構成される、C 3 0 に記載のデバイス。

[C 3 4] 情報を決定するために、前記ビデオエンコーダは、参照ピクチャリスト中のピクチャについての動きベクトル差分が0ではないことをフラグの値が常に示すことを決定するように構成され、出力用に生成するために、前記ビデオエンコーダは、前記フラグの前記値を出力用に生成するように構成される、C 3 0 に記載のデバイス。

[C 3 5] 前記ビデオエンコーダは、
ブロックサイズのユニットにおける前記動きベクトル制限の垂直動きベクトル範囲を出力用に生成するように構成される、C 3 4 に記載のデバイス。

10

20

30

40

50

[C 3 6] 前記ビュー間またはレイヤ間参照ピクチャは、第 1 のビュー間またはレイヤ間参照ピクチャを含み、前記ビデオエンコーダは、

前記現在のブロックによってビュー間またはレイヤ間参照に使用され得ない、前記第 1 のビュー間またはレイヤ間参照ピクチャを含む任意のビュー間または任意のレイヤ間参照ピクチャ中の利用不可能な参照エリアを決定し、

ここにおいて、前記動きベクトルを導出するために、前記ビデオエンコーダは、前記現在のブロックについての前記動きベクトルを、前記動きベクトルが任意のビュー間またはレイヤ間参照ピクチャ中の前記利用不可能な参照エリアを指さないように、前記決定された利用不可能なエリアに基づいて導出するように構成され、

前記利用不可能な参照エリアを示す情報を出力用に生成するように構成される、C 3 0 に記載のデバイス。

10

[C 3 7] 前記利用不可能な参照エリアを示す情報を出力用に生成するために、前記ビデオエンコーダは、ラスト走査順序で前記利用不可能な参照エリアを示す情報を出力用に生成するように構成される、C 3 6 に記載のデバイス。

[C 3 8] 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たときに、ビデオデータを復号するためのデバイスの 1 つまたは複数のプロセッサに、

現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信させ、ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、前記導出された動きベクトルが動きベクトル制限に準拠することを前記情報が確実にするように、前記ビットストリームは制約され、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

20

前記ビットストリーム中の前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出させ、

前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号させる、コンピュータ可読記憶媒体。

[C 3 9] 現在のブロックについての動きベクトルを動きベクトル予測子から導出するための情報を含むビットストリームを受信するための手段と、ここにおいて、前記動きベクトルは、ビュー間またはレイヤ間参照ピクチャを指し、ここにおいて、前記導出された動きベクトルが動きベクトル制限に準拠することを前記情報が確実にするように、前記ビットストリームは制約され、ここにおいて、前記動きベクトル制限は、前記ビュー間またはレイヤ間参照ピクチャのサブ部分を指すように前記動きベクトルの範囲を制限する、

30

前記ビットストリーム中の前記情報に基づいて、前記現在のブロックについての前記動きベクトルを導出するための手段と、

前記導出された動きベクトルに基づいて、前記現在のブロックをビュー間またはレイヤ間予測復号するための手段とを備えるデバイス。

【 図 1 】

図 1

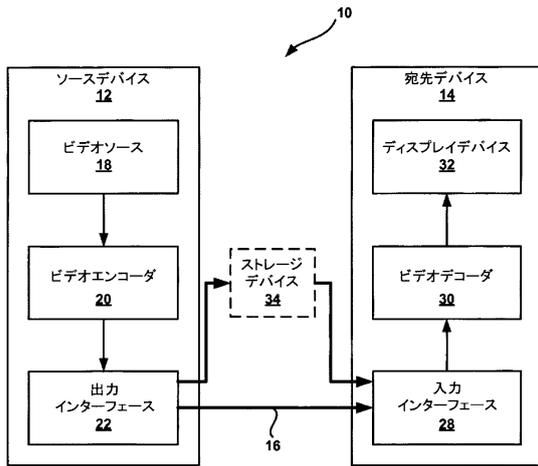


FIG. 1

【 図 2 】

図 2

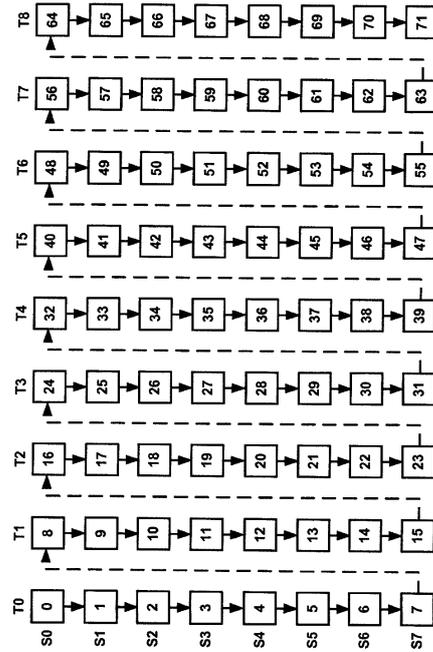


FIG. 2

【 図 3 】

図 3

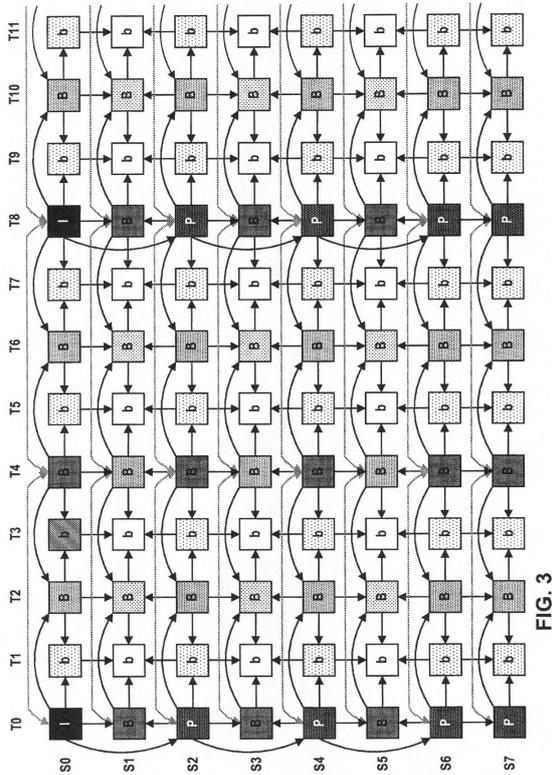


FIG. 3

【 図 4 】

図 4

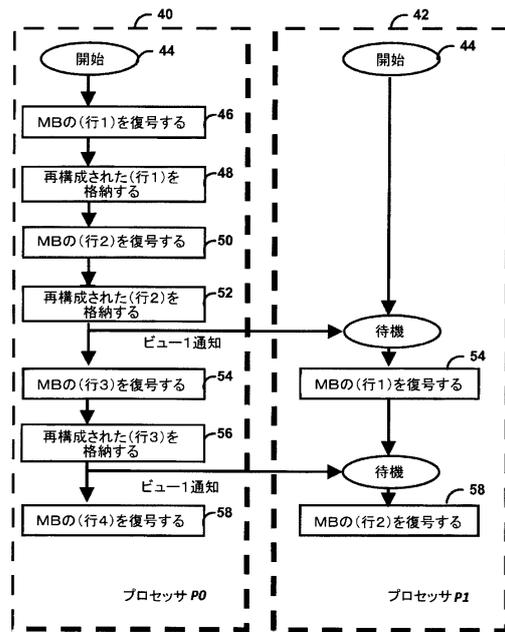


FIG. 4

【 図 5 】

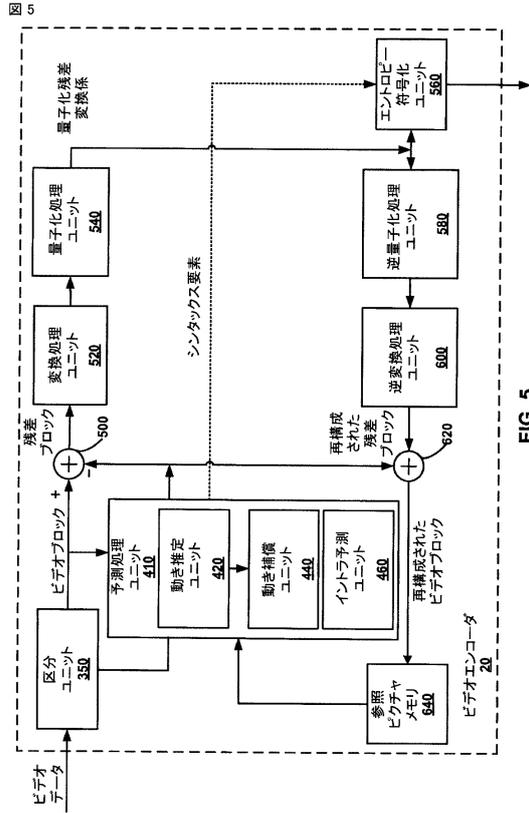


FIG. 5

【 図 6 】

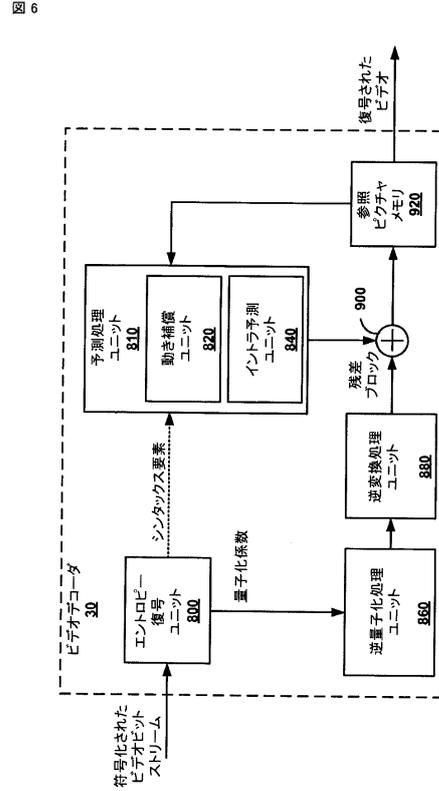


FIG. 6

【 図 7 】

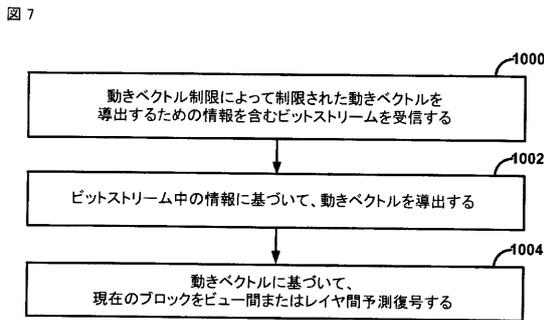


FIG. 7

【 図 8 】

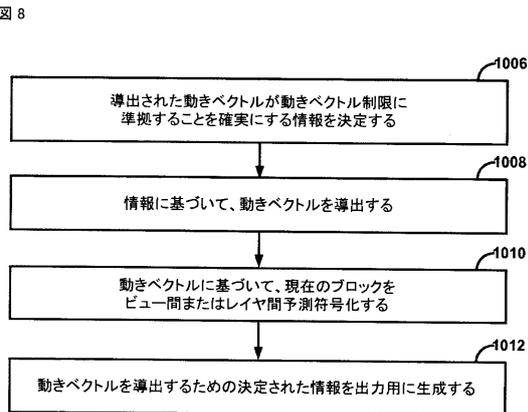


FIG. 8

フロントページの続き

- (31)優先権主張番号 61/755,274
(32)優先日 平成25年1月22日(2013.1.22)
(33)優先権主張国 米国(US)
(31)優先権主張番号 61/758,206
(32)優先日 平成25年1月29日(2013.1.29)
(33)優先権主張国 米国(US)
(31)優先権主張番号 14/146,149
(32)優先日 平成26年1月2日(2014.1.2)
(33)優先権主張国 米国(US)

- (72)発明者 セレジン、パディム
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 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 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 ラマスブラモニアン、アダルシュ・クリシュナン
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 岩井 健二

- (56)参考文献 特表2011-519226(JP,A)
国際公開第2012/176684(WO,A1)
国際公開第2012/128068(WO,A1)
国際公開第2012/124121(WO,A1)
Gerhard Tech et al., 3D-HEVC Test Model 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 1st Meeting: Stockholm, SE, 16-20 July 2012, URL:http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/1_Stockholm/wg11/JCT3V-A1005-v1.zip, 2012年9月20日, JCT3V-A1005_d0, pp.17-18
Ohji Nakagami et al., MV-HEVC: Vertical Length restriction of inter-view vector for HEVC simple 3D extension, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd Meeting: Shanghai, CN, 13 - 19 October 2012, 2012年10月5日, JCT3V-B0037, pp.1-7, URL, http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=321
Tomohiro Ikai and Tadashi Uchiumi, 3D-CE5.h related: Disparity vector restrictions, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd Meeting: Shanghai, CN, 13 - 19 October 2012, 2012年10月9日, JCT3V-B0112, pp.1-7, URL, http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=321
Tadashi Uchiumi and Tomohiro Ikai, 3D-CE5.h related: Restricted motion vector coding for inter-view prediction, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd Meeting: Shanghai, CN, 13 - 19 October 2012, 2012年10月9日, JCT3V-B0113, pp.1-4, URL, <http://phenix.it-su>

dparis.eu/jct2/doc_end_user/current_document.php?id=322

(58)調査した分野(Int.Cl. , DB名)

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