

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6779782号  
(P6779782)

(45) 発行日 令和2年11月4日(2020.11.4)

(24) 登録日 令和2年10月16日(2020.10.16)

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

請求項の数 15 (全 45 頁)

(21) 出願番号	特願2016-517339 (P2016-517339)	(73) 特許権者	595020643
(86) (22) 出願日	平成26年9月26日 (2014. 9. 26)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2016-537843 (P2016-537843A)		QUALCOMM INCORPORATED
(43) 公表日	平成28年12月1日 (2016. 12. 1)		アメリカ合衆国、カリフォルニア州 92
(86) 国際出願番号	PCT/US2014/057719		121-1714、サン・ディエゴ、モア
(87) 国際公開番号	W02015/048443		ハウス・ドライブ 5775
(87) 国際公開日	平成27年4月2日 (2015. 4. 2)	(74) 代理人	100108855
審査請求日	平成29年8月30日 (2017. 8. 30)		弁理士 蔵田 昌俊
(31) 優先権主張番号	61/883, 931	(74) 代理人	100158805
(32) 優先日	平成25年9月27日 (2013. 9. 27)		弁理士 井関 守三
(33) 優先権主張国・地域又は機関	米国 (US)	(74) 代理人	100112807
(31) 優先権主張番号	14/496, 807		弁理士 岡田 貴志
(32) 優先日	平成26年9月25日 (2014. 9. 25)		
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 MV-HEVCにおけるビュー間依存タイプ

(57) 【特許請求の範囲】

【請求項1】

マルチレイヤビデオデータを復号する方法であって、

依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、前記依存タイプシンタックス要素がシグナリングされる時、前記依存タイプシンタックス要素が、各参照レイヤに対する前記マルチレイヤビデオデータの各レイヤのために有効化された依存のタイプを識別し、前記依存タイプシンタックス要素によって識別された依存の前記タイプが、レイヤ間サンプル予測、レイヤ間動き予測、またはレイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つである、

前記第1のシンタックス要素の前記値が、前記依存タイプシンタックス要素がシグナリングされることを示すことに応答して、前記依存タイプシンタックス要素の値を受信すること、前記依存タイプシンタックス要素のそれぞれの値に基づいて参照レイヤに対する現在のレイヤのために有効化された依存の前記タイプを決定すること、および依存の前記タイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号すること、または

前記第1のシンタックス要素の前記値が、前記依存タイプシンタックス要素がシグナリングされないことを示すことに応答して、参照レイヤに対する任意の現在のレイヤの依存のタイプが依存の単一のタイプであると決定すること、及び、現在のレイヤと参照レイヤとのすべての組合せについて同じである依存の単一のタイプの別個にシグナリングされた

指示を受信することと、依存の前記単一のタイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号することと

を備える方法。

【請求項 2】

各参照レイヤに対するすべてのレイヤのために有効化された依存の前記タイプが、前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測のために使用されるが、前記現在のレイヤのレイヤ間動き予測のためには使用されない第 1 の依存タイプ、前記参照レイヤが前記現在のレイヤのレイヤ間動き予測のために使用されるが、前記現在のレイヤのレイヤ間サンプル予測のためには使用されない第 2 の依存タイプ、または前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用される第 3 の依存タイプ、のうちの 1 つを備える、請求項 1 に記載の方法。

10

【請求項 3】

各参照レイヤに対するすべてのレイヤのために有効化された依存の前記タイプがレイヤ間サンプル予測を備え、ここにおいて、レイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することが、前記参照レイヤ中で、前記現在のレイヤの前記ブロックのための予測ブロックを識別することを備える、請求項 1 に記載の方法。

【請求項 4】

ビデオパラメータセットである、パラメータセットの一部として前記第 1 のシンタックス要素を受信することと、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第 1 のシンタックス要素の前記値に回答して、依存の同じタイプが前記パラメータセットに関連するすべての他のレイヤに対するすべてのレイヤのために有効化されると決定することと

20

をさらに備える、請求項 1 に記載の方法。

【請求項 5】

マルチレイヤビデオデータを符号化する方法であって、

複数のレイヤを備えるコード化されたビデオデータについて、前記複数のレイヤの符号化を、前記複数のレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも前記複数のレイヤに対して複数の依存タイプを使用して行うべきかを決定することと、ここにおいて、依存タイプが、参照レイヤに対する現在のレイヤのために有効化された依存のタイプを識別し、レイヤ間サンプル予測、レイヤ間動き予測、またはレイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの 1 つである、

30

前記複数のレイヤの符号化を、前記同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、依存タイプシンタックス要素がシグナリングされるかどうかを表す第 1 のシンタックス要素の値を決定することと、ここにおいて、前記依存タイプシンタックス要素が、各参照レイヤに対する前記マルチレイヤビデオデータの各レイヤのために有効化された依存のタイプを識別する、

前記複数のレイヤの符号化を前記複数の依存タイプを使用して行うべきと決定したことに回答して、前記依存タイプシンタックス要素を符号化することと、

前記複数のレイヤの符号化を前記同じ依存タイプを使用して行うべきと決定したことに回答して、前記依存タイプシンタックス要素の符号化をスキップすること、及び、現在のレイヤと参照レイヤとのすべての組合せについて同じである依存の単一のタイプの指示を別個にシグナリングすること、

40

を備える方法。

【請求項 6】

前記第 1 のシンタックス要素が、パラメータセット中に含まれ、ここにおいて、前記依存タイプシンタックス要素の符号化をスキップすることが、前記パラメータセット中に前記依存タイプシンタックス要素のインスタスを含めないことを備える、請求項 5 に記載の方法。

【請求項 7】

前記同じ依存タイプが、前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測の

50

ために使用されるが、前記現在のレイヤのレイヤ間動き予測のためには使用されない第1の依存タイプ、前記参照レイヤが前記現在のレイヤのレイヤ間動き予測のために使用されるが、前記現在のレイヤのレイヤ間サンプル予測のためには使用されない第2の依存タイプ、または前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用される第3の依存タイプ、のうちの1つを備える、請求項5に記載の方法。

【請求項8】

ビデオ復号を実行する装置であって、  
ビデオデータを記憶するメモリと、

1つまたは複数のプロセッサを備えたビデオコーダとを備え、前記1つまたは複数のプロセッサが、

依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、前記依存タイプシンタックス要素がシグナリングされる時、前記依存タイプシンタックス要素が、各参照レイヤに対するマルチレイヤビデオデータの各レイヤのために有効化された依存のタイプを識別し、前記依存タイプシンタックス要素によって識別された依存の前記タイプが、レイヤ間サンプル予測、レイヤ間動き予測、またはレイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つである、

前記第1のシンタックス要素の前記値が、前記依存タイプシンタックス要素がシグナリングされることを示すことに応答して、前記依存タイプシンタックス要素の値を受信すること、前記依存タイプシンタックス要素のそれぞれの値に基づいて参照レイヤに対する任意の現在のレイヤのために有効化された依存の前記タイプを決定すること、および依存の前記タイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号すること、または

前記第1のシンタックス要素の前記値が、前記依存タイプシンタックス要素がシグナリングされないことを示すことに応答して、参照レイヤに対する現在のレイヤの依存のタイプが依存の単一のタイプであると決定すること、及び、現在のレイヤと参照レイヤとのすべての組合せについて同じである依存の単一のタイプの別個にシグナリングされた指示を受信することと、依存の前記決定されたタイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号することと

を行うように構成された、装置。

【請求項9】

前記第1のシンタックス要素の前記値が、前記依存タイプシンタックス要素がシグナリングされないと示すとき、前記第1のシンタックス要素の前記値が、各参照レイヤに対するすべてのレイヤのために有効化された依存の前記タイプをさらに識別する、請求項8に記載の装置。

【請求項10】

各参照レイヤに対するすべてのレイヤのために有効化された依存の前記タイプが、前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測のために使用されるが、前記現在のレイヤのレイヤ間動き予測のためには使用されない第1の依存タイプ、前記参照レイヤが前記現在のレイヤのレイヤ間動き予測のために使用されるが、前記現在のレイヤのレイヤ間サンプル予測のためには使用されない第2の依存タイプ、または前記参照レイヤが前記現在のレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用される第3の依存タイプ、のうちの1つを備える、請求項8に記載の装置。

【請求項11】

各参照レイヤに対するすべてのレイヤのために有効化された依存の前記タイプがレイヤ間サンプル予測を備え、ここにおいて、レイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することが、前記参照レイヤ中で、前記現在のレイヤの前記ブロックのための予測ブロックを識別することを備える、請求項8に記載の装置。

【請求項12】

10

20

30

40

50

前記 1 つまたは複数のプロセッサが、  
ビデオパラメータセットである、パラメータセットの一部として前記第 1 のシンタックス要素を受信することと、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第 1 のシンタックス要素の前記値に回答して、依存の同じタイプが前記パラメータセットに関連するすべての他のレイヤに対するすべてのレイヤのために有効化されると決定することと

を行うようにさらに構成された、請求項 8 に記載の装置。

【請求項 13】

ビデオ符号化を実行する装置であって、

ビデオデータを記憶するメモリと、

1 つまたは複数のプロセッサを備えたビデオコーダとを備え、前記 1 つまたは複数のプロセッサが、

複数のレイヤを備えるコード化されたビデオデータについて、前記複数のレイヤの符号化を、前記複数のレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも前記複数のレイヤに対して複数の依存タイプを使用して行うべきかを決定することと、ここにおいて、依存タイプが、参照レイヤに対する現在のレイヤのために有効化された依存のタイプを識別し、レイヤ間サンプル予測、レイヤ間動き予測、またはレイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの 1 つである、

前記複数のレイヤの符号化を、前記同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、依存タイプシンタックス要素がシグナリングされるかどうかを表す第 1 のシンタックス要素の値を決定することと、ここにおいて、前記依存タイプシンタックス要素が、各参照レイヤに対するマルチレイヤビデオデータの各レイヤのために有効化された依存のタイプを識別する、

前記複数のレイヤの符号化を複数の依存タイプを使用して行うべきと決定したことに回答して、前記依存タイプシンタックス要素を符号化することと、

前記複数のレイヤの符号化を前記同じ依存タイプを使用して行うべきと決定したことに回答して、前記依存タイプシンタックス要素の符号化をスキップすること、及び、現在のレイヤと参照レイヤとのすべての組合せについて同じである依存の単一のタイプの指示を別個にシグナリングすること、

を行うように構成された、装置。

【請求項 14】

集積回路、

マイクロプロセッサ、または

ビデオコーダを含むワイヤレス通信デバイス

のうちの少なくとも 1 つを備える、請求項 8 または請求項 13 に記載の装置。

【請求項 15】

1 つまたは複数のプロセッサによって実行されたとき、前記 1 つまたは複数のプロセッサに、請求項 1 - 7 のいずれかに記載の方法を実行することを行わせる命令を記憶したコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本出願は、その全体が参照により本明細書に組み込まれる、2013年9月27日に出願された米国仮出願第61/883,931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、マルチレイヤコーディングにおけるビュー間依存タイプのシグナリングに関する。

【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

10

20

30

40

50

キャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末（PDA）、ラップトップコンピュータまたはデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲームデバイス、ビデオゲームコンソール、携帯電話または衛星無線電話、いわゆる「スマートフォン」、ビデオ遠隔会議デバイス、ビデオストリーミングデバイスなどを含む、広範囲のデバイスに組み込まれ得る。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4, Part 10, アドバンスドビデオコーディング（AVC: Advanced Video Coding）、現在開発中の高効率ビデオコーディング（HEVC: High Efficiency Video Coding）規格によって定義された規格、およびそのような規格の拡張に記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。これらのビデオデバイスは、そのようなビデオ圧縮技法を実装することによって、デジタルビデオ情報のより効率的な送信、受信、符号化、復号、および/または記憶を行い得る。

10

**【0004】**

[0004]ビデオ圧縮技法は、空間的（ピクチャ内）予測および/または時間的（ピクチャ間）予測を実行して、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。ブロックベースのビデオコーディングでは、ビデオスライス（すなわち、ビデオフレームまたはビデオフレームの一部）がビデオブロックに区分され得、これらのビデオブロックは、ツリーブロック、コーディングユニット（CU: coding unit）および/またはコーディングノードと呼ばれることもある。ピクチャのイントラコード化（I）スライス中のビデオブロックは、同じピクチャ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。ピクチャのインターコード化（PまたはB）スライス中のビデオブロックは、同じピクチャ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、または他の参照ピクチャ中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

20

**【0005】**

[0005]空間的予測または時間的予測は、コーディングされるべきブロックのための予測ブロックを生じる。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ブロックとの間のピクセル差分を表す。インターコード化ブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルとコード化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とに従って符号化される。イントラコード化ブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは、ピクセル領域から変換領域に変換され、残差変換係数が生じ得、その残差変換係数は、次いで量子化され得る。最初に2次元アレイで構成される量子化変換係数は、変換係数の1次元ベクトルを生成するために走査され得、なお一層の圧縮を達成するためにエントロピーコーディングが適用され得る。

30

**【発明の概要】****【0006】**

[0006]本開示の技法によれば、マルチレイヤビデオデータのレイヤ組合せのためのレイヤ間依存タイプをシグナリングすることに関連するシグナリングオーバーヘッドは、ビデオデータのビットストリーム中に、各レイヤ組合せのための依存タイプシンタックス要素がシグナリングされるかどうかを表すシンタックス要素を含めることによって低減され得る。依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、ビデオデコーダは、すべてのレイヤ組合せのための依存のタイプが所定のタイプであると決定し、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号する。

40

**【0007】**

[0007]一例では、マルチレイヤビデオデータを復号する方法は、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、依存タイプシンタックス要素が、参照レイヤに

50

対する現在のレイヤの依存のタイプを識別する、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定することと、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号することとを行うこととを含む。

**【0008】**

[0008]別の例では、マルチレイヤビデオデータを符号化する方法は、複数のレイヤを備えるコード化されたビデオデータについて、複数のレイヤの符号化を、複数のレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することと、ここにおいて、依存タイプが、参照レイヤに対する現在のレイヤの依存のタイプを識別する、複数のレイヤの符号化を、同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を決定することとを含む。

10

**【0009】**

[0009]別の例では、ビデオコーディングを実行する装置は、ビデオデータを記憶するメモリと、1つまたは複数のプロセッサを備えるビデオコーダとを含み、1つまたは複数のプロセッサは、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定することと、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号することとを行うこととを行うように構成される。

20

**【0010】**

[0010]別の例では、1つまたは複数のプロセッサによって実行されたとき、1つまたは複数のプロセッサに、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定することと、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号することとを行うこととを行わせる命令を記憶したコンピュータ可読記憶媒体。

30

**【0011】**

[0011]現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信するための手段と、ここにおいて、依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定するための手段と、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号するための手段とを含む、マルチレイヤビデオデータを復号するための装置。

40

**【0012】**

[0012]複数のレイヤを備えるコード化されたビデオデータについて、複数のレイヤの符号化を、複数のレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定するための手段と、ここにおいて、依存タイプが、参照レイヤに対する現在のレイヤの依存のタイプを識別する、複数のレイヤの符号化を、同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、現在のレイヤのための依存タイプシンタッ

50

クス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を決定するための手段とを備える、ビデオデータをコーディングするための装置。

【0013】

[0013] 1つまたは複数の例の詳細が以下の添付の図面および説明において記載されている。他の特徴、目的、および利点は、説明および図面、ならびに特許請求の範囲から明らかになる。

【図面の簡単な説明】

【0014】

【図1】[0014]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。

10

【図2】[0015]本開示で説明する1つまたは複数の例による、例示的なマルチビューの符号化または復号の順序を示すグラフィック図。

【図3】[0016]例示的なMVCの時間的予測およびビュー間予測構造を示す図。

【図4】[0017]本開示で説明する技法を実施し得る例示的なビデオエンコーダを示すブロック図。

【図5】[0018]本開示で説明する技法を実施し得る例示的なビデオデコーダを示すブロック図。

【図6】[0019]ネットワークの一部を形成するデバイスの例示的なセットを示すブロック図。

【図7】[0020]本開示の技法による、ビデオデータを符号化する例示的な方法を示すフローチャート。

20

【図8】[0021]本開示の技法による、ビデオデータを復号する例示的な方法を示すフローチャート。

【発明を実施するための形態】

【0015】

[0022]本開示は、新生のマルチビュー、高効率ビデオコーディング(MV-HEVC: Multi-View, High Efficiency Video Coding)規格などのマルチレイヤビデオコーディングにおけるレイヤ間依存タイプのシグナリングに関する技法をもたらす。高効率ビデオコーディング(HEVC)規格では、レイヤは、概して、ビデオコーディングレイヤ(VCL)と、特定のレイヤ識別値(たとえば、HEVCにおける「nuh\_layer\_id」値、ここで、「nuh」は「NALユニットヘッダ」を指す)をすべてが有する(すなわち、各々が共有する)ネットワークアブストラクションレイヤ(NAL)ユニットと、レイヤに関連する非VCL NALユニットとのセットを指す。レイヤは、たとえば、マルチビュービデオデータをコーディングするときはビューに対応し、スケーラブルビデオコーディングの場合は様々なスケーラブルな寸法(たとえば、空間解像度、PSNR、カラービット深度または他のそのような寸法)に対応し得る。ビデオストリームの各レイヤは、レイヤ識別値に関連付けられる。したがって、異なるレイヤは、異なるレイヤ識別値に関連付けられる。

30

【0016】

[0023]レイヤ間予測は、概して、参照ピクチャの予測されているブロックに関連するレイヤ(たとえば、現在のレイヤ)のレイヤとは異なるレイヤの、サンプル値または動き情報などのデータ要素に依存する予測(たとえば、すでにコーディングされているビデオブロックからビデオブロックを予測すること)を指す。現在のレイヤは、参照レイヤに対する特定のレイヤ間依存タイプを有し得る。レイヤ間依存タイプは、現在のレイヤが参照レイヤに依存し得る方法を定義する。レイヤ間依存タイプの例としては、レイヤ間サンプル予測、レイヤ間動き予測、およびレイヤ間サンプル予測とレイヤ間動き予測との組合せがある。レイヤ間サンプル予測が有効化されるとき、ビデオコーダは、異なるレイヤ(すなわち、参照レイヤ)の参照ピクチャ中のブロックに基づいてビデオデータの現在のブロックを予測し得る。ビデオコーダは、たとえば、視差動きベクトルを使用して異なるレイヤ中のブロックを識別し得る。視差動きベクトルは、たとえば、現在のブロックと同じレイ

40

50

ヤ中のブロックである、現在のブロックの空間的または時間的に隣接するブロックの動き情報に基づいて現在のブロックについて決定され得る。

【 0 0 1 7 】

[0024]レイヤ間動き予測が有効化される時、ビデオコーダは、異なるレイヤ中のブロックの動き情報に基づいて現在のブロックについての動き情報を予測し得る。動き情報は、たとえば、動きベクトル（たとえば、視差動きベクトルまたは時間的動きベクトル）を含み得、いくつかの例では、動き情報はまた、参照ピクチャインデックスなどの追加情報を含み得る。他のレイヤ（すなわち、現在のブロックに関連するレイヤ以外のレイヤ）中のブロックの動き情報を使用して、ビデオコーダは、現在のブロックについての動き情報を決定し得、決定された動き情報に基づいて、現在のブロックと同じレイヤの参照ピクチャ中の予測ブロックを識別し得る。レイヤ間サンプル予測とレイヤ間動き予測との組合せが有効化される時、ビデオコーダは、レイヤ間サンプル予測とレイヤ間動き予測との両方を実行し得る。

10

【 0 0 1 8 】

[0025]レイヤ間依存タイプをシグナリングすることのいくつかの実装形態によれば、ビデオコーダは、組合せのためのレイヤ間依存タイプを識別するために、ビットストリーム中のレイヤの組合せごとにシンタックス要素（たとえば、HEVCにおける「direct\_dependency\_type」シンタックス要素）をコーディングする。シンタックス要素の各インスタンスが2ビット以上である場合、レイヤ間依存タイプのシグナリングは、特に、多数のレイヤをもつビットストリームの場合に望ましくないシグナリングオーバーヘッドを生成し得る。本開示の技法によれば、レイヤ間動き予測またはレイヤ間サンプル予測のうちの1つが無効化され、したがって、ビデオコーダがレイヤ間依存タイプをシグナリングする必要があるインスタンスを低減または除去し得る。ビデオエンコーダは、特定のタイプのレイヤ間予測が無効化されていることをビデオデコーダにシグナリングし得る。

20

【 0 0 1 9 】

[0026]レイヤ間サンプル予測またはレイヤ間動き予測のうちの1つを無効化することは、レイヤ間サンプル予測またはレイヤ間動き予測のうちの1つを有効化することに機能的に相当し得る。たとえば、3つのレイヤ間依存タイプ（たとえば、レイヤ間サンプル予測のみ、レイヤ間動き予測のみ、およびレイヤ間サンプル予測とレイヤ間動き予測との組合せ）の場合、ビデオデコーダが、レイヤ間サンプル予測を無効化する場合、ビデオデコーダは、レイヤ間サンプル予測のみと、レイヤ間サンプル予測とレイヤ間動き予測との組合せとの両方を無効化し、それによって、レイヤ間動き予測のみを有効化する。同様に、ビデオデコーダがレイヤ間動き予測を無効化する場合、ビデオコーダは、レイヤ間サンプル予測を有効化する。

30

【 0 0 2 0 】

[0027]本開示の技法の一例によれば、ビデオデコーダは、現在のレイヤのための第2のシンタックス要素（すなわち、依存タイプシンタックス要素）がシグナリングされるかどうかを表す（すなわち、示す）第1のシンタックス要素（たとえば、以下で詳細に説明する「direct\_dep\_type\_minus\_2」シンタックス要素）の値を受信し得る。依存タイプシンタックス要素は、参照レイヤに対する現在のレイヤの依存のタイプを識別し得る。依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に回答して、ビデオデコーダは、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定し、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号し得る。

40

【 0 0 2 1 】

[0028]いくつかの実装形態では、所定のタイプは、たとえば、ビデオコーデックの一部としてビデオエンコーダおよびデコーダ中に構成され得る（たとえば、メモリ中にローカルに記憶され得る）。そのような構成では、第1のシンタックス要素は、シングルビットであり得、ここで、第1のシンタックス要素の第1の値（たとえば、「0」の値または「

50



「1」の値)は、すべてのレイヤのためのレイヤ間依存タイプが所定のタイプであることを示す。したがって、第1のシンタックス要素の第1の値を受信したことに応答して、ビデオデコーダは、第2のシンタックス要素のインスタンスを受信することなしに、ビデオビットストリームのすべてのレイヤ組合せのためのレイヤ間依存タイプがビデオコーデックによって定義された(または、ローカルに記憶された)所定のレイヤ間依存タイプであると決定し得る。第1のシンタックス要素の第2の値(たとえば、「1」の値または「0」の値)は、レイヤ間依存タイプがレイヤ組合せごとに別個にシグナリングされることになることを示し得る。したがって、第1のシンタックス要素の第2の値を受信したことに応答して、ビデオデコーダは、レイヤ組合せごとに第2のシンタックス要素のインスタンスを受信することによって(すなわち、受信に基づいて)レイヤの組合せのためのレイヤ間依存タイプを決定し得る。

10

**【0022】**

[0029]他の実装形態では、所定のタイプは、たとえば、第1のシンタックス要素の一部として、ビデオエンコーダからビデオデコーダにシグナリングされ得る。そのような構成では、第1のシンタックス要素は、マルチビットシンタックス要素であり得、ここで、第1のシンタックス要素の第1の値は、すべてのレイヤ組合せのためのレイヤ間依存タイプがレイヤ間サンプル予測であることを示し、第1のシンタックス要素の第2の値は、すべてのレイヤ組合せのためのレイヤ間依存タイプがレイヤ間動き予測であることを示す。したがって、第1のシンタックス要素の第1の値または第2の値を受信したことに応答して、ビデオデコーダは、第2のシンタックス要素のインスタンスを受信することなしに、ビデオビットストリームのすべてのレイヤ組合せのためのレイヤ間依存タイプが第1のシンタックス要素によって示される所定のレイヤ間依存であると決定し得る。第1のシンタックス要素の第3の値は、レイヤ間依存タイプがレイヤ組合せごとに別個にシグナリングされることになることを示し得る。したがって、第1のシンタックス要素の第3の値を受信したことに応答して、ビデオデコーダは、レイヤ組合せごとに第2のシンタックス要素のインスタンスを受信することによって(受信に基づいて)レイヤの組合せのためのレイヤ間依存タイプを決定し得る。

20

**【0023】**

[0030]本開示の技法によれば、レイヤ間動き予測またはレイヤ間サンプル予測のいずれが無効化されるとき、ビデオデコーダは、いかなる追加のシグナリングもなしに(たとえば、上記で説明した第2のシンタックス要素のインスタンスを受信することなしに)、レイヤの組合せのためのレイヤ間依存タイプを決定するように構成され得る。たとえば、レイヤ間動き予測が無効化される場合、ビデオデコーダは、レイヤ組合せのための追加のシグナリングなしに、レイヤ組合せのためのレイヤ間依存タイプがレイヤ間サンプル予測であると決定するように構成され得る。既存の技法は、常に第2のシンタックス要素(すなわち、依存タイプシンタックス要素)の複数のインスタンスをコーディングし得るが、本開示の技法は、いくつかのコーディングシナリオでは、第1のシンタックス要素の1つの(すなわち、単一の)インスタンスのみをコーディングし、第2のシンタックス要素の複数のインスタンスをコーディングすることを回避し得る(すなわち、それを控え得るおよび/またはそれを必要としないことがある)。いくつかのレイヤ組合せのためのレイヤ間依存タイプ(たとえば、上記で説明した第2のシンタックス要素)をシグナリングする必要を低減または除去することによって、本開示の技法は、いくつかのコーディングシナリオでは、レイヤ間依存タイプをシグナリングすることに関連するシグナリングオーバーヘッドを低減し得る。

30

40

**【0024】**

[0031]本開示で使用する現在のブロックという用語は、概して、すでにコーディングされているブロックまたはまだコーディングされていないブロックとは対照的に、現在コーディングされているブロックを指すように意図されている。本開示で使用するコーディングという用語は、概して、ビデオ符号化またはビデオ復号のいずれかを指すために使用され得る。

50

## 【 0 0 2 5 】

[0032] 図 1 は、本開示で説明するレイヤ間依存タイプシグナリング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 に示されたように、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マルチレイヤビデオデータを含む符号化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 および宛先デバイス 14 は、ワイヤレス通信に対する機能を備え得る。

10

## 【 0 0 2 6 】

[0033] 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 に符号化ビデオデータを移動することが可能な任意のタイプの媒体またはデバイスを備え得る。一例では、リンク 16 は、ソースデバイス 12 が、符号化ビデオデータをリアルタイムで宛先デバイス 14 に直接送信することを可能にするための通信媒体を備え得る。符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス 14 に送信され得る。通信媒体は、無線周波数（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など、パケットベースのネットワークの一部を形成し得る。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス 12 から宛先デバイス 14 への通信を容易にするために有用であり得る任意の他の機器を含み得る。

20

## 【 0 0 2 7 】

[0034] 代替的に、符号化データは出力インターフェース 22 からストレージデバイス 17 に出力され得る。同様に、符号化データは、入力インターフェースによってストレージデバイス 17 からアクセスされ得る。ストレージデバイス 17 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、揮発性または不揮発性メモリ、あるいは符号化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 17 は、ソースデバイス 12 によって生成された符号化ビデオを保持し得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 17 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、符号化ビデオデータを記憶し、その符号化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、ウェブサイトのための）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、任意の標準のデータ接続を通して符号化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ビデオデータにアクセスするのに好適であるワイヤレスチャネル（たとえば、ワイヤレスローカルエリアネットワーク（WLAN）接続）、ワイヤード接続（たとえば、DSL、ケーブルモデムなど）、またはその両方の組合せを含み得る。ストレージデバイス 17 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。

30

40

## 【 0 0 2 8 】

[0035] 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

50

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の、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。

#### 【0029】

[0036]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22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12において、ビデオソース18は、ビデオキャプチャデバイス、たとえばビデオカメラ、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18がビデオカメラである場合、ソースデバイス12および宛先デバイス14は、いわゆるカメラフォンまたはビデオフォンを形成し得る。しかしながら、本開示で説明する技法は、ビデオコーディング全般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。

#### 【0030】

[0037]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20によって符号化され得る。符号化ビデオデータは、ソースデバイス12の出力インターフェース22を介して宛先デバイス14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14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17上に記憶され得る。

#### 【0031】

[0038]宛先デバイス14は、入力インターフェース28と、ビデオデコーダ30と、ディスプレイデバイス32とを含む。場合によっては、入力インターフェース28は、受信機および/またはモデムを含み得る。宛先デバイス14の入力インターフェース28は、リンク16を介して符号化ビデオデータを受信する。リンク16を介して通信され、またはストレージデバイス17上に提供された符号化ビデオデータは、ビデオデータを復号する際にビデオデコーダ30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20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。

#### 【0032】

[0039]ディスプレイデバイス32は、宛先デバイス14と一体化されるかまたはその外部にあり得る。いくつかの例では、宛先デバイス14は、一体型ディスプレイデバイスを含み、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14はディスプレイデバイスであり得る。概して、ディスプレイデバイス32は、復号ビデオデータをユーザに表示し、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプのディスプレイデバイスなどの様々なディスプレイデバイスのいずれかを備え得る。

#### 【0033】

[0040]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、そのスケラブルビデオコーディング(SVC)拡張、マルチビュービデオコーディング(MVC)拡張、およびMVCベース3DV拡張を含む、ISO/IEC MPEG-4 Visualおよび(ISO/IEC MPEG-4 AVCとしても知られる)ITU-T

10

20

30

40

50

H.264などのビデオ圧縮規格に従って動作する。いくつかの事例では、H.264/AVCのMVCベース3DV拡張に準拠するどんなビットストリームも、H.264/AVCのMVC拡張に準拠するサブビットストリームを常に含む。さらに、H.264/AVCへの3次元ビデオ(3DV)コーディング拡張、すなわちAVCベースの3DVを生成するための取り組みが進行中である。他の例では、ビデオエンコーダ20およびビデオデコーダ30は、ITU-T H.261、ISO/IEC MPEG-1 Visual、ITU-T H.262またはISO/IEC MPEG-2 Visual、およびITU-T H.264、ISO/IEC Visualに従って動作し得る。

【0034】

[0041]最近、新しいビデオコーディング規格、すなわち高効率ビデオコーディング(HEVC)の設計が、ITU-T Video Coding Experts Group(VCEG)およびISO/IEC Motion Picture Experts Group(MPEG)のJoint Collaboration Team on Video Coding(JCT-VC)によって確定された。ビデオエンコーダ20およびビデオデコーダ30は、HEVC規格に従って動作し得る。

10

【0035】

[0042]さらに、HEVCのためのスケーラブルビデオコーディング拡張と、マルチビューコーディング拡張と、3DV拡張とを作成する作業が進行中である。3DサービスをサポートするHEVCの2つの拡張は、現在ITU-T VCEGのJoint Collaboration Team on 3D Video coding(JCT-3V)およびISO/IEC MPEGによって開発されており、それぞれ、MV-HEVCおよび3D-HEVCと呼ばれる。3D-HEVCは、マルチビュービデオプラス深度フォーマットをコーディングし、HEVCコーディングモジュールに加えて新しいコーディングツールを含む。新たに導入されたコーディングツールは、テクスチャコーディングと深度コーディングの両方に適用可能である。MV-HEVCは、HEVCのブロックレベル設計を変更することなしに、複数の(テクスチャ)ビューのコーディングをサポートするHEVCのマルチビューコーディング拡張を指す。MV-HEVCの最近のワーキングドラフト(WD)、以下WD4は、本明細書によって、参照によってインコーポレートであり、2014年9月25日現在、[http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/4\\_Incheon/wg11/JCT3V-D1004-v1.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1004-v1.zip)において利用可能である。

20

30

【0036】

[0043]HEVCのスケーラブルビデオコーディング拡張はSHEVCと呼ばれることがある。以下SHEVC WD2と呼ぶ、SHEVCの最近のワーキングドラフト(WD)は、本明細書によって、参照によってインコーポレートであり、2014年9月25日現在、[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M1008-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M1008-v1.zip)において利用可能である。ビデオエンコーダ20およびビデオデコーダ30は、SHEVC、MV-HEVC、および/または3D-HEVCのいずれかに従って動作し得る。

40

【0037】

[0044]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20およびビデオデコーダ30は、それぞれ、オーディオエンコーダおよびデコーダと統合され得、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MUX-DEMUXユニット、または他のハードウェアおよびソフトウェアを含み得る。適用可能な場合、いくつかの例では、MUX-DEMUXユニットは、ITU H.223マルチプレクスサブプロトコル、またはユーザデータプロトコル(UDP)などの他のプロトコルに準拠し得る。

【0038】

[0045]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

50

マイクロプロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、ディスクリート論理、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、様々な好適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、ソフトウェアに対する命令を好適な非一時的コンピュータ可読媒体に記憶し、本開示の技法を実行するための1つまたは複数のプロセッサを使用してハードウェアにおいて命令を実行し得る。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得、そのいずれかが、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得る。

10

#### 【0039】

[0046]上記で紹介したように、JCT-VCは、最近、HEVC規格の開発を完成した。HEVC規格化の取り組みは、HEVCテストモデル(HM)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いていた。HMは、たとえば、ITU-T H.264/AVCに従う既存のデバイスに対して、ビデオコーディングデバイスのいくつかの追加の能力を仮定する。たとえば、H.264は9つのイントラ予測符号化モードを提供するが、HMは33個ものイントラ予測符号化モードを提供し得る。

#### 【0040】

[0047]概して、HMの作業モデルは、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: largest coding unit)に分割され得ることを記述する。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する。スライスは、コーディング順序でいくつかの連続する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、コード化ビデオブロックを備える。コード化ビットストリームに関連付けられた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。

20

30

#### 【0041】

[0048]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: prediction unit)および変換ユニット(TU: transform unit)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、8x8ピクセルから最大64x64ピクセル以上をもつツリーブロックのサイズにまでわたり得る。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の1つまたは複数のPUへの区分を記述し得る。区分モードは、CUがスキップであるのか、または直接モード符号化されるのか、イントラ予測モード符号化されるのか、またはインター予測モード符号化されるのかの間で異なり得る。PUは、形状が非正方形に区分され得る。CUに関連するシンタックスデータはまた、たとえば、4分木に従うCUの1つまたは複数のTUへの区分を記述し得る。TUは、形状が方形または非方形であり得る。

40

#### 【0042】

[0049]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のために定義された所与のCU内のPUのサイズに基づいてサイズ決定されるが、これは常にそうであるとは限らない。TUは、一般に、PUと同じサイズであるか、またはそれよりも小さい。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ため

50

に変換され、その変換係数は量子化され得る。

【 0 0 4 3 】

[0050]概して、PUは、予測プロセスに関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されるとき、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されるとき、PUは、PUのための動きベクトルを定義するデータを含み得る。PUのための動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルについての解像度（たとえば、1/4ピクセル精度または1/8ピクセル精度）、動きベクトルが指す参照ピクチャ、および/または動きベクトルのための参照ピクチャリスト（たとえば、リスト0、リスト1、またはリストC）を記述し得る。

10

【 0 0 4 4 】

[0051]以下でより詳細に説明するように、インターモードコーディングの一部として、ビデオエンコーダ20は、マルチレイヤビデオデータのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定し得る。ビデオエンコーダ20は、次いで、使用される依存タイプに基づいて相応にシグナリングを行うか、またはシグナリングを省略し得る。ビデオデコーダ30は、シグナリングを介して、所定のタイプとともに構成されるか、またはそれを潜在的に受信し、したがって、依存タイプがシグナリングされない場合、ビデオデコーダ30は、所定のタイプを単に適用し得る。

【 0 0 4 5 】

20

[0052]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)をも含み得る。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。残差値はピクセル差分値を備え、ピクセル差分値は、エントロピーコーディングのためのシリアル化変換係数(serialized transform coefficient)を生成するために、TUを使用して変換係数に変換され、量子化され、走査され得る。本開示では、一般に、CUのコーディングノードを指すために「ビデオブロック」という用語を使用する。いくつかの特定の場合において、本開示では、コーディングノードならびにPUおよびTUを含む、ツリーブロック、すなわち、LCUまたはCUを指すためにも「ビデオブロック」という用語を使用し得る。

30

【 0 0 4 6 】

[0053]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1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は、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、一般に、ビデオデータを符号化するために、個々のビデオスライス内のビデオブロックに作用する。ビデオブロックはCU内のコーディングノードに対応し得る。ビデオブロックは、固定サイズまたは変動サイズを有し得、指定されたコーディング規格に応じてサイズが異なり得る。

40

【 0 0 4 7 】

[0054]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と、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称PUサイズでのインター予測とをサポートする。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の区分をサポートする。非対称の区分では、CUの一方向は区分されないが、他の方向は25%および75%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 $2N \times 1$

50

、5 N PUとで水平方向に区分された2 N × 2 N CUを指す。

【0048】

[0055]本開示では、「N × N (NxN)」および「N × N (N 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 × 16 (16x16)ピクセルまたは16 × 16 (16 by 16)ピクセルを指すために互換的に使用され得る。概して、16 × 16ブロックは、垂直方向に16ピクセル(y = 16)、および水平方向に16ピクセル(x = 16)を有する。同様に、N × Nブロックは、概して、垂直方向にNピクセルを有し、水平方向にN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および列に配列され得る。さらに、ブロックは、必ずしも、水平方向に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 × Mピクセルを備え得、ここで、Mは必ずしもNに等しいとは限らない。

10

【0049】

[0056]CUのPUを使用したイントラ予測コーディングまたはインター予測コーディングに続いて、ビデオエンコーダ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のための変換係数を生成するためにTUを変換し得る。

20

【0050】

[0057]変換係数を生成するための任意の変換の後で、ビデオエンコーダ20は、変換係数の量子化を実行し得る。量子化は、一般に、係数を表すために使用されるデータの量をできるだけ低減するために変換係数が量子化され、さらなる圧縮を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、nビットの値は、量子化中にmビットの値に切り捨てられ得、ここで、nはmよりも大きい。

【0051】

[0058]いくつかの例では、ビデオエンコーダ20は、量子化変換係数を走査して、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、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次元のベクトルを形成した後、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: context-adaptive variable length coding)、コンテキスト適応型バイナリ算術コーディング(CABAC: context-adaptive binary arithmetic coding)、シンタックスベースコンテキスト適応型バイナリ算術コーディング(SBAC: syntax-based context-adaptive binary arithmetic coding)、確率間隔区分エントロピー(PIPE: Probability Interval Partitioning Entropy)コーディングまたは別のエントロピー符号化の方法に従って、1次元のベクトルをエントロピー符号化し得る。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に関連するシンタックス要素をエントロピー符号化し得る。

30

40

【0052】

[0059]CABACを実行するために、ビデオエンコーダ20は、コンテキストモデル内のコンテキストを、送信されるべきシンボルに割り当て得る。コンテキストは、たとえば、シンボルの近隣値が非0であるか否かに関係し得る。CAVLC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ための可変長コードを選択し得る。VLC中のコードワードは、比較的より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、VLCの使用は、たとえば、送信されるべき各シンボルのための等長コードワードを使用することに勝るビット節約を達成し得る。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き得る

50

## 【 0 0 5 3 】

[0060]上記で紹介したように、本開示は、M V - H E V Cを含むマルチレイヤビデオコーディングにおけるレイヤ間依存タイプのシグナリングに関する技法について説明する。本開示で説明する技法はまた、S H E V Cまたは他のコーデックなどのH E V Cの他のマルチレイヤ拡張に潜在的に適用され得る。上記で紹介したように、M V Cは、H . 2 6 4 / A V Cの拡張である。次に、将来の説明のためのコンテキストおよびバックグラウンドを提供するために、M V C仕様の態様について簡単に説明する。

## 【 0 0 5 4 】

[0061]図2は、本開示で説明する1つまたは複数の例による、例示的なM V Cの符号化または復号の順序を示すグラフィック図である。たとえば、図2に示された復号順序構成は、時間優先コーディングと呼ばれる。図2では、S 0 ~ S 7はそれぞれ、マルチビュービデオの異なるビューを指す。T 0 ~ T 8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ド化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T 0についてのビューS 0 ~ S 7(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T 1についてのビューS 0 ~ S 7(すなわち、ピクチャ8 ~ 15)のすべてを含み、以下同様である。この例では、ピクチャ0 ~ 7は、同じ時間インスタンス(すなわち、時間インスタンスT 0)におけるものであり、ピクチャ8 ~ 15は、同じ時間インスタンス(すなわち、時間インスタンスT 1)におけるものである。同じ時間インスタンスをもつピクチャは概して同時に表示され、同じ時間インスタンスのピクチャ内のオブジェクト間の水平視差と、場合によっては何らかの垂直視差は、閲覧者に、3Dボリュームを包含する画像を知覚させる。

## 【 0 0 5 5 】

[0062]典型的なM V C復号順序(すなわち、ビットストリーム順序)を図2に示す。復号順序構成は時間優先(time-first)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ド化ピクチャを含んでいるように定義される。アクセスユニットの復号順序は出力または表示の順序と同じでないことがあることに留意されたい。

## 【 0 0 5 6 】

[0063]図3は、例示的なマルチビュービデオコーディング(M V C)の予測パターンを示す概念図である。図3の例では、8個のビューが示され、ビューごとに12個の時間位置が示される。概して、図3中の各行はビューに対応し、各列は時間ロケーションを示す。ビューの各々は、他のビューに対する相対的なカメラロケーションを示すために使用され得るビュー識別子(「view\_id」)を使用して識別され得る。図3に示す例では、ビューIDは「S 0」~「S 7」として示されているが、数字のビューIDも使用され得る。さらに、時間ロケーションの各々は、ピクチャの表示順序を示すピクチャ順序カウント(POC)値を使用して識別され得る。図3に示す例では、POC値は「T 0」~「T 11」として示されている。

## 【 0 0 5 7 】

[0064]図3中のピクチャは、対応するピクチャがイントラコード化される(すなわち、Iフレームである)のか、または一方向に(すなわち、Pフレームとして)インターコード化されるのか、複数の方向に(すなわち、Bフレームとして)インターコード化されるのかを指定する、文字を含む影付きブロックを使用して示される。概して、予測は矢印によって示され、ここで、矢印の終点のピクチャは、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。たとえば、時間ロケーションT 0におけるビューS 2のPフレームは、時間ロケーションT 0におけるビューS 0のIフレームから予測される。図3に示されたピクチャの各々は、ビューコンポーネントと呼ばれることがある。

## 【 0 0 5 8 】

[0065]シングルビュービデオ符号化の場合と同様に、マルチビュービデオシーケンスの

10

20

30

40

50



ピクチャは、異なる時間位置におけるピクチャに関して予測符号化され得る。たとえば、時間位置 T 1 にあるビュー S 0 の b フレームは、時間位置 T 0 にあるビュー S 0 の I フレームからその b フレームに向けられた矢印を有し、その矢印は、b フレームが I フレームから予測されることを示す。しかしながら、さらに、マルチビュービデオ符号化のコンテキストにおいて、ピクチャはビュー間予測され得る。すなわち、ビューコンポーネントは、参照のために他のビュー中のビューコンポーネントを使用することができる。MVC では、たとえば、別のビュー中のビューコンポーネントがインター予測参照であるかのように、ビュー間予測が実現される。潜在的なビュー間参照は、SPS の MVC 拡張においてシグナリングされ得、インター予測参照またはビュー間予測参照の柔軟な順序付けを可能にする、参照ピクチャリストの構築プロセスによって修正され得る。

10

【0059】

[0066] 図 3 は、ビュー間予測の様々な例を提供する。図 3 の例では、ビュー S 1 のピクチャは、ビュー S 1 の様々な時間的位置にあるピクチャから予測されるものとして、ならびに同じ時間的位置にあるビュー S 0 およびビュー S 2 のピクチャのうちのピクチャからビュー間予測されるものとして示されている。たとえば、時間位置 T 1 にあるビュー S 1 の b フレームは、時間位置 T 0 および T 2 にあるビュー S 1 の B フレームの各々、ならびに時間位置 T 1 にあるビュー S 0 およびビュー S 2 の b フレームから予測される。

【0060】

[0067] 図 3 の例では、大文字の「B」および小文字の「b」は、異なる符号化方法ではなく、ピクチャ間の異なる階層関係を示すものとする。概して、大文字の「B」フレームは、小文字の「b」フレームよりも予測階層が比較的高い。図 3 はまた、異なるレベルの陰影を使用して予測階層の差異を示し、より陰影の量が多い（すなわち、比較的暗い）ピクチャは、より陰影が少ない（すなわち、比較的明るい）ピクチャよりも予測階層が高い。たとえば、図 3 のすべての I フレームは完全な陰影によって示されるが、P フレームはいくぶん明るい陰影を有し、B フレーム（および小文字の b フレーム）は、互いに対して様々なレベルの陰影を有するが、P フレームおよび I フレームの陰影よりも常に明るい。

20

【0061】

[0068] 一般に、階層の比較的高いピクチャが、階層の比較的低いピクチャの復号中に参照ピクチャとして使用され得るように、予測階層の比較的高いピクチャは、階層の比較的低いピクチャを復号する前に復号されるべきであるという点で、予測階層はビュー順序インデックスに関係する。ビュー順序インデックスは、アクセスユニット中のビューコンポーネントの復号順序を示すインデックスである。ビュー順序インデックスは、SPS のようなパラメータセット中で示唆され得る。

30

【0062】

[0069] このようにして、参照ピクチャとして使用されるピクチャは、その参照ピクチャを参照して符号化されたピクチャを復号する前に復号され得る。ビュー順序インデックスは、アクセスユニット中のビューコンポーネントの復号順序を示すインデックスである。MVC / AVC によれば、ビュー順序インデックス  $i$  ごとに、対応する `view_id` がシグナリングされる。ビューコンポーネントの復号は、ビュー順序インデックスの昇順に従う。すべてのビューが提示された場合、ビュー順序インデックスのセットは、0 からビューの全数よりも 1 少ない数まで連続的に順序付けされたセットを備える。

40

【0063】

[0070] いくつかの例では、全ビットストリームのサブセットが抽出されて、依然 MVC に準拠するサブビットストリームを形成することができる。たとえば、サーバによって提供されるサービス、1 つもしくは複数のクライアントのデコーダの容量、サポート、および能力、ならびに / または、1 つもしくは複数のクライアントの選好に基づいて、特定の適用例が必要とし得る多くの可能なサブビットストリームが存在する。たとえば、あるクライアントが 3 つのビューのみを必要とする場合があり、2 つのシナリオがあり得る。一例では、あるクライアントは滑らかな鑑賞体験を求めることがあり、`view_id` 値 S

50

0、S 1、および S 2 のビューを好むことがあり、別の他のクライアントはビュースケラビリティを求めることがあり、view\_id 値 S 0、S 2、および S 4 のビューを好むことがある。これらのサブビットストリームの両方が、独立した MVC ビットストリームとして復号され得、同時にサポートされ得る。

【 0 0 6 4 】

[0071] 上記のように、図 3 は 8 個のビュー ( S 0 ~ S 7 ) を示すが、MVC / AVC 拡張は最大で 1 0 2 4 個のビューをサポートし、NAL ユニットヘッダ内の view\_id を使用して、NAL ユニットが属するビューを識別する。ビュー順序インデックスは、NAL ユニットヘッダの一部としてシグナリングされ得る。すなわち、比較の目的で、ビュー順序インデックスは、MVC / AVC 拡張の NAL ユニットヘッダ内でシグナリングされる view\_id を置き換え得る。ビュー順序は、一般に、アクセスユニット内のビューの順序付けを記述し、ビュー順序インデックスは、そのアクセスユニットのビュー順序における特定のビューを識別する。すなわち、ビュー順序インデックスは、アクセスユニットの対応するビューコンポーネントの復号順序を記述する。

10

【 0 0 6 5 】

[0072] したがって、SPS は、ビューの view\_id とビューのビュー順序インデックスとの間の関係を提供し得る。ビュー順序インデックスと SPS 内のデータとを使用して、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、NAL ユニットヘッダ内の MVC / AVC の 1 0 ビットの view\_id をビュー順序インデックスによって置き換えて、これは、MVC / AVC 方式よりもビット節約につながり得る。

20

【 0 0 6 6 】

[0073] マルチビュービデオコーディングのための ( 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 ) 典型的な MVC 予測構造を図 3 に示す。予測は、矢印によって示される。図 3 の例では、矢印の終点のオブジェクトは、予測の基準のために矢印の始点のオブジェクトを使用する。

【 0 0 6 7 】

[0074] MVC では、H . 2 6 4 / 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。2 つのビューのコーディングも MVC によってサポートされ得る。MVC の潜在的な利点のうちの 1 つは、MVC エンコーダが 3 D 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、また、MVC デコーダ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、MVC デコーダを有する任意のレンダラは、3 つ以上のビューをもつ 3 D ビデオコンテンツを予想し得る。

30

【 0 0 6 8 】

[0075] 次に、MVC ビュー間予測の態様について説明する。MVC 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 1 つの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有する場合、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まったく同様に、参照ピクチャリストの任意の位置に置かれ得る。

40

【 0 0 6 9 】

[0076] 次に、MVC 拡張のためのシーケンスパラメータセット ( SPS ) の態様について説明する。図 3 に示すように、ビューコンポーネントは、参照のために他のビュー中のビューコンポーネントを使用することができ、これは、ビュー間予測と呼ばれる。MVC では、別のビュー中のビューコンポーネントがインター予測参照であるかのように、ビュー間予測が実現される。しかしながら、潜在的なビュー間参照は、( 以下の表 1 に示すように ) SPS MVC 拡張においてシグナリングされ、インター予測またはビュー間予測参照のフレキシブルな順序付けを可能にする、ビデオコーダによって実施される参照ピクチャリスト構成プロセスによって変更され得る。

【 0 0 7 0 】

50

【表 1】

seq_parameter_set_mvc_extension() {		記述子
<b>num_views_minus1</b>		ue(v)
for( i = 0; i <= num_views_minus1; i++ )		
<b>view_id[ i ]</b>		ue(v)
for( i = 1; i <= num_views_minus1; i++ ) {		
<b>num_anchor_refs_10[ i ]</b>		ue(v)
for( j = 0; j < num_anchor_refs_10[ i ]; j++ )		
<b>anchor_ref_10[ i ][ j ]</b>		ue(v)
<b>num_anchor_refs_11[ i ]</b>		ue(v)
for( j = 0; j < num_anchor_refs_11[ i ]; j++ )		
<b>anchor_ref_11[ i ][ j ]</b>		ue(v)
}		
for( i = 1; i <= num_views_minus1; i++ ) {		
<b>num_non_anchor_refs_10[ i ]</b>		ue(v)
for( j = 0; j < num_non_anchor_refs_10[ i ]; j++ )		
<b>non_anchor_ref_10[ i ][ j ]</b>		ue(v)
<b>num_non_anchor_refs_11[ i ]</b>		ue(v)
for( j = 0; j < num_non_anchor_refs_11[ i ]; j++ )		
<b>non_anchor_ref_11[ i ][ j ]</b>		ue(v)
}		
}		
<b>num_level_values_signalled_minus1</b>		ue(v)
for( i = 0; i <= num_level_values_signalled_minus1; i++ )		
{		
<b>level_idc[ i ]</b>		u(8)
<b>num_applicable_ops_minus1[ i ]</b>		ue(v)
for( j = 0; j <= num_applicable_ops_minus1[ i ]; j++ ) {		
<b>applicable_op_temporal_id[ i ][ j ]</b>		u(3)
<b>applicable_op_num_target_views_minus1[ i ][ j ]</b>		ue(v)
for( k = 0; k <=		
<b>applicable_op_num_target_views_minus1[ i ][ j ]; k++ )</b>		
<b>applicable_op_target_view_id[ i ][ j ][ k ]</b>		ue(v)
<b>applicable_op_num_views_minus1[ i ][ j ]</b>		ue(v)
}		
}		
}		
}		

表1

## 【 0 0 7 1 】

[0077]表 1 で上記に示した S P S M V C 拡張では、ビューごとに、参照ピクチャリスト 0 と参照ピクチャリスト 1 とを形成するために使用され得るビューの数がシグナリングされる。

## 【 0 0 7 2 】

[0078] S P S M V C 拡張においてシグナリングされるアンカーピクチャについての予測関係は、同じビューの ( S P S M V C 拡張においてシグナリングされる ) 非アンカーピクチャについての予測関係とは異なり得る。H . 2 6 4 では、アンカーピクチャは、す

10

20

30

40

50

すべてのスライスが同じアクセスユニット内のスライスだけを参照し得るコード化ピクチャである。言い換えれば、ビュー間予測はアンカーピクチャのために使用され得るが、インター予測（すなわち、時間的予測）はアンカーピクチャのために使用されない。H.264は、非アンカーピクチャを、アンカーピクチャでないピクチャとして定義する。

【0073】

[0079]次に、MV-HEVCにおける依存タイプの態様について説明する。MV-HEVCでは、レイヤ間依存タイプは、少なくとも3つのタイプがサポートされる方法でシグナリングされる。3つのタイプは、(MVCの場合のような)レイヤ間サンプル予測、レイヤ間動き予測、および両方の組合せである。

【0074】

[0080]以下に示すように、MV-HEVCのビデオパラメータセット(VPS)拡張では、シンタックス要素「direct\_dep\_type\_len\_minus2」は、何個のレイヤ間予測タイプがサポートされるのかを指定し、一方、シンタックス要素「direct\_dependency\_type[i][j]」は、厳密なレイヤ間予測タイプを指定する。より具体的には、シンタックス要素「direct\_dependency\_type[i][j]」は、依存レイヤ[i]のタイプがレイヤ[j]に関して有することを識別する。

【0075】

[0081]VPSセマンティクスに関するMV-HEVC WD5の一部を以下に再現する。

【0076】

10

20

【表 2】

## ビデオパラメータセット拡張シンタックス

	記述子
<code>vps_extension() {</code>	
<code>while( !byte_aligned() )</code>	
<code>    vps_extension_byte_alignment_reserved_one_bit</code>	u(1)
<code>    avc_base_layer_flag</code>	u(1)
<code>    splitting_flag</code>	u(1)
<code>    for( i = 0, NumScalabilityTypes = 0; i &lt; 16; i++ ) {</code>	
<code>        scalability_mask[ i ]</code>	u(1)
<code>        NumScalabilityTypes += scalability_mask[ i ]</code>	
<code>    }</code>	
<code>    ...</code>	
<code>    num_output_layer_sets</code>	ue(v)
<code>    for( i = 0; i &lt; num_output_layer_sets; i++ ) {</code>	
<code>        output_layer_set_idx_minus1[ i ]</code>	ue(v)
<code>        lsIdx = output_layer_set_idx_minus1[ i ] + 1</code>	
<code>        for( j = 0; j &lt;= vps_max_layer_id; j++ )</code>	
<code>            if( layer_id_included_flag[ lsIdx ][ j ] )</code>	
<code>                output_layer_flag[ lsIdx ][ j ]</code>	u(1)
<code>        }</code>	
<code>    ...</code>	
<code>    direct_dep_type_len_minus2</code>	ue(v)
<code>    for( i = 1; i &lt;= vps_max_layers_minus1; i++ )</code>	
<code>        for( j = 0; j &lt; i; j++ )</code>	
<code>            if( direct_dependency_flag[ i ][ j ] )</code>	
<code>                direct_dependency_type[ i ][ j ]</code>	u(v)
<code>    }</code>	

表2

## 【 0 0 7 7 】

`direct_dep_type_len_minus2`に2を加えたものは、`direct_dependency_type[ i ][ j ]`シンタックス要素のビット数を指定する。この仕様のこのバージョンに準拠するビットストリーム中で、`direct_dep_type_len_minus2`の値は、イコール0であるものとする。この仕様のこのバージョンでは、`direct_dep_type_len_minus2`の値が0に等しくなるものとするが、デコーダは、両端値を含む0～30の範囲内の、`direct_dep_type_len_minus2`の他の値がシンタックス中に現れることを可能にするものとする。

`direct_dependency_type[ i ][ j ]`は、`nuh_layer_id`イコール`layer_id_in_nuh[ i ]`をもつレイヤと`layer_id_in_nuh[ j ]`に等しい`nuh_layer_id`をもつレイヤとの間の依存のタイプを示す。0に等しい`direct_dependency_type[ i ][ j ]`は、`layer_id_in_nuh[ j ]`に等しい`nuh_layer_id`をもつレイヤが、`nuh_layer_id`イコール`layer_id_in_nuh[ i ]`をもつレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用されることを示す。1に等しい`direct_dependency_type[ i ][ j ]`は、`laye`

10

20

30

40

50

`r_layer_id_in_nuh[j]`に等しい`nuh_layer_id`をもつレイヤが、`nuh_layer_id`イコール`layer_id_in_nuh[i]`をもつレイヤのレイヤ間サンプル予測のために使用されるが、レイヤ間動き予測のために使用されないことを示す。2に等しい`direct_dependency_type[i][j]`は、`layer_id_in_nuh[j]`に等しい`nuh_layer_id`をもつレイヤが、`nuh_layer_id`イコール`layer_id_in_nuh[i]`をもつレイヤのレイヤ間動き予測のために使用されるが、レイヤ間サンプル予測のために使用されないことを示す。この仕様のこのバージョンでは、`direct_dependency_type[i][j]`の値は、両端値を含む0~2の範囲内にあるものとするが、デコーダは、両端値を含む $3 \sim 2^{32} - 2$ の範囲内の`direct_dependency_type[i][j]`の値がシンタックス中に出現することを可能にするものとする。

10

【0078】

[0082]いくつかのタイプのマルチレイヤビデオをコーディングするとき、ビデオエンコーダ20およびビデオデコーダ30のパフォーマンスは、いくつかのタイプのレイヤ間予測によって向上されないこと、または最小限にしか向上されないことがあり、特定のタイプのレイヤ間予測が望ましいレートひずみトレードオフを生成しないことを意味する。一例として、いくつかの事例では、MV-HEVCコーディングされたビデオは、レイヤ間動き予測から利益を得ないか、最小限にしか利益を得ないことがある。本開示の技法によれば、ビデオエンコーダ20は、レイヤ間動き予測が無効化されるべきであるという決定と、1つのタイプのレイヤ間予測がビデオデコーダ30によって仮定され得るという決定とを行い得る。ビデオエンコーダがそのような決定を行った場合でも、既存のシグナリング技法に従って、ビデオエンコーダは、依然として、ビットストリーム中のレイヤの各組合せのための依存タイプを識別するシンタックス要素(たとえば、「`direct_dependency_type`」)をシグナリングする必要があり得、依存タイプシンタックス要素の各発生は少なくとも2ビットを要する。たとえば、レイヤ間動き予測が無効化される場合に「`direct_dependency_type`」シンタックス要素は有用でないことがあるが、既存のMV-HEVC技法は、シンタックス要素のこのシグナリングを回避するための機構を含まない。

20

【0079】

[0083]本開示は、参照レイヤに対する現在のレイヤのための依存タイプのシグナリングを変更するための技法をもたらず。MV-HEVCでは、たとえば、「`direct_dep_type_len_minus_2`」シンタックス要素は、依存タイプシンタックス要素がシグナリングされないことになることを示す0に、依存タイプの数が等しくなるのを可能にするように変更され得る。依存タイプシンタックス要素がシグナリングされないとき、ビデオデコーダ30は、参照レイヤに対する現在のレイヤの依存のタイプが、所定のタイプであると決定し、したがって、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号することに進み得る。上記で説明したように、所定のタイプは、ビデオコーデックにおいて定義され得るか、または第1のシンタックス要素(たとえば、`direct_dep_type_len_minus_2`)の一部としてシグナリングされ得る。

30

40

【0080】

[0084]MV-HEVCと併せて本開示の技法を実装する場合、たとえば、「`direct_dependency_type`」シンタックス要素のシグナリングが、`direct_dep_type_len_minus_2`の値に応じて回避され得る。`direct_dep_type_len_minus_2`が第1の値に等しくなる場合、ビデオデコーダ30は、たとえば、`direct_dependency_type`シンタックス要素のインスタンスを受信することなしに、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであることを決定し得る。復号プロセスのためにビデオデコーダ30が「`direct_dependency_type`」シンタックス要素の値を必要とする(すなわち、利用する)範囲で、ビデオデコーダ30は、`direct_dependency`

50

cy\_typeの値を所定のタイプに対応する値に設定し得る。direct\_dep\_type\_len\_minus2(すなわち、第1のシンタックス要素)が第2の値に等しくなる場合、ビデオデコーダ30は、たとえば、direct\_dependency\_typeシンタックス要素のインスタンスを受信することによって、参照レイヤに対する現在のレイヤのための依存タイプを決定し得る。

【0081】

[0085]本開示の技法によれば、第0の依存タイプは、所定のタイプのレイヤ間予測に等しくされ得る(またはそれに関連付けられ得る)。したがって、ビデオデコーダ30は、direct\_dep\_type\_len\_minus2の0の値に基づいて、VPSに関連するレイヤのための依存タイプが所定のタイプであると決定し得る。所定のタイプは、たとえば、(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちのいずれかであり得る。

10

【0082】

[0086]本開示の技法によれば、VPS拡張セマンティクスは、以下に示す方法で変更され得る。以下の説明では、下線は追加を示すものであり、

【0083】

【数1】

~~取り消し線~~

20

【0084】

は削除を示すものである。以下に示す変更をMV-HEVC WD5に関連して示す。以下の例では、direct\_dep\_type\_len値は、コード化ビデオシーケンス全体のために1つの固定予測タイプが使用されることを示し得る。したがって、いくつかの例では、direct\_dependency\_typeシンタックス要素が存在する必要がないことがある。direct\_dependency\_typeシンタックス要素が、シグナリングされず、所定の値であると導出されるように、上記で説明したように、他のシグナリングも実装され得る。

【0085】

【表3】

30

ビデオパラメータセット拡張シンタックス

vps_extension() {	記述子
...	
<b>direct_dep_type_len_minus2</b>	ue(v)
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
if(direct_dependency_flag[i][j] && <u>direct_dep_type_len</u> )	
<b>direct_dependency_type[i][j]</b>	u(v)
}	

40

表3

【0086】

[0087]上記の表3の例では、シンタックス要素「direct\_dep\_type\_len\_minus2」は、値が0に等しくなり得ることを示す「direct\_dep\_type\_len」に変更されている。ビデオデコーダ30が、VPS中で、「direct\_dep\_type\_len」の0の値を受信すると、ビデオデコーダ30は、VPSに関連するレイヤのすべての組合せに所定のタイプのレイヤ間予測を割り当てる。この

50

所定のタイプは、たとえば、コーデックにプログラムされ得るか、または別個にシグナリングされ得る。上記の表3の例ならびに以下の他の例では、ビデオデコーダ30が、ビットストリーム中で`direct_dependency_type`シンタックス要素を受信する場合、ビデオデコーダ30は、レイヤ組合せごとに`direct_dependency_type`のインスタンスを受信し、ここで、表3中の`[i]`および`[j]`は、様々なレイヤを表し、`[i][j]`は、レイヤ組合せに対応する。「`direct_dependency_type_len`」の値が0であるとき、`direct_dependency_type_len`が偽になるので、論理AND演算「`direct_dependency_flag[i][j] && direct_dependency_len`」が偽になるので、表3で上記に示したループがスキップされる（すなわち、ループの命令セットが実行されない）。したがって、ループがスキップされるとき、ビデオデコーダ30は、シンタックス要素「`direct_dependency_type[i][j]`」のいかなるインスタンスも受信しない。

10

## 【0087】

[0088]本開示の技法によれば、ビデオエンコーダ20は、ビデオデコーダ30が、`direct_dependency_len`シンタックス要素の1つのインスタンスを復号することによって、`direct_dependency_type`シンタックス要素の複数のインスタンスを復号することを回避する（それを控えるおよび/またはそれを必要としない）ことが可能であり得る方法でビデオデータを符号化し、それによって、シグナリングオーバーヘッドを低減し得る。しかしながら、本開示の技法では、レイヤ組合せごとに`direct_dependency_type`を別個にシグナリングすることがコーディング品質の改善をもたらすとき、ビデオエンコーダ20が、そのような別個のシグナリングを行うことが依然として可能である。

20

## 【0088】

[0089]シンタックス要素「`direct_dependency_len`」が、上記の表3に、「`ue(v)`」の記述子をもつ可変長シンタックス要素として示されているが、シンタックス要素「`direct_dependency_len`」は、代替的に、固定長コーディングを使用してシグナリングされ得る。たとえば、上記の表3中のシンタックス要素「`direct_dependency_len`」のための記述子は、`u(5)`、`u(6)`、または任意の他の固定長に変更され得る。

30

## 【0089】

[0090]さらに、以下の表4および表5の例で示すように、シンタックス要素「`direct_dependency_len`」が0に等しくないという条件は、VPS拡張シンタックス中のより前の位置に配置され得る。表4および表5の例は、意味的に代替実装形態であるが、機能的に等価である。表4と表5との両方の例では、シンタックス要素「`direct_dependency_len`」の偽値は、「`direct_dependency_type[i][j]`」のインスタンスを受信するために使用されるループを終了する。

## 【0090】



【表4】

vps_extension() {	記述子
...	
<b>direct_dep_type_len_minus2</b>	ue(v)
for(i = 1; i <= vps_max_layers_minus1 && direct_dep_type_len; i++)	
)	
for(j = 0; j < i; j++)	
if(direct_dependency_flag[i][j])	
<b>direct_dependency_type[i][j]</b>	u(v)
}	

表4

10

【0091】

【表5】

vps_extension() {	記述子
...	
<b>direct_dep_type_len_minus2</b>	ue(v)
<u>if(direct_dep_type_len)</u>	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
if(direct_dependency_flag[i][j])	
<b>direct_dependency_type[i][j]</b>	u(v)
}	

表5

20

【0092】

[0091]「direct\_dep\_type\_len\_minus2」シンタックス要素と「direct\_dependency\_type[i][j]」シンタックス要素との説明に変更は、次のように実装され得る。

30

【0093】

【数2】

~~direct\_dep\_type\_len\_minus2に2を加えたものは、direct\_dependency\_type[i][j]シンタックス要素のビット数を指定する。この仕様のこのバージョンに準拠するビットストリーム中で、direct\_dep\_type\_len\_minus2の値は、イコール0であるものとする。この仕様のこのバージョンに準拠するビットストリーム中で、direct\_dep\_type\_len\_minus2の値は、イコール0であるものとする。この仕様のこのバージョンでは、direct\_dep\_type\_len\_minus2の値が0に等しくなるものとするが、デコダは、direct\_dep\_type\_len\_minus2の他の値可能にするにあるものとするdirect\_dep\_type\_lenの値は、シンタックス中に現れるように両端値を含む0~32-30の範囲内にあるものとする。~~

40

【0094】

注釈 - MV - HEVC仕様では、direct\_dep\_type\_lenの値は、0に

50

等しくなる必要があり得る。SHEVC仕様では、direct\_dependency\_type\_layer\_idの値は、2に等しくなる必要があり得る。

direct\_dependency\_type[i][j]は、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤとlayer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤとの間の依存のタイプを示す。0に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用されることを示す。1に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間サンプル予測のために使用されるが、レイヤ間動き予測のために使用されないことを示す。2に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間動き予測のために使用されるが、レイヤ間サンプル予測のために使用されないことを示す。この仕様のこのバージョンでは、direct\_dependency\_type[i][j]の値は、両端値を含む0~2の範囲内にあるものとするが、デコーダは、両端値を含む $3 \sim 2^{32} - 2$ の範囲内のdirect\_dependency\_type[i][j]の値がシンタックス中に出現することを可能にするものとする。direct\_dependency\_flag[i][j]が1であり、direct\_dependency\_type[i][j]が存在しないとき、direct\_dependency\_type[i][j]の値は、1に等しくなると推論される。

【0095】

[0092]代替として、direct\_dependency\_type[i][j]のセマンティクスは、値0をレイヤ間サンプル予測のみの場合に対応させるように変更され得る。そのような例では、セマンティクスは以下のように変更され得る。

direct\_dependency\_type[i][j]は、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤとlayer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤとの間の依存のタイプを示す。

【0096】

【数3】

01

【0097】

に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間サンプル予測とレイヤ間動き予測との両方のために使用されることを示す。

【0098】

【数4】

01

【0099】

に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_

layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間サンプル予測のために使用されるが、レイヤ間動き予測のために使用されないことを示す。2に等しいdirect\_dependency\_type[i][j]は、layer\_id\_in\_nuh[j]に等しいnuh\_layer\_idをもつレイヤが、nuh\_layer\_idイコールlayer\_id\_in\_nuh[i]をもつレイヤのレイヤ間動き予測のために使用されるが、レイヤ間サンプル予測のために使用されないことを示す。この仕様のこのバージョンでは、direct\_dependency\_type[i][j]の値は、両端値を含む0~2の範囲内にあるものとするが、デコーダは、両端値を含む3~ $2^{32}-2$ の範囲内のdirect\_dependency\_type[i][j]の値がシンタックス中に出現することを可能にするものとする。direct\_dependency\_flag[i][j]が1に等しくなり、direct\_dependency\_type[i][j]が存在しないとき、direct\_dependency\_type[i][j]の値は、0に等しくなると推論される。

10

「1に等しいdirect\_dependency\_type[i][j]」から開始するセンテンスの位置が「0に等しいdirect\_dependency\_type[i][j]」から開始するセンテンスとスワップされ得ることに留意されたい。

【0100】

[0093]上記の例では、「direct\_dependency\_type[i][j]」の値が0または1に等しくなると推論することが、所定のタイプのレイヤ間予測の例を構成する。

20

【0101】

[0094]図4は、本開示で説明する技法を実施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、たとえば、上記で説明したセマンティクスを実装する符号化ビデオを生成するように構成され得る。いくつかの例では、ビデオエンコーダ20は、ビデオを後処理エンティティ27に出力するように構成され得る。後処理エンティティ27は、MANEまたはスライシング/編集デバイスなどの、ビデオエンコーダ20からの符号化ビデオデータを処理し得るビデオエンティティの一例を表すように意図されている。場合によっては、後処理エンティティ27はネットワークエンティティの一例であり得る。いくつかのビデオ符号化システムでは、後処理エンティティ27およびビデオエンコーダ20は別個のデバイスの部分であり得、他の事例では、後処理エンティティ27に関して説明する機能は、ビデオエンコーダ20を備える同じデバイスによって実行され得る。

30

【0102】

[0095]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し得る。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するフレーム内またはピクチャ内のビデオの時間的冗長性を低減または除去するために時間的予測に依拠する。イントラモード(Iモード)は、いくつかの空間ベースの圧縮モードのいずれかを指すことがある。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すことがある。

40

【0103】

[0096]図4の例では、ビデオエンコーダ20は、ビデオデータメモリ31と、区分ユニット35と、予測処理ユニット41と、フィルタユニット63と、復号ピクチャバッファ64と、加算器50と、変換処理ユニット52と、量子化ユニット54と、エントロピー符号化ユニット56とを含む。予測処理ユニット41は、動き推定ユニット42と、動き補償ユニット44と、イントラ予測処理ユニット46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、逆量子化ユニット58と、逆変換処理ユニット60と、加算器62とを含む。フィルタユニット63は、デブロッキングフィルタ、適応ループフィルタ(ALF)、およびサンプル適応オフセット(SAO)フィルタなど、1つ

50

または複数のループフィルタを表すように意図されている。図2では、フィルタユニット63はループ内フィルタであるものとして示されているが、他の構成では、フィルタユニット63はループ後フィルタとして実装され得る。

#### 【0104】

[0097]ビデオデータメモリ31は、ビデオエンコーダ20の構成要素によって符号化されるべきビデオデータを記憶し得る。ビデオデータメモリ31に記憶されたビデオデータは、たとえば、ビデオソース18から取得され得る。復号ピクチャバッファ64は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオエンコーダ20によってビデオデータを符号化するために使用するための、参照ビデオデータを記憶する参照ピクチャメモリであり得る。ビデオデータメモリ31および復号ピクチャバッファ64は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM(登録商標))、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ31および復号ピクチャバッファ64は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ31は、ビデオエンコーダ20の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

#### 【0105】

[0098]図4に示すように、ビデオエンコーダ20はビデオデータメモリ31においてビデオデータを受信し、区分ユニット35はデータをビデオブロックに区分する。この区分はまた、たとえば、LCUおよびCUの4分木構造に従って、ビデオブロック区分として、ウェルとして、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を含み得る。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41は、エラー結果(たとえば、コーディングレートおよびひずみレベル)に基づいて、現在のビデオブロックのために、複数のイントラコーディングモードのうちの一つ、または複数のインターコーディングモードのうちの一つなど、複数の可能なコーディングモードのうちの一つを選択し得る。予測処理ユニット41は、得られたイントラコード化ブロックまたはインターコード化ブロックを、残差ブロックを生成するために加算器50に与え、参照ピクチャとして使用するための符号化ブロックを再構成するために加算器62に与え得る。

#### 【0106】

[0099]予測処理ユニット41内のイントラ予測処理ユニット46は、コーディングす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一つまたは複数の隣接ブロックに対して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て空間的圧縮を行い得る。予測処理ユニット41内の動き推定ユニット42および動き補償ユニット44は、時間圧縮を行うために、一つまたは複数の参照ピクチャ中の一つまたは複数の予測ブロックに対して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。予測処理ユニット41内の動き推定ユニット42および動き補償ユニット44はまた、異なるレイヤの一つまたは複数の参照ピクチャ中の一つまたは複数の予測ブロックに対して現在のビデオブロックのレイヤ間予測コーディングを実行し得る。

#### 【0107】

[0100]動き推定ユニット42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。所定のパターンは、シーケンス中のビデオスライスをPスライス、BスライスまたはGPBスライスに指定し得る。動き推定ユニット42および動き補償ユニット44は、高度に統合され得るが、概念的な目的のために別々に示してある。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトル(たとえば、時間的動きベクトルおよび視差動きベクトル)を生成するプロセスである。時間的動きベクトルは、たとえば、

10

20

30

40

50

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。視差動きベクトルは、たとえば、異なるレイヤまたはビュー中の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。

【0108】

[0101]予測ブロックは、絶対差分和(SAD: sum of absolute difference)、2乗差分和(SSD: sum of square difference)、または他の差分メトリック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にぴったり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、復号ピクチャバッファ64に記憶された参照ピクチャのサブ整数ピクセル位置の値を計算し得る。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42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。

10

【0109】

[0102]動き推定ユニット42は、PUの位置を参照ピクチャの予測ブロックの位置と比較することによって、インターコード化スライス内のビデオブロックのPUについての動きベクトルを計算する。参照ピクチャは、第1の参照ピクチャリスト(リスト0)または第2の参照ピクチャリスト(リスト1)から選択され得、それらの参照ピクチャリストの各々は、復号ピクチャバッファ64に記憶された1つまたは複数の参照ピクチャを識別する。動き推定ユニット42は、計算された動きベクトルをエントロピー符号化ユニット56と動き補償ユニット44とに送る。ビデオデータのブロックをコーディングするための動きベクトルを決定することの一部として、動き推定ユニット42は、レイヤ間サンプル予測、レイヤ間動き予測、および/またはレイヤ間サンプル予測とレイヤ間動き予測との組合せを実装するモードを含む様々なコーディングモードをテストし得る。

20

【0110】

[0103]動き補償ユニット44によって実行される動き補償は、動き推定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取り込むこと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44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ーマ差分成分の両方を含み得る。加算器50は、この減算演算を実行する1つまたは複数の構成要素を表す。動き補償ユニット44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によって使用するための、ビデオブロックとビデオスライスとに関連付けられたシンタックス要素を生成し得る。

30

【0111】

[0104]本開示で説明する技法のいくつかは、部分的に動き推定ユニット42と動き補償ユニット44とによって実装され得る。たとえば、コード化ビデオシーケンスについて、動き推定ユニット42は、たとえば、レートひずみトレードオフに基づいて、コード化ビデオシーケンスのコーディングが(依存タイプシンタックス要素の値をシグナリングすることをも含むことになる)あるタイプのレイヤ間予測の使用から利益を得ないと決定し得る。その決定に基づいて、動き推定ユニット42は、上記で説明した技法に従ってシンタックス要素を生成し得る。より具体的には、コード化ビデオシーケンスのために、あるタイプのレイヤ間予測を使用すべきでないことと決定したことに応答して、動き推定ユニット42は、場合によってはビデオエンコーダ20の他の構成要素と組み合わせて、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を生成し得る。依存タイプシンタックス要素がシグナリングされないこ

40

50

とを示す第1のシンタックス要素の値に応答して、動き推定ユニットは、依存タイプシンタックス要素の値を生成することをスキップし得る。

【0112】

[0105] PUの動き情報を表すために必要とされるビット数を低減するために、動き推定ユニット42は、マージモードまたは高度動きベクトル予測 (AMVP: advanced motion vector prediction) プロセスに従ってPUの各々のための候補リストを生成し得る。

PUのための候補リスト中の各候補は、動き情報を示し得る。候補リスト中の候補のいくつかによって示される動き情報は、他のPUの動き情報に基づき得る。たとえば、HEVCマージモードについて、5つの空間候補ロケーションと1つの時間候補ロケーションとがあり得る。動き推定ユニット42はまた、レイヤの異なるビュー中の対応するブロックに基づいて、1つまたは複数の動きベクトル候補 (たとえば、3D HEVC候補またはレイヤ間候補) を決定し得る。いくつかの例では、動き推定ユニット42はまた、すでに決定された候補から部分的な動きベクトルを組み合わせるか、候補を変更するか、または単に候補として0動きベクトルを挿入することによって追加の候補を生成し得る。これらの追加の候補は、元の候補であると見なされず、仮想候補と呼ばれることがある。マージモードでは、動き推定ユニット42は、候補のうちの1つのためのインデックスを生成し、その候補についての動き情報がブロックを符号化および復号するために使用される。AMVPモードでは、動き推定ユニット42は、候補のうちの1つのためのインデックスを生成し、その候補についての動き情報が動き予測子として使用される。したがって、AMVPモードでは、動き推定ユニット42はまた、動きベクトル差分情報を生成し、動きベクトル差分に加えて動きベクトル予測子が、ブロックを符号化するために使用される動きベクトルを識別する。

【0113】

[0106] 候補リストを構築するために動き補償ユニット44が何のタイプの候補を使用するかは、現在のレイヤと参照レイヤとの間の依存タイプによって影響を及ぼされ得る。たとえば、レイヤ間サンプル予測が有効化される場合、動き推定ユニット42は、候補リスト中に、異なるレイヤ (すなわち、参照レイヤ) の参照ピクチャ中のブロックを指す動き情報を含め得る。別の例として、レイヤ間動き予測が有効化される時、動き推定ユニット42は、現在のブロックのための候補リスト中に異なるレイヤ中のブロックの動き情報を含め得る。動き情報は、たとえば、動きベクトル (たとえば、視差動きベクトルまたは時間的動きベクトル) を含み得、いくつかの例では、参照ピクチャインデックスなどの追加情報をも含み得る。レイヤ間サンプル予測とレイヤ間動き予測との組合せが有効化されるとき、ビデオコーダは、候補リストのための1つまたは複数の候補を決定するために、レイヤ間サンプル予測とレイヤ間動き予測との両方を実行し得る。

【0114】

[0107] 上記で説明したように、動き推定ユニット42がマージモードまたはAMVPのための候補リストを構築する方法は、有効化されるレイヤ間依存のタイプに依存し得る。たとえば、レイヤ間サンプル予測のみが有効化される場合、動き推定ユニット42は、候補を決定するために異なるレイヤのブロックを見ないように構成され得る。レイヤ間動き予測のみが有効化される場合、動き推定ユニット42は、候補を決定するために異なるレイヤのブロックを見るように構成され得るが、動き推定ユニット42は、時間的動きベクトルを利用する候補の動き情報のみを含めるように構成され得る。

【0115】

[0108] イントラ予測処理ユニット46は、上記で説明したように、動き推定ユニット42と動き補償ユニット44とによって実行されるインター予測の代替として、現在ブロックをイントラ予測し得る。特に、イントラ予測処理ユニット46は、現在ブロックを符号化するために使用する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、イントラ予測処理ユニット46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ブロックを符号化し得、イントラ予測処理ユニット46 (または、いくつかの例では、モード選択ユニット40) は、テストされたモードから使用するのに適切な

イントラ予測モードを選択し得る。たとえば、イントラ予測処理ユニット46は、様々なテストされたイントラ予測モード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、符号化ブロックと、符号化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。イントラ予測処理ユニット46は、符号化された様々なブロックのひずみおよびレートから比を算出し、どのイントラ予測モードがブロックの最良のレートひずみ値を示すかを決定し得る。

【0116】

[0109]いずれの場合も、ブロックのためのイントラ予測モードを選択した後に、イントラ予測処理ユニット46は、ブロックのため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56に与え得る。エントロピー符号化ユニット56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み得る。

【0117】

[0110]予測処理ユニット41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に、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52に適用され得る。変換処理ユニット52は、離散コサイン変換（DCT: discrete cosine transform）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52は、残差ビデオデータをピクセル領域から周波数領域などの変換領域に変換し得る。

【0118】

[0111]変換処理ユニット52は、得られた変換係数を量子化ユニット54に送り得る。量子化ユニット54は、ビットレートをさらに低減するために変換係数を量子化する。量子化プロセスは、係数の一部または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、修正され得る。いくつかの例では、量子化ユニット54は、次いで、量子化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56が、走査を実行し得る。

【0119】

[0112]量子化の後に、エントロピー符号化ユニット56は量子化された変換係数をエントロピー符号化する。たとえば、エントロピー符号化ユニット56は、コンテキスト適応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56によるエントロピー符号化の後に、符号化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56はまた、コーディングされている現在のビデオスライスのため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し得る。

【0120】

[0113]逆量子化ユニット58および逆変換処理ユニット60は、それぞれ逆量子化および逆変換を適用して、参照ピクチャの参照ブロックとして後で使用するためにピクセル領

10

20

30

40

50

域において残差ブロックを再構築する。動き補償ユニット44は、残差ブロックを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44はまた、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て、動き推定において使用するサブ整数ピクセル値を計算し得る。加算器62は、復号ピクチャバッファ64に記憶するための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44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42および動き補償ユニット44によって参照ブロックとして使用され得る。

【0121】

[0114]このようにして、図4のビデオエンコーダ20は、本開示の技法による、マルチレイヤビデオデータを符号化するように構成されたビデオエンコーダの一例を表す。複数のレイヤを備えるコード化されたビデオデータの場合、ビデオエンコーダ20は、複数のレイヤの符号化を、単一の依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定し得る。依存タイプは、参照レイヤに対する現在のレイヤの依存のタイプを識別する。複数のレイヤの符号化を、単一の依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかの決定に基づいて、ビデオエンコーダ20は、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を決定し得る。複数のレイヤの符号化を単一の依存タイプを使用して行うべきと決定したことに応答して、ビデオエンコーダ20は、依存タイプシンタックス要素の符号化をスキップし得る。第1のシンタックス要素は、パラメータセット中に含まれ得、ビデオエンコーダ20は、パラメータセット中に依存タイプシンタックス要素のインスタンスを含めないことによって、依存タイプシンタックス要素の符号化をスキップし得る。複数のレイヤを符号化を複数の依存タイプを使用して行うべきと決定したことに応答して、ビデオエンコーダ20は、依存タイプシンタックス要素を符号化し得る。

【0122】

[0115]同じ依存タイプは、たとえば、(1)レイヤ間サンプル予測のみ、(2)レイヤ間動き予測のみ、または(3)レイヤ間サンプル予測とレイヤ間動き予測との両方のうちのいずれかであり得る。このコンテキストでは、同じ依存タイプを有することは、パラメータセットに関連するすべてのレイヤが同じ依存タイプを有することを意味する。したがって、ビデオエンコーダ20が、複数のレイヤの符号化を、同じ依存タイプを用いて行うべきと決定する場合、レイヤの組合せのすべてが、たとえば、上記のタイプ(1)に等しい依存タイプを有することになるか、またはレイヤの組合せのすべてが、上記のタイプ(2)に等しい依存タイプを有することになるか、またはレイヤの組合せのすべてが、上記のタイプ(3)に等しい依存タイプを有することになる。ビデオエンコーダ20が、複数のレイヤの符号化を複数の依存タイプを用いて行うべきと決定する場合、レイヤのいくつかの組合せが、タイプ(1)に等しい依存タイプを有し得、一方、レイヤの他の組合せが、タイプ(2)に等しい依存タイプを有し得、レイヤのまた他の組合せが、タイプ(3)に等しい依存タイプを有する。

【0123】

[0116]図5は、本開示で説明する技法を実施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。ビデオデコーダ30は、たとえば、上記で説明したセマンティクスを使用して符号化されたビデオデータを復号するように構成され得る。図5の例では、ビデオデコーダ30は、ビデオデータメモリ79と、エン트로ピー復号ユニット80と、予測処理ユニット81と、逆量子化ユニット86と、逆変換処理ユニット88と、加算器90と、フィルタユニット91と、コード化ピクチャバッファ92とを含む。予測処理ユニット81は、動き補償ユニット82と、イントラ予測処理ユニット84とを含む。ビデオデコーダ30は、いくつかの例では、図4から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。

10

20

30

40

50



## 【 0 1 2 4 】

[0117]ビデオデータメモリ79は、ビデオデコーダ30の構成要素によって復号されるべき、符号化ビデオビットストリームなどのビデオデータを記憶し得る。ビデオデータメモリ79に記憶されたビデオデータは、たとえば、コンピュータ可読媒体16から、たとえば、カメラなどのローカルビデオソースから、ビデオデータのワイヤードもしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって取得され得る。ビデオデータメモリ79は、符号化されたビデオビットストリームからの符号化されたビデオデータを記憶するコーディング済みピクチャバッファ(CPB)を形成し得る。復号ピクチャバッファ162は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオデコーダ30によってビデオデータを復号する際に使用するための、参照ビデオデータを記憶する参照ピクチャメモリであり得る。ビデオデータメモリ79および復号ピクチャバッファ162は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM)、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ79および復号ピクチャバッファ162は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ79は、ビデオデコーダ30の他の構成要素とともにオンチップであるか、またはそれらの構成要素に対してオフチップであり得る。

10

## 【 0 1 2 5 】

[0118]復号プロセスの間、ビデオデコーダ30は、符号化ビデオスライスのビデオブロックおよび関連するシンタックス要素を表す符号化ビデオビットストリームをビデオエンコーダ20から受信する。ビデオデコーダ30は、ネットワークエンティティ29から符号化ビデオビットストリームを受信し得る。ネットワークエンティティ29は、たとえば、上記で説明した技法のうちの1つまたは複数を実装するように構成されたサーバ、MANE、ビデオエディタ/スプライサ、または他のそのようなデバイスであり得る。ネットワークエンティティ29はビデオエンコーダ20などのビデオエンコーダを含むことも、含まないこともある。本開示で説明する技法のいくつかは、ネットワークエンティティ29が符号化ビデオビットストリームをビデオデコーダ30に送信するのに先立って、ネットワークエンティティ29によって実施され得る。いくつかのビデオ復号システムでは、ネットワークエンティティ29およびビデオデコーダ30は別個のデバイスの一部であり得るが、他の事例では、ネットワークエンティティ29に関して説明する機能は、ビデオデコーダ30を備える同じデバイスによって実行され得る。

20

30

## 【 0 1 2 6 】

[0119]ビデオデコーダ30のエントロピー復号ユニット80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、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80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81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。

## 【 0 1 2 7 】

[0120]ビデオスライスがイントラコード化(I)スライスとしてコーディングされたとき、予測処理ユニット81のイントラ予測処理ユニット84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ド化(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たとき、予測処理ユニット81の動き補償ユニット82は、エントロピー復号ユニット80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つの中の参照ピクチャのうちの1つから生成され得る。ビデオデコーダ30は、コード化ピクチャバッファ92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を使用して、参照フレームリスト、すなわち、リスト

40

50

0 およびリスト 1 を構成し得る。

【 0 1 2 8 】

[0121]動き補償ユニット 8 2 は、動きベクトルと他のシンタックス要素とをパースすることによって現在のビデオスライスのビデオブロックのための予測情報を決定し、その予測情報を使用して、復号されている現在のビデオブロックの予測ブロックを生成する。たとえば、動き補償ユニット 8 2 は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード（たとえば、イントラ予測またはインター予測）と、インター予測スライスタイプ（たとえば、B スライス、P スライス、または G P B 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ビデオブロックのための動きベクトルと、スライスの各インターコード化ビデオブロックのためのインター予測ステータスと、現在のビデオスライス中のビデオブロックを復号するための他の情報とを決定するために、受信されたシンタックス要素のいくつかを使用する。

10

【 0 1 2 9 】

[0122]動き補償ユニット 8 2 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 8 2 は、参照ブロックのサブ整数ピクセルのための補間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 2 0 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 8 2 は、受信されたシンタックス要素からビデオエンコーダ 2 0 によって使用された補間フィルタを決定し、その補間フィルタを使用して予測ブロックを生成し得る。

20

【 0 1 3 0 】

[0123]動き補償ユニット 8 2 は、レイヤ間予測を含む複数のインターモードを使用して、受信されたビデオデータを復号し得る。たとえば、レイヤ間サンプル予測を使用してコーディングされた現在のレイヤのビデオブロックについて、動き補償ユニット 8 2 は、視差動きベクトルを決定するために受信されたシンタックス要素を使用し、参照レイヤ中で、ビデオブロックのための予測ブロックを識別するために視差動きベクトルを使用し得る。レイヤ間動き予測を使用してコーディングされた現在のレイヤのビデオブロックについて、動き補償ユニット 8 2 は、参照レイヤ中の参照ブロックの位置を特定するために、受信されたシンタックス要素を使用し、参照ブロックから、ビデオブロックについての動き情報を決定し得る。動き情報を使用して、動き補償ユニット 8 2 は、現在のレイヤから参照ピクチャ中のビデオブロックのための予測ブロックを識別し得る。

30

【 0 1 3 1 】

[0124]インターコーディングの一部として、動き補償ユニット 8 2 は、動き推定ユニット 4 2 と同じ方法でマージモードおよび A M V P モードを実装し得る。したがって、動き補償ユニット 8 2 は、有効化されたレイヤ間依存のタイプに基づいて候補の任意の制限を適用することを含めて、動き推定ユニット 4 2 と同じ候補リストを構築するように構成され得る。

【 0 1 3 2 】

[0125]逆量子化ユニット 8 6 は、ビットストリーム中で与えられ、エン트로ピー復号ユニット 8 0 によって復号された量子化変換係数を逆量子化 (inverse quantize)、すなわち逆量子化 (de-quantize) する。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するための、ビデオスライス中の各ビデオブロックについてビデオエンコーダ 2 0 によって計算される量子化パラメータの使用を含み得る。逆変換処理ユニット 8 8 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 D C T、逆整数変換、または概念的に同様の逆変換プロセスを変換係数に適用する。

40

【 0 1 3 3 】

[0126]動き補償ユニット 8 2 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 3 0 は、逆変換処理ユニット 8 8 からの残差ブロックを動き補償ユニット 8 2 によって生成された対応

50

する予測ブロックと加算することによって、復号ビデオブロックを形成する。加算器90は、この加算演算を実行する1つまたは複数の構成要素を表す。所望される場合、ピクセル遷移を平滑化するか、またはさもなければ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)ループフィルタも使用され得る。フィルタユニット91は、デブロッキングフィルタ、適応ループフィルタ(ALF)、およびサンプル適応オフセット(SAO)フィルタなど、1つまたは複数のループフィルタを表すように意図されている。図5では、フィルタユニット91はループ内フィルタであるものとして示されているが、他の構成では、フィルタユニット91はループ後フィルタとして実装され得る。所与のフレームまたはピクチャ中の復号ビデオブロックは、次いで、コード化ピクチャバッファ92に記憶され、コード化ピクチャバッファ92は後続の動き補償のために使用される参照ピクチャを記憶する。コード化ピクチャバッファ92はまた、図1のディスプレイデバイス32などのディスプレイデバイス上での後の表示のために、復号ビデオを記憶する。

#### 【0134】

[0127]図5のビデオデコーダ30は、本開示の技法による、マルチレイヤビデオデータを復号するように構成されたビデオデコーダの一例を表す。マルチレイヤビデオデータを復号することの一部として、ビデオデコーダ30は、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信するように構成され得る。依存タイプシンタックス要素は、1つまたは複数の参照レイヤに対する現在のレイヤの依存のタイプを識別する。依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に回答して、ビデオデコーダ30は、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定し、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号し得る。

#### 【0135】

[0128]参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定するために、ビデオデコーダ30は、依存タイプシンタックス要素の値を推論するように構成され得る。参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定するために、ビデオデコーダ30は、現在のレイヤのための依存タイプシンタックス要素のインスタンスを復号することなしにレイヤ間依存タイプを決定し得る。所定のタイプは、たとえば、(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つであり得る。所定のタイプがレイヤ間サンプル予測である事例では、ビデオデコーダ30は、参照レイヤ中で、現在のレイヤのブロックのための予測ブロックを識別することによってレイヤ間予測を使用して現在のレイヤのブロックを復号し得る。

#### 【0136】

[0129]ビデオデコーダ30は、パラメータセットの一部として第1のシンタックス要素を受信し得、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に回答して、ビデオデコーダ30は、パラメータセットに関連するすべての他のレイヤに対する現在のレイヤの依存のタイプが、所定のタイプであると決定し得る。さらに、依存タイプシンタックス要素がシグナリングされることを示す第1のシンタックス要素の値に回答して、ビデオデコーダ30は、依存タイプシンタックス要素の値を受信し、依存タイプシンタックス要素の値に基づいて参照レイヤに対する現在のレイヤの依存のタイプを決定し、決定されたタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号し得る。依存タイプシンタックス要素の値は、参照レイヤに対する現在のレイヤの依存のタイプが(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つであることを示し得る。

#### 【0137】

[0130]図6は、ネットワーク100の一部を形成するデバイスの例示的なセットを示すブロック図である。この例では、ネットワーク100は、ルーティングデバイス104A

10

20

30

40

50

、 104B (ルーティングデバイス104) とトランスコーディングデバイス106 とを含む。ルーティングデバイス104 およびトランスコーディングデバイス106 は、ネットワーク100の一部を形成し得る少数のデバイスを表すものである。スイッチ、ハブ、ゲートウェイ、ファイアウォール、ブリッジ、および他のそのようなデバイスなどの他のネットワークデバイスも、ネットワーク100内に含まれ得る。その上、サーバデバイス102とクライアントデバイス108との間にネットワーク経路に沿って追加のネットワークデバイスが提供され得る。いくつかの例では、サーバデバイス102はソースデバイス12(図1)に対応し得る一方、クライアントデバイス108は宛先デバイス14(図1)に対応し得る。

【0138】

[0131]概して、ルーティングデバイス104は、ネットワーク100を介してネットワークデータを交換するための1つまたは複数のルーティングプロトコルを実装する。いくつかの例では、ルーティングデバイス104は、プロキシまたはキャッシュ動作を実行するように構成され得る。したがって、いくつかの例では、ルーティングデバイス104はプロキシデバイスと呼ばれることがある。概して、ルーティングデバイス104は、ネットワーク100を介したルートを発見するためにルーティングプロトコルを実行する。そのようなルーティングプロトコルを実行することによって、ルーティングデバイス104Bは、それ自体からルーティングデバイス104Aを介してサーバデバイス102へ至るネットワークルートを発見し得る。

【0139】

[0132]本開示の技法は、ルーティングデバイス104およびトランスコーディングデバイス106のようなネットワークデバイスによって実施され得るが、クライアントデバイス108によっても実施され得る。このように、ルーティングデバイス104、トランスコーディングデバイス106、およびクライアントデバイス108は、本開示の技法を実行するように構成されたデバイスの例を表す。その上、図1のデバイス、ならびに図4に示すエンコーダ20および図5に示すデコーダ30も、本開示の技法を実行するように構成され得る例示的なデバイスである。

【0140】

[0133]たとえば、サーバデバイス102は、時間レイヤ切替えポイントなどの、ランダムアクセスポイントもしくはストリーム適応ポイント、または他のストリーム適応ポイントの後にくる1つまたは複数のピクチャを符号化するためのビデオエンコーダを含み得る。たとえば、このポイントは、ビットレート、フレームレート(すなわち、時間的レイヤ切替えポイント)、または空間解像度の適応のための切替えポイントであり得る。同様に、クライアントデバイス108は、時間レイヤ切替えポイントなどの、ランダムアクセスポイントまたはストリーム適応ポイントの後にくる1つまたは複数のピクチャを復号し得る。

【0141】

[0134]図7は、本開示の技法による、マルチレイヤビデオデータを符号化する例示的な方法を示すフローチャートである。図7の技法について、ビデオエンコーダ20に関して説明するが、図7の技法は、ビデオエンコーダのいかなる特定のタイプにも限定されない。図7の例では、ビデオエンコーダ20は、エンコーダマルチレイヤビデオデータに対して構成される。複数のレイヤを含むコード化されたビデオデータの場合、ビデオエンコーダ20は、複数のレイヤを符号化するを、マルチレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定する(170)。依存タイプは、参照レイヤに対する現在のレイヤの依存のタイプを識別する。複数のレイヤの符号化を、同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、ビデオエンコーダ20は、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を決定する(172)。第1のシンタックス要素は、たとえば、パラメータセット中に含まれ得、ビデオエンコーダ20は、パラメータセット中に依

10

20

30

40

50

存タイプシンタックス要素のインスタンスを含めないことによって、依存タイプシンタックス要素の符号化をスキップし得る。複数のレイヤの符号化を同じ依存タイプを使用し行うべきと決定したことに応答して（174、はい）、ビデオエンコーダ20は、依存タイプシンタックス要素の符号化をスキップする（176）。複数のレイヤの符号化を複数の依存タイプを使用し行うべきと決定したことに応答して（174、いいえ）、ビデオエンコーダ20は、依存タイプシンタックス要素を符号化する（178）。同じ依存タイプは、たとえば、（1）レイヤ間サンプル予測、（2）レイヤ間動き予測、または（3）レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つであり得る。

【0142】

[0135] 図8は、本開示の技法による、マルチレイヤビデオデータを復号する例示的な方法を示すフローチャートである。図8の技法について、ビデオデコーダ30に関して説明するが、図8の技法は、ビデオデコーダのいかなる特定のタイプにも限定されない。図7の技法によれば、ビデオデコーダ30は、マルチレイヤビデオデータを復号し得る。ビデオデコーダ30は、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信し得る（180）。依存タイプシンタックス要素は、参照レイヤに対する現在のレイヤの依存のタイプを識別する。依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して（182、はい）、ビデオデコーダ30は、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定し（184）、所定のタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号し得る（186）。依存タイプシンタックス要素がシグナリングされることを示す第1のシンタックス要素の値に応答して（182、いいえ）、ビデオデコーダ30は、依存タイプシンタックス要素の値を受信し（183）、依存タイプシンタックス要素の値に基づいて参照レイヤに対する現在のレイヤの依存のタイプを決定し（185）、決定されたタイプに準拠するレイヤ間予測を使用して現在のレイヤのブロックを復号し得る（187）。依存タイプシンタックス要素の値は、参照レイヤに対する現在のレイヤの依存のタイプが（1）レイヤ間サンプル予測、（2）レイヤ間動き予測、または（3）レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つであることを示し得る。

【0143】

[0136] ビデオデコーダ30は、たとえば、依存タイプシンタックス要素の値を推論することによって、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定し得る。代替または追加として、ビデオデコーダ30は、現在のレイヤのための依存タイプシンタックス要素のインスタンスを復号することなしにレイヤ間依存タイプを決定することによって、参照レイヤに対する現在のレイヤの依存のタイプが所定のタイプであると決定し得る。所定のタイプは、たとえば、（1）レイヤ間サンプル予測、（2）レイヤ間動き予測、または（3）レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つであり得る。いくつかの例では、所定のタイプは、レイヤ間サンプル予測であり得、ビデオデコーダ30は、参照レイヤ中で、現在のレイヤのブロックのための予測ブロックを識別することによってレイヤ間予測を使用して現在のレイヤのブロックを復号し得る。

【0144】

[0137] ビデオデコーダ30は、たとえば、パラメータセットの一部として第1のシンタックス要素を受信し得、依存タイプシンタックス要素がシグナリングされないことを示す第1のシンタックス要素の値に応答して、ビデオデコーダ30は、パラメータセットに関連するすべての他のレイヤに対する現在のレイヤの依存のタイプが、所定のタイプであると決定し得る。

【0145】

[0138] 1つまたは複数の例では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、1つもしくは複数の命令もしくはコードとしてコンピュータ可読媒体上に記

10

20

30

40

50

憶され、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得る。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含み得る。このようにして、コンピュータ可読媒体は、概して、(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し得る。データ記憶媒体は、本開示で説明する技法の実装のための命令、コードおよび/またはデータ構造を取り出すために、1つまたは複数のコンピュータあるいは1つまたは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であり得る。コンピュータプログラム製品はコンピュータ可読媒体を含み得る。

10

## 【0146】

[0139] 限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM(登録商標)、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージ、もしくは他の磁気記憶デバイス、フラッシュメモリ、または命令もしくはデータ構造の形態の所望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る任意の他の媒体を備えることができる。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、光ファイバーケーブル、ツイストペア、デジタル加入者線(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、光ファイバーケーブル、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、非一時的な有形記憶媒体を対象とすることを理解されたい。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)、およびBlu-rayディスク(disc)を含み、ディスク(disk)は、通常、データを磁氣的に再生し、ディスク(disc)は、データをレーザーで光学的に再生する。上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。

20

## 【0147】

[0140] 命令は、1つまたは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、あるいは他の等価な集積回路またはディスクリット論理回路など、1つまたは複数のプロセッサによって実施され得る。したがって、本明細書で使用する「プロセッサ」という用語は、前述の構造、または、本明細書で説明する技法の実装に好適な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モジュールおよび/またはソフトウェアモジュール内に与えられるか、あるいは複合コーデックに組み込まれ得る。また、本技法は、1つまたは複数の回路または論理要素で十分に実装され得る。

30

## 【0148】

[0141] 本開示の技法は、ワイヤレスハンドセット、集積回路(IC)またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置で実装され得る。本開示では、開示す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、またはユニットについて説明したが、それらの構成要素、モジュール、またはユニットを、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。そうではなく、上記で説明したように、様々なユニットは、コーデックハードウェアユニット中で組み合わせられるか、または上記で説明した1つまたは複数のプロセッサを含む、好適なソフトウェアおよび/またはファームウェアとともに相互動作可能なハードウェアユニットの集合によって提供され得る。

40

## 【0149】

50

[0142] 様々な例について説明した。これらおよび他の例は、以下の特許請求の範囲に含まれる。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

[ C 1 ]

マルチレイヤビデオデータを復号する方法であって、

現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、前記依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第1のシンタックス要素の前記値に  
10 応答して、

前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが所定のタイプであると決定することと、

前記所定のタイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号することと

を備える方法。

[ C 2 ]

前記第1のシンタックス要素の前記値が前記所定のタイプを識別する、C1に記載の方法。

[ C 3 ]

前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが前記所定のタイプであると決定することが、前記現在のレイヤのための前記依存タイプシンタックス要素のインスタンスを復号することなしに前記レイヤ間依存タイプを決定することを備える、C1に記載の方法。

[ C 4 ]

前記所定のタイプが、(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つを備える、C1に記載の方法。

[ C 5 ]

前記所定のタイプがレイヤ間サンプル予測を備え、ここにおいて、レイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することが、前記参照レイヤ中で、前記現在のレイヤの前記ブロックのための予測ブロックを識別することを備える、C1に記載の方法。

[ C 6 ]

パラメータセットの一部として前記第1のシンタックス要素を受信することと、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第1のシンタックス要素の前記値に  
30 応答して、前記パラメータセットに関連するすべての他のレイヤに対する前記現在のレイヤの依存の前記タイプが、前記所定のタイプであると決定すること

と  
をさらに備える、C1に記載の方法。

[ C 7 ]

前記パラメータセットがビデオパラメータセットを備える、C6に記載の方法。

[ C 8 ]

前記依存タイプシンタックス要素がシグナリングされることを示す前記第1のシンタックス要素の前記値に  
40 応答して、

前記依存タイプシンタックス要素の値を受信することと、

前記依存タイプシンタックス要素の前記値に基づいて前記参照レイヤに対する前記現在のレイヤの依存の前記タイプを決定することと、

前記決定されたタイプに準拠するレイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することと

をさらに備える、C1に記載の方法。

10

20

30

40

50

[ C 9 ]

前記依存タイプシンタックス要素の前記値が、前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つを備えることを示す、C 8に記載の方法。

[ C 10 ]

マルチレイヤビデオデータを符号化する方法であって、  
複数のレイヤを備えるコード化されたビデオデータについて、前記複数のレイヤを符号化を、前記複数のレイヤのすべてのレイヤに対して同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することと、ここにおいて、依存タイプが、参照レイヤに対する現在のレイヤの依存のタイプを識別する、

前記複数のレイヤの符号化を、前記同じ依存タイプを使用して行うべきか、それとも複数の依存タイプを使用して行うべきかを決定することに基づいて、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を決定することと

を備える方法。

[ C 11 ]

前記複数のレイヤの符号化を前記同じ依存タイプを使用して行うべきと決定したことに応答して、前記依存タイプシンタックス要素の符号化をスキップすること

をさらに備える、C 10に記載の方法。

[ C 12 ]

前記第1のシンタックス要素が、パラメータセット中に含まれ、ここにおいて、前記依存タイプシンタックス要素の符号化をスキップすることが、前記パラメータセット中に前記依存タイプシンタックス要素のインスタスを含めないことを備える、C 10に記載の方法。

[ C 13 ]

前記複数のレイヤの符号化を前記複数の依存タイプを使用して行うべきと決定したことに応答して、前記依存タイプシンタックス要素を符号化すること

をさらに備える、方法クレーム10。

[ C 14 ]

前記同じ依存タイプが、(1)レイヤ間サンプル予測、(2)レイヤ間動き予測、または(3)レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つを備える、C 10に記載の方法。

[ C 15 ]

ビデオコーディングを実行する装置であって、

ビデオデータを記憶するメモリと、

1つまたは複数のプロセッサを備えたビデオコーダとを備え、前記1つまたは複数のプロセッサが、

現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、前記依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第1のシンタックス要素の前記値に応答して、

前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが所定のタイプであると決定することと、

前記所定のタイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号することと

を行うように構成された、装置。

[ C 16 ]

前記第1のシンタックス要素の前記値が前記所定のタイプを識別する、C 15に記載の

10

20

30

40

50



装置。

[ C 1 7 ]

前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが前記所定のタイプであると決定するために、前記1つまたは複数のプロセッサが、前記現在のレイヤのための前記依存タイプシンタックス要素のインスタンスを復号することなしに前記レイヤ間依存タイプを決定することをを行うように構成された、C 1 5に記載の装置。

[ C 1 8 ]

前記所定のタイプが、( 1 )レイヤ間サンプル予測、( 2 )レイヤ間動き予測、または( 3 )レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つを備える、C 1 5に記載の装置。

10

[ C 1 9 ]

前記所定のタイプがレイヤ間サンプル予測を備え、ここにおいて、レイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することが、前記参照レイヤ中で、前記現在のレイヤの前記ブロックのための予測ブロックを識別することを備える、C 1 5に記載の装置。

[ C 2 0 ]

前記1つまたは複数のプロセッサが、パラメータセットの一部として前記第1のシンタックス要素を受信することと、前記依存タイプシンタックス要素がシグナリングされないことを示す前記第1のシンタックス要素の前記値に回答して、前記パラメータセットに関連するすべての他のレイヤに対する前記現在のレイヤの依存の前記タイプが、前記所定のタイプであると決定することとを行うようにさらに構成された、C 1 5に記載の装置。

20

[ C 2 1 ]

前記1つまたは複数のプロセッサが、前記依存タイプシンタックス要素がシグナリングされることを示す前記第1のシンタックス要素の前記値に回答して、前記依存タイプシンタックス要素の値を受信することと、前記依存タイプシンタックス要素の前記値に基づいて前記参照レイヤに対する前記現在のレイヤの依存の前記タイプを決定することと、前記決定されたタイプに準拠するレイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することとを行うようにさらに構成された、C 1 5に記載の装置。

30

[ C 2 2 ]

前記依存タイプシンタックス要素の前記値が、前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが( 1 )レイヤ間サンプル予測、( 2 )レイヤ間動き予測、または( 3 )レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの1つを備えることを示す、C 2 1に記載の装置。

[ C 2 3 ]

集積回路、マイクロプロセッサ、またはビデオコードを含むワイヤレス通信デバイス  
のうちの少なくとも1つを備える、C 1 5に記載の装置。

40

[ C 2 4 ]

1つまたは複数のプロセッサによって実行されたとき、前記1つまたは複数のプロセッサに、現在のレイヤのための依存タイプシンタックス要素がシグナリングされるかどうかを表す第1のシンタックス要素の値を受信することと、ここにおいて、前記依存タイプシンタックス要素が、参照レイヤに対する現在のレイヤの依存のタイプを識別する、前記依存タイプシンタックス要素がシグナリングされないことを示す前記第1のシンタ

50

ックス要素の前記値に応答して、

前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが所定のタイプであると決定することと、

前記所定のタイプに準拠するレイヤ間予測を使用して前記現在のレイヤのブロックを復号することと

を行わせる命令を記憶したコンピュータ可読記憶媒体。

[ C 2 5 ]

前記第 1 のシンタックス要素の前記値が前記所定のタイプを識別する、C 2 4 に記載のコンピュータ可読記憶媒体。

[ C 2 6 ]

前記 1 つまたは複数のプロセッサが、前記現在のレイヤのための前記依存タイプシンタックス要素のインスタンスを復号することなしに前記レイヤ間依存タイプを決定することによって、前記参照レイヤに対する前記現在のレイヤの依存の前記タイプが前記所定のタイプであると決定する、C 2 4 に記載のコンピュータ可読記憶媒体。

[ C 2 7 ]

前記所定のタイプが、( 1 ) レイヤ間サンプル予測、( 2 ) レイヤ間動き予測、または ( 3 ) レイヤ間サンプル予測とレイヤ間動き予測との組合せのうちの 1 つを備える、C 2 4 に記載のコンピュータ可読記憶媒体。

[ C 2 8 ]

前記所定のタイプが、レイヤ間サンプル予測を備え、ここにおいて、前記コンピュータ可読記憶媒体が、実行されたとき、前記 1 つまたは複数のプロセッサに、

前記参照レイヤ中で、前記現在のレイヤの前記ブロックのための予測ブロックを識別することを備えるレイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することを行わせる命令をさらに記憶した、C 2 4 に記載のコンピュータ可読記憶媒体。

[ C 2 9 ]

実行されたとき、前記 1 つまたは複数のプロセッサに、

パラメータセットの一部として前記第 1 のシンタックス要素を受信することと、

前記依存タイプシンタックス要素がシグナリングされないことを示す前記第 1 のシンタックス要素の前記値に応答して、前記パラメータセットに関連するすべての他のレイヤに対する前記現在のレイヤの依存の前記タイプが、前記所定のタイプであると決定すること

を行わせる命令をさらに記憶した、C 2 4 に記載のコンピュータ可読記憶媒体。

[ C 3 0 ]

実行されたとき、前記 1 つまたは複数のプロセッサに、

前記依存タイプシンタックス要素がシグナリングされることを示す前記第 1 のシンタックス要素の前記値に応答して、

前記依存タイプシンタックス要素の値を受信することと、

前記依存タイプシンタックス要素の前記値に基づいて前記参照レイヤに対する前記現在のレイヤの依存の前記タイプを決定することと、

前記決定されたタイプに準拠するレイヤ間予測を使用して前記現在のレイヤの前記ブロックを復号することと

を行わせる命令をさらに記憶した、C 2 4 に記載のコンピュータ可読記憶媒体。

10

20

30

40

【図 1】

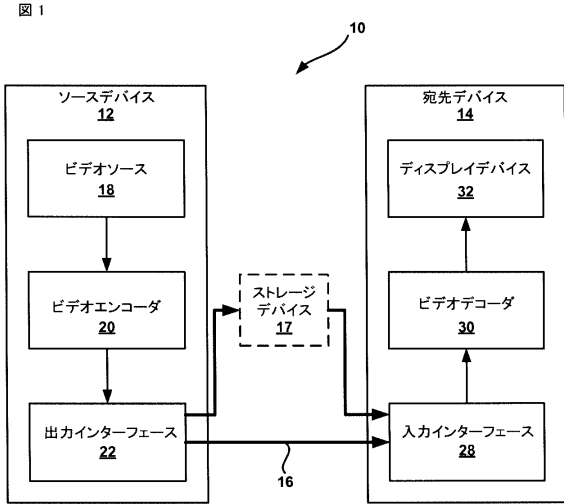


FIG. 1

【図 2】

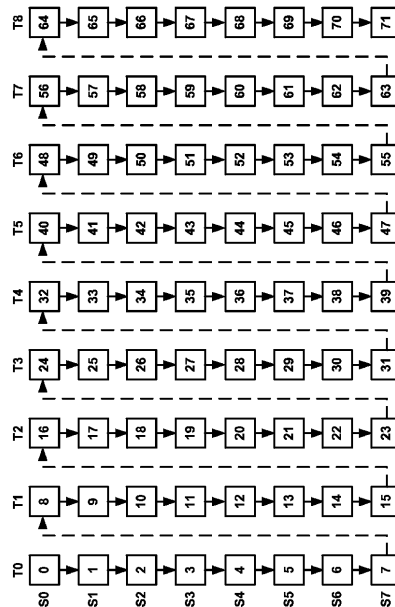


FIG. 2

【図 3】

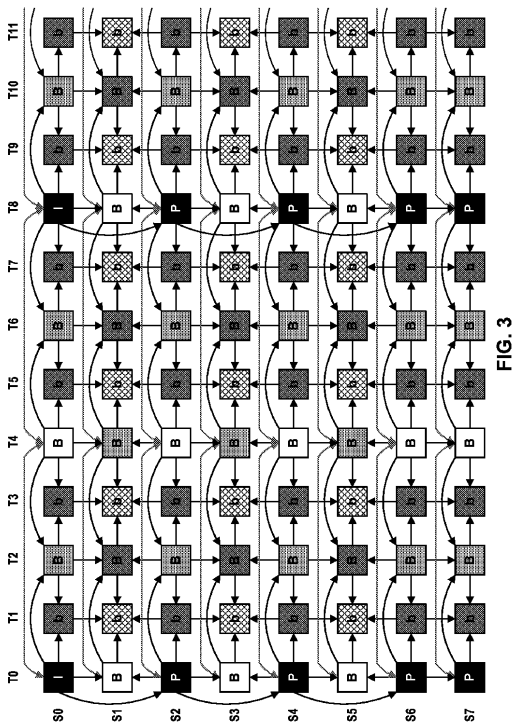


FIG. 3

【図 4】

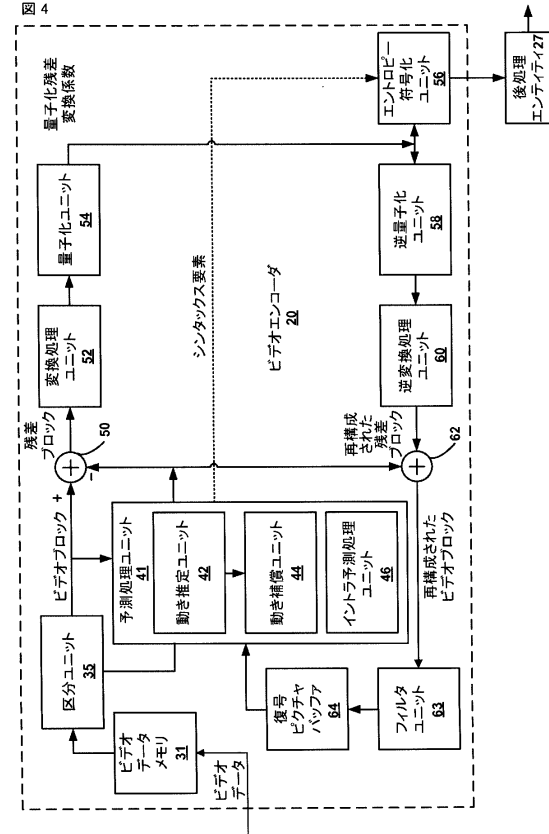


FIG. 4

【図5】

図5

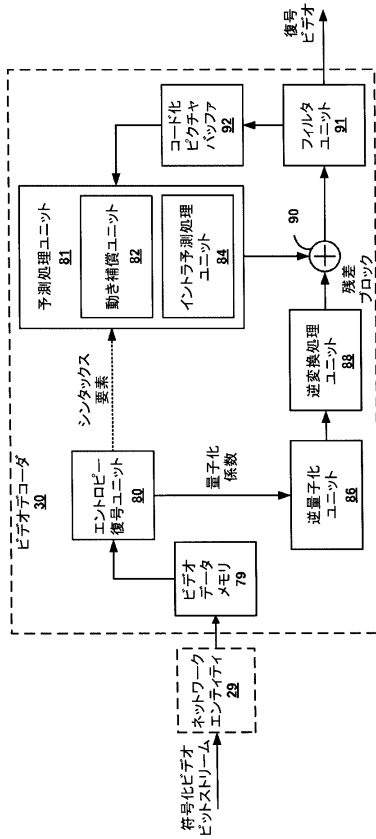


FIG. 5

【図6】

図6

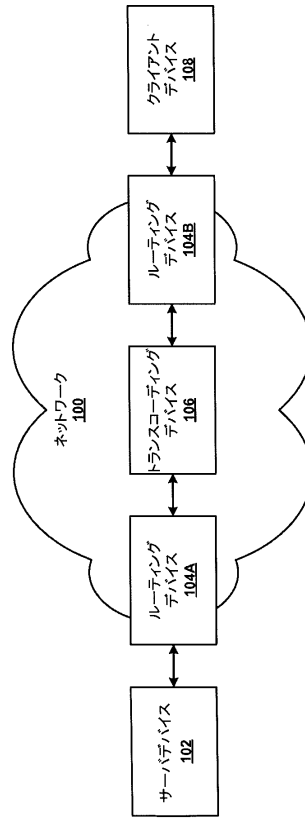


FIG. 6

【図7】

図7

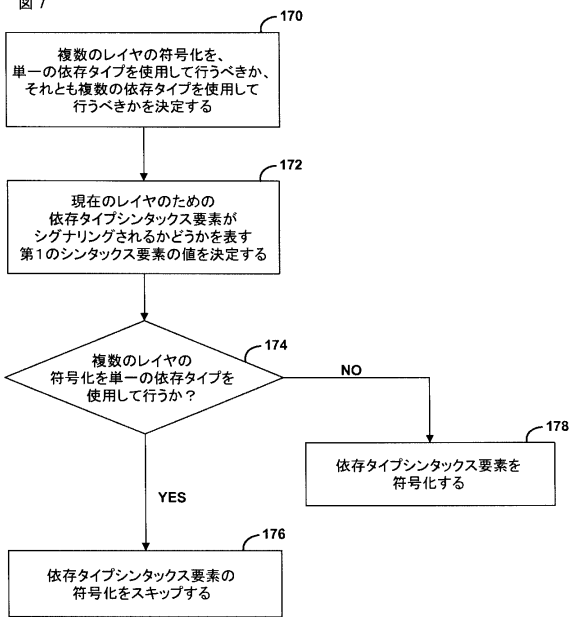


FIG. 7

【図8】

図8

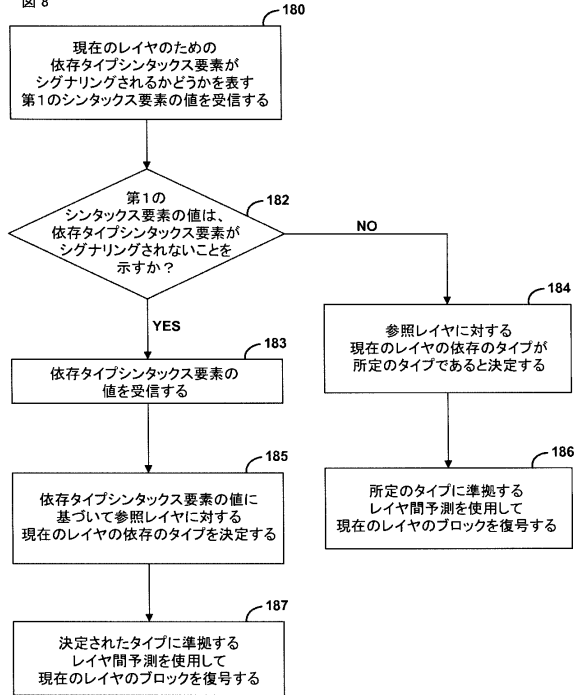


FIG. 8

---

フロントページの続き

前置審査

- (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 ワン、イエ-クイ  
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 鉢呂 健

- (56)参考文献 国際公開第 2 0 1 5 / 0 0 5 3 3 1 ( W O , A 1 )  
米国特許出願公開第 2 0 1 3 / 0 1 8 8 7 3 3 ( U S , A 1 )  
Tomohiro Ikai, Tadashi Uchiumi, MV-HEVC/SHVC HLS: On dependency type for SHVC/MV-HEVC , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Vienna, AT, 25 July - 2 Aug. 2013, [JCTVC-N0058], Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1 /SC 29/WG 11 5th Meeting: Vienna, AT, 27 July - 2 Aug. 2013, [JCT3V-E0041] , 2 0 1 3 年 7 月 1 2 日 , pp.1-3  
CHEN J; RAPA K; LI X; ET AL , DESCRIPTION OF SCALABLE VIDEO CODING TECHNOLOGY PROPOSAL BY QUALCOMM (CONFIGURATION 2) , 11. JCT-VC MEETING; 102. MPEG MEETING; 10-10-2012 - 1 9-10-2012; SHANGHAI; (JOINT 以下備考 , 2 0 1 2 年 1 0 月 2 日 , NR:JCTVC-K0036 , COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ) , U R L , HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/

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