

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第6275736号
(P6275736)

(45) 発行日 平成30年2月7日 (2018. 2. 7)

(24) 登録日 平成30年1月19日 (2018. 1. 19)

(51) Int. Cl.

F I

HO 4 N 19/597 (2014. 01)

HO 4 N 19/105 (2014. 01)

HO 4 N 19/176 (2014. 01)

HO 4 N 19/70 (2014. 01)

HO 4 N 19/597

HO 4 N 19/105

HO 4 N 19/176

HO 4 N 19/70

請求項の数 15 (全 95 頁)

(21) 出願番号	特願2015-545880 (P2015-545880)	(73) 特許権者	595020643
(86) (22) 出願日	平成25年12月6日 (2013. 12. 6)		クォアルコム・インコーポレイテッド
(65) 公表番号	特表2016-504845 (P2016-504845A)		QUALCOMM INCORPORATED
(43) 公表日	平成28年2月12日 (2016. 2. 12)		アメリカ合衆国、カリフォルニア州 92
(86) 国際出願番号	PCT/US2013/073626		121-1714、サン・ディエゴ、モア
(87) 国際公開番号	W02014/089461		ハウス・ドライブ 5775
(87) 国際公開日	平成26年6月12日 (2014. 6. 12)	(74) 代理人	100108855
審査請求日	平成28年11月8日 (2016. 11. 8)		弁理士 蔵田 昌俊
(31) 優先権主張番号	61/734, 874	(74) 代理人	100109830
(32) 優先日	平成24年12月7日 (2012. 12. 7)		弁理士 福原 淑弘
(33) 優先権主張国	米国 (US)	(74) 代理人	100158805
(31) 優先権主張番号	14/098, 357		弁理士 井関 守三
(32) 優先日	平成25年12月5日 (2013. 12. 5)	(74) 代理人	100194814
(33) 優先権主張国	米国 (US)		弁理士 奥村 元宏

最終頁に続く

(54) 【発明の名称】 スケーラブルなマルチビュービデオコーディングにおける高度な残差予測

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを復号する方法であって、

第 1 のビューに関連付けられるビデオデータのブロックをコーディングするための区分モードを決定することと、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記決定された区分モードに基づいて、ビュー間残差予測処理における残差予測子への適用のために重み付けファクタをコーディングするかどうかを決定することと、ここで、前記ビュー間残差予測処理は、前記第 1 のビューとは異なる第 2 のビューに関連付けられる残差予測子に対して、ビデオデータの前記ブロックについての残差データをコーディングすることを含み、前記残差データは、ビデオデータの前記ブロックのサンプルと、ビデオデータの前記ブロックの時間的動きベクトルによって識別される時間的参照ブロックの対応するサンプルとの差を示し、前記残差予測子は、ビデオデータの前記ブロックの視差ベクトルによって識別される視差参照ブロックのサンプルと、前記時間的動きベクトルおよび視差動きベクトルによって識別される時間的視差参照ブロックの対応するサンプルとの差を示し、

前記決定された区分モードでビデオデータの前記ブロックを復号することと、を備え、

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、前記区分モードが PART_2 N×2 Nモード、PART_2 N×Nモード、

および $PART_N \times 2N$ モードのうちの 1 つに等しくないとき、前記重み付けファクタをコーディングしないと決定することを備える、方法。

【請求項 2】

ビデオデータを符号化する方法であって、

第 1 のビューに関連付けられるビデオデータのブロックをコーディングするための区分モードを決定することと、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記決定された区分モードに基づいて、ビュー間残差予測処理における残差予測子への適用のために重み付けファクタをコーディングするかどうかを決定することと、ここで、前記ビュー間残差予測処理は、前記第 1 のビューとは異なる第 2 のビューに関連付けられる残差予測子に対して、ビデオデータの前記ブロックについての残差データをコーディングすることを含み、前記残差データは、ビデオデータの前記ブロックのサンプルと、ビデオデータの前記ブロックの時間的動きベクトルによって識別される時間的参照ブロックの対応するサンプルとの差を示し、前記残差予測子は、ビデオデータの前記ブロックの視差ベクトルによって識別される視差参照ブロックのサンプルと、前記時間的動きベクトルおよび視差動きベクトルによって識別される時間的視差参照ブロックの対応するサンプルとの差を示し、

前記決定された区分モードでビデオデータの前記ブロックを符号化することと、
を備え、

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、前記区分モードが $PART_2N \times 2N$ モード、 $PART_2N \times N$ モード、および $PART_N \times 2N$ モードのうちの 1 つに等しくないとき、前記重み付けファクタをコーディングしないと決定することを備える、方法。

【請求項 3】

前記重み付けファクタをコーディングしないという決定に基づいて、前記方法は、前記重み付けファクタの値が 0 であると自動的に決定し、前記ブロックの第 1 の残差を予測するために前記ビュー間残差予測処理を適用しないことをさらに備える、請求項 1 または請求項 2 に記載の方法。

【請求項 4】

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、前記区分モードが前記ブロックを予測することを全体として含むかどうかを決定することを備え、前記方法は、前記区分モードが前記ブロックを予測することを全体として含まないとき、前記重み付けファクタのコーディングをスキップし、前記ブロックの第 1 の残差を予測するために前記ビュー間残差予測処理を適用しないことをさらに備える、請求項 1 または請求項 2 に記載の方法。

【請求項 5】

前記方法は、前記区分モードが $PART_2N \times 2N$ モード、 $PART_2N \times N$ モード、および $PART_N \times 2N$ モードのうちの 1 つを含まないとき、前記重み付けファクタのコーディングをスキップし、前記ブロックの第 1 の残差を予測するために前記ビュー間残差予測処理を適用しないことをさらに備える、請求項 1 または請求項 2 に記載の方法。

【請求項 6】

High Efficiency Video Coding (HEVC) に適用され、

前記ブロックに対するコーディングモードに基づいて前記重み付けファクタをコーディングするかどうかを決定することをさらに備え、前記コーディングモードがスキップモードおよび統合モード以外のコーディングモードであるとき、前記重み付けファクタのコーディングをスキップし、前記ブロックの第 1 の残差を予測するために前記ビュー間残差予測処理を適用しない、請求項 1 または請求項 2 に記載の方法。

【請求項 7】

前記重み付けファクタをコーディングするという決定に基づいて、前記方法はさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタをディセーブルにすることを備え、前記1つまたは複数のディセーブルにされた重み付けファクタは前記ビュー間残差予測処理において使用されない、請求項1または請求項2に記載の方法。

【請求項8】

前記1つまたは複数の重み付けファクタをディセーブルにすることは、シーケンスパラメータセット（SPS）、ビデオパラメータセット（VPS）、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタをディセーブルにすることを備える、請求項1に従属する請求項7に記載の方法。

10

【請求項9】

前記重み付けファクタをコーディングするという決定に基づいて、前記方法はさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタを修正することを備え、前記修正された1つまたは複数の重み付けファクタは前記ビュー間残差予測処理において使用される、請求項1または請求項2に記載の方法。

【請求項10】

前記1つまたは複数の重み付けファクタを修正することは、シーケンスパラメータセット（SPS）、ビデオパラメータセット（VPS）、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタを修正することを備える、請求項1に従属する請求項9に記載の方法。

20

【請求項11】

ビデオデータを復号するための装置であって、

第1のビューに関連付けられるビデオデータのブロックをコーディングするための区分モードを決定するための手段と、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記決定された区分モードに基づいて、ビュー間残差予測処理における残差予測子への適用のために重み付けファクタをコーディングするかどうかを決定するための手段と、ここで、前記ビュー間残差予測処理は、前記第1のビューとは異なる第2のビューに関連付けられるデータ残差予測子に対して、ビデオブロックの前記ブロックについての残差データをコーディングすることを含み、前記残差データは、ビデオデータの前記ブロックのサンプルと、ビデオデータの前記ブロックの時間的動きベクトルによって識別される時間的参照ブロックの対応するサンプルとの差を示し、前記残差予測子は、ビデオデータの前記ブロックの視差ベクトルによって識別される視差参照ブロックのサンプルと、前記時間的動きベクトルおよび視差動きベクトルによって識別される時間的視差参照ブロックの対応するサンプルとの差を示し、

30

前記決定された区分モードでビデオデータの前記ブロックを復号するための手段と、
を備え、

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定するための手段は、前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つに等しくないとき、前記重み付けファクタをコーディングしないと決定するための手段を備える、装置。

40

【請求項12】

ビデオデータを符号化するための装置であって、

第1のビューに関連付けられるビデオデータのブロックをコーディングするための区分モードを決定するための手段と、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記決定された区分モードに基づいて、ビュー間残差予測処理における残差予測子への適用のために重み付けファクタをコーディングするかどうかを決定するための手段と、ここで、前記ビュー間残差予測処理は、前記第1のビューとは異なる第2のビューに関連付

50

けられるデータ残差予測子に対して、ビデオブロックの前記ブロックについての残差データをコーディングすることを含み、前記残差データは、ビデオデータの前記ブロックのサンプルと、ビデオデータの前記ブロックの時間的動きベクトルによって識別される時間的参照ブロックの対応するサンプルとの差を示し、前記残差予測子は、ビデオデータの前記ブロックの視差ベクトルによって識別される視差参照ブロックのサンプルと、前記時間的動きベクトルおよび視差動きベクトルによって識別される時間的視差参照ブロックの対応するサンプルとの差を示し、

前記決定された区分モードでビデオデータの前記ブロックを符号化するための手段と、
を備え、

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定するための手段は、前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つに等しくないとき、前記重み付けファクタをコーディングしないと決定するための手段を備える、装置。

10

【請求項13】

前記ブロックを復号するための前記手段は、前記ビュー間残差予測処理で前記ブロックを復号するための手段を備え、前記ビュー間残差予測処理で前記ブロックを復号するための前記手段は、

前記ブロックの時間的動きベクトルによって示される時間的参照ブロックを決定するための手段と、

前記ブロックの視差ベクトルによって示される視差参照ブロックを決定するための手段と、

20

前記時間的動きベクトルと前記視差ベクトルの組合せによって示される時間的視差参照ブロックを決定するための手段と、

符号化されたビットストリームから、前記ブロックに対する最終的な残差を示すデータを取得するための手段と、

前記視差参照ブロックと前記時間的視差参照ブロックとの差に基づいて、残差予測子を決定するための手段と、

前記最終的な残差、前記残差予測子、および前記時間的参照ブロックの組合せに基づいて、前記ブロックを再構築するための手段と、

を備える、請求項11に記載の装置。

30

【請求項14】

前記ブロックを符号化するための前記手段は、前記ビュー間残差予測処理で前記ブロックを符号化するための手段を備え、前記ビュー間残差予測処理で前記ブロックを符号化するための前記手段は、

前記ブロックの時間的動きベクトルによって示される時間的参照ブロックを決定するための手段と、

前記ブロックの視差ベクトルによって示される視差参照ブロックを決定するための手段と、

前記時間的動きベクトルと前記視差ベクトルの組合せによって示される時間的視差参照ブロックを決定するための手段と、

40

第1のブロックと前記時間的参照ブロックとの差を備える第1の残差を決定するための手段と、

前記視差参照ブロックと前記時間的視差参照ブロックとの差を備える残差予測子を決定するための手段と、

前記第1の残差と前記残差予測子との差に基づいて、最終的な残差を決定するための手段と、

前記最終的な残差を示すデータをビットストリーム中に符号化するための手段と、

を備える、請求項12に記載の装置。

【請求項15】

命令を記憶した非一時的コンピュータ可読媒体であって、前記命令は、実行されると、

50

1つまたは複数のプロセッサに、請求項1～10のいずれか一項に記載の方法を行わせる、非一時的コンピュータ可読媒体。

【発明の詳細な説明】

【優先権の主張】

【0001】

本出願は、その内容全体が完全に参照により組み込まれる、2012年12月7日出願された米国仮出願第61/734,874号の利益を主張する。

【技術分野】

【0002】

本開示は、ビデオコーディングに関する。

10

【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、MPEG-2、MPEG-4、ITU-T H. 263、ITU-T H. 264/MPEG-4、Part 10、Advanced Video Coding (AVC)で定義されている規格、現在開発中のHigh Efficiency Video Coding (HEVC)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のような、ビデオ圧縮技法を実装する。ビデオデバイスは、そのようなビデオ圧縮技法を実装することによって、デジタルビデオ情報をより効率的に送信し、受信し、符号化し、復号し、かつ/または記憶することができる。

20

【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(イントラピクチャ)予測および/または時間的(インターピクチャ)予測を実行する。ブロックベースのビデオコーディングの場合、ビデオスライス(すなわち、ピクチャまたはピクチャの一部)は、ツリーブロック、コーディングユニット(CU)および/またはコーディングノードとも呼ばれ得るビデオブロックに区分され得る。ピクチャのイントラコーディングされた(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ディングされた(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することができる。

30

【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予測ブロックをもたらす。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と、コーディングされたブロックと予測ブロックとの差を示す残差データとに従って符号化される。イントラコーディングされたブロックは、イントラコーディングモードと残差データとに従って符号化される。さらなる圧縮のために、残差データは空間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、次いで、残差変換係数は量子化され得る。最初に2次元アレイで構成される量子化された変換係数は、変換係数の1次元ベクトルを生成するためにスキャンされてよく、エントロピーコーディングがさらなる圧縮を達成するために適用されてよい。

40

【0006】

[0006]マルチビューコーディングビットストリームは、たとえば、複数の視点からのビ

50

ューを符号化することによって生成され得る。マルチビューコーディング態様を利用するいくつかの3次元(3D)ビデオ規格が開発されている。たとえば、異なるビューは、3Dビデオをサポートするために左目のビューと右目のビューとを伝えることができる。あるいは、いくつかの3Dビデオコーディング処理は、いわゆるマルチビュープラス深度コーディングを適用することができる。マルチビュープラス深度コーディングでは、3Dビデオビットストリームは、テクスチャビュー成分だけではなく深度ビュー成分も含み得る。たとえば、各ビューは、1つのテクスチャビュー成分と1つの深度ビュー成分とを備え得る。

【発明の概要】

【0007】

10

[0007]全般に、本開示は、High Efficiency Video Coding (HEVC) のような2次元コーデックに基づく、マルチレイヤエンコーダデコーダ(コーデック)および3次元ビデオ(3DV)コーデックのためのビュー間残差予測に関する。本開示の技法は、いくつかの例では、高度なインター残差予測(ARP)処理を改良(refine)するために使用され得る。たとえば、本開示の態様は、ARPをイネーブル(enable)/ディセーブル(disable)すること、ARPにおける補間、およびARPにおける重み付けファクタに関し得る。

【0008】

[0008]一例では、マルチレイヤビデオデータをコーディングする方法は、第1の時間的位置にあるビデオデータの第1のブロックに対して、第1のブロックをコーディングするための1つまたは複数の参照ピクチャリストが第2の異なる時間的位置にある少なくとも1つの参照ピクチャを含むかどうかを決定することと、1つまたは複数の参照ピクチャリスト中のある参照ピクチャのビデオデータの少なくとも1つの参照ブロックに対してビデオデータの第1のブロックをコーディングすることとを含み、コーディングすることは、1つまたは複数の参照ピクチャリストが第2の時間的位置にある少なくとも1つの参照ピクチャを含まないとき、ビュー間残差予測処理をディセーブルにすることを含む。

20

【0009】

[0009]別の例では、マルチレイヤビデオデータをコーディングするための装置は、第1の時間的位置にあるビデオデータの第1のブロックに対して、第1のブロックをコーディングするための1つまたは複数の参照ピクチャリストが第2の異なる時間的位置にある少なくとも1つの参照ピクチャを含むかどうかを決定し、1つまたは複数の参照ピクチャリスト中のある参照ピクチャのビデオデータの少なくとも1つの参照ブロックに対してビデオデータの第1のブロックをコーディングするように構成される、1つまたは複数のプロセッサを含み、コーディングすることは、1つまたは複数の参照ピクチャリストが第2の時間的位置にある少なくとも1つの参照ピクチャを含まないとき、ビュー間残差予測処理をディセーブルにすることを含む。

30

【0010】

[0010]別の例では、マルチレイヤビデオデータをコーディングするための装置は、第1の時間的位置にあるビデオデータの第1のブロックに対して、第1のブロックをコーディングするための1つまたは複数の参照ピクチャリストが第2の異なる時間的位置にある少なくとも1つの参照ピクチャを含むかどうかを決定するための手段と、1つまたは複数の参照ピクチャリスト中のある参照ピクチャのビデオデータの少なくとも1つの参照ブロックに対してビデオデータの第1のブロックをコーディングするための手段とを含み、コーディングすることは、1つまたは複数の参照ピクチャリストが第2の時間的位置にある少なくとも1つの参照ピクチャを含まないとき、ビュー間残差予測処理をディセーブルにすることを含む。

40

【0011】

[0011]別の例では、非一時的コンピュータ可読媒体は命令を記憶しており、この命令は、実行されると、1つまたは複数のプロセッサに、第1の時間的位置にあるビデオデータの第1のブロックに対して、第1のブロックをコーディングするための1つまたは複数の

50

参照ピクチャリストが第2の異なる時間的位置にある少なくとも1つの参照ピクチャを含むかどうかを決定させ、1つまたは複数の参照ピクチャリスト中のある参照ピクチャのビデオデータの少なくとも1つの参照ブロックに対してビデオデータの第1のブロックをコーディングさせ、コーディングすることは、1つまたは複数の参照ピクチャリストが第2の時間的位置にある少なくとも1つの参照ピクチャを含まないとき、ビュー間残差予測処理をディセーブルにすることを含む。

【0012】

[0012]別の例では、ビデオデータをコーディングする方法は、ビデオデータの現在のブロックに対する、時間的動きベクトルによって示される時間的参照ブロックの位置を決定することと、ここで、現在のブロックおよび時間的参照ブロックはビデオデータの第1のレイヤに位置し、第1のタイプの補間によって、現在のブロックの視差ベクトルによって示される視差参照ブロックの位置を補間することと、ここで、視差参照ブロックは第2の異なるレイヤに位置し、第1のタイプの補間は双線形フィルタを備え、時間的動きベクトルと視差ベクトルの組合せによって示される視差参照ブロックの時間的視差参照ブロックを決定することと、時間的参照ブロック、視差参照ブロック、および時間的視差参照ブロックに基づいて現在のブロックをコーディングすることを含む。

10

【0013】

[0013]別の例では、ビデオデータをコーディングするための装置は、ビデオデータの現在のブロックに対する、時間的動きベクトルによって示される時間的参照ブロックの位置を決定し、ここで、現在のブロックおよび時間的参照ブロックはビデオデータの第1のレイヤに位置し、第1のタイプの補間によって、現在のブロックの視差ベクトルによって示される視差参照ブロックの位置を補間し、ここで、視差参照ブロックは第2の異なるレイヤに位置し、第1のタイプの補間は双線形フィルタを備え、時間的動きベクトルと視差ベクトルの組合せによって示される視差参照ブロックの時間的視差参照ブロックを決定し、時間的参照ブロック、視差参照ブロック、および時間的視差参照ブロックに基づいて現在のブロックをコーディングするように構成される、1つまたは複数のプロセッサを含む。

20

【0014】

[0014]別の例では、ビデオデータをコーディングするための装置は、ビデオデータの現在のブロックに対する、時間的動きベクトルによって示される時間的参照ブロックの位置を決定するための手段と、ここで、現在のブロックおよび時間的参照ブロックはビデオデータの第1のレイヤに位置し、第1のタイプの補間によって、現在のブロックの視差ベクトルによって示される視差参照ブロックの位置を補間するための手段と、ここで、視差参照ブロックは第2の異なるレイヤに位置し、第1のタイプの補間は双線形フィルタを備え、時間的動きベクトルと視差ベクトルの組合せによって示される視差参照ブロックの時間的視差参照ブロックを決定するための手段と、時間的参照ブロック、視差参照ブロック、および時間的視差参照ブロックに基づいて現在のブロックをコーディングするための手段とを含む。

30

【0015】

[0015]別の例では、非一時的コンピュータ可読媒体は命令を記憶しており、この命令は、実行されると、1つまたは複数のプロセッサに、ビデオデータの現在のブロックに対する、時間的動きベクトルによって示される時間的参照ブロックの位置を決定させ、ここで、現在のブロックおよび時間的参照ブロックはビデオデータの第1のレイヤに位置し、第1のタイプの補間によって、現在のブロックの視差ベクトルによって示される視差参照ブロックの位置を補間させ、ここで、視差参照ブロックは第2の異なるレイヤに位置し、第1のタイプの補間は双線形フィルタを備え、時間的動きベクトルと視差ベクトルの組合せによって示される視差参照ブロックの時間的視差参照ブロックを決定させ、時間的参照ブロック、視差参照ブロック、および時間的視差参照ブロックに基づいて現在のブロックをコーディングさせる。

40

【0016】

[0016]別の例では、ビデオデータをコーディングする方法は、ビデオデータのブロック

50

をコーディングするための区分モードを決定することと、ここで、区分モードは予測コーディングのためのビデオデータのブロックの分割を示し、区分モードに基づいてビュー間残差予測処理のための重み付けファクタをコーディングするかどうかを決定することと、ここで、重み付けファクタがコーディングされないとき、ビュー間残差予測処理は現在のブロックに対する残差を予測するために適用されない、決定された区分モードでビデオデータのブロックをコーディングすることを含む。

【0017】

[0017]別の例では、ビデオデータをコーディングするための装置は、ビデオデータのブロックをコーディングするための区分モードを決定し、ここで、区分モードは予測コーディングのためのビデオデータのブロックの分割を示し、区分モードに基づいてビュー間残差予測処理のための重み付けファクタをコーディングするかどうかを決定することと、ここで、重み付けファクタがコーディングされないとき、ビュー間残差予測処理は現在のブロックに対する残差を予測するために適用されない、決定された区分モードでビデオデータのブロックをコーディングするように構成される、1つまたは複数のプロセッサを含む。

10

【0018】

[0018]別の例では、ビデオデータをコーディングするための装置は、ビデオデータのブロックをコーディングするための区分モードを決定するための手段と、ここで、区分モードは予測コーディングのためのビデオデータのブロックの分割を示し、区分モードに基づいてビュー間残差予測処理のための重み付けファクタをコーディングするかどうかを決定するための手段と、ここで、重み付けファクタがコーディングされないとき、ビュー間残差予測処理は現在のブロックに対する残差を予測するために適用されない、決定された区分モードでビデオデータのブロックをコーディングするための手段と、を含む。

20

【0019】

[0019]別の例では、非一時的コンピュータ可読媒体は命令を記憶しており、この命令は、実行されると、1つまたは複数のプロセッサに、ビデオデータのブロックをコーディングするための区分モードを決定させ、ここで、区分モードは予測コーディングのためのビデオデータのブロックの分割を示し、区分モードに基づいてビュー間残差予測処理のための重み付けファクタをコーディングするかどうかを決定させ、ここで、重み付けファクタがコーディングされないとき、ビュー間残差予測処理が現在のブロックに対する残差を予測するために適用されない、決定された区分モードでビデオデータのブロックをコーディングさせる。

30

【0020】

[0020]別の例では、ビデオデータをコーディングする方法は、ビデオデータの第1のレイヤ中のビデオデータの第1のブロックに対して、第1のブロックを予測するための時間的動きベクトルと、関連付けられる時間的参照ピクチャとを決定することと、ここで、時間的参照ピクチャはピクチャ順序カウンタ値を有し、第1のブロックと関連付けられる視差ベクトルによって示される視差参照ピクチャ中の視差参照ブロックを決定することと、ここで、視差参照ピクチャは、第1のブロックと第1のブロックと異なる第2のビューを含むピクチャを含むアクセスユニットに含まれ、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを、復号ピクチャバッファが含むかどうかを決定することと、ここで、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを、復号ピクチャバッファが含まないとき、時間的視差参照ピクチャは時間的動きベクトルと視差ベクトルの組合せに基づいて位置決定され、第1のブロックの残差データを予測するためのビュー間残差予測処理を修正することと、ビュー間残差予測処理と修正されたビュー間残差予測処理のうちの1つによって、ビデオデータの第1のブロックに対する残差をコーディングすることを含む。

40

【0021】

[0021]別の例では、ビデオデータをコーディングするための装置は、ビデオデータの第

50

1のレイヤ中のビデオデータの第1のブロックに対して、第1のブロックを予測するための時間的動きベクトルと、関連付けられる時間的参照ピクチャとを決定し、ここで、時間的参照ピクチャはピクチャ順序カウンタ値を有し、第1のブロックと関連付けられる視差ベクトルによって示される視差参照ピクチャ中の視差参照ブロックを決定し、ここで、視差参照ピクチャは、第1のブロックと第1のブロックと異なる第2のビューを含むピクチャを含むアクセスユニットに含まれ、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含むかどうかを決定し、ここで、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含まないとき、時間的視差参照ピクチャは時間的動きベクトルと視差ベクトルの組合せに基づいて位置決定される、第1のブロックの残差データを予測するためのビュー間残差予測処理を修正し、ビュー間残差予測処理と修正されたビュー間残差予測処理のうちの1つによって、ビデオデータの第1のブロックに対する残差をコーディングするように構成される、1つまたは複数のプロセッサを含む。

10

【0022】

[0022]別の例では、ビデオデータをコーディングするための装置は、ビデオデータの第1のレイヤ中のビデオデータの第1のブロックに対して、第1のブロックを予測するための時間的動きベクトルと、関連付けられる時間的参照ピクチャとを決定するための手段と、ここで、時間的参照ピクチャはピクチャ順序カウンタ値を有し、第1のブロックと関連付けられる視差ベクトルによって示される視差参照ピクチャ中の視差参照ブロックを決定するための手段と、ここで、視差参照ピクチャは、第1のブロックと第1のブロックと異なる第2のビューを含むピクチャを含むアクセスユニットに含まれ、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含むかどうかを決定するための手段と、ここで、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含まないとき、時間的視差参照ピクチャは時間的動きベクトルと視差ベクトルの組合せに基づいて位置決定され、第1のブロックの残差データを予測するためのビュー間残差予測処理を修正するための手段と、ビュー間残差予測処理と修正されたビュー間残差予測処理のうちの1つによって、ビデオデータの第1のブロックに対する残差をコーディングするための手段とを含む。

20

30

【0023】

[0023]別の例では、非一時的コンピュータ可読媒体は命令を記憶しており、この命令は、実行されると、1つまたは複数のプロセッサに、ビデオデータの第1のレイヤ中のビデオデータの第1のブロックに対して、第1のブロックを予測するための時間的動きベクトルと、関連付けられる時間的参照ピクチャとを決定させ、ここで、時間的参照ピクチャはピクチャ順序カウンタ値を有し、第1のブロックと関連付けられる視差ベクトルによって示される視差参照ピクチャ中の視差参照ブロックを決定させ、ここで、視差参照ピクチャは、第1のブロックと第1のブロックと異なる第2のビューを含むピクチャを含むアクセスユニットに含まれ、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含むかどうかを決定させ、ここで、第2のビュー中にあり時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを復号ピクチャバッファが含まないとき、時間的視差参照ピクチャは時間的動きベクトルと視差ベクトルの組合せに基づいて位置決定される、第1のブロックの残差データを予測するためのビュー間残差予測処理を修正させ、ビュー間残差予測処理と修正されたビュー間残差予測処理のうちの1つによって、ビデオデータの第1のブロックに対する残差をコーディングさせる。

40

【0024】

[0024]本開示の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おいて説明される。他の特徴、目的、および利点は、説明、図面、および特許請求の範囲から明らかになるであろう。

50

【図面の簡単な説明】

【0025】

【図1】本開示で説明され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】本開示で説明され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3】本開示で説明され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4】マルチビューシーケンスをコーディングすることを示す概念図。

【図5】例示的なマルチビュー予測パターンを示す概念図。

10

【図6】ビデオデータの例示的なスケーラブルレイヤを示す概念図。

【図7】現在のPUに対する例示的な空間隣接する予測ユニット（PU）を示す概念図。

【図8】ビュー間残差予測を示すブロック図。

【図9】マルチビュービデオコーディングにおける高度な残差予測（ARP）の例示的な予測構造を示す概念図。

【図10】ARPにおける、現在のブロックと、参照ブロックと、動き補償されたブロックとの例示的な関係を示す概念図。

【図11】1/4サンプルのルーマ補間のための整数サンプルと小数サンプルの位置を示す概念図。

【図12】ビデオデータのブロックをコーディングするための区分モードを示す概念図。

20

【図13】本開示の1つまたは複数の技法による、ビデオエンコーダの例示的な動作を示すフローチャート。

【図14】本開示の1つまたは複数の技法による、ビデオデコーダの例示的な動作を示すフローチャート。

【図15】本開示の1つまたは複数の技法による、ビデオエンコーダの例示的な動作を示すフローチャート。

【図16】本開示の1つまたは複数の技法による、ビデオデコーダの例示的な動作を示すフローチャート。

【図17】本開示の1つまたは複数の技法による、ビデオエンコーダの例示的な動作を示すフローチャート。

30

【図18】本開示の1つまたは複数の技法による、ビデオデコーダの例示的な動作を示すフローチャート。

【図19】本開示の1つまたは複数の技法による、ビデオエンコーダの例示的な動作を示すフローチャート。

【図20】本開示の1つまたは複数の技法による、ビデオデコーダ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】

【0026】

[0045]本開示の技法は全般に、高度な2次元（2D）コーデックに基づいて、マルチビューコーデック、3DV（たとえば、マルチビュープラス深度）コーデック、またはスケーラブルコーデックのための高度な残差予測（ARP）のコーディング効率をさらに改善するための、様々な技法に関する。たとえば、High Efficiency Video Coding（HEVC）規格が、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）によって開発されている。「HEVC Working Draft 9」と呼ばれる（本明細書ではWD9とも呼ばれる）HEVC規格のドラフトは、Bross他、「High Efficiency Video Coding（HEVC）text specification draft 9」、Joint Collaborative Team on

40

50

Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11、第11回会議、上海、中国、2012年10月に記載されており、http://phenix.int-evry.fr/jct/doc_end_user/documents/11/Shanghai/wg11/JCTVC-K1003-v10.zipから入手可能である。

【0027】

[0046] HEVCの1つの使用法は、高精細度および超高精細度(UHD)ビデオの領域におけるものであり得る。多くの高精細度(HD)ディスプレイはすでに、ステレオビデオをレンダリングすることが可能であり、UHDディスプレイの増大する解像度およびディスプレイサイズは、そのようなディスプレイをステレオビデオに対してさらにより適したものにし得る。その上、HEVCの改善された圧縮能力(たとえば、H.264/AVC Highプロファイルと比較して、同じ品質でビットレートは半分であると予測される)は、HEVCを、ステレオビデオをコーディングするための良好な候補にし得る。たとえば、ビュー間の冗長性を利用する機構を使用して、ビデオコーダ(たとえば、ビデオエンコーダまたはビデオデコーダ)は、H.264/AVC規格を使用してコーディングされる同じ品質および解像度の単一ビュー(モノスコープ)ビデオよりもさらに低いレートで、フル解像度のステレオビデオをコーディングするために、HEVCを使用することが可能であり得る。

【0028】

[0047] AVCベースのプロジェクトと同様に、VCEGおよびMPEGのJoint Collaboration Team on 3D Video Coding (JCT-3V)は、HEVCコーディング技術を使用している2つの3DV方法の研究を進めている。一方は、MV-HEVCとも呼ばれるHEVCのマルチビュー拡張であり、もう一方は、深度増強されたHEVCベースのフル3DVコーデック、すなわち3D-HEVCである。規格化の取り組みの一部は、HEVCに基づいたマルチビュー/3Dビデオコーディングの規格化を含む。最新のソフトウェア3D-HTMバージョン5.0は、https://hevc.hhi.fraunhofer.de/svn/svn_3D_VCS_software/tags/HTM-5.0/において電子的に入手可能である。以下で説明される技法は、上記の2つの提案されている3DV方法とともに実施され得る。

【0029】

[0048] いくつかの例では、本技法はまた(または代替的に)、HEVCに対するスケーラブル拡張とともに実施され得る。スケーラブルビデオコーディングでは、ビデオデータの複数のレイヤがコーディングされ得る。いくつかの例では、各レイヤは特定のビューに対応し得る。ここで、ビュースケーラビリティと空間スケーラビリティの適用は、より多くのビューに対する後方互換性のある拡張、および/または、レガシーデバイスによる復号が可能になる方法でのビューの解像度の向上を可能にするので、3Dサービスの進化において非常に有益であると考えられる。

【0030】

[0049] 2次元ビデオコーディングでは、ビデオデータ(すなわち、ピクチャのシーケンス)は、ピクチャごとに、必ずしも表示の順序ではない順序でコーディングされる。ビデオコーディングデバイスは、各ピクチャをブロックに分割し、各ブロックを個々にコーディングする。ブロックベースの予測モードは、イントラ予測とも呼ばれる空間予測とインター予測とも呼ばれる時間的予測とを含む。

【0031】

[0050] マルチビューコーディングされたデータまたはスケーラブルコーディングされたデータのような、3次元ビデオデータでは、ブロックはまた、ビュー間予測および/またはレイヤ間予測され得る。本明細書で説明されるように、ビデオ「レイヤ」は一般に、ビュー、フレームレート、解像度などの少なくとも1つの共通の特性を有するピクチャのシーケンスを指し得る。たとえば、レイヤは、マルチビュービデオデータの特定のビュー(

たとえば、視点）と関連付けられるビデオデータを含み得る。別の例として、レイヤは、スケーラブルビデオデータの特定のレイヤと関連付けられるビデオデータを含み得る。

【0032】

[0051]したがって、本開示は、ビデオデータのレイヤとビューを交換可能に指し得る。すなわち、ビデオデータのビューはビデオデータのレイヤと呼ばれることがあり、ビデオデータのレイヤはビデオデータのビューと呼ばれることがある。その上、ビュー間予測およびレイヤ間予測という用語は、ビデオデータの複数のレイヤおよび／またはビューの間の予測を交換可能に指し得る。加えて、マルチレイヤコーデック（またはマルチレイヤビデオコード）は、マルチビューコーデックまたはスケーラブルコーデックをまとめて指し得る。

10

【0033】

[0052]マルチビューまたはスケーラブルビデオコーディングでは、ブロックは、ビデオデータの別のビューまたはレイヤのピクチャから予測され得る。この方式で、異なるビューから再構築されたビュー成分に基づくビュー間予測が可能にされ得る。本開示は、特定のビューまたはレイヤの符号化されたピクチャを指すために、「ビュー成分」という用語を使用する。すなわち、ビュー成分は、（表示順序または出力順序に関して）特定の時間における特定のビューに対する符号化されたピクチャを備え得る。ビュー成分（またはビュー成分のスライス）は、ピクチャ順序カウンタ（POC）値を有することがあり、POC値は一般に、ビュー成分の表示順序（または出力順序）を示す。

【0034】

20

[0053]通常、2つのビューの同一のまたは対応するオブジェクトは同じ位置にない。「視差ベクトル」という用語は、あるビューのピクチャ中のオブジェクトの、異なるビューにおける対応するオブジェクトに対する変位を示すベクトルを指すために使用され得る。そのようなベクトルは、「変位ベクトル」とも呼ばれ得る。視差ベクトルはまた、ピクチャのビデオデータのピクセルまたはブロックに適用可能であり得る。たとえば、第1のビューのピクチャ中のピクセルは、第2のビューのピクチャ中の対応するピクセルに対して、第1のビューおよび第2のビューが撮影された異なるカメラ位置に関する特定の視差ベクトルの分だけ、変位していることがある。いくつかの例では、視差ベクトルは、あるビューから別のビューへの動き情報（参照ピクチャインデックスを伴う、または伴わない動きベクトル）を予測するために使用され得る。

30

【0035】

[0054]したがって、コーディング効率をさらに改善するために、ビデオコードはまた、ビュー間動き予測および／またはビュー間残差予測を適用することができる。ビュー間動き予測に関して、ビデオコードは、あるビューのブロックと関連付けられる動きベクトルを、第2の異なるビューのブロックと関連付けられる動きベクトルに対してコーディングすることができる。同様に、以下でより詳細に説明されるように、ビュー間残差予測では、ビデオコードは、あるビューの残差データを第2の異なるビューの残差に対してコーディングすることができる。いくつかの例では、ビュー間残差予測は、特に3D-HVCの状況では、高度な残差予測（ARP）と呼ばれ得る。

【0036】

40

[0055]ARPでは、ビデオコードは、現在のブロックを予測するための予測ブロックを決定する。現在のブロックの予測ブロックは、現在のブロックの動きベクトルによって示される位置と関連付けられる、時間的参照ピクチャのサンプルに基づき得る。時間的参照ピクチャは、現在のピクチャと同じビューと関連付けられるが、現在のピクチャとは異なる時間インスタンスと関連付けられる。いくつかの例では、ブロックのサンプルが特定のピクチャのサンプルに基づくとき、サンプルは、特定のピクチャの実際のサンプルまたは補間されたサンプルに基づき得る。

【0037】

[0056]加えて、ARPでは、ビデオコードは、現在のブロックの視差ベクトルによって示される位置にある視差参照ピクチャのサンプルに基づいて、視差参照ブロックを決定す

50

る。視差参照ピクチャは、現在のピクチャとは異なるビュー（すなわち、参照ビュー）と関連付けられるが、現在のピクチャと同じ時間インスタンスと関連付けられる。

【0038】

[0057]ビデオコーダはまた、現在のブロックの時間的視差参照ブロックを決定する。時間的参照ブロックは、現在のブロックの動きベクトルおよび視差ベクトルによって示される位置と関連付けられる時間的視差参照ピクチャのサンプルに基づく。たとえば、時間的視差参照ブロックは、視差参照ブロックに時間的動きベクトルを適用する（たとえば、時間的動きベクトルを再使用する）ことによって、位置決定され得る。したがって、時間的視差参照ピクチャは、視差参照ピクチャと同じビューと関連付けられ、現在のブロックの時間的参照ピクチャと同じアクセスユニットと関連付けられる。

10

【0039】

[0058]例示を目的に、時間的視差参照ブロックは、時間的動きベクトルを視差参照ブロックに適用することによって位置決定されるものとして本明細書では説明されるが、いくつかの例では、時間的動きベクトルは、実際には視差参照ピクチャに直接適用されないことがある。むしろ、時間的動きベクトルは、たとえば現在のブロックに対して、時間的視差参照ブロックを位置決定するために、視差ベクトルと組み合わせられ得る。たとえば、例示を目的に、視差ベクトルがDV[0]およびDV[1]として示され、時間的動きベクトルがTMV[0]およびTMV[1]として示されると仮定する。この例では、ビデオコーダ（ビデオエンコーダまたはビデオデコーダのような）は、視差ベクトルと時間的動きベクトルを組み合わせることによって、たとえばDV[0]+TMV[0]、DV[1]+TMV[1]によって、現在のブロックに対する時間的視差参照ピクチャ中の時間的視差ブロックの位置を決定することができる。したがって、「時間的動きベクトルを視差参照ブロックに適用する」ことに対する本明細書での言及は、時間的動きベクトルが視差参照ブロックの位置に直接適用されることを必ずしも要求しない。

20

【0040】

[0059]ビデオコーダは次いで、現在のブロックと関連付けられる残差、たとえば、現在のブロックと時間的参照ブロックとの差を予測するための、残差予測子を決定する。現在のブロックに対する残差予測子の各サンプルは、視差参照ブロックのサンプルと、時間的視差参照ブロックの対応するサンプルとの差を示す。いくつかの例では、ビデオコーダは、重み付けファクタ（たとえば、0、0.5、1など）を残差予測子に適用して、残差予測子の精度を上げることができる。

30

【0041】

[0060]ビデオコーダがビデオエンコーダである例では、ビデオエンコーダは、現在のブロックについての最終的な残差ブロックを決定することができる。最終的な残差ブロックは、現在のブロックのサンプルと、時間的予測ブロック中のサンプルと、残差予測子中のサンプルとの差を示すサンプルを備える。ビデオエンコーダは、ビットストリーム中に、最終的な残差ブロックを表すデータを含め得る。ビデオコーダがビデオデコーダである例では、ビデオデコーダは、最終的な残差ブロック、残差予測子、および時間的予測ブロックに基づいて、現在のブロックを再構築することができる。

【0042】

40

[0061]ARPはビュー間（またはレイヤ間）残差予測のコーディング効率を改善することができるが、さらなる改良が可能である。たとえば、本開示のいくつかの技法は、ARP重み付けファクタに関する。上で述べられたように、ビデオコーダは、重み付けファクタを残差予測子に適用することができる。一般に、重み付けファクタは、現在のブロックをコーディングするための参照ピクチャリスト中に時間的参照ピクチャがあるかどうかに関係なく、常にビットストリーム中でシグナリングされる。しかしながら、時間的参照ピクチャがないときに重み付けファクタをシグナリングすることは、不必要に複雑さを上げて効率を下げることもあり、それは、時間的参照ピクチャがなければ時間的予測およびARPを適用するための関連付けられる残差もないからである。

【0043】

50

[0062] 参照ピクチャリスト中に（たとえば、リスト0にもリスト1にも）時間的参照ピクチャがない可能性がある1つの例は、ランダムアクセスピクチャをコーディングするときである。以下でより詳細に説明されるように、ランダムアクセスピクチャは、時間的に予測されない。ランダムアクセスピクチャは通常、イントラ予測だけ、またはビュー間予測だけが行われる（ビュー間参照ピクチャのみが参照ピクチャリストに含まれる）。したがって、上で述べられたように、重み付けファクタのシグナリングは不必要かつ非効率的であり、それは、予測子を決定するための残差がないからである。

【0044】

[0063] 本開示の態様によれば、ビデオコーダ（ビデオエンコーダまたはビデオデコーダのような）は、現在コーディングされているブロックに対する参照ピクチャリスト中の参照ピクチャに基づいて、ARP（あるレイヤの残差を第2の異なるレイヤの残差に対してコーディングすることを含む）をイネーブルまたはディセーブルにすることができる。ある例では、ビデオコーダは、現在コーディングされているブロックに対する参照ピクチャリスト（たとえば、リスト0またはリスト1）が任意の時間的参照ピクチャを含むかどうかに基づいて、ARPをイネーブルまたはディセーブルにすることができる。本開示の態様によれば、インター予測されたスライスに対する参照ピクチャリストがビュー間参照ピクチャのみを含む場合、ビデオコーダは、スライスのブロックをコーディングするときにARPをディセーブルにすることができる。そのような例では、ビデオコーダがビデオエンコーダを備えるとき、ビデオエンコーダは、ビットストリーム中のスライス内のすべてのブロック（たとえば、以下でより詳細に説明されるように、High Efficiency Video Coding (HEVC) の状況ではコーディングユニットまたは予測ユニット）に対する重み付けファクタをシグナリングしなくてよい（重み付けファクタのシグナリングをスキップ（省略）してよい）。同様に、ビデオコーダがビデオデコーダを備えるとき、ビデオデコーダは、重み付けファクタの復号を同様にスキップし、重み付けファクタが0に等しいと自動的に決定する（すなわち、推測する）ことができる。

【0045】

[0064] 上で説明された技法は、ランダムアクセスピクチャの状況で適用され得る。たとえば、本開示の態様によれば、ビデオコーダは、現在コーディングされているビュー成分がランダムアクセスビュー成分かどうかに基づいて、ARPをイネーブルまたはディセーブルにすることができる。上で述べられたように、ランダムアクセスピクチャはイントラ予測またはビュー間予測されるので、ランダムアクセスビュー成分は時間的参照ピクチャを有しない。したがって、ビデオコーダは、ランダムアクセスビュー成分の各ブロックに対してARPをディセーブルにすることができる。やはり、ビデオエンコーダは、ビットストリーム中で重み付けファクタをシグナリングしなくてよい（重み付けファクタのシグナリングをスキップしてよい）。同様に、ビデオコーダがビデオデコーダを備えるとき、ビデオデコーダは同様に、重み付けファクタの復号をスキップし、重み付けファクタが0に等しいと推測することができる。

【0046】

[0065] 別の例では、本開示の態様によれば、ビデオコーダは、少なくとも1つの参照ピクチャが現在コーディングされているブロックと同じビューからのものである場合、ARPをイネーブルにすることができる。加えて、または代替的に、ビデオコーダは、両方の参照ピクチャ（RefPicList0中の参照ピクチャおよびRefPicList1中の参照ピクチャに対応する）が利用可能であればそれらが現在コーディングされているブロックと同じビューのものであるときにのみ、ARPをイネーブルにすることができる。加えて、または代替的に、ビデオコーダは、ブロックがビュー間参照ピクチャとともにビュー間コーディングされる場合、ブロックに対するARPをディセーブルにすることができる。上で述べられたように、ARPがディセーブルにされるとき、重み付けファクタはシグナリングされない。

【0047】

[0066] 本開示の技法はまた、ARPにおける補間に関する。たとえば、ARPを実行す

るとき（たとえば、重み付けファクタが0ではない）、ビデオエンコーダとビデオデコーダの両方が、残差予測子生成処理の間に追加の動き補償処理を使用することができる。したがって、動きベクトルが小数ピクセル（小数ペル）位置を示す場合、ビデオコーダは、2つの小数ペル補間処理、たとえば、時間的参照ブロックを位置決定するための一方の補間処理と、視差時間的参照ブロックを位置決定するための他方の補間処理とを実行する。加えて、ビデオコーダは、視差参照ブロックを決定するときに、さらに別の小数ペル補間処理を適用することができる。H E V Cでは、8タップのフィルタがルーマ成分に対して規定され、一方、4タップのフィルタがクロマ成分に対して規定される。そのような補間処理は、A R Pと関連付けられる計算の複雑さを上げ得る。

【0048】

[0067]本開示の態様によれば、A R Pの動き補償処理は、特に参照ブロックのサブピクセル（サブペル）補間に関して、簡略化され得る。たとえば、ビデオコーダは、動き補償の間に予測信号を生成するために使用される処理（たとえば、時間的参照ブロックを決定するために使用される処理）と同様または同一の方法で、視差参照ブロックを決定することができる。すなわち、ビデオコーダは、現在のブロックの視差ベクトルとともに、再構築された参照ビューピクチャを使用して、視差参照ブロックを決定することができる。

【0049】

[0068]いくつかの例では、本開示の態様によれば、ビデオコーダは、A R P中の参照ブロックの位置を決定するための、1つまたは複数のタイプの補間を使用することができる。たとえば、ビデオコーダは、双線形フィルタのようなローパスフィルタを使用して、視差参照ブロックの位置を補間することができる。加えて、または代替的に、ビデオコーダは、ローパスフィルタを使用して、時間的視差参照ブロックの位置を補間することができる。さらに別の例では、ビデオコーダは、ローパスフィルタを使用して、時間的参照ブロックの位置を補間することができる。したがって、本開示の態様によれば、ビデオコーダは、双線形フィルタを使用して、A R P中の1つまたは複数の参照ブロックの位置を補間することができ、これは、H E V Cによって規定される高次のタップフィルタを適用することより、計算上、より効率的であり得る。本明細書では双線形フィルタに対する言及が行われるが、1つまたは複数の他のローパスフィルタも使用されてよいこと、またはそれらが代替的に使用されてよいことを理解されたい。本開示の態様によれば、ビデオコーダは、ルーマ成分、クロマ成分、またはルーマ成分とクロマ成分の両方の任意の組合せに、上で説明されたローパスフィルタを適用することができる。

【0050】

[0069]本開示の技法はまた、特定のコーディングモードおよび／または区分モードに対するA R P重み付けファクタをシグナリングすることに関する。たとえば、一般に、重み付けファクタは、 $PART_2N \times 2N$ 、 $PART_2N \times N$ 、 $PART_N \times 2N$ などを含むすべての区分モード（たとえば、図12に示される例に関してより詳細に説明されるような）、および、スキップ、統合（merge）、高度な動きベクトル予測（A M V P）を含むすべてのインターコーディングされるモードに対して、シグナリングされ得る。すべての区分モードおよびインターモードに対する重み付けファクタをシグナリングすることは不必要に複雑さを上げ効率を下げることもあり、それは、A R Pがいくつかの区分モードまたはインターモードでは効率的に適用されないことがあるからである。

【0051】

[0070]本開示の態様によれば、A R Pは、現在コーディングされているブロックの区分モードおよび／またはコーディングモードに基づいて、イネーブルまたはディセーブルにされ得る。たとえば、重み付けファクタは、ある区分モードおよび／またはあるコーディングモードのみに対してシグナリングされるだけであり得る。重み付けファクタがビットストリームに含まれない場合、ビデオデコーダは、重み付けファクタの復号をスキップし、重み付けファクタの値が0である（したがってA R Pをディセーブルにする）と推測することができる。本開示の態様によれば、いくつかの例では、 $PART_2N \times 2N$ に等しくない区分モードを伴う任意のインターコーディングされたブロックに対する重み付け

ファクタはシグナリングされなくてよい。別の例では、PART__2N×2N、PART__2N×N、PART__N×2N以外の区分モードを伴うインターコーディングされたブロックに対する重み付けファクタはシグナリングされなくてよい。さらに別の例では、加えて、または代替的に、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタは、シグナリングされなくてよい。

【0052】

【0071】本開示の技法はまた、重み付けファクタがビットストリーム中でシグナリングされる方式を改良することに関する。たとえば、一般に、ビデオコーデは、3つの固定の重み付けファクタの固定セット（たとえば、0、0.5、および1）から重み付けファクタを選択することができる。しかしながら、いくつかの例では、3つの固定の重み付けファクタは、現在のビューとその参照ビューとの品質の差が原因で、十分な予測の効率を達成するのに十分な柔軟性をもたらさないことがある。現在のビューと参照ビューとの品質の差は、特にスケーラブルビデオコーディングに関しては、動的であり得る。逆に、3つの重み付けファクタは、いくつかのスライスまたはピクチャにより必要とされるものを超えることがある。すなわち、いくつかのスライスまたはピクチャは、複雑さとコーディング効率の改善との間の最適なバランスを達成するために、3つの重み付けファクタから選択する必要はないことがある。

【0053】

【0072】本開示の態様によれば、重み付けファクタに対するより柔軟な手法が実施され得る。たとえば、利用可能な重み付けファクタの数は、（たとえば、シーケンスパラメータセット（SPS）のようなパラメータセット中の）シーケンスレベルで変更され得る。例示を目的とするある例では、たとえば0.5および／または1の1つまたは複数の重み付けファクタをディセーブルにするためのインジケータが、SPS中でシグナリングされ得る。別の例では、そのようなインジケータは、ビデオパラメータセット（VPS）中でシグナリングされ、すべての非ベースビューに対して適用可能であってよい。さらに別の例では、そのようなインジケータは、各々の非ベースビューに対してVPS拡張においてシグナリングされ得る。別の例では、そのようなインジケータは、1つまたは複数の重み付けファクタをディセーブルにするために、ピクチャパラメータセット（PPS）、スライスヘッダ、またはビューパラメータセットにおいて提供され得る。重み付けファクタがディセーブルにされているとき、残りの重み付けファクタを表すためにより少数のビットが使用されてよく、これによってビットを節約する。

【0054】

【0073】他の態様によれば、1つまたは複数の重み付けファクタを修正および／または置換するための、インジケータが提供され得る。ある例では、ビデオコーデは、0.5という重み付けファクタを0.75という重み付けファクタで置換することができる。このインジケータは、スライスヘッダ、SPS、ピクチャパラメータセット（PPS）、またはVPSでシグナリングされ得る。

【0055】

【0074】本開示の技法はまた、復号ピクチャバッファ（以下で図2および図3に関してより詳細に説明されるように、参照ピクチャメモリとも交換可能に呼ばれ得る）の参照ピクチャおよび／または参照ピクチャリストに基づいて、ARP処理をイネーブルにするかディセーブルにするかを決定することに関する。たとえば、上で述べられたように、残差予測子を決定するための時間的視差参照ブロックは通常、時間的動きベクトルを視差参照ブロックに適用することによって位置決定される。しかしながら、いくつかの例では、復号ピクチャバッファは、時間的動きベクトルを視差参照ブロックに適用することによって示されるピクチャを含まないことがある。すなわち、復号ピクチャバッファは、視差参照ブロックと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じピクチャ順序カウンタ（POC）値も有する、ピクチャを含まないことがある。

【0056】

【0075】いくつかの例では、ピクチャが復号ピクチャバッファに含まれる場合であっても、参照ピクチャリストまたは視差参照ブロックを含むスライスの参照ピクチャリストは、時間的動きベクトルを視差参照ブロックに適用することによって示されるピクチャ、たとえば、可能性のある時間的視差参照ピクチャを含まないことがある。そのような例では、時間的視差参照ブロックを位置決定することは、コーディング処理に誤差および／または遅延をもたらすことがある。

【0057】

【0076】本開示の態様によれば、ビデオコーダは、復号ピクチャバッファおよび／または参照ピクチャリストのピクチャに基づいて、ARPをイネーブルまたはディセーブルにすることができる。たとえば、現在のブロックをコーディングするための復号ピクチャバッファが、現在のブロックの時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中のピクチャを含まないとき、ビデオコーダはARP処理を修正することができる。

10

【0058】

【0077】別の例では、加えて、または代替的に、視差参照ブロックの参照ピクチャリストが、現在のブロックの時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中にピクチャを含まないとき、ビデオコーダはARP処理を修正することができる。すなわち、現在の参照ピクチャリストのインデックスがXであるとする（Xは0または1である）、一例では、視差参照ブロックのXに等しいリストインデックスを伴う参照ピクチャリストが、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じPOCを有する参照ピクチャを含まない場合、ビデオコーダはARP処理を修正することができる。別の例では、視差参照ブロックの参照ピクチャリストのいずれもが（たとえば、リスト0もリスト1も）、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じPOCを有する参照ピクチャを含まない場合、ビデオコーダはARP処理を修正することができる。

20

【0059】

【0078】いくつかの例では、ビデオコーダは、現在のブロックがARPを使用してコーディングされないように、ARP処理をディセーブルにすることによってARP処理を修正することができる。すなわち、残差予測子は生成されず、または常に0に設定される。他の例では、ビデオコーダは、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、ARP処理を修正することができる。たとえば、ビデオコーダは、スケーリングされた動きベクトルが、視差参照ピクチャに適用されると、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にある時間的視差参照ピクチャを識別するように、時間的動きベクトルをスケーリングすることができる。上で説明された技法は、参照ピクチャリストに含まれないピクチャ中の視差参照ブロックをビデオコーダが位置決定しようとするのを防ぐことができる。

30

【0060】

【0079】図1は、高度な残差予測（ARP）のための本開示の技法を利用し得る例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示されるように、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与えるソースデバイス12を含む。特に、ソースデバイス12は、コンピュータ可読媒体16を介してビデオデータを宛先デバイス14に与える。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック（すなわち、ラップトップ）コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、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応し得る。

40

【0061】

50

[0080]宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化されたビデオデータを受信することができる。コンピュータ可読媒体16は、符号化されたビデオデータをソースデバイス12から宛先デバイス14に移動することが可能な任意のタイプの媒体またはデバイスを備え得る。一例では、コンピュータ可読媒体16は、ソースデバイス12が、符号化されたビデオデータを宛先デバイス14にリアルタイムで直接送信することを可能にするための通信媒体を備え得る。

【0062】

[0081]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、高周波(RF)スペクトルあるいは1つまたは複数の物理伝送線路のような、任意のワイヤレスまたは有線の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

【0063】

[0082]いくつかの例では、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からアクセスされ得る。ストレージデバイスは、ハードドライブ、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のメモリ、あるい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のような、様々な分散されたまたはローカルにアクセスされるデータ記憶媒体のいずれかを含み得る。さらなる例では、ストレージデバイスは、ソースデバイス12によって生成された符号化されたビデオを記憶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。

20

【0064】

[0083]宛先デバイス14は、ストリーミングまたはダウンロードを介して、ストレージデバイスから記憶されたビデオデータにアクセスし得る。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に送信することができる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を含む。宛先デバイス14は、インターネット接続を含む、任意の標準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)、有線接続(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み得る。ストレージデバイス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、またはそれらの組合せであり得る。

30

【0065】

[0084]本開示の技法は、必ずしもワイヤレスの用途または設定に限定されるとは限らない。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、dynamic adaptive streaming over HTTP(DASH)などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されたデジタルビデオ、データ記憶媒体に記憶されたデジタルビデオの復号、または他の用途のような、種々のマルチメディア用途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/またはビデオ電話などの用途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

【0066】

50

[0085]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。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。本開示によれば、ソースデバイス12のビデオエンコーダ20は、マルチビューコーディングにおける動きベクトル予測のための技法を適用するように構成され得る。他の例では、ソースデバイスおよび宛先デバイスは、他のコンポーネントまたは構成を含み得る。たとえば、ソースデバイス12は、外部カメラなどの外部ビデオソース18からビデオデータを受信し得る。同様に、宛先デバイス14は、一体型ディスプレイデバイスを含むのではなく、外部ディスプレイデバイスとインターフェースをとり得る。

【0067】

[0086]図1の示されるシステム10は一例にすぎない。高度な残差予測のための技法は、任意のデジタルビデオ符号化および／または復号デバイスによって実行され得る。一般に、本開示の技法はビデオ符号化デバイスによって実行されるが、本技法は、通常「コーデック」と呼ばれるビデオエンコーダ／デコーダによっても実行され得る。その上、本開示の技法は、ビデオプロセッサによっても実行され得る。ソースデバイス12および宛先デバイス14は、ソースデバイス12が、宛先デバイス14に送信するためのコーディングされたビデオデータを生成するような、コーディングデバイスの例にすぎない。いくつかの例では、デバイス12、14の各々がビデオ符号化コンポーネントとビデオ復号コンポーネントとを含むように、デバイス12、14は、実質的に対称的な方式で動作することができる。したがって、システム10は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、またはビデオ電話のために、ビデオデバイス12とビデオデバイス14との間の一方または双方のビデオ送信をサポートし得る。

【0068】

[0087]ソースデバイス12のビデオソース18は、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、および／またはビデオコンテンツプロバイダからビデオを受信するためのビデオフィードインターフェースを含み得る。さらなる代替として、ビデオソース18は、ソースビデオとしてのコンピュータグラフィックススペースのデータ、またはライブビデオとアーカイブされたビデオとコンピュータにより生成されたビデオとの組合せを生成し得る。場合によっては、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、いわゆるカメラ付き携帯電話またはビデオ付き携帯電話を形成し得る。しかしながら、上で言及されたように、本開示で説明される技法は、一般にビデオコーディングに適用可能であり、ワイヤレスおよび／または有線の用途に適用され得る。各々の場合において、キャプチャされたビデオ、以前にキャプチャされたビデオ、またはコンピュータにより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。次いで、符号化されたビデオ情報は、出力インターフェース22によってコンピュータ可読媒体16に出力され得る。

【0069】

[0088]コンピュータ可読媒体16は、ワイヤレスブロードキャストまたは有線ネットワーク送信などの一時媒体、あるいはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-rayディスク、または他のコンピュータ可読媒体などの記憶媒体（すなわち、非一時的記憶媒体）を含み得る。いくつかの例では、ネットワークサーバ（図示されず）は、ソースデバイス12から符号化されたビデオデータを受信し、たとえば、ネットワーク送信を介して、その符号化されたビデオデータを宛先デバイス14に与え得る。同様に、ディスクスタンピング設備のような、媒体製造設備のコンピューティングデバイスは、ソースデバイス12から符号化されたビデオデータを受信し、その符号化されたビデオデータを含むディスクを生成し得る。したがって、様々な例では、コンピュータ可読媒体16は、様々な形態の1つまたは複数のコンピュータ可読媒体を含むと理解され得る。

【0070】

[0089]宛先デバイス14の入力インターフェース28は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ビデオエンコーダ20によって定義され、またビデオデコーダ30によって使用される、ブロックおよび他のコーディングされたユニット、たとえば、GOPの特性および／または処理を記述するシンタックス要素を含む、シンタックス情報を含み得る。ディスプレイデバイス32は、復号されたビデオデータをユーザに対して表示し、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプのディスプレイデバイスのような、様々なディスプレイデバイスのいずれかを備え得る。

【0071】

[0090]図1には示されないが、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オーディオデコーダと統合されてよく、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の、適切なMUX-DEMUXユニット、または他のハードウェアおよびソフトウェアを含み得る。適用可能な場合、MUX-DEMUXユニットは、ITU H. 223マルチプレクサプロトコル、またはユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠し得る。

【0072】

[0091]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、適用可能なとき、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(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)、ディスクリット論理回路、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のような、種々の好適なエンコーダまたはデコーダ回路のいずれかとして実装され得る。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に含まれてよく、そのいずれもが複合ビデオエンコーダ／デコーダ(コーデック)の一部として統合されてよい。ビデオエンコーダ20および／またはビデオデコーダ30を含むデバイスは、集積回路、マイクロプロセッサ、および／または携帯電話のようなワイヤレス通信デバイスを備え得る。

【0073】

[0092]本開示では全般に、ビデオエンコーダ20が、ある情報をビデオデコーダ30などの別のデバイスに「シグナリング」することに言及することがある。しかしながら、ビデオエンコーダ20はあるシンタックス要素をビデオデータの様々な符号化された部分と関連付けることによって情報をシグナリングし得ることを理解されたい。すなわち、ビデオエンコーダ20は、ビデオデータの様々な符号化された部分のヘッダに、あるシンタックス要素を格納することによって、データを「シグナリング」することができる。いくつかの場合には、そのようなシンタックス要素は、ビデオデコーダ30によって受信され復号される前に、符号化され記憶され(たとえば、記憶デバイス24に記憶され)得る。したがって、「シグナリング」という用語は全般に、圧縮されたビデオデータを復号するためのシンタックスまたは他のデータの通信を、そのような通信がリアルタイムで発生するかほぼリアルタイムで発生するかある期間にわたって発生するかにかかわらず指すことがあり、ある期間にわたる通信は、シンタックス要素を符号化の時点で媒体に記憶し、次いで、シンタックス要素がこの媒体に記憶された後の任意の時点で復号デバイスによって取り出され得るときに、発生し得る。

【0074】

[0093]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、代替的にMPEG-4、Part 10、Advanced Video Coding(AVC)と呼ばれるITU-T H. 264規格のような、プロプライエタリ規格または業界規格、あるいはそのような規格の拡張に従って動作し得る。ITU-T H. 264/MPEG-4(AVC)規格は、Joint Video Team(JVT)として知られる共同パートナーシップの成果としてISO/IEC Moving Picture

10

20

30

40

50

Experts Group (MPEG) とともに ITU-T Video Coding Experts Group (VCEG) によって策定された。

【0075】

[0094]ビデオエンコーダ20およびビデオデコーダ30は、加えて、または代替的に、High Efficiency Video Coding (HEVC) 規格のような別のビデオコーディング規格に従って動作し得る。「HEVC Working Draft 9」と呼ばれるHEVC規格のドラフトは、Bross他、「High Efficiency Video Coding (HEVC) text specification draft 9」、Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP 3 and ISO/IEC JTC1/SC29/WG11、第11回会議、上海、中国、2012年10月に記載されている。

10

【0076】

[0095]さらに、上で述べられたように、HEVC向けのスケーラブルビデオコーディング拡張、マルチビューコーディング拡張、および3DV拡張を作成するための作業が進行中である。したがって、いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30はマルチビュービデオコーディングを実行することができる。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、HEVCのマルチビュー拡張(MV-HEVCと呼ばれる)、深度増強されたHEVCベースのフル3DVコーデック(3D-HEVCと呼ばれる)、または、HEVCのスケーラブルビデオコーディング拡張(SHEVC (スケーラブルHEVC) またはHSVC (高効率スケーラブルビデオコーディング) と呼ばれる)を実装することができる。

20

【0077】

[0096]以下で説明される技法は、上で述べられたHEVC拡張の1つまたは複数とともに実装され得る。3D-HEVCでは、テクスチャビューと深度ビューの両方に対する、コーディングユニット/予測ユニットレベルでのコーディングツールを含む新たなコーディングツールが、含まれサポートされ得る。2013年11月21日時点で、3D-HEVCのためのソフトウェア(すなわち、3D-HTMバージョン5.0)は、次のリンク https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-5.0/ からダウンロードされ得る。

30

【0078】

[0097]一般に、HEVCの動き補償ループは、H.264/AVCにおける動き補償ループと同じである。たとえば、動き補償ループにおける現在のフレーム

【数1】

$$\hat{f}$$

【0079】

の再構築は、逆量子化された係数rと時間的予測Pを足したもの

【数2】

40

$$\hat{f} = r + P$$

【0080】

に等しい。上の式では、Pは、Pフレームのための単予測的インター予測またはBフレームのための双予測的インター予測を示す。

【0081】

[0098]しかしながら、HEVCにおける動き補償のユニットは、以前のビデオコーディング規格におけるユニットとは異なる。たとえば、以前のビデオコーディング規格におけ

50

るマクロブロックの概念は、H E V Cでは存在しない。むしろ、マクロブロックは、一般的な4分木方式に基づく柔軟な階層構造によって置き換えられる。この方式の中で、3つのタイプのブロック、すなわちコーディングユニット(CU)、予測ユニット(PU)、および変換ユニット(TU)が定義される。CUは領域分割の基本ユニットである。CUの概念はマクロブロックの概念に類似するが、CUは最大サイズに制限されず、コンテンツの適応性を向上させるために4つの等しいサイズのCUへの再帰的分割を可能にする。PUはインター／イントラ予測の基本ユニットである。いくつかの例では、PUは、不規則な画像パターンを効果的にコーディングするために、単一のPUの中に複数の任意の形状の区分を含み得る。TUは変換の基本ユニットである。CUのTUは、CUのPUとは独立に定義され得る。しかしながら、TUのサイズは、TUが属するCUに限定される。3つの異なる概念へのブロック構造のこの分離は、各々がその役割に従って最適化されることを可能にでき、このことはコーディング効率の改善をもたらし得る。

10

【0082】

[0099]H E V Cおよび他のビデオコーディング仕様では、ビデオシーケンスは通常、一連のピクチャを含む。ピクチャは「フレーム」とも呼ばれることもある。ピクチャは、 S_L 、 S_{Cb} 、および S_{Cr} と示される3つのサンプルアレイを含み得る。 S_L は、ルーマサンプルの2次元アレイ(すなわち、ブロック)である。 S_{Cb} は、Cbクロミナンスサンプルの2次元アレイである。 S_{Cr} は、Crクロミナンスサンプルの2次元アレイである。クロミナンスサンプルは、本明細書では「クロマ」サンプルとも呼ばれることもある。他の例では、ピクチャは、モノクロームであってよく、ルーマサンプルのアレイのみを含み得る。

20

【0083】

[0100]ピクチャの符号化された表現を生成するために、ビデオエンコーダ20はコーディングツリーユニット(CTU)のセットを生成し得る。CTUの各々は、ルーマサンプルのコーディングツリーブロックと、クロマサンプルの2つの対応するコーディングツリーブロックと、それらのコーディングツリーブロックのサンプルをコーディングするために使用されるシンタックス構造とを備え得る。3つの別個のカラープレーンを有する1つまたは複数のモノクロームピクチャでは、CTUは、単一のコーディングツリーブロックと、そのコーディングツリーブロックのサンプルをコーディングするために使用されるシンタックス構造とを備え得る。コーディングツリーブロックは、サンプルの $N \times N$ のブロックであり得る。CTUは「ツリーブロック」または「最大コーディングユニット」(LCU)とも呼ばれることもある。H E V CのCTUは、H. 264/AVCのような、他の規格のマクロブロックに広い意味で類似し得る。しかしながら、CTUは、必ずしも特定のサイズに限定されず、1つまたは複数のCUを含み得る。スライスは、ラスタースキャン順序で連続的に順序付けられた整数個のCTUを含み得る。

30

【0084】

[0101]コーディングされたスライスは、スライスヘッダとスライスデータとを備え得る。スライスのスライスヘッダは、スライスについての情報を提供するシンタックス要素を含むシンタックス構造であり得る。スライスデータは、スライスのコーディングされたCTUを含み得る。

【0085】

40

[0102]本開示は、サンプルの1つまたは複数のブロックのサンプルをコーディングするために使用される1つまたは複数のサンプルブロックとシンタックス構造とを指すために、「ビデオユニット」または「ビデオブロック」または「ブロック」という用語を使用し得る。例示的なタイプのビデオユニットまたはブロックは、CTU、CU、PU、変換ユニット(TU)、マクロブロック、マクロブロック区分などを含み得る。いくつかの状況では、PUの議論は、マクロブロック区分のマクロブロックの議論と交換され得る。

【0086】

[0103]コーディングされたCTUを生成するために、ビデオエンコーダ20は、CTUのコーディングツリーブロックに対して4分木区分を再帰的に実行して、コーディングツリーブロックをコーディングブロックに分割することができ、したがって「コーディング

50

ツリーユニット」という名称である。コーディングブロックは、サンプルの $N \times N$ のブロックである。CUは、ルーマサンプルアレイとCbサンプルアレイとCrサンプルアレイとを有するピクチャのルーマサンプルのコーディングブロックと、そのピクチャのクロマサンプルの2つの対応するコーディングブロックと、それらのコーディングブロックのサンプルをコーディングするために使用されるシンタックス構造とを備え得る。3つの別個のカラープレーンを有する1つまたは複数のモノクロームピクチャでは、CUは、単一のコーディングブロックと、そのコーディングブロックのサンプルをコーディングするために使用されるシンタックス構造とを備え得る。

【0087】

[0104]ビデオエンコーダ20は、CUのコーディングブロックを1つまたは複数の予測ブロックに区分することができる。予測ブロックは、同じ予測が適用されるサンプルの方形（すなわち、正方形または非正方形）ブロックである。CUのPUは、ルーマサンプルの予測ブロックと、クロマサンプルの2つの対応する予測ブロックと、それらの予測ブロックを予測するために使用されるシンタックス構造とを備え得る。3つの別個のカラープレーンを有する1つまたは複数のモノクロームピクチャでは、PUは、単一の予測ブロックと、その予測ブロックを予測するために使用されるシンタックス構造とを備え得る。ビデオエンコーダ20は、CUの各PUのルーマ予測ブロック、Cb予測ブロック、およびCr予測ブロックに対する、予測ルーマブロック、予測Cbブロック、および予測Crブロックを生成することができる。したがって、本開示では、CUは1つまたは複数のPUに区分されると言われ得る。説明を簡単にするために、本開示は、PUの予測ブロックのサイズを、単にPUのサイズと呼ぶことがある。

【0088】

[0105]ビデオエンコーダ20は、イントラ予測またはインター予測を使用して、PUの予測ブロックを生成し得る。ビデオエンコーダ20がイントラ予測を使用してPUの予測ブロックを生成する場合、ビデオエンコーダ20は、PUと関連付けられたピクチャのサンプルに基づいてPUの予測ブロックを生成し得る。本開示では、「に基づいて」という句は、「に少なくとも部分的に基づいて」を示し得る。

【0089】

[0106]ビデオエンコーダ20がインター予測を使用してPUの予測ブロックを生成する場合、ビデオエンコーダ20は、PUと関連付けられたピクチャ以外の1つまたは複数のピクチャの復号されたサンプルに基づいて、PUの予測ブロックを生成し得る。ブロックの予測ブロック（たとえば、PU）を生成するためにインター予測が使用されるとき、本開示は、ブロックを「インターコーディングされる」または「インター予測される」ものとして呼ぶことがある。インター予測は、単予測的（すなわち、単予測）または双予測的（すなわち、双予測）であり得る。単予測または双予測を実行するために、ビデオエンコーダ20は、現在のピクチャに対して、第1の参照ピクチャリスト（RefPicList0）と第2の参照ピクチャリスト（RefPicList1）とを生成し得る。参照ピクチャリストの各々は、1つまたは複数の参照ピクチャを含み得る。参照ピクチャリストが構築された後（すなわち、利用可能であれば、RefPicList0およびRefPicList1）、参照ピクチャリストに対する参照インデックスは、参照ピクチャリストに含まれる任意の参照ピクチャを識別するために使用され得る。

【0090】

[0107]単予測を使用するとき、ビデオエンコーダ20は、参照ピクチャ内の参照位置を決定するために、RefPicList0とRefPicList1のいずれかまたは両方の中の参照ピクチャを探索し得る。さらに、単予測を使用するとき、ビデオエンコーダ20は、参照位置に対応するサンプルに少なくとも部分的に基づいて、PUの予測ブロックを生成し得る。その上、単予測を使用するとき、ビデオエンコーダ20は、PUの予測ブロックと参照位置との間の空間的変位を示す単一の動きベクトルを生成し得る。この動きベクトルは、PUの予測ブロックと参照位置との間の水平方向の変位を規定する水平成分を含み、PUの予測ブロックと参照位置との間の垂直方向の変位を規定する垂直成分を

10

20

30

40

50

含み得る。

【0091】

[0108] 双予測を使用してPUを符号化するとき、ビデオエンコーダ20は、RefPicList0中の参照ピクチャ中の第1の参照位置と、RefPicList1中の参照ピクチャ中の第2の参照位置とを決定し得る。ビデオエンコーダ20は、第1の参照位置および第2の参照位置に対応するサンプルに少なくとも部分的に基づいて、PUの予測ブロックを生成することができる。その上、双予測を使用してPUを符号化するとき、ビデオエンコーダ20は、PUの予測ブロックと第1の参照位置との間の空間的変位を示す第1の動きベクトルと、PUの予測ブロックと第2の参照位置との間の空間的変位を示す第2の動きベクトルとを生成することができる。

10

【0092】

[0109] ビデオエンコーダ20がインター予測を使用してPUの予測ブロックを生成する場合、ビデオエンコーダ20は、PUと関連付けられたピクチャ以外の1つまたは複数のピクチャのサンプルに基づいて、PUの予測ブロックを生成することができる。たとえば、ビデオエンコーダ20は、PUに対して単予測的インター予測（すなわち、単予測）または双予測的インター予測（すなわち、双予測）を実行することができる。

【0093】

[0110] ビデオエンコーダ20がPUに対して単予測を実行する例では、ビデオエンコーダ20は、PUの動きベクトルに基づいて、参照ピクチャ中の参照位置を決定することができる。ビデオエンコーダ20は次いで、PUの予測ブロックを決定することができる。PUの予測ブロック中の各サンプルは、参照位置と関連付けられ得る。いくつかの例では、PUの予測ブロック中のサンプルは、当該PUと同じサイズを有し左上の角が参照位置であるサンプルのブロック内にそのサンプルがあるとき、その参照位置と関連付けられ得る。予測ブロック中の各サンプルは、参照ピクチャの実際のサンプルまたは補間されたサンプルであり得る。

20

【0094】

[0111] 予測ブロックのルーマサンプルが参照ピクチャの補間されたルーマサンプルに基づく例では、ビデオエンコーダ20は、8タップの補間フィルタを参照ピクチャの実際のルーマサンプルに適用することによって、補間されたルーマサンプルを生成することができる。予測ブロックのクロマサンプルが参照ピクチャの補間されたクロマサンプルに基づく例では、ビデオエンコーダ20は、4タップの補間フィルタを参照ピクチャの実際のクロマサンプルに適用することによって、補間されたクロマサンプルを生成することができる。一般に、フィルタのタップの数は、フィルタを数学的に表すために必要とされる係数の数を示す。よりタップ数の大きいフィルタは、よりタップ数の少ないフィルタより、一般に複雑である。

30

【0095】

[0112] ビデオエンコーダ20がPUに対して双予測を実行する例では、PUは2つの動きベクトルを有する。ビデオエンコーダ20は、PUの動きベクトルに基づいて、2つの参照ピクチャ中の2つの参照位置を決定することができる。ビデオエンコーダ20は次いで、上で説明された方式で、2つの参照位置と関連付けられる参照ブロックを決定することができる。ビデオエンコーダ20は次いで、PUの予測ブロックを決定することができる。予測ブロック中の各サンプルは、参照ブロック中の対応するサンプルの加重平均であり得る。サンプルの重みは、PUを含むピクチャからの参照ピクチャの時間的距離に基づき得る。

40

【0096】

[0113] ビデオエンコーダ20は、様々な区分モードに従ってCUを1つまたは複数のPUに区分することができる。たとえば、CUのPUの予測ブロックを生成するためにイントラ予測が使用される場合、CUは、PART_2N×2NモードまたはPART_N×Nモードに従って区分され得る。PART_2N×2Nモードでは、CUは1つのPUしか有しない。PART_N×Nモードでは、CUは長方形の予測ブロックを有する4つの

50

等しいサイズのPUを有する。CUのPUの予測ブロックを生成するためにインター予測が使用される場合、CUは、PART_2N×2Nモード、PART_N×Nモード、PART_2N×Nモード、PART_N×2Nモード、PART_2N×nUモード、PART_2N×uDモード、PART_nL×2Nモード、またはPART_nR×2Nモードに従って区分され得る。PART_2N×NモードおよびPART_N×2Nモードでは、CUは長方形の予測ブロックを有する2つの等しいサイズのPUに区分される。PART_2N×nUモード、PART_2N×uDモード、PART_nL×2Nモード、およびPART_nR×2Nモードの各々では、CUは長方形の予測ブロックを有する2つの等しくないサイズのPUに区分される。

【0097】

10

[0114]ビデオエンコーダ20がCUの1つまたは複数のPUの予測ルーマブロックと、予測Cbブロックと、予測Crブロックとを生成した後、ビデオエンコーダ20は、CUのルーマ残差ブロックを生成することができる。CUのルーマ残差ブロック中の各サンプルは、CUの予測ルーマブロックのうちの1つの中のルーマサンプルとCUの元のルーマコーディングブロック中の対応するサンプルとの差を示す。さらに、ビデオエンコーダ20はCUのCb残差ブロックを生成することができる。CUのCb残差ブロック中の各サンプルは、CUの予測Cbブロックのうちの1つの中のCbサンプルと、CUの元のCbコーディングブロック中の対応するサンプルとの差を示し得る。ビデオエンコーダ20はまた、CUのCr残差ブロックを生成することができる。CUのCr残差ブロック中の各サンプルは、CUの予測Crブロックのうちの1つの中のCrサンプルと、CUの元のCr

20

【0098】

[0115]さらに、ビデオエンコーダ20は、4分木区分を使用して、CUのルーマ残差ブロック、Cb残差ブロック、およびCr残差ブロックを、1つまたは複数のルーマ変換ブロック、Cb変換ブロック、およびCr変換ブロックに分解することができる。変換ブロックは、同じ変換が適用されるサンプルの方形（たとえば、正方形または非正方形）ブロックである。CUのTUは、ルーマサンプルの変換ブロックと、クロマサンプルの2つの対応する変換ブロックと、それらの変換ブロックサンプルを変換するために使用されるシンタックス構造とを備え得る。したがって、CUの各TUは、ルーマ変換ブロック、Cb変換ブロックおよびCr変換ブロックと関連付けられ得る。TUと関連付けられたルーマ変換ブロックは、CUのルーマ残差ブロックのサブブロックであり得る。Cb変換ブロックはCUのCb残差ブロックのサブブロックであり得る。Cr変換ブロックはCUのCr残差ブロックのサブブロックであり得る。3つの別個のカラープレーンを有する1つまたは複数のモノクロームピクチャでは、TUは、単一の変換ブロックと、その変換ブロックのサンプルを変換するために使用されるシンタックス構造とを備え得る。

30

【0099】

[0116]ビデオエンコーダ20は、TUのルーマ変換ブロックに1回または複数回の変換を適用して、TUのルーマ係数ブロックを生成することができる。係数ブロックは変換係数の2次元アレイであり得る。変換係数はスカラー量であり得る。ビデオエンコーダ20は、TUのCb変換ブロックに1回または複数回の変換を適用して、TUのCb係数ブロックを生成することができる。ビデオエンコーダ20は、TUのCr変換ブロックに1回または複数回の変換を適用して、TUのCr係数ブロックを生成することができる。

40

【0100】

[0117]係数ブロック（たとえば、ルーマ係数ブロック、Cb係数ブロックまたはCr係数ブロック）を生成した後に、ビデオエンコーダ20は、係数ブロックを量子化することができる。量子化は、一般に、変換係数を表すために使用されるデータの量をできるだけ低減するために変換係数が量子化され、さらなる圧縮を実現する処理を指す。ビデオエンコーダ20は、CUと関連付けられた量子化パラメータ（QP）値に基づいて、CUのTUと関連付けられた係数ブロックを量子化することができる。ビデオエンコーダ20は、CUと関連付けられたQP値を調整することによって、CUと関連付けられた係数ブロッ

50

クに適用される量子化の程度を調整することができる。いくつかの例では、C Uと関連付けられるQ P値は、全体として現在のピクチャまたはスライスと関連付けられ得る。ビデオエンコーダ20が係数ブロックを量子化した後に、ビデオエンコーダ20は、量子化された変換係数を示すシンタックス要素をエントロピー符号化することができる。たとえば、ビデオエンコーダ20は、量子化された変換係数を示すシンタックス要素に対してコンテキスト適応型バイナリ算術コーディング(C A B A C)を実行することができる。

【0101】

[0118]ビデオエンコーダ20は、ビデオデータの表現(すなわち、コーディングされたピクチャおよび関連付けられたデータ)を形成するビットのシーケンスを含むビットストリームを出力することができる。ビットストリームは、一連のネットワーク抽象化レイヤ(N A L)ユニットを備え得る。N A Lユニットは、N A Lユニット中のデータのタイプの指示と、必要に応じてエミュレーション防止ビットが散在させられているローバイトシーケンスペイロード(R B S P)の形態でそのデータを含むバイトとを含む、シンタックス構造である。N A Lユニットの各々は、N A Lユニットヘッダを含み、R B S Pをカプセル化する。N A Lユニットヘッダは、N A Lユニットタイプコードを示すシンタックス要素を含み得る。N A LユニットのN A Lユニットヘッダによって規定されるN A Lユニットタイプコードは、N A Lユニットのタイプを示す。R B S Pは、N A Lユニット内にカプセル化された整数個のバイトを含むシンタックス構造であり得る。いくつかの例では、R B S Pは0ビットを含む。

【0102】

[0119]異なるタイプのN A Lユニットは、異なるタイプのR B S Pをカプセル化し得る。たとえば、異なるタイプのN A Lユニットは、ビデオパラメータセット(V P S)、シーケンスパラメータセット(S P S)、ピクチャパラメータセット(P P S)、コーディングされたスライス、S E Iなどに対して、異なるR B S Pをカプセル化し得る。(パラメータセットおよびS E IメッセージのためのR B S Pではなく)ビデオコーディングデータのためのR B S Pをカプセル化するN A Lユニットは、ビデオコーディングレイヤ(V C L)N A Lユニットと呼ばれ得る。

【0103】

[0120]H E V Cでは、S P Sは、コーディングされたビデオシーケンス(C V S)のすべてのスライスに適用される情報を含み得る。H E V Cでは、C V Sは、瞬時復号リフレッシュ(I D R)ピクチャ、あるいはブローkunリンクアクセス(B L A)ピクチャ、あるいは、I D RまたはB L Aピクチャではないすべての後続のピクチャを含むビットストリーム中の最初のピクチャであるクリーンランダムアクセス(C R A)ピクチャから開始し得る。すなわち、H E V Cでは、C V Sは、復号順序で、ビットストリーム中の第1のアクセスユニットであるC R Aアクセスユニットと、I D RアクセスユニットまたはB L Aアクセスユニットと、それに続いて、後続のI D RまたはB L Aアクセスユニットを含まないがそれまでのすべての後続のアクセスユニットを含む、0個以上の非I D Rおよび非B L Aアクセスユニットとからなり得る、アクセスユニットのシーケンスを備え得る。

【0104】

[0121]V P Sは、0個以上のC V S全体に適用されるシンタックス要素を備えるシンタックス構造である。S P Sは、S P Sがアクティブであるとき、アクティブであるV P Sを識別するシンタックス要素を含み得る。したがって、V P Sのシンタックス要素は、S P Sのシンタックス要素よりも一般的に適用可能であり得る。P P Sは、0個以上のコーディングされたピクチャに適用されるシンタックス要素を備えるシンタックス構造である。P P Sは、P P Sがアクティブであるとき、アクティブであるS P Sを識別するシンタックス要素を含み得る。スライスのスライスヘッダは、スライスがコーディングされているときにアクティブであるP P Sを示す、シンタックス要素を含み得る。

【0105】

[0122]ビデオデコーダ30は、ビデオエンコーダ20によって生成されたビットストリームを受信することができる。加えて、ビデオデコーダ30は、ビットストリームを解析

して、ビットストリームからシンタックス要素を取得することができる。ビデオデコーダ 30 は、ビットストリームから取得されたシンタックス要素に少なくとも部分的に基づいて、ビデオデータのピクチャを再構築することができる。ビデオデータを再構築するための処理は、全般に、ビデオエンコーダ 20 によって実行される処理の逆であり得る。たとえば、ビデオデコーダ 30 は、PU の動きベクトルを使用して、現在の CU の PU の予測ブロックを決定することができる。加えて、ビデオデコーダ 30 は、現在の CU の TU と関連付けられる係数ブロックを逆量子化することができる。ビデオデコーダ 30 は、現在の CU の TU と関連付けられる変換ブロックを再構築するために、係数ブロックに対して逆変換を実行することができる。ビデオデコーダ 30 は、現在の CU の PU の予測ブロックのサンプルを現在の CU の TU の変換ブロックの対応するサンプルに加算することによって、現在の CU のコーディングブロックを再構築することができる。ピクチャの各 CU のコーディングブロックを再構築することによって、ビデオデコーダ 30 はピクチャを再構築することができる。

10

【0106】

[0123]いくつかの例では、ビデオエンコーダ 20 は、統合（マージmerge）モードまたは高度な動きベクトル予測（AMVP）モードを使用して、PU の動き情報をシグナリングすることができる。言い換えると、HEVC では、動きパラメータの予測のために 2 つのモードがあり、一方は統合／スキップモードであり、他方は AMVP である。動き予測は、1 つまたは複数の他のビデオユニットの動き情報に基づく、ビデオユニット（たとえば、PU）の動き情報の決定を備え得る。PU の動き情報（すなわち、動きパラメータ）は、PU の動きベクトルと、PU の参照インデックスと、1 つまたは複数の予測方向インジケータとを含み得る。

20

【0107】

[0124]ビデオエンコーダ 20 が統合モードを使用して現在の PU の動き情報をシグナリングするとき、ビデオエンコーダ 20 は、統合候補リストを生成する。言い換えると、ビデオエンコーダ 20 は、動きベクトル予測子リストの構築処理を実行することができる。統合候補リストは、現在の PU に空間的または時間的に隣接する PU の動き情報を示す、統合候補のセットを含む。すなわち、統合モードでは、動きパラメータ（参照インデックス、動きベクトルなど）の候補リストが構築され、候補は、空間的に隣接するブロックおよび時間的に隣接するブロックからであり得る。

30

【0108】

[0125]さらに、統合モードでは、ビデオエンコーダ 20 は、統合候補リストから統合候補を選択することができ、現在の PU の動き情報として、選択された統合候補によって示される動き情報を使用することができる。ビデオエンコーダ 20 は、選択された統合候補の統合候補リスト中の位置をシグナリングすることができる。たとえば、ビデオエンコーダ 20 は、選択された統合候補の統合リスト内の位置を示すインデックスを送信する（すなわち、候補インデックスを統合する）ことによって、選択された動きベクトルパラメータをシグナリングすることができる。

【0109】

[0126]ビデオデコーダ 30 は、ビットストリームから、候補リストへのインデックス（すなわち、統合候補インデックス）を取得することができる。加えて、ビデオデコーダ 30 は、同じ統合候補リストを生成することができ、統合候補インデックスに基づいて、選択された統合候補を決定することができる。ビデオデコーダ 30 は次いで、選択された統合候補の動き情報を使用して、現在の PU の予測ブロックを生成することができる。すなわち、ビデオデコーダ 30 は、候補リストインデックスに少なくとも部分的に基づいて、候補リスト中の選択された候補を決定することができ、選択された候補は、現在の PU の動き情報（たとえば、動きベクトル）を規定する。このようにして、デコーダ側において、インデックスが復号されると、インデックスが指す対応するブロックのすべての動きパラメータは、現在の PU によって継承され得る。

40

【0110】

50

[0127]スキップモードは統合モードと同様である。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、ビデオエンコーダ20およびビデオデコーダ30が統合モードで統合候補リストを使用するのと同じ方法で、統合候補リストを生成し使用する。しかしながら、ビデオエンコーダ20がスキップモードを使用して現在のPUの動き情報をシグナリングするとき、ビデオエンコーダ20は、現在のPUに対する残差データを何らシグナリングしない。したがって、ビデオデコーダ30は、残差データを使用せずに、統合候補リスト中の選択された候補の動き情報によって示される参照ブロックに基づいて、PUの予測ブロックを決定することができる。スキップモードは統合モードと同じ動きベクトル導出処理を有するので、本文書で説明される技法は、統合モードとスキップモードの両方に適用され得る。

10

【0111】

[0128]AMVPモードは、ビデオエンコーダ20が候補リストを生成することができ候補リストから候補を選択することができるという点で、統合モードと同様である。しかしながら、ビデオエンコーダ20がAMVPモードを使用して現在のPUのRefPicListX(Xは0または1である)動き情報をシグナリングするとき、ビデオエンコーダ20は、現在のPUに対するRefPicListX動きベクトル予測子(MVP)フラグをシグナリングすることに加えて、現在のPUに対するRefPicListX動きベクトル差分(MVD)と現在のPUに対するRefPicListX参照インデックスとをシグナリングすることができる。現在のPUに対するRefPicListX MVPフラグは、AMVP候補リスト中の選択されたAMVP候補の位置を示し得る。現在のPUに対するRefPicListX MVDは、現在のPUのRefPicListX動きベクトルと、選択されたAMVP候補の動きベクトルとの差を示し得る。このようにして、ビデオエンコーダ20は、RefPicListX MVPフラグと、RefPicListX参照インデックス値と、RefPicListX MVDとをシグナリングすることによって、現在のPUのRefPicListX動き情報をシグナリングすることができる。言い換えると、現在のPUの動きベクトルを表すビットストリーム中のデータは、参照インデックスと、候補リストに対するインデックスと、MVDとを表すデータを含み得る。したがって、選択された動きベクトルは、候補リストへのインデックスを送信することによってシグナリングされ得る。加えて、参照インデックス値および動きベクトル差分もシグナリングされ得る。

20

30

【0112】

[0129]さらに、現在のPUの動き情報がAMVPモードを使用してシグナリングされる時、ビデオデコーダ30は、ビットストリームから、現在のPUに対するMVDとMVPフラグとを取得することができる。ビデオデコーダ30は、同じAMVP候補リストを生成することができ、MVPフラグに基づいて、選択されたAMVP候補を決定することができる。ビデオデコーダ30は、MVDを、選択されたAMVP候補によって示される動きベクトルに加算することによって、現在のPUの動きベクトルを復元することができる。すなわち、ビデオデコーダ30は、選択されたAMVP候補によって示される動きベクトルおよびMVDに基づいて、現在のPU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は次いで、復元された動きベクトルまたは現在のPUの動きベクトルを使用して、現在のPUの予測ブロックを生成することができる。

40

【0113】

[0130]ビデオコーダが現在のPUに対するAMVP候補リストを生成するとき、ビデオコーダは、現在のPUに空間的に隣接する位置(すなわち、空間的隣接PU)を包含するPUの動き情報に基づいて1つまたは複数のAMVP候補を、現在のPUに時間的に隣接するPU(すなわち、時間的隣接PU)の動き情報に基づいて1つまたは複数のAMVP候補を導出することができる。AMVPでは、各々の動きの仮定に対する動きベクトル予測子の候補リストは、コーディングされた参照インデックスに基づいて導出され得る。本開示では、PU(または他のタイプのビデオユニット)は、PUと関連付けられる予測ブロック(またはビデオユニットと関連付けられる他のタイプのサンプルブロック)がある

50

位置を含む場合、その位置を「包含する」と言われ得る。候補リストは、同じ参照インデックスと、時間的参照ピクチャにおいて同じ位置にあるブロックの隣接ブロックの動きパラメータ（すなわち、動き情報）に基づいて導出される時間的動きベクトル予測子とに関連付けられる、隣接ブロックの動きベクトルを含む。

【0114】

【0131】コーディング効率をさらに改善するために、ビデオコーダはまた、ビュー間動き予測および／またはビュー間残差予測を適用することができる。ビュー間動き予測に関して、ビデオコーダは、たとえば、上で説明された統合／スキップモードまたはAMVPモードを使用して、あるビューのブロックと関連付けられる動きベクトルを、第2の異なるビューのブロックと関連付けられる動きベクトルに対してコーディングすることができる。同様に、ビュー間残差予測におけるように、ビデオコーダは、あるビューの残差データを第2の異なるビューの残差に対してコーディングすることができる。いくつかの例では、ビュー間残差予測は、以下でより詳細に説明されるように、高度な残差予測（ARP）処理を適用することによって達成され得る。

【0115】

【0132】ビュー間残差予測では、ビデオエンコーダ20および／またはビデオデコーダ30は、現在のブロックを予測するための予測ブロックを決定することができる。現在のブロックの予測ブロックは、現在のブロックの動きベクトルによって示される位置と関連付けられる、時間的参照ピクチャのサンプルに基づき得る。時間的参照ピクチャは、現在のピクチャと同じビューと関連付けられるが、現在のピクチャとは異なる時間インスタンスと関連付けられる。いくつかの例では、ブロックのサンプルが特定のピクチャのサンプルに基づくとき、サンプルは、特定のピクチャの実際のサンプルまたは補間されたサンプルに基づき得る。

【0116】

【0133】ビデオエンコーダ20および／またはビデオデコーダ30はまた、現在のブロックの視差ベクトルによって示される位置にある視差参照ピクチャのサンプルに基づいて、視差参照ブロックを決定する。視差参照ピクチャは、現在のピクチャとは異なるビュー（すなわち、参照ビュー）と関連付けられるが、現在のピクチャと同じ時間インスタンスと関連付けられる。

【0117】

【0134】ビデオエンコーダ20および／またはビデオデコーダ30はまた、現在のブロックの時間的視差参照ブロックを決定する。時間的参照ブロックは、現在のブロックの動きベクトルおよび視差ベクトルによって（たとえば、動きベクトルと視差ベクトルの組合せによって）示される位置と関連付けられる時間的視差参照ピクチャのサンプルに基づく。すなわち、ビデオエンコーダ20および／またはビデオデコーダ30は、動きベクトルと視差ベクトルを組み合わせ、組み合わせられたベクトルを現在のブロックに適用し、時間的視差参照ピクチャ中の時間的視差参照ブロックを位置決定することができる。したがって、時間的視差参照ピクチャは、視差参照ピクチャと同じビューと関連付けられ、時間的参照ピクチャと同じアクセスユニットと関連付けられる。

【0118】

【0135】ビデオエンコーダ20および／またはビデオデコーダ30は次いで、現在のブロックと関連付けられる残差、たとえば、現在のブロックと時間的参照ブロックとの差を予測するための、残差予測子を決定する。現在のブロックに対する残差予測子の各サンプルは、視差参照ブロックのサンプルと、時間的視差参照ブロックの対応するサンプルとの差を示す。いくつかの例では、ビデオエンコーダ20および／またはビデオデコーダ30は、重み付けファクタ（たとえば、0、0.5、1など）を残差予測子に適用して、残差予測子の精度を上げることができる。

【0119】

【0136】ビデオエンコーダ20は、現在のブロックに対する最終的な残差ブロックを決定することができる。最終的な残差ブロックは、現在のブロックのサンプルと、時間的予測

ブロック中のサンプルと、残差予測子中のサンプルとの差を示すサンプルを備える。ビデオエンコーダ20は、ビットストリーム中に、最終的な残差ブロックを表すデータを含め得る。ビデオデコーダ、ビデオデコーダは、最終的な残差ブロック（たとえば、符号化されたビットストリームから取得されるような）、残差予測子、および時間的予測ブロックに基づいて、現在のブロックを再構築することができる。

【0120】

[0137]ARPはビュー間（またはレイヤ間）残差予測のコーディング効率を改善することができるが、さらなる改良が可能である。たとえば、本開示のいくつかの技法は、ARP重み付けファクタに関する。上で述べられたように、ビデオコーダは、重み付けファクタを残差予測子に適用することができる。一般に、重み付けファクタは、現在のブロックをコーディングするための参照ピクチャリスト中に時間的参照ピクチャがあるかどうかに関係なく、常にビットストリーム中でシグナリングされる。しかしながら、時間的参照ピクチャがないときに重み付けファクタをシグナリングすることは、不必要に複雑さを上げて効率を下げることもあり、それは、時間的参照ピクチャがなければ時間的予測およびARPを適用するための関連付けられる残差がないからである。

10

【0121】

[0138]本開示の態様によれば、ビデオエンコーダ20および／またはビデオデコーダ30は、第1の時間的位置にあるビデオデータの第1のブロックに対して、第1のブロックをコーディングするための参照ピクチャリスト（たとえば、RefPicList0およびRefPicList1）が第2の異なる時間的位置にある少なくとも1つの参照ピクチャを含むかどうかを決定することができる。ビデオエンコーダ20および／またはビデオデコーダ30はまた、参照ピクチャリスト中の参照ピクチャのビデオデータの少なくとも1つの参照ブロックに対して、ビデオデータの第1のブロックをコーディングすることができる。しかしながら、ビデオエンコーダ20および／またはビデオデコーダ30は、参照ピクチャリストが第2の時間的位置にある少なくとも1つの参照ピクチャを含まないとき、ビュー間残差予測処理をディセーブルにすることができる。

20

【0122】

[0139]ビデオエンコーダ20は、ビットストリームで重み付けファクタをシグナリングしなくてよく（重み付けファクタのシグナリングをスキップしてよく）、これによって、ビュー間残差予測が使用されないことを示す。そのような例では、ビデオエンコーダ20は、残差を予測することなく残差をコーディングすることができる。同様に、ビュー間予測がディセーブルにされるとき、ビデオデコーダ30は、重み付けファクタが0に等しいと自動的に決定し（すなわち推測し）、重み付けファクタの復号をスキップすることができる。このようにして、ビデオエンコーダ20および／またはビデオデコーダ30は、現在コーディングされているブロックに対する参照ピクチャリスト中の参照ピクチャに基づいて、ビュー間残差予測（たとえば、ARP）をイネーブルまたはディセーブルにすることができる。

30

【0123】

[0140]上で説明された技法は、ランダムアクセスピクチャの状況で適用され得る。たとえば、本開示の態様によれば、ビデオエンコーダ20および／またはビデオデコーダ30は、現在コーディングされているビュー成分がランダムアクセスビュー成分かどうかに基づいて、ビュー間残差予測をイネーブルまたはディセーブルにすることができる。すなわち、たとえば、ビデオエンコーダ20および／またはビデオデコーダ30は、関連付けられる時間的参照ピクチャを有しない、ランダムアクセスピクチャのすべてのブロックに対してのとき、ビュー間残差予測をディセーブルにすることができる。

40

【0124】

[0141]本開示の技法はまた、ビュー間残差予測における補間に関する。たとえば、ビュー間残差予測を実行するとき、ビデオエンコーダ20とビデオデコーダ30の両方が、コーディングの間に追加の動き補償処理を使用することができる。したがって、動きベクトルが小数ペル位置を示す場合、ビデオコーダは、2つの小数ペル補間処理、たとえば、時

50

間的参照ブロックを位置決定するための一方の補間処理と、視差時間的参照ブロックを位置決定するための他方の補間処理とを実行する。加えて、ビデオエンコーダ20および／またはビデオデコーダ30は、視差参照ブロックを決定するときに、さらに別の小数ペル補間処理を適用することができる。H E V Cでは、例として、8タップのフィルタがルーマ成分に対して規定され、一方、4タップのフィルタがクロマ成分に対して規定される。そのような補間処理は、ビュー間残差予測と関連付けられる計算上の複雑さを上げ得る。

【0125】

[0142]本開示の態様によれば、ビュー間残差予測の動き補償処理は、特に参照ブロックのサブペル補間に関して、簡略化され得る。たとえば、ビデオエンコーダ20および／またはビデオデコーダ30は、第1のタイプの補間によって、時間的動きベクトルによって示される時間的参照ブロックの、ビデオデータの現在のブロックに対する位置を補間することができ、現在のブロックおよび時間的参照ブロックは、ビデオデータの第1のレイヤに位置する。加えて、ビデオエンコーダ20および／またはビデオデコーダ30は、第2のタイプの補間によって、現在のブロックの視差ベクトルによって示される視差参照ブロックの位置を補間することができ、視差参照ブロックは第2の異なるレイヤに位置し、第2のタイプの補間は双線形フィルタを備える。ビデオエンコーダ20および／またはビデオデコーダ30はまた、時間的動きベクトルを視差参照ブロックに適用することによって示される視差参照ブロックの時間的視差参照ブロックを決定し、時間的参照ブロック、視差参照ブロック、および時間的視差参照ブロックに基づいて現在のブロックをコーディングする（たとえば、ビュー間残差予測を使用して現在のブロックの残差をコーディングする）ことができる。

【0126】

[0143]いくつかの例によれば、第1のタイプの補間はまた、双線形フィルタのようなローパスフィルタを備え得る。別の例では、双線形フィルタは、時間的視差参照ブロックの位置を補間するために使用され得る。したがって、本開示の態様によれば、ビデオエンコーダ20および／またはビデオデコーダ30は、双線形フィルタのようなローパスフィルタを使用して、ビュー間残差予測における1つまたは複数の参照ブロックの位置を補間することができる。再び、双線形フィルタに対して言及が行われるが、他の例では、ビデオエンコーダ20および／またはビデオデコーダ30は、H E V C（特に、W D 9で規定されるフィルタ）によって規定されるより高次のタップフィルタを適用することよりも計算上効率的である、いくつかの他のローパスフィルタを適用することができる。本開示の態様によれば、ビデオエンコーダ20および／またはビデオデコーダ30は、ルーマ成分、クロマ成分、またはルーマ成分とクロマ成分の両方の任意の組合せに、上で説明されたローパスフィルタを適用することができる。

【0127】

[0144]本開示の技法はまた、特定のコーディングモードおよび／または区分モードに対するA R P重み付けファクタをシグナリングすることに関する。たとえば、一般に、重み付けファクタは、 $PART_2N \times 2N$ 、 $PART_2N \times N$ 、 $PART_N \times 2N$ などを含むすべての区分モード（たとえば、図12に示される例に関してより詳細に説明されるような）、および、スキップ、統合、高度な動きベクトル予測（A M V P）を含むすべてのインターコーディングされるモードに対して、シグナリングされ得る。すべての区分モードおよびインターモードに対する重み付けファクタをシグナリングすることは不必要に複雑さを上げ効率を下げることもあり、それは、A R Pがいくつかの区分モードまたはインターモードでは効率的に適用されないことがあるからである。

【0128】

[0145]本開示の態様によれば、ビュー間残差予測は、現在コーディングされているブロックの区分モードおよび／またはコーディングモードに基づいて、イネーブルまたはディセーブルにされ得る。たとえば、ビデオエンコーダ20および／またはビデオデコーダ30は、ビデオデータのブロックをコーディングするための区分モードを決定することができ、区分モードは、予測コーディングのためのビデオデータのブロックの分割を示す。加

えて、ビデオエンコーダ20および／またはビデオデコーダ30は、区分モードに基づいて、ビュー間残差予測処理のために重み付けファクタをコーディングするかどうかを決定することができ、ここで、重み付けファクタがコーディングされないとき、ビュー間残差予測処理は、現在のブロックに対する残差を予測するために適用されない。ビデオエンコーダ20および／またはビデオデコーダ30は次いで、決定された区分モードを使用して、ビデオデータのブロックをコーディングすることができる。

【0129】

[0146]本開示の態様によれば、いくつかの例では、PART₂N×2Nに等しくない区分モードを伴う任意のインターコーディングされたブロックに対する重み付けファクタはシグナリングされなくてよい。別の例では、加えて、または代替的に、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタは、シグナリングされなくてよい。

10

【0130】

[0147]本開示の技法はまた、重み付けファクタがビットストリーム中でシグナリングされる方式を改良することに関する。たとえば、一般に、ビデオエンコーダ20および／またはビデオデコーダ30は、3つの固定の重み付けファクタの固定のセット（たとえば、0、0.5、および1）から重み付けファクタを選択することができる。しかしながら、いくつかの例では、3つの固定の重み付けファクタは、現在のビューとその参照ビューとの品質の差が原因で、十分な予測の効率を達成するのに十分な柔軟性をもたらさないことがある。現在のビューと参照ビューとの品質の差は、特にスケーラブルビデオコーディングに関しては、動的であり得る。逆に、3つの重み付けファクタは、いくつかのスライスまたはピクチャにより必要とされるものを超えることがある。すなわち、いくつかのスライスまたはピクチャは、複雑さとコーディング効率の改善との間の最適なバランスを達成するために、3つの重み付けファクタから選択する必要はないことがある。

20

【0131】

[0148]本開示の態様によれば、重み付けファクタに対するより柔軟な手法が実施され得る。たとえば、利用可能な重み付けファクタの数は、（たとえば、シーケンスパラメータセット（SPS）のようなパラメータセット中の）シーケンスレベルで変更され得る。例示を目的とするある例では、たとえば0.5および／または1の1つまたは複数の重み付けファクタをディセーブルにするためのインジケータが、SPS中でシグナリングされ得る。別の例では、そのようなインジケータは、VP S中でシグナリングされ、すべての非ベースビューに対して適用可能であってよい。さらに別の例では、そのようなインジケータは、各々の非ベースビューに対してビデオパラメータセット（VP S）拡張においてシグナリングされ得る。別の例では、そのようなインジケータは、1つまたは複数の重み付けファクタをディセーブルにするために、ピクチャパラメータセット（PPS）、スライスヘッダ、またはビューパラメータセットにおいて提供され得る。重み付けファクタがディセーブルにされているとき、残りの重み付けファクタを表すためにより少数のビットが使用されてよく、これによってビットを節約する。

30

【0132】

[0149]他の態様によれば、1つまたは複数の重み付けファクタを修正および／または置換するための、インジケータが提供され得る。ある例では、ビデオコーダは、0.5という重み付けファクタを0.75という重み付けファクタで置換することができる。このインジケータは、スライスヘッダ、SPS、ピクチャパラメータセット（PPS）、またはVP Sでシグナリングされ得る。

40

【0133】

[0150]上で述べられたように、残差予測子を決定するための時間的視差参照ブロックは通常、時間的動きベクトルを視差参照ブロックに適用することによって位置決定される。すなわち、ビデオコーダは、時間的動きベクトルと視差ベクトルを組み合わせ、たとえば現在のブロックに対して、組合せに基づいて時間的視差参照ブロックを位置決定することができる。しかしながら、いくつかの例では、復号ピクチャバッファおよび／または現

50

在のブロックをコーディングするための参照ピクチャリストは、時間的動きベクトルを視差参照ブロックに適用することによって示されるピクチャを含まないことがある。

【0134】

【0151】本開示の態様によれば、ビデオコーダは、復号ピクチャバッファおよび／または参照ピクチャリストのピクチャに基づいて、ARPをイネーブルまたはディセーブルにすることができる。たとえば、ビデオエンコーダ20および／またはビデオデコーダ30は、ビデオデータの第1のレイヤの中のビデオデータの第1のブロックに対して、第1のブロックを予測するための時間的動きベクトルと関連付けられる時間的参照ピクチャとを決定することができ、時間的参照ピクチャは、ピクチャ順序カウンタ値を有する。加えて、ビデオエンコーダ20および／またはビデオデコーダ30は、第1のブロックを含むピクチャを含むアクセスユニットのピクチャ中の視差参照ブロックを決定することができる。ビデオエンコーダ20および／またはビデオデコーダ30は、時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを、復号ピクチャバッファが含むかどうかを決定することができ、ここで、時間的視差参照ピクチャは時間的動きベクトルと視差ベクトルの組合せに基づいて位置決定され、復号ピクチャバッファが時間的参照ピクチャのピクチャ順序カウンタ値を有する時間的視差参照ピクチャを含まないとき、ビデオエンコーダ20および／またはビデオデコーダ30は、第1のブロックの残差データを予測するためのビュー間残差予測処理を修正することができる。

【0135】

【0152】いくつかの例では、ビデオエンコーダ20および／またはビデオデコーダ30は、現在のブロックがビュー間残差予測を使用してコーディングされないようにビュー間残差予測処理をディセーブルにすることによって、ビュー間残差予測処理を修正することができる。他の例では、ビデオエンコーダ20および／またはビデオデコーダ30は、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、ビュー間残差予測処理を修正することができる。たとえば、ビデオエンコーダ20および／またはビデオデコーダ30は、スケーリングされた動きベクトルが、視差参照ピクチャに適用されると（たとえば、または、視差ベクトルと組み合わせられると）、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にある時間的視差参照ピクチャを識別するように、時間的動きベクトルをスケーリングすることができる。上で説明された技法は、参照ピクチャリストに含まれないピクチャ中の視差参照ブロックをビデオエンコーダ20および／またはビデオデコーダ30が位置決定しようとするのを防ぐことができる。

【0136】

【0153】図2は、高度な残差予測のための本開示で説明され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、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ピクチャ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（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し得る。単方向予測（Pモード）または双方向予測（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。

【0137】

【0154】上で述べられたように、ビデオエンコーダ20は、マルチビュービデオコーディングを実行するように適合され得る。たとえば、ビデオエンコーダ20は、MVC、MV-HEVC、3D-HEVC、および／またはH SVCビデオコーディング規格に従って、ビデオデータの複数のスケーラブルレイヤを符号化するように構成され得る。したがって、ビデオエンコーダ20は、時間インスタンス中の各ビューがビデオデコーダ30のようなデコーダによって処理され得るように、MV-HEVCをコーディングするように構成され得る。HEVC-3Dでは、各ビューに対するテクスチャマップ（すなわち、ルーマ

値およびクロマ値)を符号化することに加えて、ビデオエンコーダ20はさらに、各ビューに対する深度マップを符号化することができる。

【0138】

[0155]いずれの場合でも、図2に示されるように、ビデオエンコーダ20は、符号化されるべきビデオデータを受信する。図2の例では、ビデオエンコーダ20は、モード選択ユニット40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、参照ピクチャメモリ64とを含む。次に、モード選択ユニット40は、動き推定ユニット42と、動き補償ユニット44と、イントラ予測ユニット46と、区分ユニット48とを含む。ビデオブロックの再構築のために、ビデオエンコーダ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。再構築されたビデオからブロックネサアティファクトを除去するためにブロック境界をフィルタリングするための、デブロッキングフィルタ(図2に図示されず)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通常、加算器62の出力をフィルタリングする。デブロッキングフィルタに加えて、追加のループフィルタ(ループ内またはループ後)も使用され得る。そのようなフィルタは、簡潔にするために示されていないが、所望される場合、(ループ内フィルタとして)加算器50の出力をフィルタリングし得る。

10

【0139】

[0156]符号化処理中に、ビデオエンコーダ20は、コーディングされるべきピクチャまたはスライスを受信する。ピクチャまたはスライスは複数のビデオブロックに分割され得る。動き推定ユニット42および動き補償ユニット44は、時間圧縮を行うために、1つまたは複数の参照ピクチャ中の1つまたは複数のブロックに対する受信されたビデオブロックのイントラ予測コーディングを実行する。イントラ予測ユニット46は、代替的に、空間圧縮を行うために、コーディングされるべきブロックと同じピクチャ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受信されたビデオブロックのイントラ予測コーディングを実行することができる。ビデオエンコーダ20は、たとえば、ビデオデータのブロックごとに適切なコーディングモードを選択するために、複数のコーディングパスを実行することができる。

20

【0140】

[0157]その上、区分ユニット48は、以前のコーディングパスにおける以前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分することができる。たとえば、区分ユニット48は、最初にピクチャまたはスライスをLCUに区分し、レートひずみ分析(たとえば、レートひずみ最適化)に基づいてLCUの各々を複数のサブCUに区分することができる。モード選択ユニット40はさらに、LCUを複数のサブCUに区分することを示す4分木データ構造を生成することができる。4分木のリーフノードCUは、1つまたは複数のPUと1つまたは複数のTUとを含み得る。

30

【0141】

[0158]モード選択ユニット40は、たとえば、誤差結果に基づいて、コーディングモード、すなわちイントラまたはインターのうちの1つを選択し、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、加算器50に提供して残差ブロックデータを生成し、かつ加算器62に提供して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築することができる。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのようなシンタックス情報のようなシンタックス要素を、エントロピー符号化ユニット56に与える。

40

【0142】

[0159]動き推定ユニット42、レイヤ間予測ユニット43、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されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトルを生成する処理である。動きベクトルは、たとえば、現在のピクチャ(または他のコーディングされたユニット)内のコーディングされている現在のブロックに対する参照ピクチャ

50

ャ（または他のコーディングされたユニット）内の予測ブロックに対する、現在のピクチャ内のビデオブロックのPUの変位を示すことができる。

【0143】

[0160]予測ブロックは、絶対値差分和（SAD）、2乗差分和（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ブロックと厳密に一致することが判明しているブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャバッファとも呼ばれ得る参照ピクチャメモリ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することができる。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、または他の小数ピクセル位置の値を補間することができる。したがって、動き推定ユニット42は、フルピクセル位置と小数ピクセル位置とに対する動き探索を実行し、小数ピクセル精度で動きベクトルを出力することができる。

10

【0144】

[0161]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、インターコーディングされたスライス中のビデオブロックのPUのための動きベクトルを計算する。したがって、一般に、動きベクトルのためのデータは、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（ref_idx）と、水平成分と、垂直成分とを含み得る。参照ピクチャは、第1の参照ピクチャリスト（リスト0）、第2の参照ピクチャリスト（リスト1）、または組み合わされた参照ピクチャリスト（リストc）から選択されてよく、それらの各々が、参照ピクチャメモリ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。

20

【0145】

[0162]動き推定ユニット42は、参照ピクチャの予測ブロックを識別する動きベクトルを生成し、エントロピー符号化ユニット56と動き補償ユニット44とに送ることができる。すなわち、動き推定ユニット42は、予測ブロックを含んでいる参照ピクチャリストを識別する動きベクトルデータと、予測ブロックのピクチャを識別する参照ピクチャリストへのインデックスと、識別されたピクチャ内の予測ブロックを位置決定するための水平成分および垂直成分とを生成し、送ることができる。

【0146】

[0163]いくつかの例では、現在のPUに対する実際の動きベクトルを送るのではなく、レイヤ間予測ユニット43は、動きベクトルを通信するために必要とされるデータの量をさらに低減するために、動きベクトルを予測し得る。この場合、動きベクトル自体を符号化および通信するのではなく、レイヤ間予測ユニット43は、既知の（または知り得る）動きベクトルに対する動きベクトル差分（MVD）を生成することができる。現在の動きベクトルを定義するためにMVDとともに使用され得る既知の動きベクトルは、いわゆる動きベクトル予測子（MVP）によって定義され得る。一般に、有効なMVPであるために、予測のために使用されている動きベクトルは、現在コーディングされている動きベクトルと同じ参照ピクチャを指さなければならない。

30

【0147】

[0164]レイヤ間予測ユニット43は、マルチビューコーディングにおいて、たとえば、MVDの生成または統合のために、動きベクトル予測子を識別することができる。たとえば、レイヤ間予測ユニット43は、現在のブロックに対する動きベクトルを予測するために、現在のブロックとは異なるビュー成分中のブロックから視差動きベクトルを識別することができる。他の例では、レイヤ間予測ユニット43は、現在のブロックに対する動きベクトルを予測するために、現在のブロックとは異なるビュー成分中のブロックから時間的動きベクトルを識別することができる。

40

【0148】

[0165]本開示の態様によれば、レイヤ間予測ユニット43は、レイヤ間残差予測を実行することができる。たとえば、レイヤ間予測ユニット43は、あるレイヤの残差データを、第2の異なるレイヤの残差データに対してコーディングすることができる。いくつかの

50

例では、レイヤ間予測ユニット43はまず、現在のブロックを予測するための予測ブロックを決定することができる。現在のブロックの予測ブロックは、現在のブロックの動きベクトルによって示される位置と関連付けられる、時間的参照ピクチャのサンプルに基づき得る。時間的参照ピクチャは、現在のピクチャと同じレイヤと関連付けられるが、現在のピクチャとは異なる時間インスタンスと関連付けられる。

【0149】

[0166]レイヤ間予測ユニット43はまた、現在のブロックの視差ベクトルによって示される位置にある視差参照ピクチャのサンプルに基づいて、視差参照ブロックを決定する。視差参照ピクチャは、現在のピクチャとは異なるレイヤ（すなわち、参照レイヤ）と関連付けられるが、現在のピクチャと同じ時間インスタンスと関連付けられる。レイヤ間予測ユニット43はまた、現在のブロックの時間的視差参照ブロックを決定する。時間的参照ブロックは、現在のブロックの動きベクトルおよび視差ベクトルによって（たとえば、動きベクトルと視差ベクトルの組合せによって）示される位置と関連付けられる時間的視差参照ピクチャのサンプルに基づく。したがって、時間的視差参照ピクチャは、視差参照ピクチャと同じビューと関連付けられ、時間的参照ピクチャと同じアクセスユニットと関連付けられる。

10

【0150】

[0167]レイヤ間予測ユニット43は次いで、現在のブロックと関連付けられる残差、たとえば、現在のブロックと時間的参照ブロックとの差を予測するための、残差予測子を決

20

【0151】

[0168]レイヤ間予測ユニット43は、現在のブロックに対する最終的な残差ブロックを決定することができる。最終的な残差ブロックは、現在のブロックのサンプルと、時間的予測ブロック中のサンプルと、残差予測子中のサンプルとの差を示すサンプルを備える。ビデオエンコーダ20は、ビットストリーム中に、最終的な残差ブロックを表すデータを含め得る。

【0152】

30

[0169]本開示の態様によれば、レイヤ間予測ユニット43は、現在コーディングされているブロックに対する参照ピクチャリスト中の参照ピクチャに基づいて、ビュー間残差予測（あるレイヤの残差を第2の異なるレイヤの残差に対してコーディングすることを含む）をイネーブルまたはディセーブルにすることができる。ある例では、レイヤ間予測ユニット43は、現在コーディングされているブロックに対する参照ピクチャリスト（たとえば、RefPicList0またはRefPicList1）が任意の時間的参照ピクチャを含むかどうかに基づいて、ビュー間残差予測をイネーブルまたはディセーブルにすることができる。本開示の態様によれば、インター予測されたブロックに対する参照ピクチャリストがビュー間参照ピクチャのみを含む場合、レイヤ間予測ユニット43は、レイヤ間予測ユニット43をディセーブルにすることができる。いくつかの例では、レイヤ間予測ユニット43は、ランダムアクセスビュー成分の各ブロックに対して、レイヤ間予測ユニット43をディセーブルにすることができる。

40

【0153】

[0170]別の例では、視差参照ブロックの参照ピクチャリストが、時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中の参照ピクチャを含まないとき、レイヤ間予測ユニット43はビュー間残差予測を修正することができる。ビュー間残差予測を修正するかどうかの決定は、参照ピクチャリスト（たとえば、RefPicList0および／またはRefPicList1）の一方または両方に基づき得る。すなわち、すなわち、現在の参照ピクチャリストのインデックスがXであるとする（Xは0または1である）、一例では、視差参照ブロックのXに等しいリストインデックスを伴う参照ピク

50

チャリストが視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じPOCを有する参照ピクチャを含まない場合、レイヤ間予測ユニット43はARP処理を修正することができる。別の例では、視差参照ブロックの参照ピクチャリストのいずれもが（たとえば、リスト0もリスト1も）、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じPOCを有する参照ピクチャを含まない場合、レイヤ間予測ユニット43はARP処理を修正することができる。

【0154】

[0171]いくつかの例では、レイヤ間予測ユニット43は、ビュー間残差予測をディセーブルにすることによって、ビュー間残差予測を修正することができる。他の例では、レイヤ間予測ユニット43は、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、ビュー間残差予測処理を修正することができる。たとえば、レイヤ間予測ユニット43は、動きベクトルと視差ベクトルのスケーリングされた組合せが、視差参照ピクチャに適用されると、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にある時間的視差参照ピクチャを識別するように、時間的動きベクトルをスケーリングすることができる。

10

【0155】

[0172]参照ピクチャリストに関して説明されるが、レイヤ間予測ユニット43は、加えて、または代替的に、参照ピクチャメモリ64（すなわち、復号ピクチャバッファ）が時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中のピクチャを含まない場合、ビュー間残差予測を修正および／またはディセーブル化することができる。

20

【0156】

[0173]さらに別の例では、本開示の態様によれば、レイヤ間予測ユニット43は、特にサブペル位置を補間するときに参照ブロックが位置決定される方式を簡略化することができる。たとえば、レイヤ間予測ユニット43は、双線形フィルタのようなローパスフィルタを使用して、視差参照ブロックの位置を補間することができる。加えて、または代替的に、レイヤ間予測ユニット43は、双線形フィルタのようなローパスフィルタを使用して、時間的視差参照ブロックの位置を補間することができる。さらに別の例では、本開示の態様によれば、動き推定ユニット42および／または動き補償ユニット44は、双線形フィルタのようなローパスフィルタを使用して、時間的参照ブロックの位置を補間することができる。

30

【0157】

[0174]さらに別の例では、本開示の態様によれば、レイヤ間予測ユニット43は、ビュー間残差予測のみを適用することができるので、特定のコーディングモードおよび／または区分モードだけに対して、重み付けファクタをシグナリングすることができる。たとえば、レイヤ間予測ユニット43は、PART_2 N×2 Nに等しくない区分モードを伴う任意のインターコーディングされるブロックだけに対して、重み付けファクタをシグナリングすることができる。別の例では、加えて、または代替的に、レイヤ間予測ユニット43は、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタを、シグナリングしなくてよい。

40

【0158】

[0175]動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定された動きベクトルおよび／またはレイヤ間予測ユニット43からの情報に基づいて、予測ブロックをフェッチすることまたは生成することを伴い得る。動き補償ユニット44は、いくつかの例では、ビュー間予測を適用することができる。やはり、動き推定ユニット42、レイヤ間予測ユニット43、および動き補償ユニット44は、いくつかの例では、機能的に統合され得る。現在のビデオブロックのPU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を位置決定することができる。

【0159】

50

[0176]加算器50は、以下で論じられるように、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42はルーマ成分に対して動き推定を実行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方のためにルーマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40はまた、ビデオスライスのビデオブロックを復号する際にビデオデコーダ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付けられる、シンタックス要素を生成することができる。

【0160】

[0177]イントラ予測ユニット46は、上で説明されたように、動き推定ユニット42と動き補償ユニット44とによって実行されるインター予測の代替として、現在のブロックをイントラ予測することができる。特に、イントラ予測ユニット46は、現在のブロックを符号化するために使用するようイントラ予測モードを決定することができる。いくつかの例では、イントラ予測ユニット46は、たとえば、別個の符号化パスの間に、様々なイントラ予測モードを使用して現在のブロックを符号化することができ、イントラ予測ユニット46（または、いくつかの例では、モード選択ユニット40）は、使用するのに適したイントラ予測モードをテストされたモードから選択することができる。

【0161】

[0178]たとえば、イントラ予測ユニット46は、様々なテストされたイントラ予測モードに対するレートひずみ分析を使用してレートひずみ値を計算し、テストされたモードの中で最良のレートひずみ特性を有するイントラ予測モードを選択することができる。レートひずみ分析は、一般に、符号化されたブロックと、符号化されたブロックを生成するために符号化された元の符号化されていないブロックとの間のひずみ（または誤差）の量、ならびに符号化されたブロックを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46は、どのイントラ予測モードがブロックに対し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することができる。

【0162】

[0179]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に与えることができる。エントロピー符号化ユニット56は、選択されたイントラ予測モードを示す情報を符号化することができる。ビデオエンコーダ20は、送信される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修正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対して使用する、最確イントラ予測モード、イントラ予測モードインデックステーブル、および修正されたイントラ予測モードインデックステーブルの指示とを含み得る、構成データを含め得る。

【0163】

[0180]ビデオエンコーダ20は、コーディングされている元のビデオブロックから、モード選択ユニット40からの予測データ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。加算器50は、この減算演算を実行する1つまたは複数のコンポーネントを表す。変換処理ユニット52は、離散コサイン変換（DCT）または概念的に同様の変換などの変換を残差ブロックに適用し、残差変換係数値を備えるビデオブロックを生成する。変換処理ユニット52は、DCTと概念的に同様である他の変換を実行することができる。ウェーブレット変換、整数変換、サブバンド変換または他のタイプの変換も使用され得る。いずれの場合も、変換処理ユニット52は、変換を残差ブロックに適用し、残差変換係数のブロックを生成する。変換は、残差情報を、ピクセル値領域から周波数領域などの変換領域に変換することができる。

【0164】

[0181]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に送ることができる。量子化ユニット54は、ビットレートをさらに低減するために変換係数を量子化する。量子化処理は、係数の一部またはすべてと関連付けられるビット深度を低減することができる。量子化の程度は、量子化パラメータを調整することによって修正され得る。いくつかの例では、量子化ユニット54は、次いで、量子化された変換係数を含む行列のスキャンを実行することができる。代替的に、エントロピー符号化ユニット56がスキャンを実行してよい。

【0165】

[0182]量子化の後、エントロピー符号化ユニット56は、量子化された変換係数をエントロピーコーディングする。たとえば、エントロピー符号化ユニット56は、コンテキスト適応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)、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コーディング技法を実行することができる。

【0166】

[0183]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、たとえば、参照ブロックとして後で使用するために、ピクセル領域中で残差ブロックを再構築する。動き補償ユニット44は、残差ブロックを参照ピクチャメモリ64のピクチャのうちの1つの予測ブロックに加算することによって、参照ブロックを計算することができる。動き補償ユニット44はまた、再構築された残差ブロックに1つまたは複数の補間フィルタを適用して、動き推定において使用するサブ整数ピクセル値を計算することができる。

【0167】

[0184]加算器62は、再構築された残差ブロックを、動き補償ユニット44によって生成された動き補償予測ブロックに加算して、参照ピクチャメモリ64に記憶するための再構築されたビデオブロックを生成する。再構築されたビデオブロックは、後続のピクチャ中のブロックをインターコーディングするための参照ブロックとして、動き推定ユニット42および動き補償ユニット44によって使用され得る。

【0168】

[0185]図3は、マルチビューコーディングにおいて動きベクトルを予測するための本開示で説明され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。図3の例では、ビデオデコーダ30は、エントロピー復号ユニット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、参照ピクチャメモリ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、レイヤ間予測ユニット83と、イントラ予測ユニット84とを含む。

【0169】

[0186]上で述べられたように、ビデオデコーダ30は、マルチビュービデオコーディングを実行するように適合され得る。いくつかの例では、ビデオデコーダ30は、マルチビューHEVCを復号するように構成され得る。HEVC-3Dでは、各ビューに対するテクスチャマップ(すなわち、ルーマ値およびクロマ値)を復号することに加えて、ビデオデコーダ30はさらに、各ビューに対する深度マップを復号することができる。

【0170】

[0187]いずれにしても、復号処理の間に、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のビデオブロックと、関連付けられ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のエントロピー復号ユニット80は、量子化された係数と、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することができる。

【0171】

[0188]たとえば、背景として、ビデオデコーダ30は、ネットワークを介した送信のために、いわゆる「ネットワーク抽象化レイヤユニット」またはNALユニットに圧縮された、圧縮ビデオデータを受信することができる。各NALユニットは、NALユニットに記憶されるデータのタイプを識別するヘッダを含み得る。一般にNALユニットに記憶される、2つのタイプのデータがある。NALユニットに記憶される第1のタイプのデータはビデオコーディングレイヤ（VCL）データであり、これは圧縮ビデオデータを含む。NALユニットに記憶される第2のタイプのデータは非VCLデータと呼ばれ、これは、多数のNALユニットに共通のヘッダデータを定義するパラメータセットなどの追加の情報と、補足エンハンスメント情報（SEI）とを含む。

10

【0172】

[0189]たとえば、パラメータセットは、（たとえば、シーケンスパラメータセット（SPS）中の）シーケンスレベルヘッダ情報と、（たとえば、ピクチャパラメータセット（PPS）中の）まれに変化するピクチャレベルヘッダ情報とを含み得る。パラメータセット中に含まれている、まれに変化する情報は、シーケンスまたはピクチャごとに繰り返される必要がなく、それによりコーディング効率が改善される。加えて、パラメータセットの使用はヘッダ情報の帯域外送信を可能にし、それにより誤り耐性のための冗長送信の必要をなくす。

【0173】

[0190]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81のイントラ予測ユニット84は、シグナリングされたイントラ予測モードと、現在のピクチャの以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に対する予測データを生成することができる。ピクチャ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と他のシンタックス要素とに基づいて、現在のビデオスライスのビデオブロックの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つから生成される。ビデオデコーダ30は、参照ピクチャメモリ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、参照ピクチャリストと、リスト0と、リスト1と

20

30

【0174】

[0191]動き補償ユニット82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対する予測情報を決定し、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に対する予測ブロックを生成する。たとえば、動き補償ユニット82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築情報と、スライスの各々のインター符号化されたビデオブロックのための動きベクトルと、スライスの各々のインターコーディングされた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。いくつかの例では、動き補償ユニット82は、レイヤ間予測ユニット83からある動き情報を受信することができる。

40

【0175】

[0192]レイヤ間予測ユニット83は、現在のブロックのための動き情報を取り出す場所を示す予測データを受信することができる。たとえば、レイヤ間予測ユニット83は、MVPインデックス（mvp_flag）、MVD、統合フラグ（merge_flag）、および／または統合インデックス（merge_idx）などの動きベクトル予測情報を受信し、そのような情報を使用して、現在のブロックを予測するために使用される動き

50

情報を識別することができる。すなわち、ビデオエンコーダ20に関して上で述べられたように、本開示の態様によれば、レイヤ間予測ユニット83は、MVPインデックス(mvp_flag)とMVDとを受信し、そのような情報を使用して、現在のブロックを予測するために使用される動きベクトルを決定することができる。レイヤ間予測ユニット83は、MVPまたは統合候補のリストを生成することができる。MVPおよび／または統合候補は、現在復号されているビデオブロックとは異なるビューの中に位置する1つまたは複数のビデオブロックを含み得る。

【0176】

[0193]本開示の態様によれば、レイヤ間予測ユニット83は、レイヤ間残差予測を実行することができる。たとえば、レイヤ間予測ユニット83は、あるレイヤの残差データを、第2の異なるレイヤの残差データに対してコーディングすることができる。いくつかの例では、レイヤ間予測ユニット83はまず、現在のブロックを予測するための予測ブロックを決定することができる。現在のブロックの予測ブロックは、現在のブロックの動きベクトルによって示される位置と関連付けられる、時間的参照ピクチャのサンプルに基づき得る。時間的参照ピクチャは、現在のピクチャと同じレイヤと関連付けられるが、現在のピクチャとは異なる時間インスタンスと関連付けられる。

10

【0177】

[0194]レイヤ間予測ユニット83はまた、現在のブロックの視差ベクトルによって示される位置にある視差参照ピクチャのサンプルに基づいて、視差参照ブロックを決定する。視差参照ピクチャは、現在のピクチャとは異なるレイヤ(すなわち、参照レイヤ)と関連付けられるが、現在のピクチャと同じ時間インスタンスと関連付けられる。レイヤ間予測ユニット83はまた、現在のブロックの時間的視差参照ブロックを決定する。時間的参照ブロックは、現在のブロックの動きベクトルおよび視差ベクトルによって(たとえば、動きベクトルと視差ベクトルの組合せによって)示される位置と関連付けられる時間的視差参照ピクチャのサンプルに基づく。したがって、時間的視差参照ピクチャは、視差参照ピクチャと同じビューと関連付けられ、時間的参照ピクチャと同じアクセスユニットと関連付けられる。

20

【0178】

[0195]レイヤ間予測ユニット83は次いで、現在のブロックと関連付けられる残差、たとえば、現在のブロックと時間的参照ブロックとの差を予測するための、残差予測子を決

30

【0179】

[0196]レイヤ間予測ユニット83は、符号化されたビットストリームから、現在のブロックに対する最終的な残差ブロックを示すデータを取得することができる。レイヤ間予測ユニット83は、最終的な残差ブロックと、時間的予測ブロックと、残差予測子の中のサンプルとを組み合わせることによって、現在のブロックを再構築することができる。

【0180】

40

[0197]本開示の態様によれば、レイヤ間予測ユニット83は、現在コーディングされているブロックに対する参照ピクチャリスト中の参照ピクチャに基づいて、ビュー間残差予測(あるレイヤの残差を第2の異なるレイヤの残差に対してコーディングすることを含む)をイネーブルまたはディセーブルにすることができる。ある例では、レイヤ間予測ユニット83は、現在コーディングされているブロックに対する参照ピクチャリストが任意の時間的参照ピクチャを含むかどうかに基づいて、ビュー間残差予測をイネーブルまたはディセーブルにすることができる。本開示の態様によれば、インター予測されたブロックに対する参照ピクチャリストがビュー間参照ピクチャのみを含む場合、レイヤ間予測ユニット83は、レイヤ間予測ユニット83をディセーブルにすることができる。いくつかの例では、レイヤ間予測ユニット83は、ランダムアクセスビュー成分の各ブロックに対して

50

、レイヤ間予測ユニット 83 をディセーブルにすることができる。

【0181】

[0198]別の例では、視差参照ブロックの参照ピクチャリストが、時間的参照ピクチャと同じ POC を有する視差参照ピクチャと同じビュー中の参照ピクチャを含まないとき、レイヤ間予測ユニット 83 はビュー間残差予測を修正することができる。ビュー間残差予測を修正するかどうかの決定は、参照ピクチャリスト（たとえば、RefPicList0 および／または RefPicList1）の一方または両方に基づき得る。すなわち、すなわち、現在の参照ピクチャリストのインデックスが X であるとする（X は 0 または 1 である）、一例では、視差参照ブロックの X に等しいリストインデックスを伴う参照ピクチャリストが、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じ POC を有する参照ピクチャを含まない場合、レイヤ間予測ユニット 83 は A R P 処理を修正することができる。別の例では、視差参照ブロックの参照ピクチャリストのいずれもが（たとえば、リスト 0 もリスト 1 も）、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じ POC を有する参照ピクチャを含まない場合、レイヤ間予測ユニット 83 は A R P 処理を修正することができる。

10

【0182】

[0199]いくつかの例では、レイヤ間予測ユニット 83 は、ビュー間残差予測をディセーブルにすることによって、ビュー間残差予測を修正することができる。他の例では、レイヤ間予測ユニット 83 は、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、ビュー間残差予測処理を修正することができる。たとえば、レイヤ間予測ユニット 83 は、動きベクトルと視差ベクトルのスケーリングされた組合せが、視差参照ピクチャに適用されると、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にある時間的視差参照ピクチャを識別するように、時間的動きベクトルをスケーリングすることができる。

20

【0183】

[0200]さらに別の例では、本開示の態様によれば、レイヤ間予測ユニット 83 は、特にサブペル位置を補間するときに参照ブロックが位置決定される方式を簡略化することができる。たとえば、レイヤ間予測ユニット 83 は、双線形フィルタのようなローパスフィルタを使用して、視差参照ブロックの位置を補間することができる。加えて、または代替的に、レイヤ間予測ユニット 83 は、双線形フィルタのようなローパスフィルタを使用して、時間的視差参照ブロックの位置を補間することができる。さらに別の例では、本開示の態様によれば、動き補償ユニット 82 は、双線形フィルタのようなローパスフィルタを使用して、時間的参照ブロックの位置を補間することができる。

30

【0184】

[0201]さらに別の例では、本開示の態様によれば、レイヤ間予測ユニット 83 は、ビュー間残差予測のみを適用することができるので、特定のコーディングモードおよび／または区分モードだけに対して、重み付けファクタをシグナリングすることができる。たとえば、レイヤ間予測ユニット 83 は、PART_2 N×2 N に等しくない区分モードを伴う任意のインターコーディングされたブロックだけに対して、重み付けファクタをシグナリングすることができる。別の例では、加えて、または代替的に、レイヤ間予測ユニット 83 は、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタを、シグナリングしなくてよい。

40

【0185】

[0202]逆量子化ユニット 86 は、ビットストリーム中で提供され、エントロピー復号ユニット 80 によって復号された、量子化された変換係数を逆量子化 (inverse quantize) 、すなわち、逆量子化 (de-quantize) する。逆量子化処理は、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して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ことを含み得る。

50

【0186】

[0203] 逆変換処理ユニット88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処理を変換係数に適用する。本開示の態様によれば、逆変換処理ユニット88は、変換が残差データに適用された方式を決定することができる。すなわち、たとえば、逆変換処理ユニット88は、受信されたビデオデータのブロックと関連付けられる残差ルーマサンプルと残差クロマサンプルとに、変換（たとえば、DCT、整数変換、ウェーブレット変換、または1つもしくは複数の他の変換）が適用された方式を表すRQTを決定することができる。

【0187】

[0204] 動き補償ユニット82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88からの残差ブロックを動き補償ユニット82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。加算器90は、この加算演算を実行する1つまたは複数のコンポーネントを表す。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リングするためのデブロック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、または別様にビデオ品質を改善するために、他のループフィルタも（コーディンググループ中またはコーディンググループ後のいずれかで）使用され得る。次いで、所与のピクチャ中の復号されたビデオブロックは、その後の動き補償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92に記憶される。参照ピクチャメモリ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に後で提示するために、復号されたビデオを記憶する。

【0188】

[0205] 図4は、例示的なマルチビュー復号順序を示す概念図である。マルチビュー復号順序はビットストリームの順序であり得る。図4の例では、各正方形がビュー成分に対応する。正方形の列は、アクセスユニットに対応する。各アクセスユニットは、時間インスタンス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得る。正方形の行は、ビューに対応する。図4の例では、アクセスユニットがT0～T11と標識され、ビューがS0～S7と標識される。アクセスユニットの各ビュー成分は次のアクセスユニットの任意のビュー成分の前に復号されるので、図4の復号順序は時間優先コーディングと呼ばれ得る。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一ではないことがある。

【0189】

[0206] マルチビューコーディングはビュー間予測をサポートすることができる。ビュー間予測は、H.264/AVC、HEVC、または他のビデオコーディング仕様において使用されるインター予測と同様であり、同じシンタックス要素を使用し得る。しかしながら、ビデオコードが（マクロブロックまたはPUのような）現在のビデオユニットに対してビュー間予測を実行するとき、ビデオコードは、参照ピクチャとして、現在のビデオユニットと同じアクセスユニット中にあるが異なるビュー中にあるピクチャを使用することができる。対照的に、従来のインター予測は、参照ピクチャとして異なるアクセスユニット中のピクチャのみを使用する。

【0190】

[0207] 図5は、MVC、マルチビューHEVC、および3D-HEVC（マルチビュープラス深度）とともに使用され得る例示的なMVC予測パターンを示す概念図である。以下でのMVCへの言及は全般にMVCに当てはまり、H.264/MVCには限定されない。

【0191】

[0208] 図5の例では、8個のビュー（S0～S7）が示され、ビューごとに12個の時間的位置（T0～T11）が示される。一般に、図5の各行はビューに対応し、各列は時間的位置を示す。ビューの各々は、他のビューに対する相対的なカメラ位置を示すために

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5に示された例では、ビューIDは「S0」～「S7」として示されているが、数字のビューIDが使用されることもある。加えて、時間的位置の各々は、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5に示された例では、POC値は「T0」～「T11」として示されている。

【0192】

[0209]マルチビューコーディングされたビットストリームは、特定のデコーダによって復号可能である、いわゆる基本ビューを有してよく、ステレオビューペアがサポートされ得るが、いくつかのマルチビュービットストリームは、3Dビデオ入力として3つ以上のビューをサポートすることができる。したがって、特定のデコーダを有するクライアントのレンダラは、複数のビューを伴う3Dビデオコンテンツを予想することができる。

10

【0193】

[0210]図5のピクチャは、対応するピクチャがイントラコーディングされる（すなわち、Iフレームである）か、または一方向に（すなわち、Pフレームとして）インターコーディングされるか、または複数の方向に（すなわち、Bフレームとして）インターコーディングされるかを指定する、文字を含む影付きブロックを使用して示される。一般に、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、時間的位置T0にあるビューS2のPフレームは、時間的位置T0にあるビューS0のIフレームから予測される。

【0194】

20

[0211]シングルビュービデオの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ピクチャは、異なる時間的位置におけるピクチャに関して予測的に符号化され得る。たとえば、時間的位置T1におけるビューS0のbフレームは、時間的位置T0におけるビューS0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、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、加えて、マルチビュービデオの符号化の状況において、ピクチャはビュー間予測され得る。すなわち、ビュー成分は、参照のために他のビュー中のビュー成分を使用することができる。たとえば、別のビュー中のビュー成分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得る。可能性のあるビュー間参照は、シーケンスパラメータセット（SPS）MVC拡張においてシグナリングされてよく、インター予測またはビュー間予測の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処理によって修正され得る。

30

【0195】

[0212]図5は、ビュー間予測の様々な例を提供する。図5の例では、ビューS1のピクチャは、ビューS1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、かつ同じ時間的位置にあるビューS0およびビューS2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、時間的位置T1にあるビューS1のbフレームは、時間的位置T0およびT2にあるビューS1のBフレームの各々、ならびに時間的位置T1にあるビューS0およびビューS2のbフレームから予測される。

【0196】

[0213]図5の例では、大文字「B」および小文字「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ことが意図される。一般に、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5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5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。

40

【0197】

[0214]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

50

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成分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などのパラメータセット中で示唆され得る。

【0198】

[0215]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成分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 i ごとに、対応する $view_id$ がシグナリングされる。ビュー成分の復号は、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る場合、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。

10

【0199】

[0216]準拠するサブビットストリームを形成するために、ビットストリーム全体のサブセットが抽出され得る。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ことがあり、2 つの状況があり得る。一例では、あるクライアントは滑らかなビュー体験を必要とすることがあり、 $view_id$ 値 $S0$ 、 $S1$ 、および $S2$ のビューを選好することがあり、一方、別のクライアントはビュースケラビリティを必要とし、 $view_id$ 値 $S0$ 、 $S2$ 、および $S4$ のビューを選好することがある。これらのサブビットストリームの両方が、独立したビットストリームとして復号され得るとともに、同時にサポートされ得ることに留意されたい。

20

【0200】

[0217]ビュー間予測に関して、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される。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、参照ピクチャリストの任意の位置に置かれ得る。

30

【0201】

[0218]したがって、マルチビュービデオコーディングの状況では、2 種類の動きベクトルが存在する。動きベクトルの 1 つの種類は、時間的参照ピクチャを指す通常の動きベクトルである。通常の時間的動きベクトルに対応するインター予測のタイプは、動き補償された予測 (MCP) と呼ばれ得る。ビュー間予測参照ピクチャが動き補償のために使用されるとき、対応する動きベクトルは「視差動きベクトル」と呼ばれる。言い換えると、視差動きベクトルは、異なるビュー中のピクチャ（すなわち、視差参照ピクチャまたはビュー間参照ピクチャ）を指す。視差動きベクトルに対応するインター予測のタイプは、「視差補償された予測」または「DCP」と呼ばれ得る。

40

【0202】

[0219]上で言及されたように、HEVC のマルチビュー拡張（すなわち、MV-HEVC）および HEVC の 3DV 拡張（すなわち、3D-HEVC）が開発中である。MV-HEVC および 3D-HEVC は、ビュー間動き予測とビュー間残差予測とを使用して、コーディング効率を改善することができる。ビュー間動き予測では、ビデオコードは、現在の PU とは異なるビュー中の PU の動き情報に基づいて、現在の PU の動き情報を決定する（すなわち、予測する）ことができる。ビュー間残差予測では、ビデオコードは、図 5 に示される予測構造を使用して、現在の CU とは異なるビュー中の残差データに基づいて、現在の CU の残差ブロックを決定することができる。

50

【0203】

[0220] ビュー間動き予測とビュー間残差予測とを可能にするために、ビデオコーダは、ブロック（たとえば、PU、CUなど）に対する視差ベクトルを決定することができる。一般に、視差ベクトルは、2つのビューの間の変位を推定するものとして使用される。ビデオエンコーダ20またはビデオデコーダ30のようなビデオコーダは、ブロックに対する視差ベクトルを使用して、ビュー間動き予測または残差予測のために別のビュー中の参照ブロック（本明細書では視差参照ブロックと呼ばれ得る）を位置決定することができ、またはビュー間動き予測のために視差ベクトルを視差動きベクトルに変換することができる。

【0204】

10

[0221] 図6は、スケーラブルビデオコーディングを示す概念図である。図6はH. 264/AVCおよびSVCに関して説明されるが、H. 264/AVCを含む他のマルチレイヤビデオコーディング方式を訴えて、同様のレイヤがコーディングされ得ることを理解されたい。別の例では、多規格コーデックを使用して同様のレイヤがコーディングされ得る。たとえば、ベースレイヤはH. 264/AVCを使用してコーディングされ得るが、エンハンスメントレイヤはHEVCに対するスケーラブルなHLSのみの拡張を使用してコーディングされ得る。したがって、以下でのSVCへの言及は全般にスケーラブルビデオコーディングに当てはまり、H. 264/SVCには限定されない。

【0205】

20

[0222] SVCでは、たとえば、空間スケーラビリティ、時間スケーラビリティ、および品質スケーラビリティ（ビットレートまたは信号対雑音比（SNR）として表される）を含むスケーラビリティが、3次元において可能にされ得る。一般に、任意の次元における表現に追加することによって、より良い表現が普通は達成され得る。たとえば、図6の例では、7.5Hzのフレームレートと64キロボイト毎秒（KBPS）のビットレートとを有するQuarter Common Intermediate Format（QCIF）において、レイヤ0がコーディングされる。加えて、レイヤ1は15Hzのフレームレートと64KBPSのビットレートとを有するQCIFにおいてコーディングされ、レイヤ2は15Hzのフレームレートと256KBPSのビットレートとを有するCIFにおいてコーディングされ、レイヤ3は7.5Hzのフレームレートと512KBPSのビットレートとを有するQCIFにおいてコーディングされ、レイヤ4は30Hzのフレームレートとメガバイト毎秒（MBPS）のビットレートとを有する4CIFにおいてコーディングされる。図5に示されるレイヤの特定の数、コンテンツ、および構成は、例示のみを目的に与えられることを理解されたい。

30

【0206】

[0223] いずれにしても、ビデオエンコーダ（ビデオエンコーダ20のような）がそのようなスケーラブルな方法でコンテンツを符号化すると、ビデオデコーダ（ビデオデコーダ30のような）は、たとえばクライアントまたは送信チャネルに依存し得る適用例の要件に従って、実際の配信されるコンテンツを適合させるように抽出器ツールを使用することができる。

【0207】

40

[0224] SVCでは、最低の空間レイヤと品質レイヤとを有するピクチャは通常、H. 264/AVCに適合する。図6の例では、最低の空間レイヤと品質レイヤとを伴うピクチャ（QCIF解像度をもつ、レイヤ0およびレイヤ1中のピクチャ）は、H. 264/AVCに適合し得る。それらの中で、最低の時間レベルのピクチャは時間ベースレイヤ（レイヤ0）を形成する。この時間ベースレイヤ（レイヤ0）は、より高い時間レベル（レイヤ1）のピクチャによって増強され得る。

【0208】

[0225] H. 264/AVC適合レイヤに加えて、空間スケーラビリティおよび/または品質スケーラビリティを与えるためにいくつかの空間および/または品質のエンハンスメントレイヤが追加され得る。各々の空間または品質のエンハンスメントレイヤ自体は、H

50

． 264／AVC 適合レイヤと同じ時間スケーラビリティ構造を伴い、時間的にスケーラブルであり得る。

【0209】

[0226]ビュー間残差予測はビデオデータの「ビュー」に関して説明され得るが、同様の技法は、図6に示されるスケーラブル構造のレイヤのような、データの複数のレイヤに適用され得ることを理解されたい。たとえば、ビデオコーダ（ビデオエンコーダ20および／またはビデオデコーダ30のような）は、あるレイヤの残差を別のレイヤを使用して予測することができる。いくつかの例では、本技法は、H S V CのようなH E V Cのスケーラブルな拡張とともに実施され得る。

【0210】

[0227]特に、以下でより詳細に説明されるように、ビデオエンコーダ20は、いくつかのコーディング区分モードおよび／またはいくつかのコーディングモードのためだけに、CUに対する重み付けファクタをシグナリングすることができる。重み付けファクタがシグナリングされないとき、ビデオデコーダ30は、重み付けファクタの復号をスキップし、重み付けファクタが0であると自動的に決定する（すなわち、推測する）ことができる。

【0211】

[0228]一例では、PART__2N×2Nに等しくない区分モードを伴うインターコーディングされたCUに対する重み付けファクタはシグナリングされなくてよい。代替的な例では、PART__2N×2N、PART__2N×2、およびPART__N×2Nに等しくない区分モードを伴うインターコーディングされたCUに対する重み付けファクタはシグナリングされなくてよい。さらに別の例では、加えて、または代替的に、スキップおよび／または統合に等しくないコーディングモードを伴う任意のインターコーディングされたCUに対する重み付けファクタは、シグナリングされなくてよい。

【0212】

[0229]他の態様によれば、ビデオコーダは重み付けファクタを修正することができる。たとえば、1つまたは複数の重み付けファクタ（たとえば0.5および／または1）をディセーブルにするためのインジケータが、シーケンスレベルでシグナリングされ得る。いくつかの例では、インジケータは、各々の非ベースビューに対してVPS拡張においてシグナリングされ得る。他の例では、インジケータは、VPS中でシグナリングされてよく、すべての非ベースビューに対して適用可能であってよい。さらに他の例では、インジケータは、ピクチャパラメータセット（PPS）、スライスヘッダ、またはビューパラメータセット中でシグナリングされ得る。

【0213】

[0230]別の例では、インジケータは、重み付けファクタの1つまたは複数を修正するためにシグナリングされ得る。たとえば、インジケータは、ビデオデコーダ30に、最初の重み付けファクタ（たとえば、0.5）を新しい重み付けファクタ（たとえば、0.75）で置換させ得る。この修正インジケータは、PPS、スライスヘッダ、またはVPSでシグナリングされ得る。

【0214】

[0231]さらに他の態様によれば、ビデオコーダは、復号ピクチャバッファのピクチャ、および／または、図6に示されるスケーラブル構造の中のピクチャをコーディングするための参照ピクチャリストに基づいて、ARPをイネーブルまたはディセーブルにすることができる。たとえば、現在のPUをコーディングするための復号ピクチャバッファが、時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中のピクチャを含まないとき、ビデオコーダはPUのためのARP処理を修正することができる。

【0215】

[0232]別の例では、加えて／代替的に、視差参照ブロックの参照ピクチャリストの一方または両方が、時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中の参照ピクチャを含まないとき、ビデオコーダはPUのためのARP処理を修正するこ

10

20

30

40

50

とができる。

【0216】

[0233]いくつかの例では、ビデオコーダは、現在のPUがARPを使用してコーディングされないように、ARP処理をディセーブルにすることによってARP処理を修正することができる。他の例では、ビデオコーダは、時間的動きベクトルをスケーリングして別の利用可能な時間的視差参照ピクチャを識別することによって、ARP処理を修正することができる。

【0217】

[0234]図7は、現在のPU100の視差ベクトルを決定するために使用され得る、現在のPU100に対する例示的な空間隣接PUを示す概念図である。図7の例では、空間隣接PUは、 A_0 、 A_1 、 B_0 、 B_1 、および B_2 として示される位置を包含するPUであり得る。

10

【0218】

[0235]上で述べられたように、ビデオコーダ（ビデオエンコーダ20またはビデオデコーダ30のような）は、ビュー間動き予測および／またはビュー間残差予測を実行することができる。これらの2つのコーディングツールをイネーブルにするために、第1のステップは、視差ベクトルを導出する。

【0219】

[0236]いくつかの例では、ビデオコーダは、隣接ブロックベース視差ベクトル（NB DV：Neighboring Blocks Based Disparity Vector）の方法を使用して、ブロックの視差ベクトルを導出することができる。たとえば、PUの視差ベクトルを導出するために、NB DVと呼ばれる処理が、3D-HEVCのテストモデル（すなわち、3D-HTM）において使用され得る。NB DV処理は、空間的および時間的に隣接するブロック（隣接PU A_0 、 A_1 、 B_0 、 B_1 、および B_2 のような）からの視差動きベクトルを使用して、現在のブロックの視差ベクトルを導出する。隣接ブロック（たとえば、現在のブロックに空間的または時間的に隣接するブロック）は、ビデオコーディングにおいてほとんど同じ動き情報と視差情報とを共有する可能性が高いので、現在のブロックは、現在のブロックの視差ベクトルの予測子として、隣接ブロックにおける動きベクトル情報を使用することができる。

20

【0220】

[0237]ビデオコーダがNB DV処理を実行するとき、ビデオコーダは、固定の確認順序で、空間隣接ブロックおよび時間隣接ブロックの動きベクトルを確認することができる。ビデオコーダが空間隣接ブロックまたは時間隣接ブロックの動きベクトルを確認するとき、ビデオコーダは、空間隣接ブロックまたは時間隣接ブロックの動きベクトルが視差動きベクトルかどうかを決定することができる。ピクチャのブロックの視差動きベクトルは、ピクチャの視差参照ピクチャ内の位置を指す動きベクトルである。

30

【0221】

[0238]所与のピクチャの視差参照ピクチャは、所与のピクチャと同じアクセスユニットと関連付けられるが所与のピクチャとは異なるビューと関連付けられる、ピクチャであり得る。ビデオコーダが視差動きベクトルを識別すると、ビデオコーダは、確認処理を終了することができる。ビデオコーダは、返された視差動きベクトルを視差ベクトルに変換することができる。たとえば、ビデオコーダは、視差動きベクトルの水平成分に等しい現在のブロックの視差ベクトルの水平成分を設定することができ、視差ベクトルの垂直成分を0に設定することができる。

40

【0222】

[0239]NB DV処理を実行することによってビデオコーダが現在のブロックの視差ベクトルを導出することが不可能である場合（すなわち、視差ベクトルが見つからない場合）、ビデオコーダは、現在のブロックの視差ベクトルとして0視差ベクトルを使用することができる。0視差ベクトルは、0に等しい水平成分と垂直成分の両方を有する視差ベクトル

50

ルである。したがって、NB D V処理が利用不可能な結果を返すときであっても、視差ベクトルを必要とするビデオコードの他のコーディング処理は、現在のブロックに対して0視差ベクトルを使用することができる。

【0223】

[0240]いくつかの例では、NB D V処理を実行することによってビデオコードが現在のブロックの視差ベクトルを導出することが不可能である場合、ビデオコードは、現在のブロックに対するビュー間残差予測をディセーブルにすることができる。しかしながら、NB D V処理を実行することによってビデオコードが現在のブロックの視差ベクトルを導出することが可能かどうかに関係なく、ビデオコードは、現在のPUに対してビュー間動き予測を使用することができる。すなわち、すべての事前に定義された隣接ブロックを確認した後で視差ベクトルが見つからない場合、ビュー間動き予測のために0視差ベクトルが使用され得るが、ビュー間残差予測は対応するCUに対してディセーブルにされ得る。

10

【0224】

[0241]上で述べられたように、たとえば、 A_0 、 A_1 、 B_0 、 B_1 、または B_2 によって示されるPUを含む、5個の空間隣接ブロックが、視差ベクトルの導出のために使用され得る。加えて、1つまたは複数の時間隣接ブロックが、視差ベクトルの導出のために使用され得る。この場合、現在のビューからのすべての参照ピクチャが、候補ピクチャとして扱われる。候補ピクチャの数は、たとえば、4個の参照ピクチャにさらに制約され得る。同じ位置にある参照ピクチャがまず確認され、候補ピクチャの残りは、参照インデックス(`refIdx`)の昇順で確認される。`RefPicList0[refIdx]`と`RefPicList1[refIdx]`の両方が利用可能であるとき、`RefPicListX[refIdx]`は他のピクチャに先行し、ここでXは`collocated_from_10_flag`に等しい。

20

【0225】

[0242]各候補ピクチャに対して、時間隣接ブロックを導出するために3つの候補領域が決定される。ある領域が2つ以上の 16×16 のブロックを包含するとき、そのような領域中のすべての 16×16 のブロックが、ラスタースキャン順序で確認される。CPU（現在のPUまたは現在のCUの同じ位置にある領域）、CLCU（現在のPUの同じ位置にある領域を包含する最大コーディングユニット(LCU)）、およびBR（CPUの右下の 4×4 のブロック）という、3つの候補領域が定義される。

30

【0226】

[0243]ビデオコードは、視差ベクトルに対する空間隣接ブロックおよび／または時間隣接ブロックを、特定の順序で確認することができる。いくつかの例では、ビデオコードは、最初に空間隣接ブロック(A_0 、 A_1 、 B_0 、 B_1 、および B_2)を確認し、続いて時間隣接ブロックを確認することができる。空間隣接ブロックの1つが視差動きベクトルを有する場合、ビデオコードは確認処理を終了することができ、現在のPUの最終的な視差ベクトルとして視差動きベクトルを使用することができる。

【0227】

[0244]ビデオコードは、候補ピクチャの候補領域の各々を確認することができる。一例では、候補ピクチャが第1の非ベースビューの中にある場合、ビデオコードは、CPU、CLCU、およびBRの順序で候補領域を確認することができる。この例では、候補ピクチャが第2の非ベースビューの中にある場合、ビデオコードは、BR、CPU、およびCLCUの順序で候補領域を確認することができる。

40

【0228】

[0245]この例では、第1の非ベースビューと関連付けられるピクチャの復号は、ベースビューと関連付けられるピクチャの復号に依存し得るが、他のビューと関連付けられるピクチャの復号には依存しないことがある。さらに、この例では、第2の非ベースビューと関連付けられるピクチャの復号はまた、ベースビューと関連付けられるピクチャの復号のみに依存し得る。他の例では、第2の非ベースビューと関連付けられるピクチャの復号はさらに、第1の非ベースビューに依存し得るが、他のビューがもしあればそれと関連付け

50

られるピクチャには依存しないことがある。

【0229】

[0246]候補領域が2つ以上の 16×16 のブロックを包含するとき、ビデオコーダは、ラスタースキャン順序に従って、候補領域中のすべての 16×16 のブロックを確認することができる。ビデオコーダが候補領域（または候補領域内の 16×16 のブロック）を確認するとき、ビデオコーダは、候補領域を包含するPUが視差動きベクトルを規定するかどうかを決定することができる。候補領域を包含するPUが視差動きベクトルを規定する場合、ビデオコーダは、PUの視差動きベクトルに基づいて、現在のビデオユニットの視差ベクトルを決定することができる。

【0230】

[0247]ビュー間動き予測は、AMVPモードと統合モードの両方に適用され得る。たとえば、上で述べられたように、AMVPモードは、ビュー間動きベクトル予測子が候補リストに追加されるような方法で拡張されている。NB DVから導出された視差ベクトルに基づいて、ビデオコーダは、視差ベクトルと現在のブロックの中間サンプルの位置とを加算することによって、参照ビュー中の参照ブロックを決定する。現在のブロックの参照インデックスがビュー間参照ピクチャを指す場合、ビデオコーダは、ビュー間動きベクトル予測子を、対応する視差ベクトルに等しく設定することができる。現在の参照インデックスが時間的参照ピクチャを指し、参照ブロックが、現在の参照インデックスと同じアクセスユニットを指す動き仮定を使用する場合には、ビデオコーダは、この動き仮定と関連付けられる動きベクトルを、ビュー間動きベクトル予測子として使用することができる。他の場合には、ビデオコーダは、ビュー間動きベクトル予測子をディセーブルなものとして標識することができ、ビデオコーダは、動きベクトル予測子候補のリストに動きベクトルを含めなくてよい。

【0231】

[0248]統合／スキップモードに関して、動きパラメータの候補リストは、ビュー間動き予測を使用して取得される動きパラメータセットによって拡張される。たとえば、ビデオコーダは、上で述べられたAMVPモードと同じ方法で、参照ビュー中の参照ブロックの動きベクトル候補を導出することができる。導出された動きベクトルがイネーブルであり、その参照ピクチャが現在のPU／CUの参照ピクチャリスト中のあるエントリのピクチャ順序カウント（POC）値と等しいPOC値を有する場合、動き情報（予測方向、参照ピクチャ、および動きベクトル）が、POCに基づいて参照インデックスを変換した後で、統合候補リストに追加され得る。そのような候補は、ビュー間予測された動きベクトルと呼ばれ得る。それ以外の場合、視差ベクトルはビュー間視差動きベクトルに変換され、ビデオコーダは、利用可能であるときのビュー間予測された動きベクトルと同じ位置において、ビュー間視差動きベクトルを統合候補リスト中へ追加することができる。

【0232】

[0249]ビュー間動き予測と同様の方式で、ビュー間残差予測は、図8および図9に関して以下でより詳細に説明されるように、各CUの視差ベクトルに基づく。

【0233】

[0250]図8は、マルチビュービデオコーディングの例示的な予測構造を示す概念図である。例として、ビデオコーダ（ビデオエンコーダ20またはビデオデコーダ30のような）は、時間 T_0 におけるビューV1中のブロック P_e を使用してブロックを予測することによって、時間 T_8 におけるビューV1中のブロックをコーディングすることができる。ビデオコーダは、 P_e から現在のブロックの元のピクセル値を減算し、これによって、現在のブロックの残差サンプルを取得することができる。

【0234】

[0251]加えて、ビデオコーダは、視差ベクトル104によって参照ビュー（ビューV0）における参照ブロックを位置決定することができる。参照ブロック I_b の元のサンプル値とその予測されるサンプル P_b の差は、以下の式で r_b によって示されるような、参照ブロックの残差サンプルと呼ばれる。いくつかの例では、ビデオコーダは、現在の残差から

r_b を減算し、得られた差の信号を変換コーディングするだけでよい。したがって、ビュー間残差予測が使用されるとき、動き補償ループは次の式で表され得る。

【数 3】

$$\hat{I}_e = r_e + P_e + r_b$$

【0 2 3 5】

ここで、現在のブロックの再構築

【数 4】

10

$$\hat{I}_e$$

【0 2 3 6】

は、逆量子化された係数 r_e に、予測 P_e と量子化正規化された残差係数 r_b とを足したものに等しい。ビデオコーダは、 r_b を残差予測子として扱うことができる。したがって、動き補償と同様に、 r_b は現在の残差から減算されてよく、得られた差の信号のみが変換コーディングされる。

【0 2 3 7】

[0252] ビデオコーダは、C U ごとにビュー間残差予測の使用を示すために、フラグを条件的にシグナリングすることができる。たとえば、ビデオコーダは、残差参照領域によって包含される、または部分的に包含される、すべての変換ユニット (T U) を網羅することができる。これらの T U のいずれかがインターコーディングされ、0 ではないコーディングされたブロックフラグ (C B F) の値 (ルーマ C B F またはクロマ C B F) を含む場合、ビデオコーダは、関連する残差参照を利用可能なものとして標識することができ、残差予測を適用することができる。この場合、ビデオコーダは、C U シンタックスの一部としてビュー間残差予測の使用を示す、フラグをシグナリングすることができる。このフラグが 1 に等しい場合、現在の残差信号は、補間された可能性のある参照残差信号を使用して予測され、差だけが、変換コーディングを使用して送信される。それ以外の場合、現在のブロックの残差は、H E V C 変換コーディングを使用して従来通りにコーディングされる。

20

30

【0 2 3 8】

[0253] 2 0 1 2 年 7 月 1 0 日に出願された米国仮出願第 6 1 / 6 7 0 , 0 7 5 号、および、2 0 1 2 年 9 月 2 7 日に出願された米国仮出願第 6 1 / 7 0 6 , 6 9 2 号は、スケーラブルビデオコーディングのための一般化された残差予測 (G R P) を提案する。これらの仮特許出願はスケーラブルビデオコーディングに注目するが、これらの仮特許出願で説明される G R P 技法は、マルチビュービデオコーディング (たとえば、M V - H E V C および 3 D - H E V C) に適用可能であり得る。

【0 2 3 9】

[0254] 単予測の状況では、G R P の概略的な考え方は、

40

$$I_c = r_c + P_c + w * r_r$$

として定式化され得る。

【0 2 4 0】

[0255] 上の式において、 I_c は現在のレイヤ (またはビュー) 中の現在のフレームの再構築を示し、 P_c は同じレイヤ (またはビュー) からの時間的予測を表し、 r_c はシグナリングされる残差を示し、 r_r は参照レイヤからの残差予測を示し、 w は重み付けファクタである。いくつかの例では、重み付けファクタは、ビットストリームにおいてコーディングされること、または、以前にコーディングされた情報に基づいて導出されることが必要であり得る。G R P のためのこのフレームワークは、シングルループ復号とマルチループ復号の両方の場合に適用され得る。マルチループ復号は、再構築されアップサンプリン

50

グされたより低分解能の信号を使用した、ブロックの予測の制約されないバージョンを伴う。エンハンスメントレイヤ中の1つのブロックを復号するために、以前のレイヤ中の複数のブロックがアクセスされる必要がある。

【0241】

[0256]たとえば、ビデオデコーダ30がマルチループ復号を使用するとき、GRPはさらに、

$$I_c = r_c + P_c + w * (I_r - P_r)$$

として定式化され得る。

【0242】

[0257]上の式では、 P_r は参照レイヤ中の現在のピクチャに対する時間的予測を示し、 P_c は同じレイヤ（またはビュー）からの時間的予測を表し、 r_c はシグナリングされた残差を示し、 w は重み付けファクタであり、 I_r は参照レイヤ中の現在のピクチャの完全な再構築を示す。

【0243】

[0258]上の式は、ビットストリーム中でシグナリングされ得る、または、以前にコーディングされた情報に基づいて導出され得る、重み付けファクタを含む。いくつかの例では、ビデオエンコーダ20は、ビットストリーム中で、GRPにおいて使用される重み付けインデックスをCUごとにシグナリングすることができる。各重み付けインデックスは、0以上の1つの重み付けファクタに対応し得る。現在のCUに対する重み付けファクタが0に等しいとき、現在のCUの残差ブロックは、従来のHEVC変換コーディングを使用してコーディングされる。そうではなく、現在のCUに対する重み付けファクタが0より大きいとき、現在の残差信号（すなわち、現在のCUの残差ブロック）は、重み付けファクタによって乗算された参照残差信号を使用して予測されてよく、差だけが変換コーディングを使用して送信される。いくつかの例では、参照残差信号は補間される。

【0244】

[0259]L. Zhang 他、「3D-CE5. h related: Advanced residual prediction for multiview coding」、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、第2回会合：上海、中国、2012年10月13～19日、文書JCT3V-B0051（以後「JCT3V-B0051」）は、ビュー間残差予測のコーディング効率をさらに改善するための、高度な残差予測（ARP）方法を提案した。いくつかの例では、ARPは、CUレベルの代わりにPUレベルで実行され得る。ARPと上で説明された残差予測方式を区別するために、上で説明された残差予測方式は、「CUベースのビュー間残差予測」と呼ばれ得る。

【0245】

[0260]図9は、マルチビュービデオコーディングにおけるARPの例示的な予測構造を示す概念図である。図6は、現在のピクチャ110、時間的参照ピクチャ112、視差参照ピクチャ114、および時間的視差参照ピクチャ116という、4つのピクチャを含む。現在のピクチャ110は、ビューV1と関連付けられ、時間インスタンス T_j と関連付けられる。時間的参照ピクチャ112は、ビューV1と関連付けられ、時間インスタンス T_i と関連付けられる。視差参照ピクチャ114は、ビューV0と関連付けられ、時間インスタンス T_j と関連付けられる。時間的視差参照ピクチャ116は、ビューV0と関連付けられ、時間インスタンス T_i と関連付けられる。

【0246】

[0261]現在のピクチャ110は、「 D_c 」として示される現在のPUを含む。言い換えると、 D_c は現在のビュー（ビュー1）の中の現在のブロックを表す。 D_c は、時間的参照ピクチャ112の中のある位置を示す、時間的動きベクトル V_D を有する。ビデオエンコーダ20は、時間的動きベクトル V_D によって示される位置と関連付けられるピクチャ1

10

20

30

40

50

1 2 中のサンプルに基づいて、時間的参照ブロック D_r を決定することができる。したがって、 D_r は時間 T_i における同じビュー（ビュー 1）からの D_c の時間的予測ブロックを示し、 V_D は D_c から D_r への動きを示す。

【0247】

[0262] さらに、ビデオエンコーダ 20 は、 D_c の視差ベクトルによって示される位置と関連付けられる視差参照ピクチャ 114 中のサンプルに基づいて、視差参照ブロック B_c を決定することができる。したがって、 B_c は参照ブロック（すなわち、時間 T_j における参照ビュー（ビュー 0））の中の D_c の表現）を示す。 B_c の左上の位置は、導出された視差ベクトルを D_c の左上の位置に加算することによって、導出された視差ベクトルとともに計算され得る。 D_c および B_c は、2 つの異なるビューにおける同じオブジェクトの投影であり得るので、 D_c および B_c は同じ動き情報を共有するはずである。したがって、時間 T_i におけるビュー 0 の中の B_c の時間的予測ブロック B_r は、 V_D の動き情報を適用することによって B_c から位置決定され得る。

【0248】

[0263] ビデオエンコーダ 20 は、時間的視差ピクチャ 116 中の時間的視差参照ブロック B_r (B_c の予測ブロック) を決定することができる。上で示されたように、時間的視差ピクチャ 116 は、 B_r と同じビュー（すなわち、ビュー V_0 ）と関連付けられ、 D_r と同じ時間インスタンス（すなわち、時間インスタンス T_i ）と関連付けられる。ビデオエンコーダ 20 は、 D_c の動きベクトル V_D によって示される位置にあるサンプルに基づいて、 B_r を決定することができる。したがって、 B_r の左上の位置は、動きベクトル V_D を B_c の左上の位置に加算することによって、再使用される動きベクトル V_D とともに計算され得る。 B_c の左上の位置は、 D_c の左上の位置と視差ベクトルとの合計に等しくてよい。したがって、 B_r の左上の位置は、 D_c の左上の位置と視差ベクトルと動きベクトル V_D の座標との合計に等しくてよい。このようにして、図 9 において矢印 118 によって示されるように、ビデオエンコーダ 20 は、 B_r を決定するための動きベクトル V_D を再使用することができる。

【0249】

[0264] さらに、ARP において、第 1 の残差ブロック中の各サンプルは、 D_c 中のサンプルと D_r の対応するサンプルとの差を示し得る。第 1 の残差ブロックは、 D_c の元の残差ブロックと呼ばれ得る。第 2 の残差ブロック中の各サンプルは、 B_c 中のサンプルと B_r 中の対応するサンプルとの差を示し得る。第 2 の残差ブロックは「残差予測子」と呼ばれ得る。ビデオエンコーダ 20 は動きベクトル V_D を使用して B_r を決定するので、残差予測子は B_c の実際の残差データとは異なり得る。

【0250】

[0265] ビデオエンコーダ 20 が残差予測子を決定した後で、ビデオエンコーダ 20 は、重み付けファクタによって残差予測子を乗算することができる。言い換えると、 V_D の動き情報を伴う B_c の残差は、重み付けファクタによって乗算され、現在の残差のための残差予測子として使用される。重み付けファクタは、0、0.5、または 1 に等しくてよい。したがって、3 つの重み付けファクタ（すなわち、0、0.5、および 1）が ARP において使用され得る。

【0251】

[0266] ビデオエンコーダ 20 が重み付けファクタによって残差予測子を乗算した後、残差予測子は、重み付けられた残差予測子と呼ばれ得る。ビデオエンコーダ 20 は、最終的な重み付けファクタとして、現在の CU（すなわち、現在の PU を含む CU）に対して最小限のレートひずみコストをもたらす重み付けファクタを選択することができる。ビデオエンコーダ 20 は、ビットストリーム中に、重み付けインデックスを示すデータを CU レベルで含め得る。重み付けインデックスは、現在の CU に対する最終的な重み付けファクタ（すなわち、重み付けられた残差予測子を生成するために使用された重み付けファクタ）を示し得る。いくつかの例では、0、1、および 2 という重み付けインデックスは、0、1、および 0.5 という重み付けファクタにそれぞれ対応する。現在の CU に対して 0

という重み付けファクタを選択することは、現在のCUのPUのいずれに対してもARPを使用しないことと等価である。

【0252】

[0267]ビデオエンコーダ20は次いで、現在のPUの最終的な残差ブロックを決定することができる。現在のPUの最終的な残差ブロック中の各サンプルは、元の残差ブロック中のサンプルと、重み付けられた残差予測子中の対応するサンプルとの差を示し得る。現在のCU（すなわち、現在のPUを含むCU）の残差ブロックは、現在のCUの他のPUの残差ブロックがもしあればそれらとともに、現在のPUの最終的な残差ブロックを含み得る。本開示の他の箇所で説明されるように、ビデオエンコーダ20は、1つまたは複数の変換ブロックの間で、現在のCUの残差ブロックを区分することができる。変換ブロックの各々は現在のCUのTUと関連付けられ得る。各変換ブロックに対して、ビデオエンコーダ20は、変換ブロックに1つまたは複数の変換を適用して、変換係数ブロックを生成することができる。ビデオエンコーダ20は、ビットストリーム中に、変換係数ブロックの量子化された変換係数を表すデータを含め得る。

10

【0253】

[0268]したがって、ARPでは、2つのビューの残差の間での高い相関を確実にするために、ビデオエンコーダ20は、現在のPUの動きを、参照ビューピクチャ中の対応するブロックに適用して、ビュー間残差予測のために使用されるべき基本ビュー中の残差を生成することができる。このようにして、現在のPUおよび参照ビュー中の対応する参照ブロックに対して、動きが揃えられる。その上、予測誤差がさらに減るように、適応重み付けファクタが残差信号に適用される。

20

【0254】

[0269]現在のPUが双予測される場合、現在のPUは、RefPicList0動きベクトルと、RefPicList1動きベクトルと、RefPicList0参照インデックスと、RefPicList1参照インデックスとを有する。本開示は、現在のPUのRefPicList0参照インデックスによって示される参照ピクチャを、現在のPUのRefPicList0ターゲット参照ピクチャと呼ぶことがある。現在のPUのRefPicList1動きベクトルは、現在のPUのRefPicList1ターゲット参照ピクチャ中の参照位置を示し得る。本開示は、現在のPUのRefPicList1参照インデックスによって示される参照ピクチャを、現在のPUのRefPicList1ターゲット参照ピクチャと呼ぶことがある。現在のPUのRefPicList1動きベクトルは、現在のPUのRefPicList1ターゲット参照ピクチャ中の参照位置を示し得る。

30

【0255】

[0270]したがって、ビデオエンコーダ20が双予測されたPUに対してARPを実行するとき、ビデオエンコーダ20は、現在のPUのRefPicList0動きベクトルに基づいて、現在のPUのRefPicList0ターゲット参照ピクチャ中の参照位置を決定することができる。本開示は、この参照位置を、現在のPUのRefPicList0参照位置と呼ぶことがある。ビデオエンコーダ20は次いで、現在のPUのRefPicList0参照位置と関連付けられる現在のPUのRefPicList0ターゲット参照ピクチャの、実際のサンプルまたは補間されたサンプルを含む、参照ブロックを決定することができる。本開示は、この参照ブロックを、現在のPUのRefPicList0参照ブロックと呼ぶことがある。

40

【0256】

[0271]加えて、ビデオエンコーダ20は、現在のPUのRefPicList1動きベクトルに基づいて、現在のPUのRefPicList1ターゲット参照ピクチャ中の参照位置を決定することができる。本開示は、この参照位置を、現在のPUのRefPicList1参照位置と呼ぶことがある。ビデオエンコーダ20は次いで、現在のPUのRefPicList0参照位置と関連付けられる現在のPUのRefPicList1ターゲット参照ピクチャの、実際のサンプルまたは補間されたサンプルを含む、参照ブロッ

50

クを決定することができる。本開示は、この参照ブロックを、現在のPUのRefPicList1参照ブロックと呼ぶことがある。

【0257】

[0272]ビデオエンコーダ20は、現在のPUのRefPicList0参照ブロックおよび現在のPUのRefPicList1参照ブロックに基づいて、現在のPUの時間的予測ブロックを決定することができる。たとえば、現在のPUの時間的予測ブロック中の各サンプルは、現在のPUのRefPicList0参照ブロックおよび現在のPUのRefPicList1参照ブロック中の対応するサンプルの加重平均を示し得る。

【0258】

[0273]さらに、ビデオエンコーダ20が双予測されたPUに対してARPを実行するとき、ビデオエンコーダ20は、現在のPUのRefPicList0動きベクトルおよび視差参照ブロックの視差参照フレーム内の位置に基づいて、時間的視差参照ピクチャ中の時間的視差参照位置を決定することができる。本開示は、この時間的視差参照位置およびこの時間的視差参照ピクチャを、RefPicList0時間的視差参照位置およびRefPicList0時間的視差参照ピクチャとそれぞれ呼ぶことがある。RefPicList0時間的視差参照ピクチャは、現在のPUのRefPicList0ターゲット参照ピクチャと同じPOC値を有し得る。ビデオエンコーダ20は次いで、RefPicList0時間的視差参照位置と関連付けられるRefPicList0時間的視差参照ピクチャの、実際のサンプルまたは補間されたサンプルを含む、サンプルブロックを決定することができる。本開示は、このサンプルブロックを、RefPicList0時間的視差参照ブロックと呼ぶことがある。

【0259】

[0274]加えて、ビデオエンコーダ20は、現在のPUのRefPicList1動きベクトルおよび視差参照ブロックの視差参照フレーム内の位置に基づいて、時間的視差参照ピクチャ中の時間的視差参照位置を決定することができる。本開示は、この時間的視差参照位置およびこの時間的視差参照ピクチャを、RefPicList1時間的視差参照位置およびRefPicList1時間的視差参照ピクチャとそれぞれ呼ぶことがある。RefPicList1時間的視差参照ピクチャは、現在のPUのRefPicList1ターゲット参照ピクチャと同じPOC値を有し得る。現在のPUのRefPicList0ターゲット参照ピクチャおよび現在のPUのRefPicList1ターゲット参照ピクチャは異なることがあるので、RefPicList1時間的視差参照ピクチャは、RefPicList0時間的視差参照ピクチャとは異なることがある。ビデオエンコーダ20は次いで、RefPicList1時間的視差参照位置と関連付けられるRefPicList1時間的視差参照ピクチャの、実際のサンプルまたは補間されたサンプルを含む、サンプルブロックを決定することができる。本開示は、このサンプルブロックを、RefPicList1時間的視差参照ブロックと呼ぶことがある。

【0260】

[0275]次に、ビデオエンコーダ20は、RefPicList0時間的視差参照ブロックおよびRefPicList1時間的視差参照ブロックに基づいて、視差予測ブロックを決定することができる。いくつかの例では、視差予測ブロック中の各サンプルは、RefPicList0時間的視差参照ブロックおよびRefPicList1時間的視差参照ブロック中の対応するサンプルの加重平均である。ビデオエンコーダ20は次いで、残差予測子を決定することができる。残差予測子は、サンプルのブロックであり得る。残差予測子中の各サンプルは、視差参照ブロック中のサンプルと、視差予測ブロック中の対応するサンプルとの差を示し得る。ビデオエンコーダ20は次いで、重み付けファクタを残差予測子に適用することによって、重み付けられた残差予測子を生成することができる。ビデオエンコーダ20は次いで、現在のPUの最終的な残差ブロックを決定することができる。現在のPUの最終的な残差ブロック中の各サンプルは、現在のPUの元の予測ブロック中のサンプルと現在のPUの時間的予測ブロック中の対応するサンプルとの差と、重み付けられた残差予測子とを示し得る。ビデオエンコーダ20は、ビットストリーム中で

現在のPUの最終的な残差ブロックをシグナリングすることができる。

【0261】

[0276]ビデオデコーダ30は、PUおよび双予測されたPUに対してARPを実行するとき、同様の処理を実行することができる。たとえば、ビデオデコーダ30は、上で説明されたサンプルの方式で、現在のPUの時間的予測ブロックと重み付けられた残差予測子とを決定することができる。ビデオデコーダ30は、ビットストリーム中でシグナリングされるデータに基づいて、現在のPUの最終的な残差ブロックを決定することができる。ビデオデコーダ30は次いで、現在のPUの最終的な残差ブロックと、現在のPUの時間的予測ブロックと、重み付けられた残差予測子とを加算することによって、現在のPUの予測ブロックを再構築することができる。

10

【0262】

[0277]図10は、上で説明された、現在のブロックと、対応するブロックと、動き補償されたブロックとの関係を示す。言い換えると、図10は、ARPにおける、現在のブロックと、参照ブロックと、動き補償されたブロックとの例示的な関係を示す概念図である。図10の例では、ビデオコーダは現在、現在のピクチャ131中の現在のPU130をコーディングしている。現在のピクチャ131は、ビューV1および時間インスタンスT1と関連付けられる。

【0263】

[0278]さらに、図10の例では、ビデオコーダは、現在のPU130の視差ベクトルによって示される位置と関連付けられる参照ピクチャ133の実際のサンプルまたは補間されたサンプルを備える、参照ブロック132（すなわち、対応するブロック）を決定することができる。たとえば、参照ブロック132の左上の角は、現在のPU130の視差ベクトルによって示される位置であり得る。時間的視差参照ブロック145は、現在のPU130の予測ブロックと同じサイズを有し得る。

20

【0264】

[0279]図10の例では、現在のPU130は、第1の動きベクトル134と第2の動きベクトル136とを有する。動きベクトル134は、時間的参照ピクチャ138の中のある位置を示す。時間的参照ピクチャ138は、ビューV1（すなわち、現在のピクチャ131と同じビュー）および時間インスタンスT0と関連付けられる。動きベクトル136は、時間的参照ピクチャ140の中のある位置を示す。時間的参照ピクチャ140は、ビューV1および時間インスタンスT3と関連付けられる。

30

【0265】

[0280]上で説明されたARP方式によれば、ビデオコーダは、参照ピクチャ133と同じビューと関連付けられ時間的参照ピクチャ138と同じ時間インスタンスと関連付けられる参照ピクチャ（すなわち、参照ピクチャ142）を決定することができる。加えて、ビデオコーダは、動きベクトル134を参照ブロック132の左上の角の座標に加算して、時間的視差参照位置を導出することができる。ビデオコーダは、時間的視差参照ブロック143（すなわち、動き補償されたブロック）を決定することができる。時間的視差参照ブロック143中のサンプルは、動きベクトル134から導出された時間的視差参照位置と関連付けられる、参照ピクチャ142の実際のサンプルまたは補間されたサンプルであり得る。時間的視差参照ブロック143は、現在のPU130の予測ブロックと同じサイズを有し得る。

40

【0266】

[0281]同様に、ビデオコーダは、参照ピクチャ134と同じビューと関連付けられ時間的参照ピクチャ140と同じ時間インスタンスと関連付けられる参照ピクチャ（すなわち、参照ピクチャ144）を決定することができる。加えて、ビデオコーダは、動きベクトル136を参照ブロック132の左上の角の座標に加算して、時間的視差参照位置を導出することができる。ビデオコーダは次いで、時間的視差参照ブロック145（すなわち、動き補償されたブロック）を決定することができる。時間的視差参照ブロック145中のサンプルは、動きベクトル136から導出された時間的視差参照位置と関連付けられる、

50

参照ピクチャ 1 4 4 の実際のサンプルまたは補間されたサンプルであり得る。時間的視差参照ブロック 1 4 5 は、現在の P U 1 3 0 の予測ブロックと同じサイズを有し得る。

【0 2 6 7】

[0282]さらに、図 1 0 の例では、ビデオコーダは、時間的視差参照ブロック 1 4 3 および時間的視差参照ブロック 1 4 5 に基づいて、視差予測ブロックを決定することができる。ビデオコーダは次いで、残差予測子を決定することができる。残差予測子中の各サンプルは、参照ブロック 1 3 2 中のサンプルと、視差予測ブロック中の対応するサンプルとの差を示し得る。

【0 2 6 8】

[0283]本開示の態様によれば、ビデオコーダ（ビデオエンコーダまたはビデオデコーダのような）は、現在コーディングされているブロックに対する参照ピクチャリスト中の参照ピクチャに基づいて、A R P（あるレイヤの残差を第 2 の異なるレイヤの残差に対してコーディングすることを含む）をイネーブルまたはディセーブルにすることができる。ある例では、ビデオコーダは、現在コーディングされているブロックに対する参照ピクチャリストが任意の時間的参照ピクチャを含むかどうかに基づいて、A R P をイネーブルまたはディセーブルにすることができる。本開示の態様によれば、インター予測されたブロックに対する参照ピクチャリストがビュー間参照ピクチャのみを含む場合、ビデオコーダは A R P をディセーブルにすることができる。そのような例では、ビデオコーダがビデオエンコーダを備えるとき、ビデオエンコーダは、ビットストリーム中で重み付けファクタをシグナリングしなくてよい（重み付けファクタのシグナリングをスキップしてよい）。同様に、ビデオコーダがビデオデコーダを備えるとき、ビデオデコーダは同様に、重み付けファクタの復号をスキップし、重み付けファクタが 0 に等しいと推測することができる。

【0 2 6 9】

[0284]上で説明された技法は、ランダムアクセスピクチャの状況で適用され得る。たとえば、本開示の態様によれば、ビデオコーダは、現在コーディングされているビュー成分がランダムアクセスビュー成分かどうかに基づいて、A R P をイネーブルまたはディセーブルにすることができる。

【0 2 7 0】

[0285]ランダムアクセスビュー成分に関して、H E V C では一般に、N A L ユニットタイプによって識別され得る 4 つのピクチャタイプがある。4 つのピクチャタイプは、瞬時復号リフレッシュ（I D R）ピクチャ、C R A ピクチャ、時間レイヤアクセス（T L A）ピクチャ、および、I D R ピクチャ、C R A ピクチャまたは T L A ピクチャではないコーディングされたピクチャを含む。I D R ピクチャおよびコーディングされたピクチャは、H. 2 6 4 / A V C 仕様から継承されたピクチャタイプである。C R A および T L A ピクチャタイプは、H E V C 規格に対する新たな追加である。C R A ピクチャは、ビデオシーケンスの中央の任意のランダムアクセスポイントから始まる復号を容易にするピクチャタイプであり、I D R ピクチャを挿入するよりも効率的であり得る。T L A ピクチャは、イネーブルな時間レイヤ切替えポイントを示すために使用され得るピクチャタイプである。

【0 2 7 1】

[0286]放送またはストリーミングのようなビデオ用途では、切替えは、ビデオデータの異なるチャンネルの間で起こることがあり、ビデオデータの特定の部分へのジャンプが起こることがある。そのような例では、切替えおよび／またはジャンプの間の最小の遅延を達成することが有益であり得る。この特徴は、ビデオビットストリーム中で一定の間隔でランダムアクセスピクチャを有することによって可能にされ得る。H. 2 6 4 / A V C と H E V C の両方において規定されている I D R ピクチャは、ランダムアクセスのために使用され得る。しかしながら、I D R ピクチャは、コーディングされたビデオシーケンスを開始し、復号ピクチャバッファ（D P B）（図 2 および図 3 に関して以下で説明されるように、参照ピクチャメモリとも呼ばれ得る）からピクチャを除去する。したがって、復号順序で I D R ピクチャに後続するピクチャは、参照として I D R ピクチャより前に復号されるピクチャを使用することができない。その結果、ランダムアクセスのために I D R ピク

チャに依存するビットストリームは、より低いコーディング効率を有することがある。コーディング効率を改善するために、復号順序でCRAピクチャに後続するが出力順序でCRAピクチャに先行するピクチャが、CRAピクチャより前に復号されたピクチャを参照として使用することを、HEVCにおけるCRAピクチャが可能にする。

【0272】

[0287] HEVCでは、CRAピクチャで開始するビットストリームは適合ビットストリームと見なされる。ビットストリームがCRAピクチャで始まる時、CRAピクチャの先行ピクチャは、利用不可能な参照ピクチャを参照することがあり、したがって、正確に復号されないことがある。しかしながら、HEVCは、開始するCRAピクチャの先行ピクチャが出力されないことを規定しており、したがって「クリーンランダムアクセス」という名称である。ビットストリーム適合要件を確立するために、HEVCは、非出力の先行ピクチャの復号のために利用不可能な参照ピクチャを生成するための復号処理を規定する。しかしながら、適合するデコーダの実装形態は、復号処理がビットストリームの開始から実行されるときと比較して、これらの適合するデコーダが同一の出力を生成し得る限り、その復号処理に従う必要はない。HEVCでは、適合するビットストリームはIDRピクチャをまったく含まなくてもよく、したがって、コーディングされたビデオシーケンスのサブセットまたは不完全なコーディングされたビデオシーケンスを含み得る。

10

【0273】

[0288] IDRピクチャおよびCRAピクチャのほかに、他のタイプのランダムアクセスポイントピクチャ、たとえば、ブロークンリンクアクセス(BLA)ピクチャがある。ランダムアクセスポイントピクチャの主要なタイプの各々について、ランダムアクセスポイントピクチャがシステムによってどのように扱われ得る可能性があるかに応じて、サブタイプがあり得る。ランダムアクセスポイントピクチャの各サブタイプは、異なるNALユニットタイプを有する。

20

【0274】

[0289] 一般に、HEVCの拡張(MV-HEVC、3D-HEVC、またはSHVCのような)に関して、ビュー成分がランダムアクセスポイントかどうかは、ビュー成分のNALユニットタイプに依存し得る。そのタイプが、ランダムアクセスポイントピクチャのためのHEVC基本仕様において定義されているタイプに属する場合、現在のビュー成分はランダムアクセスポイントビュー成分(または簡単のために、現在のビューのランダムアクセスポイントピクチャ)である。

30

【0275】

[0290] いくつかの例では、ランダムアクセス機能は、時間次元における(したがってビュー内部での)いくつかの予測がHEVC基本仕様と同様にディセーブルにされるかまたは制約されるかのいずれかである方法で、時間的予測のみに適用される。しかしながら、ランダムアクセスポイントビュー成分のためのビュー間予測は、H.264/MVCにおけるアンカーピクチャと同様に、コーディング効率を改善することが依然として可能であり、そのように一般に実行される。したがって、ランダムアクセスポイント(RAP)ビュー成分は、ビュー間予測を使用する場合、PピクチャまたはBピクチャであり得る。

【0276】

[0291] 本開示の態様によれば、(ビデオエンコーダ20またはビデオデコーダ30などの)ビデオコーダは、ランダムアクセスビュー成分の各ブロックに対するビュー間残差予測をディセーブルにすることができる。そのような例では、ビデオエンコーダ20は、ビットストリーム中で重み付けファクタをシグナリングしなくてよい(重み付けファクタのシグナリングをスキップしてよい)。ビデオデコーダ30は、重み付けファクタの復号を同様にスキップし、重み付けファクタが0に等しいと自動的に決定することができる。

40

【0277】

[0292] 別の例では、本開示の態様によれば、ビデオコーダは、少なくとも1つの参照ピクチャが現在コーディングされているブロックと同じビューからのものである場合、ARPをイネーブルにすることができる。加えて、または代替的に、ビデオコーダは、両方の

50

参照ピクチャ（RefPicList 0中の参照ピクチャおよびRefPicList 1中の参照ピクチャに対応する）が利用可能であればそれらが現在コーディングされているブロックと同じビューのものであるときにのみ、ARPをイネーブルにすることができる。加えて、または代替的に、ビデオコーダは、ブロックがビュー間参照ピクチャとともにビュー間コーディングされる場合、ブロックに対するARPをディセーブルにすることができる。上で述べられたように、ARPがディセーブルにされるとき、重み付けファクタはシグナリングされない。

【0278】

[0293]いくつかの例では、現在のブロックをコーディングするための復号ピクチャバッファが、時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中のピクチャを含まないとき、ビデオコーダはARP処理を修正することができる。

10

【0279】

[0294]別の例では、加えて、または代替的に、視差参照ブロックの参照ピクチャリストの一方または両方が、時間的参照ピクチャと同じPOCを有する視差参照ピクチャと同じビュー中の参照ピクチャを含まないとき、ビデオコーダはARP処理を修正することができる。たとえば、視差参照ブロックを含むスライスに対する現在の参照ピクチャリストのインデックスがXであるとする（Xは0または1である）、一例では、視差参照ブロックのXに等しいリストインデックスを伴う参照ピクチャリストが、視差参照ピクチャと同じビューの中にあり現在のブロックの時間的参照ピクチャと同じPOCを有する参照ピクチャを含まない場合、ビデオコーダはARP処理を修正することができる。別の例では、

20

【0280】

[0295]いくつかの例では、ビデオコーダは、現在のブロックがARPを使用してコーディングされないように、ARP処理をディセーブルにすることによってARP処理を修正することができる。他の例では、ビデオコーダは、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、ARP処理を修正することができる。たとえば、ビデオコーダは、視差ベクトルに組み合わせられると、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にあるスケーリングされた組合せ時間的視差参照ピクチャのように、時間的動きベクトルをスケーリングすることができる。上で説明された技法は、復号ピクチャバッファまたは参照ピクチャリストの一方または両方に含まれないピクチャ中の視差参照ブロックをビデオコーダが位置決定しようとするのを防ぐことができる。

30

【0281】

[0296]本開示の他の態様によれば、ARPは、現在コーディングされているブロックの区分モードおよび／またはコーディングモードに基づいて、イネーブルまたはディセーブルにされ得る。たとえば、重み付けファクタは、ある区分モードおよび／またはあるコーディングモードのみに対してシグナリングされるだけであり得る。重み付けファクタがビットストリームに含まれない場合、ビデオデコーダは、重み付けファクタの復号をスキップし、重み付けファクタの値が0である（したがってARPをディセーブルにする）と推測することができる。本開示の態様によれば、いくつかの例では、PART₂N×2Nに等しくない区分モードを伴う任意のインターコーディングされたブロックに対する重み付けファクタはシグナリングされなくてよい。別の例では、PART₂N×2N、PART₂N×N、PART₂N×N以外の区分モードを伴うインターコーディングされたブロックに対する重み付けファクタはシグナリングされなくてよい。さらに別の例では、加えて、または代替的に、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタは、シグナリングされなくてよい。

40

50

【0282】

[0297]本開示のさらに他の態様によれば、重み付けファクタに対するより柔軟な手法が実施され得る。たとえば、利用可能な重み付けファクタの数は、（たとえば、シーケンスパラメータセット（SPS）のようなパラメータセット中の）シーケンスレベルで変更され得る。例示を目的とするある例では、たとえば0.5および／または1の1つまたは複数の重み付けファクタをディセーブルにするためのインジケータ、が、SPS中でシグナリングされ得る。別の例では、そのようなインジケータは、VPS中でシグナリングされ、すべての非ベースビューに対して適用可能であってよい。さらに別の例では、そのようなインジケータは、各々の非ベースビューに対してビデオパラメータセット（VPS）拡張においてシグナリングされ得る。別の例では、そのようなインジケータは、1つまたは複数の重み付けファクタをディセーブルにするために、ピクチャパラメータセット（PPS）、スライスヘッダ、またはビューパラメータセットにおいて提供され得る。重み付けファクタがディセーブルにされているとき、残りの重み付けファクタを表すためにより少数のビットが使用されてよく、これによってビットを節約する。

10

【0283】

[0298]他の態様によれば、1つまたは複数の重み付けファクタを修正および／または置換するための、インジケータが提供され得る。ある例では、ビデオコードは、0.5という重み付けファクタを0.75という重み付けファクタで置換することができる。このインジケータは、スライスヘッダ、SPS、ピクチャパラメータセット（PPS）、またはVPSでシグナリングされ得る。

20

【0284】

[0299]本開示の態様によれば、1つの例示的な実装形態において、ビデオコードは、3D-HTMバージョン5.0（上で述べられた）に記載されるような、修正されたビュー間残差予測処理を使用することができる。たとえば、本開示の態様によれば、1つまたは複数のシンタックス要素が、ビュー間残差予測が適用されることを示すために使用され得る。

【0285】

[0300]ある例では、重み付けファクタのインデックスを示す1つまたは複数のシンタックス要素（たとえば、weighting_factor_indexシンタックス要素）がCUの一部としてシグナリングされ得る。この例では、CUのシンタックスは（たとえば、3D-HTMバージョン5.0に対して）修正されてよく、重み付けファクタのシンタックス要素は、現在のビューが従属テクスチャビューであること、現在のCUがイントラコーディングされていないこと、および現在のCUがPART_2N×2Nに等しい区分モードを有することという条件を満たす場合にのみ、シグナリングされ得る。このシンタックス要素がビットストリーム中に存在しないとき、重み付けファクタは0に等しいと推測される。1つの例示的なCUシンタックス表が以下に示される。

30

【表 1】

coding_unit(x0, y0, log2CbSize , ctDepth) {	記述子
if(transquant_bypass_enable_flag) {	
cu_transquant_bypass_flag	ae(v)
}	
...	
} else { /* PART_NxN */	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x1, y0 , log2CUSize)	
prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	
}	
if(!depth_flag && layer_id && PredMode != MODE_INTRA && PartMode == PART_2Nx2N)	
weighting_factor_index	ae(v)
...	
if(!pcm_flag) {	
if(PredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
no_residual_syntax_flag	ae(v)
if(!no_residual_syntax_flag) {	
MaxTrafoDepth = (PredMode[x0][y0] == MODE_INTRA ? max_transform_hierarchy_depth_intra + IntraSplitFlag : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0 x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

10

20

30

40

【 0 2 8 6 】

[0301]別の例示的なCUシンタックス表が以下に示される。

【表 2】

coding_unit(x0, y0, log2CbSize , ctDepth) {	記述子
if(transquant_bypass_enable_flag) {	
cu_transquant_bypass_flag	ae(v)
}	
...	
} else { /* PART_NxN */	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x1, y0 , log2CUSize)	
prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	
}	
if(!depth_flag && layer_id && PredMode != MODE_INTRA && PartMode == PART_2Nx2N && TempMVAvai && DispVectAvai)	
weighting_factor_index	ae(v)
...	
if(!pcm_flag) {	
if(PredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
no_residual_syntax_flag	ae(v)
if(!no_residual_syntax_flag) {	
MaxTrafoDepth = (PredMode[x0][y0] == MODE_INTRA ?	
max_transform_hierarchy_depth_intra + IntraSplitFlag :	
max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0 x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

10

20

30

40

【0287】

[0302]上の例では、現在のCUが同じビューからの少なくとも1つの参照ピクチャから予測されるとき、TempMVAvaiは1に等しく設定され得る。それ以外の場合、それは0に等しく設定される。加えて、視差ベクトルが発見され得る場合、DispVectAvaiは1に等しく設定され得る。それ以外の場合、それは0に等しい。

50

【0288】

[0303]別の例では、重み付けファクタのシンタックス要素は、現在のビューが従属テクスチャビューであること、現在のCUがイントラコーディングされていないこと、現在のCUがPART_2N×2Nに等しい区分モードを有すること、導出された視差ベクトルが利用可能であること、および、少なくとも1つの区分が時間的動きベクトルを有すること、たとえば参照ピクチャが同じビューからのものであることという条件が満たされるときにのみ、シグナリングされ得る。このシンタックス要素がビットストリーム中に存在しないとき、重み付けファクタは0に等しいと推測される。

【0289】

[0304]さらに別の例では、重み付けファクタのシンタックス要素は、現在のビューが従属テクスチャビューであること、現在のCUがイントラコーディングされていないこと、現在のCUがPART_2N×2Nに等しい区分モードを有すること、導出された視差ベクトルが利用可能であること、および、現在のCUのすべてのPUの中の少なくとも1つの区分が時間的動きベクトルを有すること、たとえば参照ピクチャが同じビューからのものであることという条件が満たされるときにのみ、シグナリングされ得る。このシンタックス要素がビットストリーム中に存在しないとき、重み付けファクタは0に等しいと推測される。

10

【0290】

[0305]さらに別の例では、重み付けファクタのシンタックス要素は、現在のビューが従属テクスチャビューであること、および導出された視差ベクトルが利用可能であることという条件が満たされるときにのみ、シグナリングされ得る。

20

【0291】

[0306]本開示の態様によれば、重み付けファクタは種々の方法でシグナリングされ得る。たとえば、上で述べられたように、シンタックス要素weighting_factor_indexは、高度な残差予測のために使用される重み付けファクタに対するインデックスを示し得る。存在しないとき、高度な残差予測は、現在のCUに対してディセーブルにされ得る。たとえば、重み付けファクタが0に等しい場合、現在のブロックの残差はHEVC変換コーディングを使用して従来通りにコーディングされ、HEVC仕様（たとえば、上で識別されたようなWD9など）の8.5.2.2項におけるような仕様が、予測サンプルを得るために呼び出される。重み付けファクタインデックスが存在する場合、現在の残差信号は、重み付けファクタによって乗算された、補間された可能性のある参照残差信号を使用して予測され、差のみが送信され、HEVC仕様（たとえば、WD9）の修正された8.5.2.2.1項および8.5.2.2.2項に関して以下で説明される処理が、時間的参照ピクチャが利用される各予測リストに対して呼び出され得る。

30

【0292】

[0307]いくつかの例では、重み付けファクタインデックスは、重み付けファクタにマッピングされ得る。このようにして、ビデオコードは、ビュー間残差予測においてより柔軟な方法を重み付けファクタに対して実施することができる。たとえば、例示を目的に、シグナリングされるべきN個の異なる重み付けファクタがあると仮定し、Nは2、3、4などに等しいとする。これらの重み付けファクタの各々は最初に、以下の表1の例に示されるように、固有の重み付けインデックスにマッピングされてよく、ここで、 W_0 、 W_1 、 W_2 、...、 W_{N-1} は、値の昇順に並んだ重み付けファクタである。

40

【表 3】

表1 -重み付けファクタインデックスと重み付けファクタとの間のマッピング

weighting_factor_index の値	重み付け ファクタの値
0	W_0
1	W_1
2	W_2
3	...
4	...
5	...
...	...
N-1	W_{N-1}

10

20

【0293】

[0308]別の例では、 W_0 、 W_1 、 W_2 、...、 W_{N-1} は、コーディングの間に計算され得る、使用されている重み付けファクタの確率の降順で、重み付けファクタを表し得る。

【0294】

[0309]別の例示的なマッピングが以下の表2に示され、0、1、0.5に等しい重み付けファクタはそれぞれ、0、1、2によってインデックスが付けられる。すべての残りの重み付けファクタは、値の昇順または確率の降順に基づいてインデックスが付けられ得る。

30

【表 4】

表 2 - 重み付けファクタインデックスと重み付けファクタとの間のマッピング

weighting_factor_index の 値	重み付け ファクタの値
0	0
1	1
2	0.5
3	...
4	...
5	...
...	...
N-1	...

10

20

【0295】

[0310]ビデオデコーダ30は、符号化されたビットストリームからの重み付けファクタインデックスを解析してインデックスの値を決定することができる。一例では、各重み付けファクタは、重み付けファクタインデックスによって識別されてよく、重み付けファクタインデックスは、HEVC仕様（たとえば、WD9）のセクション9.3.2.2に記載されるように、truncated unary 二値化を使用してシグナリングされ得る。別の例では、重み付けファクタは、重み付けファクタの確率の降順に基づいて固有の重み付けインデックスにまずマッピングされ、次いで、truncated unary 二値化によってコーディングされ得る。

30

【0296】

[0311]さらに別の例では、二値化処理は、以下の表3に従って定義され得る。

【表 5】

表 3 - 重み付けファクタの2値化 (Nは4より大きい)

Weighting_factor_index の値	ビン値						
0	0						
1	1	0	0				
2	1	0	1				
3	1	1	0				
4	1	1	1	0			
5	1	1	1	1	0		
...	1	1	1	1	1	0	
N-1	1	1	1	1	1	...	1
binIdx	0	1	2	3	4	...	N-3

40

50

【0297】

[0312]ここで、3からN-1の値に対応する重み付けファクタインデックスのビンストリングは、「11」というプレフィックスと、`weighting_factor_index`の値から3を減算することによってインデックスが付けられるサフィックスと一致し、このとき`truncated_unary`二値化が使用される。

【0298】

[0313]全体で4つの重み付けファクタがあるとき、二値化処理は次の表によって定義され得る。

【表6】

表4 - 重み付けファクタの2値化 (N=4)

Weighting_factor_index の値	ビン値		
0	0		
1	1	0	0
2	1	0	1
3	1	1	
binIdx	0	1	2

10

20

【0299】

[0314]全体で3つの重み付けファクタ、たとえば、0、0.5、および1があるとき、二値化処理は次の表によって定義され得る。

【表7】

表5 - 重み付けファクタの2値化 (N=3)

Weighting_factor_index の値	ビンストリング	
0	0	
1	1	0
2	1	1
binIdx	0	1

30

【0300】

[0315]コンテキスト初期化に関して、コンテキストの1つのセットが、重み付けファクタインデックスをコーディングするために使用され得る。高度なビュー間残差予測モードは、PスライスとBスライスの両方に適用され得る。Pスライスの重み付けインデックスのコンテキストのための初期確率は、Bスライスの初期確率とは異なり得る。あるいは、すべてのコンテキストモデルが、異なるビン値、たとえば0および1に対して、等しい確率で初期化される。

40

【0301】

[0316]コンテキスト選択に関して、例示を目的に、現在のピクチャの左上サンプルに対する現在のルーマコーディングブロックの左上ルーマサンプルをルーマ位置(xC, yC)が特定すると仮定する。現在のコーディングブロックのすぐ左に位置するコーディングブロックの利用可能性を規定する変数`availableL`は、HEVC仕様の6.4.1項で規定されるようなZスキャン順序でブロックに対する利用可能性導出処理を呼び出すことによって導出され、入力および出力が`availableL`に割り当てられるとき

50

位置 (xCurr, yCurr) は (xC, yC) に等しく設定され隣接位置 (xN, yN) は (xC-1, yC) に等しく設定されると、さらに仮定する。

【0302】

[0317]上の例では、現在のコーディングブロックのすぐ上に位置するコーディングブロックの利用可能性を規定する変数availableAは、HEVC仕様（たとえば、WD9）の6.4.1項で規定されるようなZスキャン順序でブロックに対する利用可能性導出処理を呼び出すことによって導出されてよく、入力および出力がavailableAに割り当てられるとき位置 (xCurr, yCurr) は (xC, yC) に等しく設定され隣接位置 (xN, yN) は (xC, yC-1) に等しく設定される。

【0303】

[0318]本開示の態様によれば、condTermFlagN (NはLまたはAであり得る) は次のように導出され得る。

【0304】

— mbPAddrNが利用可能でありブロックmbPAddrNに対する重み付けファクタが0に等しくない場合、condTermFlagNは1に等しく設定される。

【0305】

— それ以外の場合 (mbPAddrNが利用可能ではなく、または、ブロックmbPAddrNに対する重み付けファクタが0に等しい場合)、condTermFlagNは0に等しく設定される。

【0306】

[0319]加えて、ctxIdxが重み付けファクタインデックスをコーディングするために使用されるべきコンテキストインデックスであると仮定する。この例では、コーディングされるべき各ピンに対するctxIdxのインクリメント (ctxIdxInc) は、 $ctxIdxInc = M * condTermFlagL + N * condTermFlagA$ によって導出され、ここでMまたはNは1または2であり得る。あるいは、ctxIdxIncは、 $ctxIdxInc = condTermFlagA$ によって導出され得る。あるいは、ctxIdxIncは、 $ctxIdxInc = condTermFlagL$ によって導出され得る。あるいは、ctxIdxIncは0に固定され得る。

【0307】

[0320]上で述べられたように、いくつかの例では、重み付けファクタは修正され得る。たとえば、利用可能な重み付けファクタの数は、（たとえば、シーケンスパラメータセット (SPS) のようなパラメータセット中の）シーケンスレベルで変更され得る。例示を目的とするある例では、たとえば0.5および／または1の1つまたは複数の重み付けファクタをディセーブルにするためのインジケータが、SPS中でシグナリングされ得る。別の例では、そのようなインジケータは、VPS中でシグナリングされ、すべての非ベースビューに対して適用可能であってよい。さらに別の例では、そのようなインジケータは、各々の非ベースビューに対してビデオパラメータセット (VPS) 拡張においてシグナリングされ得る。別の例では、そのようなインジケータは、1つまたは複数の重み付けファクタをディセーブルにするために、ピクチャパラメータセット (PPS)、スライスヘッダ、またはビューパラメータセットにおいて提供され得る。

【0308】

[0321]他の態様によれば、1つまたは複数の重み付けファクタを修正および／または置換するための、インジケータが提供され得る。ある例では、ビデオコードは、0.5という重み付けファクタを0.75という重み付けファクタで置換することができる。このインジケータは、スライスヘッダ、SPS、ピクチャパラメータセット (PPS)、またはVPSでシグナリングされ得る。

【0309】

[0322]一例では、ビデオパラメータセットが次のように（たとえば、3D-HTMバージョン5.0に対して）修正され得る。

10

20

30

40

【表 8】

ビデオパラメータセットの拡張

vps_extension() {	記述子
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
}	
...	
for(i = 0; i <= vps_max_layers_minus1; i++) {	
if(i) {	
multi_view_mv_pred_flag[i]	u(1)
~~multi_view_residual_pred_flag[i]~~	u(1)
advanced_residual_pred_flag[i]	u(1)
if(advanced_residual_pred_flag[i]) {	
weight_factor_change_flag[i]	u(1)
if(weight_factor_change_flag[i])	
diff_weight[i]	se(v)
}	
}	
if(i%1) {	
enable_dmm_flag[i]	u(1)
use_mvi_flag[i]	u(1)
}	
}	
}	

10

20

30

【0310】

[0323]上の例では、1に等しい**advanced_residual_pred_flag[i]**は、iに等しい**layer_id**を伴う現在のテクスチャビューに対して高度な残差予測（ARP）が使用され得ることを規定し得る。0に等しい**advanced_residual_pred_flag[i]**は、iに等しい**layer_id**を伴う現在のテクスチャビューに対してARPが使用されないことを規定する。存在しないとき、**advanced_residual_pred_flag[i]**は0に等しいと推測され得る。

40

【0311】

[0324]別の例では、フラグ、すなわち**advanced_residual_pred_flag**は、VPSの拡張において一度シグナリングされてよく、すべての非ベーステクスチャビューに対して適用可能であり得る。この例では、1に等しい**weight_factor_change_flag[i]**は、2に等しい重み付けファクタインデック

50

スに対応する重み付けファクタが現在のレイヤに対して変更されることを規定し得る。加えて、0に等しい`weight_factor_change_flag[i]`は、2に等しい重み付けファクタインデックスに対応する重み付けファクタが現在のレイヤに対して変更されないことを規定し得る。加えて、`diff_weight[i]`は、2に等しい重み付けファクタインデックスに対する新たな重み付けファクタと元の重み付けファクタとの差（場合によってはスケーリングを伴う）を規定し得る。`diff_weight[i]`の範囲は、両端を含めて-2~4であり得る。

【0312】

[0325]上の例では、ビデオコーダは、次のように新たな重み付けファクタを導出することができる。

【数5】

$$W_2 = (W_0 * 4 + \text{diff_weight}[i]) \div 4$$

【0313】

上の例では、重み付けファクタ W_2 が W_0 または W_1 に等しいとき、適用可能なビューの中の任意のCUの重み付けファクタインデックスは、常に2より小さい。

【0314】

[0326]さらに別の例では、上で説明されたシンタックス要素は、シーケンスパラメータセットを参照する非ベーステクスチャビューに対して同じ機能を達成するために、`advanced_residual_pred_flag`、`weight_factor_change_flag`、および`diff_weight`として、シーケンスパラメータセットまたはシーケンスパラメータセットの拡張においてシグナリングされ得る。

【0315】

[0327]図11は、ビデオデータ中のサンプル位置を示す。一般に、サンプル位置は、ビデオコーディングにおいては動きベクトルまたは視差ベクトルによって識別され得る。ビデオコーダ（ビデオエンコーダ20および／またはビデオデコーダ30のような）は、予測コーディングを目的に、識別された位置と関連付けられるサンプルを使用することができる。図11の例では、整数サンプルは大文字で示されるが、小数サンプル位置は小文字で示される。図11の例は全般に1/4サンプルルーマ補間を示すが、同様の補間はクロマ成分に適用され得る。

【0316】

[0328]ビデオコーダ（ビデオエンコーダ20またはビデオデコーダ30のような）がPUに対してARPを実行するとき、ビデオコーダは、3つのブロック（すなわち、図9の B_r 、 B_c 、および D_r ）にアクセスする必要がある。上で述べられたように、動きベクトルが小数ペル位置を示す場合、ビデオコーダは、2つの小数ペル補間処理、たとえば、時間的参照ブロックを位置決定するための一方の補間処理と、視差時間的参照ブロックを位置決定するための他方の補間処理とを実行する。加えて、ビデオコーダは、視差参照ブロックを決定するときに、さらに別の小数ペル補間処理を適用することができる。HEVCは、動き補償されたブロックを決定するとき、小数サンプル補間処理のために、8/4タップのルーマ／クロマ補間フィルタを使用することができる。

【0317】

[0329]本開示の態様によれば、ARPの動き補償処理は、特に参照ブロックのサブペル補間に関して、簡略化され得る。いくつかの例では、本開示の態様によれば、ビデオコーダは、ARP中の参照ブロックの位置を決定するための、1つまたは複数のタイプの補間を使用することができる。たとえば、ビデオコーダは、双線形フィルタのようなローパスフィルタを使用して、参照ブロックの位置を補間することができる。一般に、双線形フィルタ（すなわち、双線形補間）は、通常の2次元グリッド上で2つの変数（たとえば、 x および y ）の関数を補間するための線形補間の拡張である。したがって、双線形フィルタは2タップのフィルタであり得る。

10

20

30

40

50

【0318】

[0330]いくつかの例では、ビデオコーダは、視差参照ブロックと時間的視差参照ブロックとを生成するとき、双線形フィルタを使用することができる。したがって、小数サンプル補間処理のためにHEVCにおいて使用される8/4タップのルーマ/クロマ補間フィルタは、残差予測子を生成するとき、すなわち、図9に示される B_r と B_c とを生成するとき、双線形フィルタによって置き換えられ得る。

【0319】

[0331]加えて、いくつかの例では、ビデオコーダは、現在のPUの動き補償されたブロックを生成するとき、双線形フィルタを使用することができる。すなわち、小数サンプル補間処理のためにHEVCにおいて使用される8/4タップのルーマ/クロマ補間フィルタは、現在のPUの動き補償されたブロックを生成するとき、すなわち、図9に示される D_r を生成するとき、双線形フィルタによって置き換えられ得る。したがって、現在のPUの予測ブロックを決定するとき、ビデオコーダは、時間的参照ピクチャのルーマ成分および/またはクロマ成分に双線形フィルタを適用することができる。

10

【0320】

[0332]1つの代替的な例では、ビデオコーダは、上で説明された双線形フィルタを、ルーマ成分のみに、またはクロマ成分のみに適用することができる。別の例では、ビデオコーダは、ルーマ成分とクロマ成分の両方に双線形フィルタを適用することができる。

【0321】

[0333]図11に示される例では、ルーマサンプル補間処理への入力は、フルサンプルユニット(x_{Int_L} , y_{Int_L})におけるルーマ位置と、小数サンプルユニット(x_{Frac_L} , y_{Frac_L})におけるルーマ位置と、ルーマ参照サンプルアレイ $refPicLX_L$ とを含み得る。加えて、補間処理の出力は、予測されたルーマサンプル値 $predSampleLX_L[x_L, y_L]$ である。

20

【0322】

[0334]影付きブロック内の大文字 $A_{i,j}$ で標識された位置は、ルーマサンプルの所与の2次元アレイ $refPicLX_L$ 内のフルサンプル位置におけるルーマサンプルを表す。これらのサンプルは、予測されたルーマサンプル値 $predSampleLX_L[x_L, y_L]$ を生成するために使用され得る。ルーマサンプルの所与のアレイ $refPicLX_L$ 内の対応するルーマサンプル $A_{i,j}$ の各々の位置($x_{A_{i,j}}$, $y_{A_{i,j}}$)は、次のように導出され得る。

30

【数6】

$$xA_{i,j} = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x_{Int_L} + i) \quad (6-1)$$

$$yA_{i,j} = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y_{Int_L} + j) \quad (6-2)$$

【0323】

[0335]影なしブロック内の小文字で標識された位置は、1/4ピクセル(1/4ペル)サンプル小数位置におけるルーマサンプルを表す。小数サンプルユニット(x_{Frac_L} , y_{Frac_L})中のルーマ位置のオフセットは、フルサンプル位置および小数サンプル位置における生成されたルーマサンプルのうちのいずれが、予測されたルーマサンプル値 $predSampleLX_L[x_L, y_L]$ に割り当てられるかを規定する。この割り当ては、以下に示される表6-1で規定される割り当てに従って実行され得る。 $predSampleLX_L[x_L, y_L]$ の値は出力である。

40

【0324】

[0336]変数 $shift1$ 、 $shift2$ 、および $shift3$ は、HEVC 8.5.2.2.2項と同じ方法で導出され得る。フルサンプル位置($x_{A_{i,j}}$, $y_{A_{i,j}}$)におけるルーマサンプル $A_{i,j}$ を仮定すると、小数サンプル位置におけるルーマサンプル「 $a_{0,0}$ 」~「 $r_{0,0}$ 」は、以下の式によって導出され得る。

【0325】

50

— $a_{0,0}$ 、 $b_{0,0}$ 、 $c_{0,0}$ 、 $d_{0,0}$ 、 $h_{0,0}$ 、および $n_{0,0}$ と標識されたサンプルは、2 タップのフィルタを最も近い整数位置のサンプルに適用することによって導出され得る。

【数 7】

$$a_{0,0} = (48 * A_{0,0} + 16 * A_{1,0}) \gg \text{shift1} \quad (6-3)$$

$$b_{0,0} = (32 * A_{0,0} + 32 * A_{1,0}) \gg \text{shift1} \quad (6-4)$$

$$c_{0,0} = (16 * A_{0,0} + 48 * A_{1,0}) \gg \text{shift1} \quad (6-5)$$

$$d_{0,0} = (48 * A_{0,0} + 16 * A_{0,1}) \gg \text{shift1} \quad (6-6)$$

$$h_{0,0} = (32 * A_{0,0} + 32 * A_{0,1}) \gg \text{shift1} \quad (6-7)$$

$$n_{0,0} = (16 * A_{0,0} + 48 * A_{0,1}) \gg \text{shift1} \quad (6-8)$$

10

【0 3 2 6】

— $e_{0,0}$ 、 $i_{0,0}$ 、 $p_{0,0}$ 、 $f_{0,0}$ 、 $j_{0,0}$ 、 $q_{0,0}$ 、 $g_{0,0}$ 、 $k_{0,0}$ 、および $r_{0,0}$ と標識されるサンプルは、8 タップのフィルタをサンプル $a_{0,i}$ 、 $b_{0,i}$ 、および $c_{0,i}$ に適用することによって導出されることが可能であり、ここで垂直方向に $i = -3 \sim 4$ である。

【数 8】

$$e_{0,0} = (48 * a_{0,0} + 16 * a_{0,1}) \gg \text{shift2} \quad (6-9)$$

$$i_{0,0} = (32 * a_{0,0} + 32 * a_{0,1}) \gg \text{shift2} \quad (6-10)$$

$$p_{0,0} = (16 * a_{0,0} + 48 * a_{0,1}) \gg \text{shift2} \quad (6-11)$$

$$f_{0,0} = (48 * b_{0,0} + 16 * b_{0,1}) \gg \text{shift2} \quad (6-12)$$

$$j_{0,0} = (32 * b_{0,0} + 32 * b_{0,1}) \gg \text{shift2} \quad (6-13)$$

$$q_{0,0} = (16 * b_{0,0} + 48 * b_{0,1}) \gg \text{shift2} \quad (6-14)$$

$$g_{0,0} = (48 * c_{0,0} + 16 * c_{0,1}) \gg \text{shift2} \quad (6-15)$$

$$k_{0,0} = (32 * c_{0,0} + 32 * c_{0,1}) \gg \text{shift2} \quad (6-16)$$

$$r_{0,0} = (16 * c_{0,0} + 48 * c_{0,1}) \gg \text{shift2} \quad (6-17)$$

20

30

【表 9】

表 6-1 – ルーマ予測サンプル $\text{predSampleLX}_L[x_L, y_L]$ の割当て

xFracL	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
yFracL	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predSampleLX_L[x_L, y_L]	A << shift3	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

40

【0 3 2 7】

[0337] 上で述べられたように、ルーマ成分に関して説明されるが、ビデオコードは、同様の方式でクロマブロック中のサンプルを位置決定することができる。

【0 3 2 8】

[0338] いくつかの例では、視差参照ブロックおよび時間的視差参照ブロックの位置は、

50

HEVC仕様の8.5.2.2.1項および8.5.2.2.2項で規定されるような動き補償が適用された後で決定され得る。たとえば、現在のブロックに対して、予測されるルーマサンプルアレイは predSampleLX_L として識別されてよく、クロマサンプルアレイは predSampleLX_{cb} および predSampleLX_{cr} として識別されてよい。この例では、重み付けファクタが0に等しくない場合、ビデオコーダは、処理の終わりにおいて次の動作を実行することができる。

【0329】

ー 各参照ピクチャリストXに対して（Xは0または1である）、参照ピクチャがビュー間参照ピクチャではない場合、予測されるサンプル値をさらに修正するために次のことが適用される。

【0330】

1. 視差ベクトル導出処理を呼び出して、ターゲット参照ビューを指す視差ベクトルを取得する。

【0331】

2. 同じアクセスユニット内のピクチャターゲット参照ビュー中の視差ベクトルによって参照ブロックを位置決定する。視差ベクトルが小数位置を指す（すなわち、参照ブロックの左上の位置（図9の B_c ）が小数の位置である）場合、双線形フィルタが、参照ブロックを補間するために適用される。

【0332】

3. 現在のブロックの動き情報を再使用して、参照ブロックに対する動き情報を導出する。参照ブロックの導出された動きベクトルと、参照ブロックに対する参照ビュー中の導出された参照ピクチャとに基づいて、参照ブロックに対して動き補償を適用して残差ブロックを導出する。現在のブロックと、参照ブロックと、動き補償されたブロックとの関係が図9に示される。

【0333】

現在のブロックの参照インデックスを ref_idx_lx と示す。

【0334】

$\text{refPicListX}[\text{ref_idx_lx}]$ と同じPOCを有しターゲット参照ビューの中にある参照ピクチャを、復号ピクチャバッファ中で選択する。

【0335】

現在のブロックの動きベクトルと同一となるように参照ブロックの動きベクトルを導出する。

【0336】

動きベクトルが小数位置を指す場合、すなわち、参照ブロックの左上の位置と動きベクトルを足したものが小数位置（図9における B_r の左上の位置）である場合、双線形補間が適用される。

【0337】

4. 重み付けファクタを残差ブロックに適用して、 predARPSampleLX_L 、 $\text{predARPSampleLX}_{cb}$ 、および $\text{predARPSampleLX}_{cr}$ と示される、重み付けられた残差ブロックを得る。

【0338】

5. 重み付けられた残差ブロックの値を予測されたサンプルに加算する。

【0339】

$$\begin{aligned} \text{predSampleLX}_L &= \text{predSampleLX}_L + \text{predARPSampleLX}_L \\ \text{predSampleLX}_{cb} &= \text{predSampleLX}_{cb} + \text{predARPSampleLX}_{cb} \\ \text{predSampleLX}_{cr} &= \text{predSampleLX}_{cr} + \text{predARPSampleLX}_{cr} \end{aligned}$$

上の演算は行列／ベクトルの加算演算であることに留意されたい。

10

20

30

40

50

【0340】

[0339] 高度なビュー間残差予測が適用されるかされないかに関係なく、H E V C仕様（たとえば、W D 9）の8. 5. 2. 2. 3項に規定されるような重み付けられたサンプル予測処理が、双方向予測されたP Uに適用される。

【0341】

[0340] 上で述べられたように、本開示のいくつかの態様によれば、視差参照ブロックの参照ピクチャリストが、時間的参照ピクチャと同じP O Cを有する視差参照ピクチャと同じビュー中のピクチャを含まないとき、ビデオコーダはA R P処理を修正することができる。

【0342】

[0341] いくつかの例では、ビデオコーダは、現在のブロックがA R Pを使用してコーディングされないように、A R P処理をディセーブルにすることによってA R P処理を修正することができる。他の例では、ビデオコーダは、時間的動きベクトルをスケーリングして別の時間的視差参照ピクチャを識別することによって、A R P処理を修正することができる。たとえば、ビデオコーダは、スケーリングされた動きベクトルが、視差参照ピクチャに適用されると、参照ピクチャリストに含まれ視差参照ピクチャに時間的に最も近い位置にある時間的視差参照ピクチャを識別するように、時間的動きベクトルをスケーリングすることができる。

【0343】

[0342] 図12は、予測ユニットと関連付けられ得る区分モード（P Uサイズを定義し得る）を全般的に示す。たとえば、特定のC Uのサイズが $2N \times 2N$ であると仮定すると、C Uは、区分モード $2N \times 2N$ （160）と、 $N \times N$ （162）と、 $hN \times 2N$ （164）と、 $2N \times hN$ （166）と、 $N \times 2N$ （168）と、 $2N \times N$ （170）と、 $nL \times 2N$ （172）と、 $nR \times 2N$ （174）と、 $2N \times nU$ （176）と、 $2N \times nD$ （178）とを使用して予測され得る。図12の例に示される区分モードは単に説明のために提示されており、ビデオデータが予測される方式を示すために他の区分モードが使用されてよい。

【0344】

[0343] いくつかの場合には、ビデオコーダ（たとえば、ビデオエンコーダ20および／またはビデオデコーダ30など）は、区分モード160と162とを使用して、イントラ予測またはインター予測を実行することができる。たとえば、ビデオコーダは、 $2N \times 2N$ P U（区分モード160）を使用してC U全体を予測することができる。別の例では、ビデオコーダは、4つの $N \times N$ サイズのP U（区分モード162）を使用してC Uを予測することができ、4つのセクションの各々は、異なる予測技法を適用される可能性がある。

【0345】

[0344] 加えて、イントラコーディングに関して、ビデオコーダは、短距離イントラ予測（S D I P）と呼ばれる技法を実行することができる。S D I Pが利用可能である場合、C Uは、平行なP Uを使用して予測され得る（区分モード164および166）。すなわち、S D I Pは一般に、C Uが平行なP Uに分割されることを可能にする。コーディングユニット（C U）を非正方形の予測ユニット（P U）に分割することによって、予測されるピクセルと参照ピクセルとの間の距離は短くされ得る。

【0346】

[0345] インターコーディングに関して、対称区分モード160および162に加えて、ビデオコーダは、P Uの並行配列（区分モード168および170）、または種々のA M P（非対称動き区分）モードを実施することができる。A M Pモードに関して、ビデオコーダは、区分モード $nL \times 2N$ （172）と、 $nR \times 2N$ （174）と、 $2N \times nU$ （176）と、 $2N \times nD$ （178）とを使用して、C Uを非対称的に区分することができる。非対称区分では、C U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C Uの部分は、「n」とその後ろに付く「U p」、「D

10

20

30

40

50

own」、「Left」、または「Right」という表示とによって示される。

【0347】

[0346]本開示の他の態様によれば、ARPは、現在コーディングされているブロックの区分モードおよび／またはコーディングモードに基づいて、イネーブルまたはディセーブルにされ得る。たとえば、重み付けファクタは、ある区分モードおよび／またはあるコーディングモードのみに対してシグナリングされるだけであり得る。重み付けファクタがビットストリームに含まれない場合、ビデオデコーダは、重み付けファクタの復号をスキップし、重み付けファクタの値が0である（したがってARPをディセーブルにする）と推測することができる。

【0348】

[0347]ある例では、例示的なコーディングユニットのシンタックス表に関して上で述べられたように、本開示のいくつかの態様によれば、PART₂N×2N（区分モード160）に等しくない区分モードを伴う任意のインターコーディングされるブロックに対する重み付けファクタはシグナリングされなくてよい。別の例では、PART₂N×2N（区分モード160）、PART₂N×N（区分モード170）、およびPART_N×2N（区分モード168）以外の区分モードを伴うインターコーディングされたブロックに対する重み付けファクタはシグナリングされなくてよい。さらに別の例では、加えて、または代替的に、スキップモードおよび／または統合モードに等しくないコーディングモードを伴う任意のインターコーディングされたブロックに対する重み付けファクタは、シグナリングされなくてよい。

【0349】

[0348]図13は、本開示の技法による、現在のブロックを符号化するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部、たとえば、現在のPUを備え得る。ビデオエンコーダ20（図1および図2）に関して説明されるが、他のデバイスが図13の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0350】

[0349]この例では、ビデオエンコーダ20は、最初に、動きベクトルを使用して現在のブロックを予測する（190）。たとえば、ビデオエンコーダ20は、現在のブロックの1つまたは複数の予測ユニット（PU）を計算し得る。この例では、ビデオエンコーダ20が現在のブロックをインター予測すると仮定される。たとえば、動き推定ユニット42は、前にコーディングされたピクチャ、たとえば、ビュー間ピクチャおよび時間的ピクチャの動き探索を実行することによって、現在のブロックの動きベクトルを計算し得る。したがって、動き推定ユニット42は、現在のブロックを符号化するために、時間的動きベクトルまたは視差動きベクトルを生成し得る。

【0351】

[0350]ビデオエンコーダ20は次いで、現在のブロックをコーディングするための参照ピクチャリスト（たとえば、現在のブロックが双予測されるとき、RefPicList0およびRefPicList1）が現在のブロックの時間的位置以外の時間的位置にある1つまたは複数の参照ピクチャを含むかどうかを決定することができる（191）。いくつかの例では、本開示の他の箇所では説明されるように、ビデオエンコーダ20は、現在のブロックがランダムアクセスピクチャに含まれるかどうかを決定することによって、そのような決定を行うことができる。

【0352】

[0351]参照ピクチャリストが現在のブロックの時間的位置とは異なる時間的位置に参照ピクチャを含む場合（ステップ191のはいの分岐）、ビデオエンコーダ20は、上で説明されたARP処理のようなビュー間残差予測処理をイネーブルにし得る。この例では、ビデオエンコーダ20は、インター残差予測を実行して、現在のブロックの残差データを予測することができる（192）。たとえば、上で述べられたように、ビデオエンコーダ20は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、時

間的動きベクトルと視差動きベクトルを組み合わせてことによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオエンコーダ20は、重み付けファクタを残差予測子に適用することができる。ビデオエンコーダ20は次いで、現在のブロックに対する残差ブロックを計算することができる(194)。

【0353】

[0352]参照ピクチャリストが現在のブロックの時間的位置とは異なる時間的位置に参照ピクチャを含まない場合(ステップ191のいいえの分岐)、ビデオエンコーダ20は、上で説明されたARP処理のようなビュー間残差予測処理をディセーブルにすることができる(194)。この例では、ビデオエンコーダ20は、ビュー間残差予測処理のための重み付けファクタをシグナリングしなくてよい。すなわち、例示を目的とする例では、ビデオエンコーダ20は、ビットストリーム中でweighting_factor_indexシンタックス要素をシグナリングしなくてよい。

10

【0354】

[0353]いずれの場合でも、ビデオエンコーダ20は、たとえば、変換ユニット(TU)を生成するために、現在のブロックに対する残差ブロックを計算する(194)。ビュー間残差予測が使用されないときに残差ブロックを計算するために、ビデオエンコーダ20は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、残差を生成することができる。ビュー間残差予測が使用されるときに残差ブロックを計算するために、ビデオエンコーダ20は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、第1の残差を生成することができる。ビデオエンコーダ20は次いで、第1の残差と残差予測子との差に基づいて、最終的な残差を計算することができる。

20

【0355】

[0354]ビデオエンコーダ20は次いで、残差ブロックの係数を変換し、量子化することができる(196)。次に、ビデオエンコーダ20は、残差ブロックの量子化された変換係数をスキャンすることができる(198)。スキャンの間、またはスキャンの後、ビデオエンコーダ20は、たとえば、ビュー間残差予測がイネーブルにされ適用される例におけるビュー間残差予測の重み付け値を含む、変換係数をエントロピー符号化することができる(200)。ビデオエンコーダ20は次いで、ビュー間残差予測がイネーブルにされ適用される例において、ブロックの係数および重み付け値に対するエントロピーコーディングされたデータを出力することができる(202)。

30

【0356】

[0355]図14は、本開示の技法による、ビデオデータの現在のブロックを復号するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部(たとえば、PU)を備え得る。ビデオデコーダ30(図1および図3)に関して説明されるが、他のデバイスが図14の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0357】

[0356]最初に、ビデオデコーダ30は、現在のブロックに対する変換係数および動きベクトルについてのデータを受信する(210)。やはり、この例は、現在のブロックがインター予測されると仮定する。エントロピー復号ユニット80は、ブロックの係数および動きベクトルについてのデータをエントロピー復号する(212)。

40

【0358】

[0357]ビデオデコーダ30は次いで、現在のブロックをコーディングするための参照ピクチャリスト(たとえば、現在のブロックが双予測されるとき、RefPicList0およびRefPicList1)が現在のブロックの時間的位置以外の時間的位置にある1つまたは複数の参照ピクチャを含むかどうかを決定することができる(214)。いくつかの例では、本開示の他の箇所で説明されるように、ビデオデコーダ30は、現在のブ

50

ロックがランダムアクセスピクチャに含まれるかどうかを決定することによって、そのような決定を行うことができる。

【0359】

[0358]参照ピクチャリストが現在のブロックの時間的位置とは異なる時間的位置に参照ピクチャを含む場合（ステップ214のはいの分岐）、ビデオデコーダ30は、上で説明されたARP処理のようなビュー間残差予測処理をイネーブルにし得る。この例では、ビデオデコーダ30は、インター残差予測を実行して、現在のブロックの残差データを予測することができる（216）。たとえば、上で述べられたように、ビデオデコーダ30は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、時間的動きベクトルと視差動きベクトルを組み合わせることによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオデコーダ30はまた、ビットストリームにおいてシグナリングされるような重み付けファクタを残差予測子に適用することができる。

10

【0360】

[0359]参照ピクチャリストが現在のブロックの時間的位置とは異なる時間的位置に参照ピクチャを含まない場合（ステップ214のいいえの分岐）、または、ビュー間残差予測によって残差データを予測した後（216）、ビデオデコーダ30は、上で説明されたARP処理のようなビュー間残差予測処理をディセーブルにすることができ、動きベクトルを使用した現在のブロックの予測へと飛ぶことができる（218）。

【0361】

[0360]いずれの場合にも、ビデオデコーダ30は次いで、復号された動きベクトルを使用して、現在のブロックを予測することができる（218）。ビデオデコーダ30は、次いで、量子化された変換係数のブロックを作成するために、再生成された係数を逆スキャンすることができる（220）。ビデオデコーダ30は、次いで、残差ブロックを生成するために係数を逆量子化し、逆変換することができる（222）。ビデオデコーダ30は、最終的に、予測ブロックと残差ブロックを組み合わせることによって現在のブロックを復号することができる（224）。たとえば、ビュー間残差予測が適用されない例では、ビデオデコーダ30は単に、予測ブロックと復号された残差を組み合わせることができる。ビュー間残差予測が適用される例では、ビデオデコーダ30は、予測ブロックと、復号された残差（最終的な残差を表す）と、残差予測子とを組み合わせることができる。

20

30

【0362】

[0361]図15は、本開示の技法による、現在のブロックを符号化するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部、たとえば、現在のPUを備え得る。ビデオエンコーダ20（図1および図2）に関して説明されるが、他のデバイスが図15の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0363】

[0362]この例では、ビデオエンコーダ20は、現在のブロックに対する時間的動きベクトルによって示される時間的参照ブロックの位置を決定する（240）。たとえば、ビデオエンコーダ20は、現在のブロックの1つまたは複数の予測ユニット（PU）を計算し得る。この例では、ビデオエンコーダ20が現在のブロックをインター予測すると仮定される。たとえば、動き推定ユニット42は、前にコーディングされたピクチャ、たとえば、ビュー間ピクチャおよび時間的ピクチャの動き探索を実行することによって、現在のブロックの動きベクトルを計算し得る。したがって、動き推定ユニット42は、現在のブロックを符号化するために、時間的動きベクトルまたは視差動きベクトルを生成し得る。

40

【0364】

[0363]ビデオエンコーダ20はまた、視差参照ブロックの位置を補間することができる（242）。たとえば、ビデオエンコーダ20は、視差ベクトルを決定して、現在のブロックと同じPOC値を有するが第2の異なるビューの中に位置する、視差参照ブロックを位置決定することができる。いくつかの例では、本開示の態様によれば、整数位置にない

50

視差参照ブロックの位置を視差ベクトルが識別する場合、ビデオエンコーダ20は、双線形フィルタを適用して、視差参照ブロックの位置を補間することができる。

【0365】

[0364]加えて、ビデオエンコーダ20は、時間的視差参照ブロックの位置を決定することができる(244)。たとえば、ビデオエンコーダ20は、時間的動きベクトルと視差動きベクトルを組み合わせ、時間的視差参照ブロックの位置を決定することができる。やはり、いくつかの例では、本開示の態様によれば、整数位置にない時間的視差参照ブロックの位置を組合せが識別する場合、ビデオエンコーダ20は、双線形フィルタを適用して、時間的視差参照ブロックの位置を補間することができる。

【0366】

[0365]ビデオエンコーダ20は次いで、現在のブロックに対する残差予測子を決定することができる(246)。ビデオエンコーダ20は、視差参照ブロックと時間的視差参照ブロックとの差に基づいて、残差予測子を決定することができる。ビデオエンコーダ20は、重み付けファクタを得られた残差予測子に適用することができる。

【0367】

[0366]ビデオエンコーダ20は次いで、ブロックに対する最終的な残差を決定することができる(248)。たとえば、ビデオエンコーダ20は、現在のブロックのサンプルと時間的参照ブロックとの差に基づいて、第1の残差を決定することができる。ビデオエンコーダ20は次いで、第1の残差と残差予測子との差に基づいて、最終的な残差を決定することができる。

【0368】

[0367]ビデオエンコーダ20は次いで、残差ブロックの係数を変換し、量子化することができる(250)。次に、ビデオエンコーダ20は、残差ブロックの量子化された変換係数をスキャンすることができる(252)。スキャンの間、またはスキャンの後、ビデオエンコーダ20は、たとえば、ビュー間残差予測の重み付け値を含む、変換係数をエントロピー符号化することができる(254)。ビデオエンコーダ20は次いで、ブロックの係数および重み付け値についてのエントロピーコーディングされたデータを出力することができる(256)。

【0369】

[0368]図16は、本開示の技法による、ビデオデータの現在のブロックを復号するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部分(たとえば、PU)を備え得る。ビデオデコーダ30(図1および図3)に関して説明されるが、他のデバイスが図14の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0370】

[0369]最初に、ビデオデコーダ30は、現在のブロックに対する変換係数および動きベクトルについてのデータを受信する(260)。やはり、この例は、現在のブロックがインター予測されると仮定する。エントロピー復号ユニット80は、ブロックの係数および動きベクトルについてのデータをエントロピー復号する(262)。

【0371】

[0370]ビデオデコーダ30は次いで、復号された動きベクトルを使用して現在のブロックを予測することができる(264)。ビデオデコーダ30はまた、量子化された変換係数のブロックを作成するために再生成された係数を逆スキャンすることができる(266)。ビデオデコーダ30はまた、残差ブロックを生成するために係数を逆量子化し、逆変換することができる(268)。

【0372】

[0371]ビデオデコーダ30はまた、視差参照ブロックの位置を補間することができる(270)。たとえば、ビデオデコーダ30は、視差ベクトルを決定して、現在のブロックと同じPOC値を有するが第2の異なるビューの中に位置する、視差参照ブロックを位置決定することができる。いくつかの例では、本開示の態様によれば、整数位置にない視差

10

20

30

40

50

参照ブロックの位置を視差ベクトルが識別する場合、ビデオデコーダ30は、双線形フィルタを適用して、視差参照ブロックの位置を補間することができる。

【0373】

[0372] 加えて、ビデオデコーダ30は、時間的視差参照ブロックの位置を決定することができる(272)。たとえば、ビデオデコーダ30は、時間的動きベクトルと視差動きベクトルを組み合わせ、時間的視差参照ブロックの位置を決定することができる。やはり、いくつかの例では、本開示の態様によれば、整数位置にない時間的視差参照ブロックの位置を組合せが識別する場合、ビデオデコーダ30は、双線形フィルタを適用して、時間的視差参照ブロックの位置を補間することができる。

【0374】

[0373] ビデオデコーダ30は次いで、現在のブロックに対する残差予測子を決定することができる(274)。ビデオデコーダ30は、視差参照ブロックと時間的視差参照ブロックとの差に基づいて、残差予測子を決定することができる。ビデオデコーダ30は、重み付けファクタを得られた残差予測子に適用することができる。

【0375】

[0374] ビデオデコーダ30は、最終的に、予測ブロックと残差を組み合わせることによって現在のブロックを復号することができる(276)。たとえば、ビデオデコーダ30は、予測ブロックと、復号された残差(最終的な残差を表す)と、残差予測子とを組み合わせることができる。

【0376】

[0375] 図17は、本開示の技法による、現在のブロックを符号化するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部、たとえば、現在のPUを備え得る。ビデオエンコーダ20(図1および図2)に関して説明されるが、他のデバイスが図17の方法と同様の方法を実行するように構成され得ることを理解されたい。

【0377】

[0376] この例では、ビデオエンコーダ20は、最初に、現在のブロックを予測するための区分モードを決定する(280)。たとえば、ビデオエンコーダ20は、現在のブロックに対して、1つのPUを計算するか(たとえば、 $2N \times 2N$ の区分モード)、2つ以上のPUを計算するか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が現在のブロックをインター予測すると仮定される。たとえば、動き推定ユニット42は、前にコーディングされたピクチャ、たとえば、ビュー間ピクチャおよび時間的ピクチャの動き探索を実行することによって、現在のブロックの動きベクトルを計算し得る。したがって、動き推定ユニット42は、現在のブロックを符号化するために、時間的動きベクトルまたは視差動きベクトルを生成し得る。

【0378】

[0377] ビデオエンコーダ20は次いで、決定された区分モードに基づいて、重み付けファクタを示すデータを符号化する(かつビュー間残差予測を実行する)かどうかを決定することができる(282)。いくつかの例では、ビデオエンコーダ20は、区分モードが $2N \times 2N$ の区分モード以外のモードである場合、ビュー間残差予測をディセーブルにして、重み付けファクタの符号化をスキップすることができる。

【0379】

[0378] ビデオエンコーダ20が重み付けファクタを符号化する場合、ビデオエンコーダ20は、ビュー間残差予測を実行して、現在のブロックの残差データを予測することができる(284)。たとえば、上で述べられたように、ビデオエンコーダ20は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、時間的動きベクトルと視差動きベクトルを組み合わせることによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオエンコーダ20は、重み付けファクタを残差予測子に適用することができる。ビデオエンコーダ20は次いで、現在のブロックに対する残差ブロックを計算すること

10

20

30

40

50

ができる（２８６）。

【０３８０】

【0379】ビデオエンコーダ２０が重み付けファクタを符号化しない場合（ステップ２８２のいいえの分岐）、ビデオエンコーダ２０は、ビュー間残差予測をディセーブルにすることができ、現在のブロックの残差ブロックの計算に飛ぶことができる（２８６）。この例では、ビデオエンコーダ２０は、ビュー間残差予測処理のための重み付けファクタをシグナリングしなくてよい。すなわち、例示を目的とする例では、ビデオエンコーダ２０は、ビットストリーム中で`weighting_factor_index`シンタックス要素をシグナリングしなくてよい。

【０３８１】

【0380】いずれの場合でも、ビデオエンコーダ２０は、たとえば、変換ユニット（TU）を生成するために、現在のブロックに対する残差ブロックを計算する（２８６）。ビュー間残差予測が使用されないときに残差ブロックを計算するために、ビデオエンコーダ２０は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、残差を生成することができる。ビュー間残差予測が使用されるときに残差ブロックを計算するために、ビデオエンコーダ２０は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、第１の残差を生成することができる。ビデオエンコーダ２０は次いで、第１の残差と残差予測子との差に基づいて、最終的な残差を計算することができる。

【０３８２】

【0381】ビデオエンコーダ２０は次いで、残差ブロックの係数を変換し、量子化することができる（２８８）。次に、ビデオエンコーダ２０は、残差ブロックの量子化された変換係数をスキャンすることができる（２９０）。スキャンの間、またはスキャンの後、ビデオエンコーダ２０は、たとえば、ビュー間残差予測がイネーブルにされ適用される例におけるビュー間残差予測の重み付け値を含む、変換係数をエントロピー符号化することができる（２９２）。ビデオエンコーダ２０は次いで、ビュー間残差予測がイネーブルにされ適用される例において、ブロックの係数および重み付け値についてのエントロピーコーディングされたデータを出力することができる（２９４）。

【０３８３】

【0382】図１８は、本開示の技法による、ビデオデータの現在のブロックを復号するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部（たとえば、PU）を備え得る。ビデオデコーダ３０（図１および図３）に関して説明されるが、他のデバイスが図１４の方法と同様の方法を実行するように構成され得ることを理解されたい。

【０３８４】

【0383】この例では、ビデオデコーダ３０は、最初に、現在のブロックを予測するための区分モードを決定する（３００）。たとえば、ビデオデコーダ３０は、現在のブロックに対して、１つのPUを決定するか（たとえば、 $2N \times 2N$ の区分モード）、２つ以上のPUを決定するかを決定することができる。ブロックのその区分構造は、符号化されたビットストリームでシグナリングされ得る。ビデオデコーダ３０はまた、現在のブロックに対する変換係数および動きベクトルについてのデータをエントロピー復号する（３０２）。やはり、この例は、現在のブロックがインター予測されると仮定する。

【０３８５】

【0384】ビデオデコーダ３０は次いで、決定された区分モードに基づいて、重み付けファクタを復号する（たとえば、符号化されたビットストリームから解析する）（かつビュー間残差予測を実行する）かどうかを決定することができる（３０４）。いくつかの例では、ビデオデコーダ２０は、区分モードが $2N \times 2N$ の区分モード以外のモードである場合、ビュー間残差予測をディセーブルにして、重み付けファクタの復号をスキップすることができる。すなわち、たとえば、区分モードが $2N \times 2N$ の区分モード以外のモードであるとき、ビデオデコーダ３０は、重み付けファクタが０であると自動的に決定する（すな

わち、推測する）ことができる。

【0386】

[0385]ビデオデコーダ30が重み付けファクタを復号する場合、ビデオデコーダ30は、ビュー間残差予測を実行して、現在のブロックの残差データを予測することができる（306）。たとえば、上で述べられたように、ビデオデコーダ30は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、現在のブロックの動きベクトルを視差参照ブロックに適用することによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオデコーダ30はまた、ビットストリームにおいてシグナリングされるような重み付けファクタを残差予測子に適用することができる。

10

【0387】

[0386]ビデオデコーダ30が重み付けファクタを復号しない場合（ステップ304のいいえの分岐）、ビデオデコーダ30は、ビュー間残差予測処理をディセーブルにすることができる。ビデオデコーダ30は、動きベクトルを使用して現在のブロックの予測に飛ぶことができる。

【0388】

[0387]いずれの場合にも、ビデオデコーダ30は次いで、復号された動きベクトルを使用して、現在のブロックを予測することができる（308）。ビデオデコーダ30は、次いで、量子化された変換係数のブロックを作成するために、再生成された係数を逆スキャンすることができる（310）。ビデオデコーダ30は、次いで、残差ブロックを生成するために係数を逆量子化し、逆変換することができる（312）。ビデオデコーダ30は、最終的に、予測ブロックと残差ブロックを組み合わせることによって現在のブロックを復号することができる（314）。たとえば、ビュー間残差予測が適用されない例では、ビデオデコーダ30は単に、予測ブロックと復号された残差を組み合わせるすることができる。ビュー間残差予測が適用される例では、ビデオデコーダ30は、予測ブロックと、復号された残差（最終的な残差を表す）と、残差予測子とを組み合わせるすることができる。

20

【0389】

[0388]図19は、本開示の技法による、現在のブロックを符号化するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部、たとえば、現在のPUを備え得る。ビデオエンコーダ20（図1および図2）に関して説明されるが、他のデバイスが図19の方法と同様の方法を実行するように構成され得ることを理解されたい。

30

【0390】

[0389]この例では、ビデオエンコーダ20は、現在のブロックに対する時間的動きベクトルと参照ピクチャとを決定する（320）。たとえば、ビデオエンコーダ20は、現在のブロックの1つまたは複数の予測ユニット（PU）を計算し得る。この例では、ビデオエンコーダ20が現在のブロックをインター予測すると仮定される。たとえば、動き推定ユニット42は、前にコーディングされたピクチャ、たとえば、ビュー間ピクチャおよび時間的ピクチャの動き探索を実行することによって、現在のブロックの動きベクトルを計算し得る。したがって、動き推定ユニット42は、現在のブロックを符号化するために、時間的動きベクトルまたは視差動きベクトルを生成し得る。

40

【0391】

[0390]ビデオエンコーダ20は次いで、現在のブロックと同じアクセスユニット中の視差参照ブロックを決定することができる（322）。たとえば、ビデオエンコーダ20は、視差ベクトルを決定して、現在のブロックと同じPOC値を有するが第2の異なるビューの中に位置する、視差参照ブロックを位置決定することができる。

【0392】

[0391]ビデオエンコーダ20は、復号ピクチャバッファ（本明細書では参照ピクチャメモリとも呼ばれる）が時間的参照ピクチャのPOC値に等しいPOCを有するピクチャを含むかどうかを決定することができる（324）。たとえば、ビデオエンコーダ20は、

50

時間的動きベクトルと視差動きベクトルの組合せによって示されるピクチャが復号ピクチャバッファに含まれるかどうかを決定することができる。いくつかの例では、可能性のある時間的視差参照ピクチャが復号ピクチャバッファに含まれる場合であっても、ビデオエンコーダ20はさらに、視差参照ブロックに対する一方または両方の参照ピクチャリストにピクチャが含まれるかどうかを決定することができる。

【0393】

[0392]復号ピクチャバッファ（および／または視差参照ブロックの一方または両方の参照ピクチャリスト）に可能性のある時間的視差参照ピクチャが含まれる場合（324）、ビデオエンコーダ20は、ビュー間残差予測処理を実行して、現在のブロックの残差データを予測することができる（326）。たとえば、上で述べられたように、ビデオエンコーダ20は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、現在のブロックの動きベクトルを視差参照ブロックに適用することによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオエンコーダ20は、重み付けファクタを残差予測子に適用することができる。ビデオエンコーダ20は次いで、現在のブロックに対する残差ブロックを計算することができる（330）。 10

【0394】

[0393]復号ピクチャバッファに可能性のある時間的視差参照ピクチャが含まれない（または、視差参照ブロックの一方または両方の参照ピクチャリストに含まれない）場合（ステップ324のいいえの分岐）、ビデオエンコーダ20は、ビュー間残差予測処理を修正することができる（328）。いくつかの例では、ビデオエンコーダ20は、処理をディセーブルにすることによって処理を修正することができる。他の例では、ビデオエンコーダ20は、利用可能な参照ピクチャ（復号ピクチャバッファおよび／または参照ピクチャリストに含まれる参照ピクチャ）を選択し、それに従って時間的動きベクトルをスケールリングすることができる。 20

【0395】

[0394]いずれの場合でも、ビデオエンコーダ20は、たとえば、変換ユニット（TU）を生成するために、現在のブロックに対する残差ブロックを計算する（330）。ビュー間残差予測が使用されないときに残差ブロックを計算するために、ビデオエンコーダ20は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、残差を生成することができる。ビュー間残差予測が使用されるときに残差ブロックを計算するために、ビデオエンコーダ20は、元のコーディングされていないブロックと現在のブロックの予測ブロックとの差分を計算して、第1の残差を生成することができる。ビデオエンコーダ20は次いで、第1の残差と残差予測子との差に基づいて、最終的な残差を計算することができる。 30

【0396】

[0395]ビデオエンコーダ20は次いで、残差ブロックの係数を変換し、量子化することができる（332）。次に、ビデオエンコーダ20は、残差ブロックの量子化された変換係数をスキャンすることができる（334）。スキャンの間、またはスキャンの後、ビデオエンコーダ20は、たとえば、ビュー間残差予測がイネーブルにされ適用される例におけるビュー間残差予測の重み付け値を含む、変換係数をエントロピー符号化することができる（336）。ビデオエンコーダ20は次いで、ビュー間残差予測がイネーブルにされ適用される例において、ブロックの係数および重み付け値についてのエントロピーコーディングされたデータを出力することができる（338）。 40

【0397】

[0396]図20は、本開示の技法による、ビデオデータの現在のブロックを復号するための例示的な方法を示すフローチャートである。現在のブロックは、現在のCUまたは現在のCUの一部（たとえば、PU）を備え得る。ビデオデコーダ30（図1および図3）に関して説明されるが、他のデバイスが図14の方法と同様の方法を実行するように構成され得ることを理解されたい。 50

【0398】

[0397]最初に、ビデオデコーダ30は、現在のブロックに対する変換係数および動きベクトルについてのデータを受信する(350)。やはり、この例は、現在のブロックがインター予測されると仮定する。ビデオデコーダ30は、受信された動きベクトルを使用して時間的参照ピクチャを位置決定することができる。

【0399】

[0398]ビデオデコーダ30は次いで、現在のブロックと同じアクセスユニット中の視差参照ブロックを決定することができる(352)。たとえば、ビデオデコーダ30は、視差ベクトルを決定して、現在のブロックと同じPOC値を有するが第2の異なるビューの中に位置する、視差参照ブロックを位置決定することができる。いくつかの例では、ビデオデコーダ30は、ビットストリームに含まれるデータに基づいて視差ベクトルを決定することができる。他の例では、ビデオデコーダ30は、ビデオエンコーダ20と同じ処理を適用して、視差ベクトルを決定することができる。

10

【0400】

[0399]ビデオデコーダ30は、復号ピクチャバッファ(本明細書では参照ピクチャメモリとも呼ばれる)が時間的参照ピクチャのPOC値に等しいPOC値を有するピクチャを含むかどうかを決定することができる(354)。たとえば、ビデオデコーダ30は、時間的動きベクトルと視差動きベクトルの組合せによって示されるピクチャが復号ピクチャバッファに含まれるかどうかを決定することができる。いくつかの例では、可能性のある時間的視差参照ピクチャが復号ピクチャバッファに含まれる場合であっても、ビデオデコーダ30はさらに、視差参照ブロックに対する1つまたは複数の参照ピクチャリストにピクチャが含まれるかどうかを決定することができる。

20

【0401】

[0400]復号ピクチャバッファ(および/または視差参照ブロックの参照ピクチャリスト)に可能性のある時間的視差参照ピクチャが含まれる場合、ビデオデコーダ30は、ビュー間残差予測処理を実行して、現在のブロックの残差データを予測することができる(356)。たとえば、上で述べられたように、ビデオデコーダ30は、第1のブロックの視差ベクトルによって示される視差参照ブロックを決定し、現在のブロックの動きベクトルを視差参照ブロックに適用することによって時間的視差参照ブロックを決定し、時間的視差参照ブロックと視差参照ブロックとの差に基づいて残差予測子を決定することができる。ビデオデコーダ30はまた、ビットストリームにおいてシグナリングされるような重み付けファクタを残差予測子に適用することができる。

30

【0402】

[0401]復号ピクチャバッファ(および/または、視差参照ブロックの参照ピクチャリスト)に可能性のある時間的視差参照ピクチャが含まれない場合(ステップ354のいいえの分岐)、ビデオデコーダ30は、ビュー間残差予測処理を修正することができる(358)。いくつかの例では、ビデオデコーダ30は、処理をディセーブルにすることによって処理を修正することができる。他の例では、ビデオデコーダ30は、利用可能な参照ピクチャ(復号ピクチャバッファおよび/または参照ピクチャリストに含まれる参照ピクチャ)を選択し、それに従って時間的動きベクトルをスケーリングすることができる。

40

【0403】

[0402]いずれの場合にも、ビデオデコーダ30は次いで、復号された動きベクトルを使用して、現在のブロックを予測することができる(360)。ビデオデコーダ30は、次いで、量子化された変換係数のブロックを作成するために、再生成された係数を逆スキャンすることができる(362)。ビデオデコーダ30は、次いで、残差ブロックを生成するために係数を逆量子化し、逆変換することができる(364)。ビデオデコーダ30は、最終的に、予測ブロックと残差ブロックを組み合わせることによって現在のブロックを復号することができる(366)。たとえば、ビュー間残差予測が適用されない例では、ビデオデコーダ30は単に、予測ブロックと復号された残差を組み合わせたことができる。ビュー間残差予測が適用される例では、ビデオデコーダ30は、予測ブロックと、復号

50

された残差（最終的な残差を表す）と、残差予測子とを組み合わせることができる。

【0404】

[0403]例によっては、本明細書で説明された技法のうちのいずれかの、いくつかの動作またはイベントは、異なる順序で実行されてよく、追加、統合、または完全に除外され得る（たとえば、すべての説明された動作またはイベントが、本技法の実施のために必要であるとは限らない）ことを認識されたい。その上、いくつかの例では、動作またはイベントは、連続的にではなく、同時に、たとえば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて実行され得る。

【0405】

[0404]本開示のいくつかの態様が、説明のために開発中のH E V C規格に関して説明された。しかしながら、本開示で説明される技法は、他の規格またはまだ開発されていないプロプライエタリビデオコーディング処理を含む、他のビデオコーディング処理のために有用であり得る。

【0406】

[0405]上で説明された技法は、ビデオエンコーダ20（図1および図2）および／またはビデオデコーダ30（図1および図3）によって実行されてよく、ビデオエンコーダ20とビデオデコーダ30の両方が全般にビデオコーダと呼ばれ得る。同様に、ビデオコーディングは、適宜、ビデオ符号化またはビデオ復号を指し得る。

【0407】

[0406]例に応じて、本明細書で説明された方法のうちのいずれかのいくつかの動作またはイベントは、異なる順序で実行されてよく、互いに追加、統合、または完全に除外され得る（たとえば、すべての説明された動作またはイベントが、本方法の実施のために必要であるとは限らない）ことを理解されたい。その上、いくつかの例では、動作またはイベントは、連続的にではなく、同時に、たとえば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて実行され得る。さらに、本開示のいくつかの態様は、明快にするために単一のモジュールまたはユニットによって実行されるものとして説明されているが、本開示の技法はビデオコーダと関連付けられるユニットまたはモジュールの組合せによって実行され得ることを理解されたい。

【0408】

[0407]技法の様々な態様の特定の組合せが上で説明されたが、これらの組合せは、本開示で説明される技法の例を単に示すために与えられる。したがって、本開示の技法は、これらの例示的な組合せに限定されるべきではなく、本開示で説明される技法の様々な態様の任意の想起可能な組合せを包含し得る。

【0409】

[0408]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てよく、あるいは、コンピュータ可読媒体を介して送信され、ハードウェアベースの処理ユニットによって実行されてよい。コンピュータ可読媒体は、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支援する任意の媒体を含む、データ記憶媒体または通信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を含み得る。

【0410】

[0409]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、あるいは（2）信号また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、本開示で説明された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。

【0411】

[0410]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。

10

【0412】

[0411]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（登録商標）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

【0413】

[0412]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された技法の実施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、本明細書で説明された機能は、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／もしくはソフトウェアのモジュール内で提供され、また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素子中で完全に実装され得る。

30

【0414】

[0413]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コンポーネント、モジュール、またはユニットが説明されたが、それらのコンポーネント、モジュール、またはユニットは、異なるハードウェアユニットによる実現を必ずしも必要としない。むしろ、上で説明されたように、様々なユニットが、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、上で説明された1つまたは複数のプロセッサを含めて、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るか、または相互動作するハードウェアユニットの集合によって与えられ得る。

40

【0415】

[0414]本開示の様々な態様が説明されてきた。これらおよび他の態様は以下の特許請求の範囲内に入る。

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。

〔C1〕

ビデオデータをコーディングする方法であって、

ビデオデータのブロックをコーディングするための区分モードを決定することと、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

50

前記区分モードに基づいて、ビュー間残差予測処理のために重み付けファクタをコーディングするかどうかを決定することと、ここで、前記重み付けファクタがコーディングされないとき、前記ビュー間残差予測処理は、前記ブロックに対する残差を予測するために適用されない、

前記決定された区分モードとともにビデオデータの前記ブロックをコーディングすることと、
を備える、方法。

[C 2]

前記ブロックをコーディングすることは前記ブロックを復号することを備え、前記重み付けファクタがコーディングされないとき、前記重み付けファクタの値が0であると自動的に決定することをさらに備える、C 1に記載の方法。

10

[C 3]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、

前記区分モードが前記ブロックを予測することを全体として含むかどうかを決定することと、

前記区分モードが前記ブロックを予測することを全体として含まないとき、前記重み付けファクタのコーディングをスキップすることと、を備える、C 1に記載の方法。

[C 4]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、

前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つを含むかどうかを決定することと、

前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つを含まないとき、前記重み付けファクタのコーディングをスキップすることと、を備える、C 1に記載の方法。

20

[C 5]

前記ブロックに対するコーディングモードに基づいて前記重み付けファクタをコーディングするかどうかを決定することをさらに備え、前記コーディングモードがスキップモードおよび統合モード以外のコーディングモードであるとき、前記重み付けファクタのコーディングをスキップすること、C 1に記載の方法。

30

[C 6]

前記重み付けファクタがコーディングされるとき、前記方法はさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタをディセーブルにすることを備え、前記1つまたは複数のディセーブルにされた重み付けファクタは前記ビュー間残差予測処理において使用されない、C 1に記載の方法。

[C 7]

前記1つまたは複数の重み付けファクタをディセーブルにすることは、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタをディセーブルにすることを備える、C 6に記載の方法。

40

[C 8]

前記重み付けファクタがコーディングされるとき、前記方法はさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタを修正することを備え、前記修正された1つまたは複数の重み付けファクタは前記ビュー間残差予測処理において使用される、C 1に記載の方法。

[C 9]

前記1つまたは複数の重み付けファクタを修正することは、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、VPS拡張、ピクチャパラメータセ

50

ット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタを修正することを備える、C 8に記載の方法。

[C 1 0]

ビデオデータをコーディングするための装置であって、

ビデオデータを記憶するメモリと、

前記ビデオデータのブロックをコーディングするための区分モードを決定し、ここで、前記区分モードは、予測コーディングのための前記ビデオデータの前記ブロックの分割を示し、

前記区分モードに基づいて、ビュー間残差予測処理のために重み付けファクタをコーディングするかどうかを決定し、ここで、前記重み付けファクタがコーディングされないとき、前記ビュー間残差予測処理は、前記ブロックに対する残差を予測するために適用されない、

前記決定された区分モードとともに前記ビデオデータの前記ブロックをコーディングする

ように構成される1つまたは複数のプロセッサと、を備える、装置。

[C 1 1]

前記ブロックをコーディングするために、前記1つまたは複数のプロセッサは、前記ブロックを復号するように構成され、前記重み付けファクタがコーディングされないとき、前記1つまたは複数のプロセッサは、前記重み付けファクタの値が0であると自動的に決定するように構成される、C 1 0に記載の装置。

[C 1 2]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定するために、前記1つまたは複数のプロセッサは、

前記区分モードが前記ブロックを予測することを全体として含むかどうかを決定するように構成され、

前記区分モードが前記ブロックを予測することを全体として含まないとき、前記1つまたは複数のプロセッサが、前記重み付けファクタのコーディングをスキップするように構成される、C 1 0に記載の装置。

[C 1 3]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定することは、

前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つを含むかどうかを決定することを備え、

前記区分モードがPART_2N×2Nモード、PART_2N×Nモード、およびPART_N×2Nモードのうちの1つを含まないとき、前記1つまたは複数のプロセッサが、前記重み付けファクタのコーディングをスキップするように構成される、C 1 0に記載の装置。

[C 1 4]

前記1つまたは複数のプロセッサがさらに、

前記ブロックに対するコーディングモードに基づいて前記重み付けファクタをコーディングするかどうかを決定するように構成され、前記コーディングモードがスキップモードおよび統合モード以外のコーディングモードであるとき、前記重み付けファクタのコーディングをスキップすること、C 1 0に記載の装置。

[C 1 5]

前記重み付けファクタがコーディングされる時、前記1つまたは複数のプロセッサはさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタをディセーブルにするように構成され、前記1つまたは複数のディセーブルにされた重み付けファクタは前記ビュー間残差予測処理において使用されない、C 1 0に記載の装置。

[C 1 6]

前記1つまたは複数の重み付けファクタをディセーブルにするために、前記1つまたは

10

20

30

40

50

複数のプロセッサは、シーケンスパラメータセット（SPS）、ビデオパラメータセット（VPS）、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタをディセーブルにするように構成される、C15に記載の装置。

[C17]

前記重み付けファクタがコーディングされる時、前記1つまたは複数のプロセッサはさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタを修正するように構成され、前記修正された1つまたは複数の重み付けファクタは前記ビュー間残差予測処理において使用される、C10に記載の装置。

[C18]

前記1つまたは複数の重み付けファクタを修正するために、前記1つまたは複数のプロセッサは、シーケンスパラメータセット（SPS）、ビデオパラメータセット（VPS）、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタを修正するように構成される、C17に記載の装置。

[C19]

前記ブロックをコーディングするために、前記1つまたは複数のプロセッサは、前記ビュー間残差予測処理によって前記ブロックを復号するように構成され、前記ビュー間残差予測処理によって前記ブロックを復号するために、前記1つまたは複数のプロセッサは、前記ブロックの時間的動きベクトルによって示される時間的参照ブロックを決定し、前記ブロックの視差ベクトルによって示される視差参照ブロックを決定し、前記時間的動きベクトルと前記視差ベクトルの組合せによって示される時間的視差参照ブロックを決定し、

符号化されたビットストリームから、前記ブロックに対する最終的な残差を示すデータを取得し、

前記視差参照ブロックと前記時間的視差参照ブロックとの差に基づいて、残差予測子を決定し、

前記最終的な残差、前記残差予測子、および前記時間的参照ブロックの組合せに基づいて、前記ブロックを再構築するように構成される、C10に記載の装置。

[C20]

前記ブロックをコーディングするために、前記1つまたは複数のプロセッサは、前記ビュー間残差予測処理によって前記ブロックを符号化するように構成され、前記ビュー間残差予測処理によって前記ブロックを符号化するために、前記1つまたは複数のプロセッサは、

前記ブロックの時間的動きベクトルによって示される時間的参照ブロックを決定し、前記ブロックの視差ベクトルによって示される視差参照ブロックを決定し、前記時間的動きベクトルと前記視差ベクトルの組合せによって示される時間的視差参照ブロックを決定し、

前記第1のブロックと前記時間的参照ブロックとの差を備える第1の残差を決定し、前記視差参照ブロックと前記時間的視差参照ブロックとの差を備える残差予測子を決定し、

前記第1の残差と前記残差予測子との差に基づいて、最終的な残差を決定し、前記最終的な残差を示すデータをビットストリーム中に符号化するように構成される、C10に記載の装置。

[C21]

ビデオデータをコーディングするための装置であって、ビデオデータのブロックをコーディングするための区分モードを決定するための手段と、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記区分モードに基づいて、ビュー間残差予測処理のために重み付けファクタをコーデ

10

20

30

40

50

ィングするかどうかを決定するための手段と、ここで、前記重み付けファクタがコーディングされないとき、前記ビュー間残差予測処理は、前記ブロックに対する残差を予測するために適用されない、

前記決定された区分モードとともにビデオデータの前記ブロックをコーディングするための手段と、

を備える、装置。

[C 2 2]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定するための前記手段は、

前記区分モードが前記ブロックを予測することを全体として含むかどうかを決定するための手段と、

前記区分モードが前記ブロックを予測することを全体として含まないとき、前記重み付けファクタのコーディングをスキップするための手段と、を備える、C 2 1に記載の装置。

[C 2 3]

前記重み付けファクタがコーディングされる時、前記装置はさらに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタをディセーブルにするための手段をさらに備え、前記1つまたは複数のディセーブルにされた重み付けファクタは前記ビュー間残差予測処理において使用されない、C 2 1に記載の装置。

[C 2 4]

前記1つまたは複数の重み付けファクタをディセーブルにするための前記手段は、シーケンスパラメータセット(SPS)、ビデオパラメータセット(VPS)、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタをディセーブルにするための手段を備える、C 2 3に記載の装置。

[C 2 5]

前記重み付けファクタがコーディングされる時、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタを修正するための手段をさらに備え、前記修正された1つまたは複数の重み付けファクタは前記ビュー間残差予測処理において使用される、C 2 1に記載の装置。

[C 2 6]

前記1つまたは複数の重み付けファクタを修正することは、シーケンスパラメータセット(SPS)、ビデオパラメータセット(VPS)、VPS拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタを修正することを備える、C 2 5に記載の装置。

[C 2 7]

命令を記憶した非一時的コンピュータ可読媒体であって、前記命令は、実行されると、1つまたは複数のプロセッサに、

ビデオデータのブロックをコーディングするための区分モードを決定させ、ここで、前記区分モードは、予測コーディングのためのビデオデータの前記ブロックの分割を示し、

前記区分モードに基づいて、ビュー間残差予測処理のために重み付けファクタをコーディングするかどうかを決定させ、ここで、前記重み付けファクタがコーディングされないとき、前記ビュー間残差予測処理は、前記ブロックに対する残差を予測するために適用されない、

前記決定された区分モードとともにビデオデータの前記ブロックをコーディングさせる、非一時的コンピュータ可読媒体。

[C 2 8]

前記区分モードに基づいて前記重み付けファクタをコーディングするかどうかを決定するために、前記命令は、前記1つまたは複数のプロセッサに、

前記区分モードが前記ブロックを予測することを全体として含むかどうかを決定させ、

10

20

30

40

50

前記区分モードが前記ブロックを予測することを全体として含まないとき、前記1つまたは複数のプロセッサは、前記重み付けファクタのコーディングをスキップするように構成される、C 2 7に記載の非一時的コンピュータ可読媒体。

[C 2 9]

前記命令はさらに、前記1つまたは複数のプロセッサに、

前記ブロックに対するコーディングモードに基づいて前記重み付けファクタをコーディングするかどうかを決定させ、ここで、前記コーディングモードがスキップモードおよび統合モード以外のコーディングモードであるとき、前記重み付けファクタのコーディングをスキップする、C 2 7に記載の非一時的コンピュータ可読媒体。

[C 3 0]

前記重み付けファクタがコーディングされる時、前記命令はさらに、前記1つまたは複数のプロセッサに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタをディセーブルにさせ、前記1つまたは複数のディセーブルにされた重み付けファクタが前記ビュー間残差予測処理において使用されない、C 2 7に記載の非一時的コンピュータ可読媒体。

[C 3 1]

前記1つまたは複数の重み付けファクタをディセーブルにするために、前記命令は、前記1つまたは複数のプロセッサに、シーケンスパラメータセット (S P S)、ビデオパラメータセット (V P S)、V P S拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタをディセーブルにさせる、C 3 0に記載の非一時的コンピュータ可読媒体。

[C 3 2]

前記重み付けファクタがコーディングされる時、前記命令はさらに、前記1つまたは複数のプロセッサに、前記ビュー間残差予測処理の1つまたは複数の重み付けファクタを修正させ、前記修正された1つまたは複数の重み付けファクタは前記ビュー間残差予測処理において使用される、C 2 7に記載の非一時的コンピュータ可読媒体。

[C 3 3]

前記1つまたは複数の重み付けファクタを修正するために、前記命令は、前記1つまたは複数のプロセッサに、シーケンスパラメータセット (S P S)、ビデオパラメータセット (V P S)、V P S拡張、ピクチャパラメータセット、およびスライスヘッダのうちの1つに含まれるデータに基づいて、前記1つまたは複数の重み付けファクタを修正させる、C 3 2に記載の非一時的コンピュータ可読媒体。

10

20

30

【図 1】

図 1

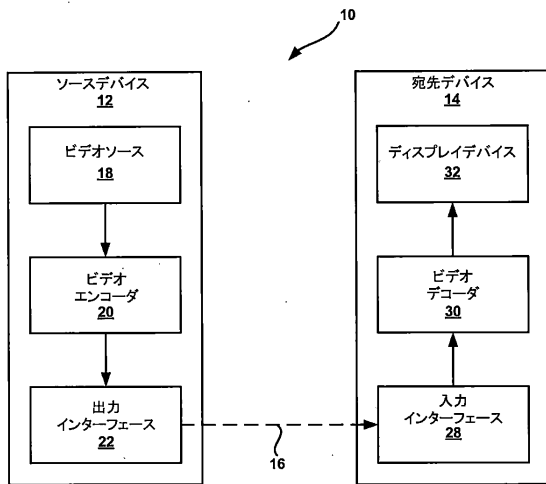


FIG. 1

【図 2】

図 2

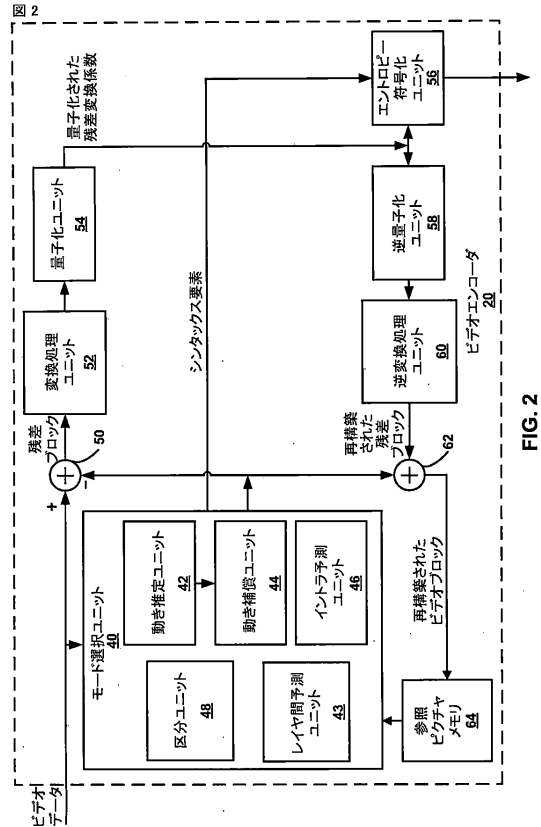


FIG. 2

【図 3】

図 3

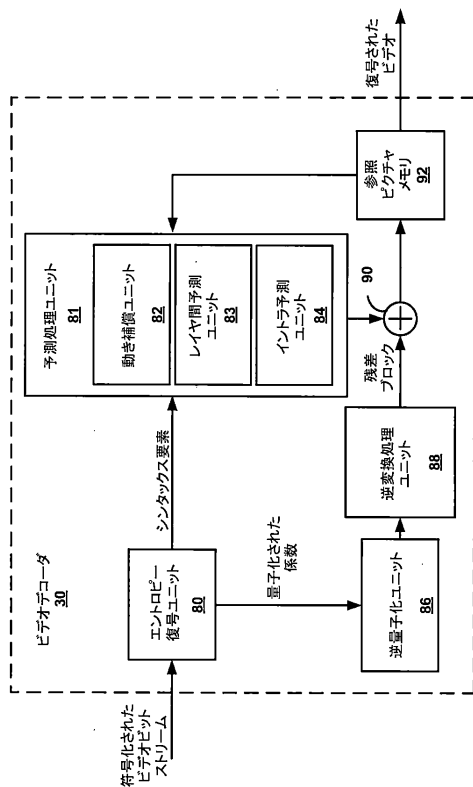


FIG. 3

【図 4】

図 4

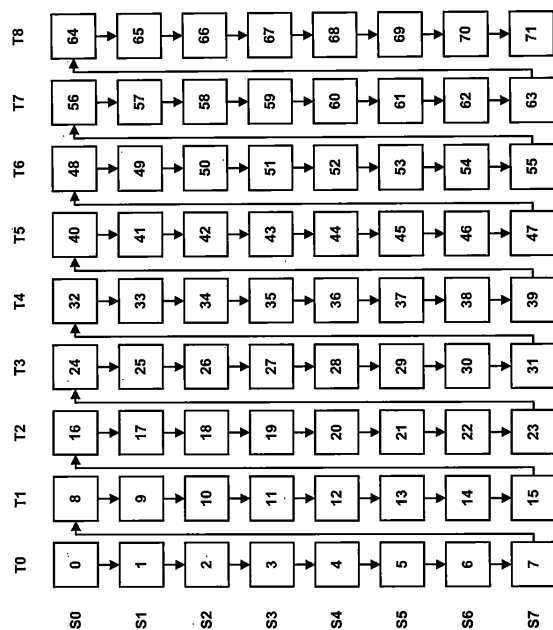
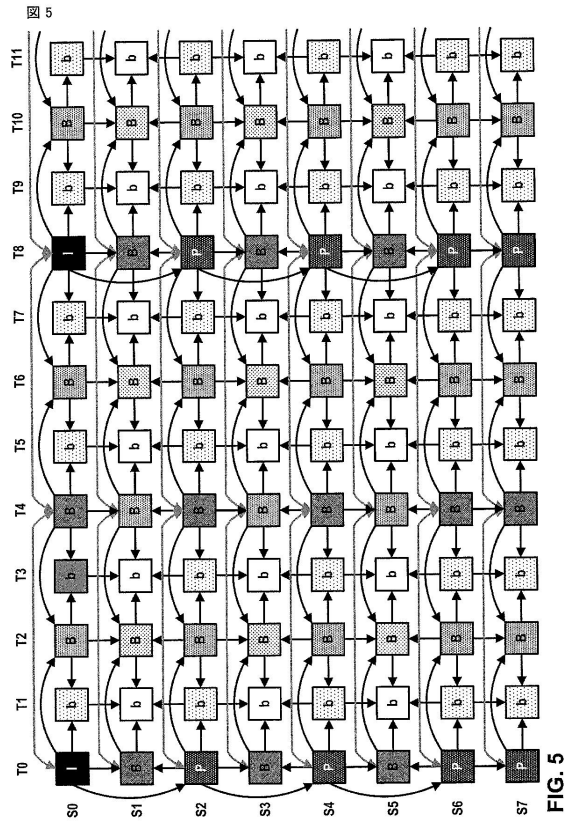
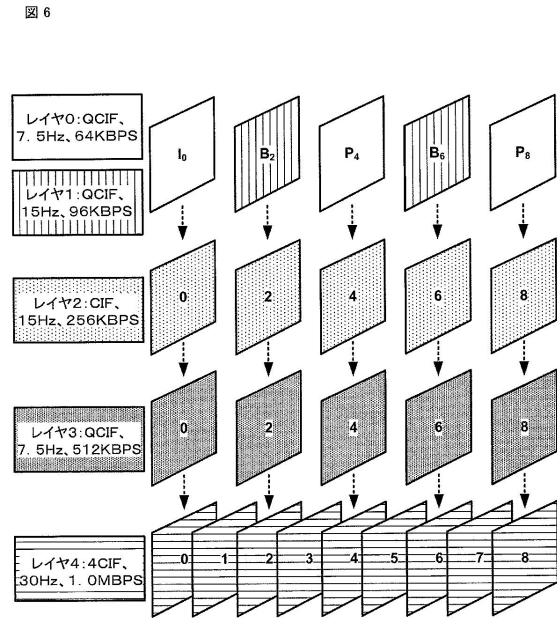


FIG. 4

【図 5】

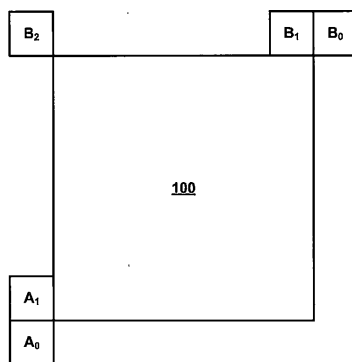


【図 6】



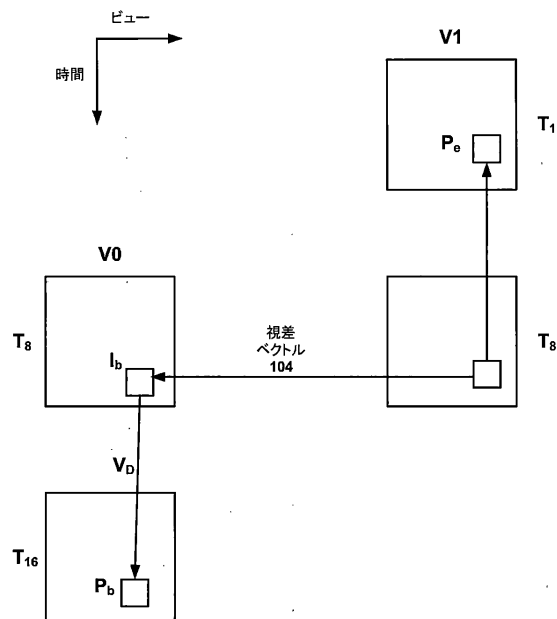
【図 7】

図 7



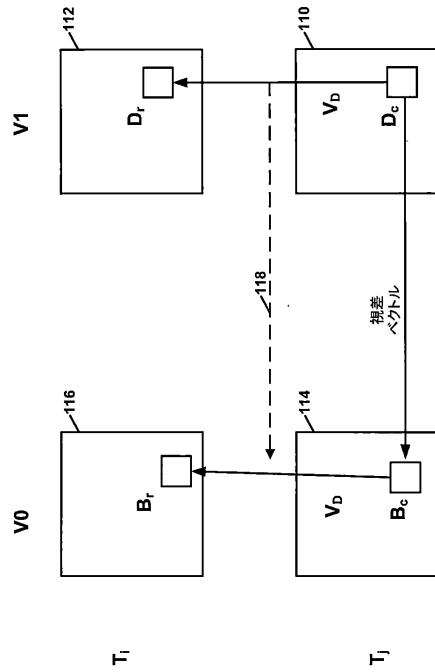
【図 8】

図 8



【図 9】

图 9



【図 10】

图 10

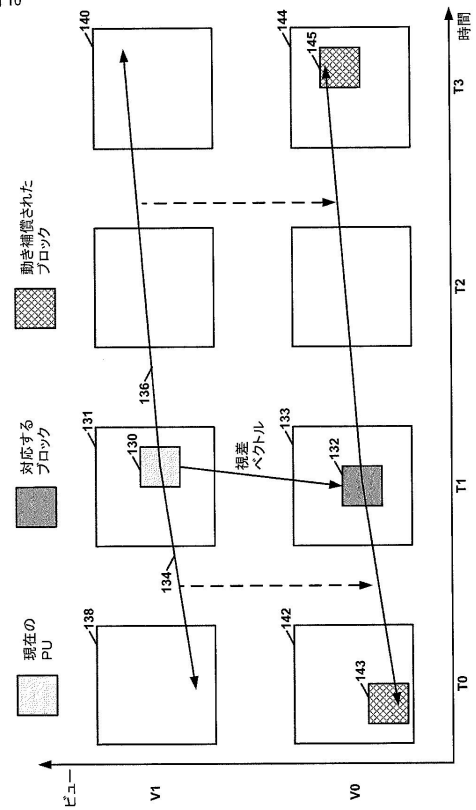


FIG. 10

【図 1 1】

图 11

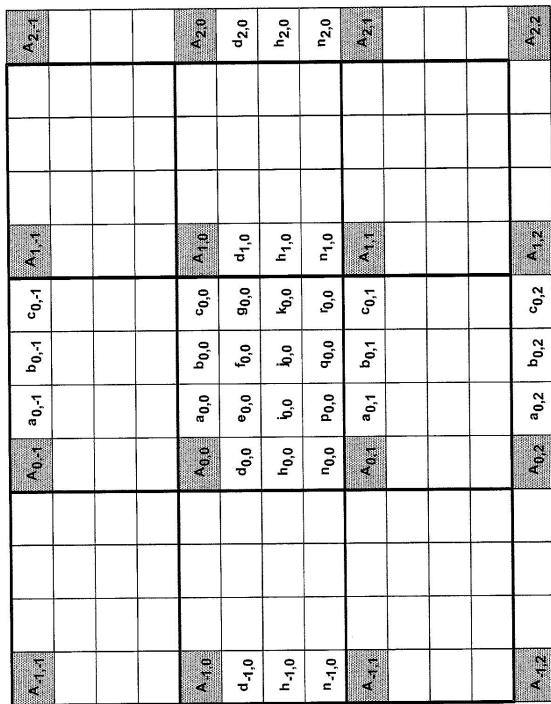


FIG. 11

【图 1 2】

图 12

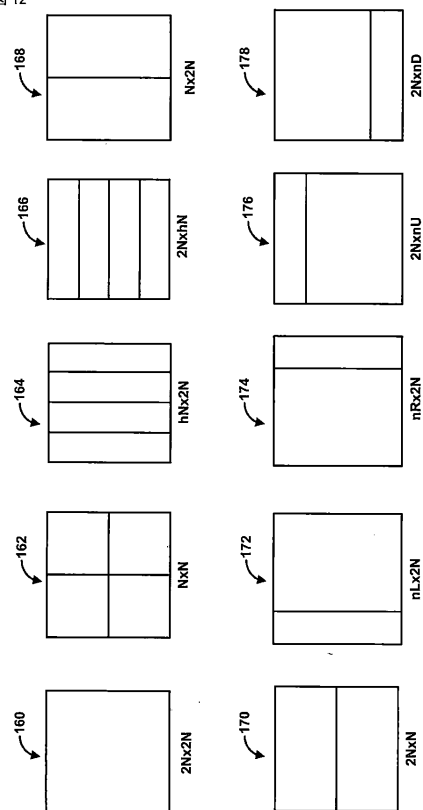


FIG. 12

【図 13】

図 13

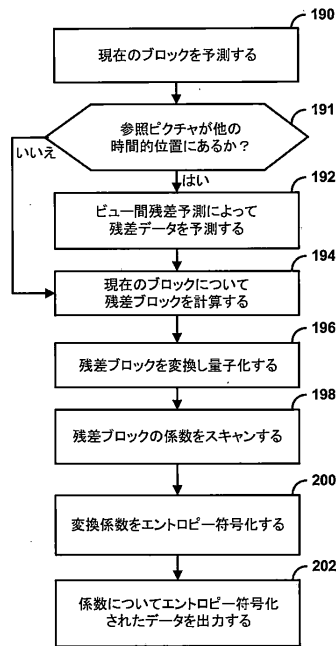


FIG. 13

【図 14】

図 14

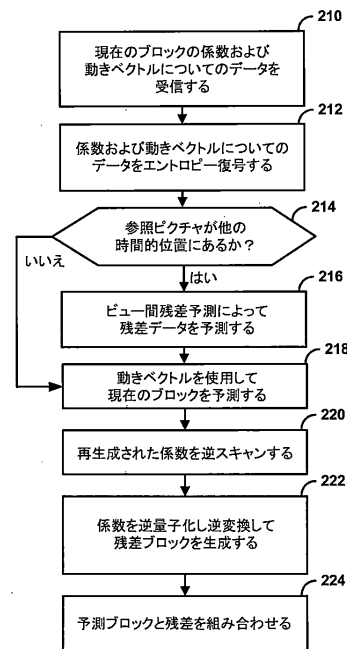


FIG. 14

【図 15】

図 15

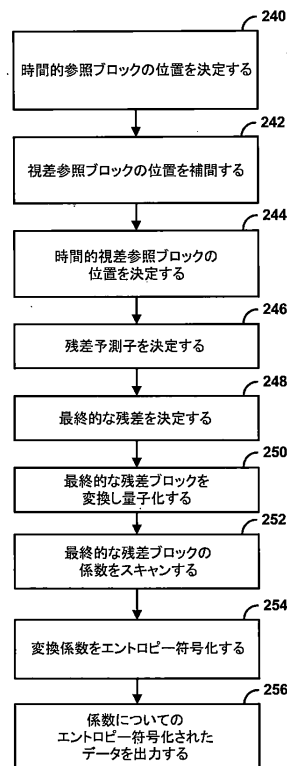


FIG. 15

【図 16】

図 16

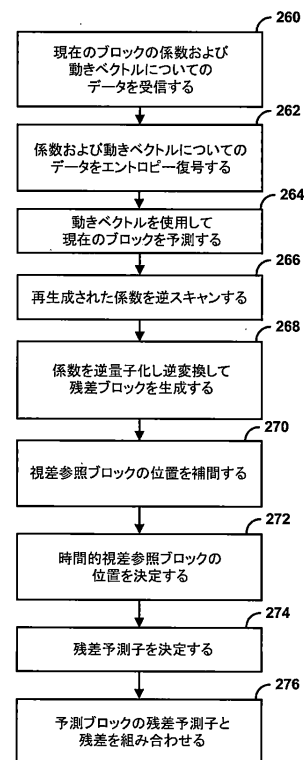


FIG. 16

【図 17】

図 17

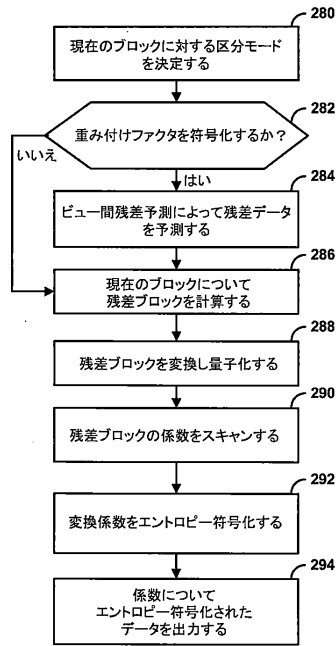


FIG. 17

【図 18】

図 18

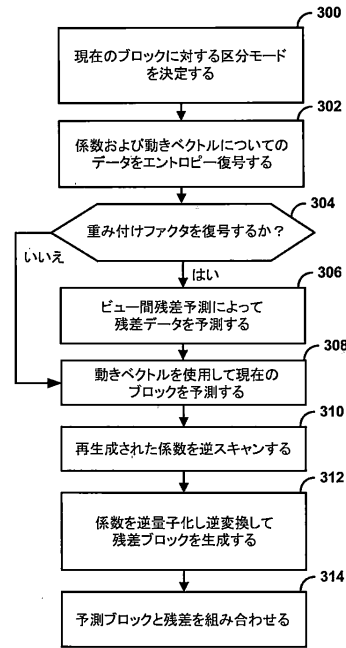


FIG. 18

【図 19】

図 19

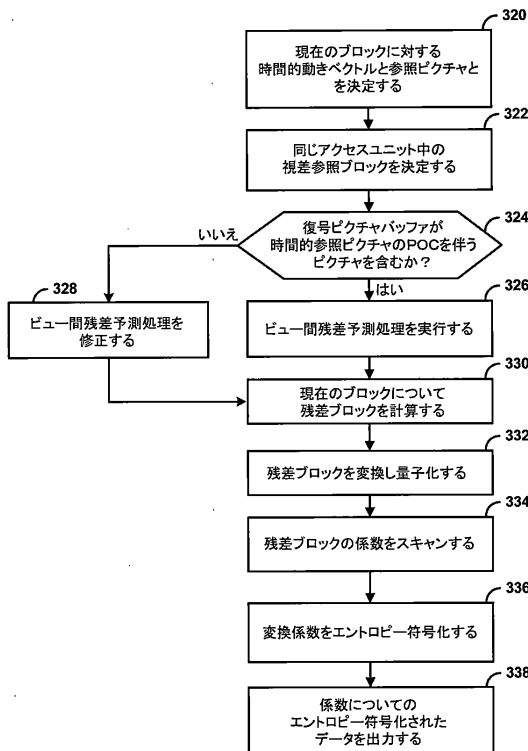


FIG. 19

【図 20】

図 20

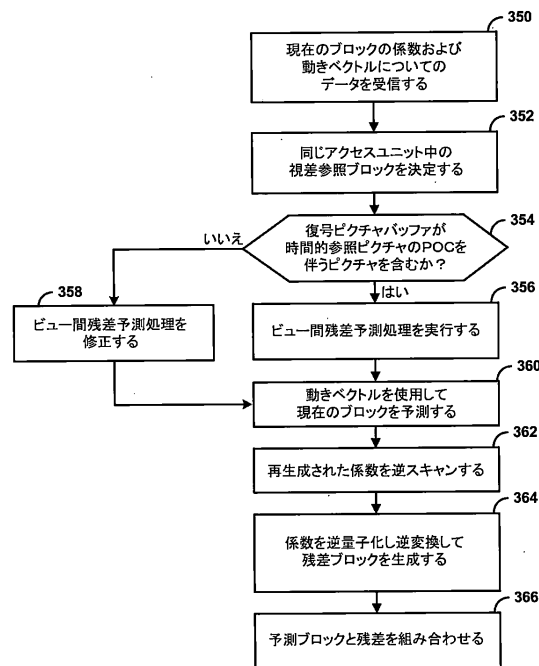


FIG. 20

フロントページの続き

- (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 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 カークゼウィックス、マルタ
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 赤穂 州一郎

- (56)参考文献 米国特許出願公開第2010/0091845 (US, A1)
特開2007-158770 (JP, A)
Jicheng An, et al., Removal of the parsing dependency of inter-view residual prediction, 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 2nd Meeting: Shanghai, SE, 13-19 Oct 2012, URL:http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/2-Shanghai/wg11/JCT3V-B0093-v1.zip, 2012年10月9日, JCT3V-B0093, 1~5.
Li Zhang, et al., 3D-CE5.h related: Advanced residual prediction for multiview coding, 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 2nd Meeting: Shanghai, CN, 13-19 Oct. 2012, URL:http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/2-Shanghai/wg11/JCT3V-B0051-v1.zip, 2012年10月10日, JCT3V-B0051, 1~4.
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 201, 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, 1~82.
Ying Chen, et al., Single-loop decoding and motion skip study in JMV, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JCT1/SC29/WG11 and ITU-T SG16 Q.6) 25th Meeting: ShenZhen, CN, 21-26 October, 2007, 2007年10月25日, JVT-Y053_r1, 1-12.
HEVC 3DVC extension text specification draft 1 based on High efficiency video coding(HEVC) text specification draft 7, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting: Stockholm, SE, 16-20 July 2012, URL:http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/1-Stockholm/wg11/JCT3V-A0012-v1.zip, 2012年7月11日, JCT3V-A0012-v1, 86,87,114,300.

- (58)調査した分野(Int.Cl., DB名)
H04N 19/00 - 19/98