

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6783799号
(P6783799)

(45) 発行日 令和2年11月11日 (2020. 11. 11)

(24) 登録日 令和2年10月26日 (2020. 10. 26)

(51) Int. Cl.	F I
HO 4 N 19/593 (2014. 01)	HO 4 N 19/593
HO 4 N 19/55 (2014. 01)	HO 4 N 19/55
HO 4 N 19/57 (2014. 01)	HO 4 N 19/57
HO 4 N 19/523 (2014. 01)	HO 4 N 19/523

請求項の数 16 (全 56 頁)

(21) 出願番号	特願2017-558550 (P2017-558550)	(73) 特許権者	507364838
(86) (22) 出願日	平成28年5月11日 (2016. 5. 11)		クアルコム、インコーポレイテッド
(65) 公表番号	特表2018-521539 (P2018-521539A)		アメリカ合衆国 カリフォルニア 921
(43) 公表日	平成30年8月2日 (2018. 8. 2)		21 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2016/031888		イブ 5775
(87) 国際公開番号	W02016/183216	(74) 代理人	100108453
(87) 国際公開日	平成28年11月17日 (2016. 11. 17)		弁理士 村山 靖彦
審査請求日	平成31年4月22日 (2019. 4. 22)	(74) 代理人	100163522
(31) 優先権主張番号	62/159, 839		弁理士 黒田 晋平
(32) 優先日	平成27年5月11日 (2015. 5. 11)	(72) 発明者	チャオ・パン
(33) 優先権主張国・地域又は機関	米国 (US)		アメリカ合衆国・カリフォルニア・902
(31) 優先権主張番号	62/173, 248		92・マリーナ・デル・レイ・ヴィア・マ
(32) 優先日	平成27年6月9日 (2015. 6. 9)		リーナ・4169・ユニット・307
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ビデオデータの特定のピクチャ内のインターコーディングのための探索範囲決定

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを符号化する方法であって、

前記ビデオデータの現在のピクチャ内の現在のブロックのための前記ビデオデータの前記現在のピクチャ内の予測子ブロックを識別するベクトルに使用されることになる解像度を決定するステップと、

前記決定された解像度に基づいて、前記現在のブロックのための探索範囲を決定するステップであって、前記現在のブロックのための前記探索範囲を決定するステップが、

前記ベクトルのために使用されることになる前記解像度が整数ピクセルである場合、初期探索範囲を前記現在のブロックのための前記探索範囲として使用し、前記初期探索範囲は前記現在のピクチャの再構成された領域であり、

前記ベクトルのために使用されることになる前記解像度が分数ピクセルである場合、前記探索範囲を決定するステップが、

前記初期探索範囲の右側境界および下側境界からMサンプルだけ前記初期探索範囲を縮小するステップと、

前記初期探索範囲の上側境界および左側境界からNサンプルだけ前記初期探索範囲を縮小するステップと、

前記縮小された初期探索範囲を前記現在のブロックのための前記探索範囲として使用するステップと

を備える、前記現在のブロックのための探索範囲を決定するステップと、

10

20

前記探索範囲内から、前記現在のブロックのための予測子ブロックを選択するステップと、

前記現在のブロックのための前記選択された予測子ブロックを識別する前記ベクトルを決定するステップと、

符号化ビデオビットストリーム内に、前記ベクトルの表現を符号化するステップとを備える、方法。

【請求項 2】

Mが2であり、Nが2である、請求項1に記載の方法。

【請求項 3】

前記ベクトルが、分数ピクセル解像度を有する彩度動きベクトルであり、前記ベクトルの前記表現を符号化するステップが、前記彩度動きベクトルが導出可能である輝度動きベクトルを符号化するステップを備える、請求項1に記載の方法。

10

【請求項 4】

前記ベクトルのために使用されることになる前記解像度を決定するステップが、前記現在のブロックのための彩度予測子ブロックを識別する彩度動きベクトルのために使用されることになる解像度を決定するステップを備え、前記方法が、

前記現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きい場合、分数ピクセル解像度が前記ベクトルのために使用されることになることを決定するステップをさらに備える、請求項1に記載の方法。

【請求項 5】

20

前記現在のブロックの彩度サンプリングフォーマットが4:2:2または4:2:0である場合、前記現在のブロックのための輝度サンプリングに対する彩度サンプリングの前記比が1よりも大きい、請求項4に記載の方法。

【請求項 6】

前記符号化ビデオビットストリーム内に、前記ベクトルのために使用されることになる前記解像度を示す構文要素を符号化するステップをさらに備える、請求項1に記載の方法。

【請求項 7】

前記ベクトルが分数ピクセル解像度を有する場合、分数ピクセル補間を使用し、前記予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、前記予測子ブロックのピクセル値を決定するステップと、

30

前記予測子ブロックの前記ピクセル値に基づいて前記現在のブロックを再構成するステップとをさらに備える、請求項1に記載の方法。

【請求項 8】

ビデオデータを復号する方法であって、

符号化ビデオビットストリームから、前記ビデオデータの現在のピクチャ内の現在のブロックのための前記ビデオデータの前記現在のピクチャ内の予測子ブロックを識別するベクトルの表現を取得するステップと、

前記ベクトルが分数ピクセル解像度または整数ピクセル解像度のどちらを有するかを決定するステップと、

40

前記ベクトルに基づいて、探索範囲内から前記現在のブロックのための予測子ブロックを決定するステップであって、

前記ベクトルが整数ピクセル解像度を有するとき、前記現在のブロックのための前記探索範囲が初期探索範囲を備え、前記初期探索範囲は前記現在のピクチャの再構成された領域であり、

前記ベクトルが分数ピクセル解像度を有するとき、前記現在のブロックのための前記探索範囲が、前記初期探索範囲の右側境界および下側境界からMサンプルだけ前記初期探索範囲を縮小し、前記初期探索範囲の上側境界および左側境界からNサンプルだけ前記初期探索範囲を縮小することによって決定された縮小探索範囲を備える、ステップと、

50

前記ベクトルが分数ピクセル解像度を有することを決定することに対応して、分数ピクセル補間を使用し、前記予測子ブロックの外部であるが、前記縮小探索範囲内からのサンプルピクセル値に少なくとも部分的に基づいて、前記予測子ブロックのピクセル値を決定するステップと、

前記予測子ブロックの前記ピクセル値に基づいて前記現在のブロックを再構成するステップと

を備える、方法。

【請求項 9】

Mが2であり、Nが2である、請求項8に記載の方法。

【請求項 10】

前記ベクトルが、彩度動きベクトルであり、前記動きベクトルの前記表現を取得するステップが、

前記符号化ビデオビットストリームから、輝度動きベクトルの表現を取得するステップと、

前記輝度動きベクトルに基づいて前記彩度動きベクトルを決定するステップとを備える、請求項8に記載の方法。

【請求項 11】

前記ベクトルが、彩度動きベクトルであり、前記方法が、

前記現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きい場合、分数ピクセル解像度が前記彩度動きベクトルのために使用されることを決定するステップをさらに備える、請求項8に記載の方法。

【請求項 12】

現在のブロックの彩度サンプリングフォーマットが4:2:2または4:2:0である場合、前記現在のブロックのための輝度サンプリングに対する彩度サンプリングの前記比が1よりも大きい、請求項11に記載の方法。

【請求項 13】

ビデオデータを符号化するための装置であって、

前記ビデオデータの現在のピクチャ内の現在のブロックのための前記ビデオデータの前記現在のピクチャ内の予測子ブロックを識別するベクトルのために使用されることになる解像度を決定するための手段と、

前記決定された解像度に基づいて、前記現在のブロックのための探索範囲を決定するための手段であって、

前記ベクトルのために使用されることになる前記解像度が整数ピクセルである場合、前記探索範囲を決定するための前記手段が、初期探索範囲を前記現在のブロックのための前記探索範囲として使用するように構成され、

前記ベクトルのために使用されることになる前記解像度が分数ピクセルである場合、前記探索範囲を決定するための前記手段が、

前記初期探索範囲の右側境界および下側境界からMサンプルだけ前記初期探索範囲を縮小し、

前記初期探索範囲の上側境界および左側境界からNサンプルだけ前記初期探索範囲を縮小し、

前記縮小された初期探索範囲を前記現在のブロックのための前記探索範囲として使用するように構成された、前記現在のブロックのための前記探索範囲を決定するための手段と、

前記探索範囲内から、前記現在のブロックのための予測子ブロックを選択するための手段と、

前記現在のブロックのための前記選択された予測子ブロックを識別する前記ベクトルを決定するための手段と、

符号化ビデオビットストリーム内に、前記ベクトルの表現を符号化するための手段とを備える、装置。

10

20

30

40

50

【請求項 14】

ビデオデータを復号するための装置であって、

符号化ビデオビットストリームから、前記ビデオデータの現在のピクチャ内の現在のブロックのための前記ビデオデータの前記現在のピクチャ内の予測子ブロックを識別するベクトルの表現を取得するための手段と、

前記ベクトルが分数ピクセル解像度または整数ピクセル解像度のどちらを有するのかを決定するための手段と、

前記ベクトルに基づいて、探索範囲内から前記現在のブロックのための予測子ブロックを決定するための手段であって、

前記ベクトルが整数ピクセル解像度を有するとき、前記現在のブロックのための前記探索範囲が、初期探索範囲を備え、 10

前記ベクトルが分数ピクセル解像度を有するとき、前記現在のブロックのための前記探索範囲が、前記初期探索範囲の右側境界および下側境界からMサンプルだけ前記初期探索範囲を縮小し、前記初期探索範囲の上側境界および左側境界からNサンプルだけ前記初期探索範囲を縮小することによって決定された縮小探索範囲を備える、手段と、

前記ベクトルが分数ピクセル解像度を有することを決定することに応答して、分数ピクセル補間を使用し、前記予測子ブロックの外部であるが、前記縮小探索範囲内からのサンプルピクセル値に少なくとも部分的に基づいて、前記予測子ブロックのピクセル値を決定するための手段と、

前記予測子ブロックの前記ピクセル値に基づいて前記現在のブロックを再構成するための手段と 20
を備える、装置。

【請求項 15】

実行されたとき、ビデオ符号化デバイスの1つまたは複数のプロセッサに請求項 1 から 7 のうちいずれか 1 項の方法を実行させる命令を記憶するコンピュータ可読記憶媒体。

【請求項 16】

実行されたとき、復号デバイスの1つまたは複数のプロセッサに請求項 8 から 12 のうちいずれか 1 項の方法を実行させる命令を記憶するコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

30

【0001】

関連出願

本出願は、2015年5月11日に出願した米国仮出願第62/159,839号、2015年6月9日に出願した米国仮出願第62/173,248号、および、2015年6月12日に出願した米国仮出願第62/175,179号に関するものであり、これらの各々の内容全体は、参照により本明細書に組み込まれている。

【0002】

本開示は、ビデオ符号化およびビデオ復号に関する。

【背景技術】

【0003】

40

デジタルビデオ機能は、デジタルテレビジョン、デジタル直接放送システム、ワイヤレス放送システム、携帯情報端末(PDA)、ラップトップまたはデスクトップコンピュータ、タブレットコンピュータ、電子書籍リーダー、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、セルラーまたは衛星無線電話、いわゆる「スマートフォン」ビデオ会議デバイス、ビデオストリーミングデバイスなどを含む、広範囲のデバイスに組み込まれることが可能である。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、高度ビデオコーディング(AVC:Advanced Video Coding)、ITU-T H.265、高効率ビデオコーディング(HEVC:High Efficiency Video Coding)、およびそのような規格の拡張によって定義される規格に記載のものなどの、ビデオ圧縮技法を実装する。ビデオデバイスは、そのよ 50

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、受信、符号化、復号、および/または記憶することがある。

【 0 0 0 4 】

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(ピクチャ内)予測および/または時間(ピクチャ間)予測を実行する。ブロックベースのビデオコーディングに関して、ビデオスライス(すなわち、ビデオピクチャ、またはビデオピクチャの一部)は、ビデオブロックに分割されることがあり、ビデオブロックはまた、ツリーブロック、コーディング単位(CU: coding unit)、および/またはコーディングノードと呼ばれることもある。ピクチャのイントラコーディング(I)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに対する空間予測を用いて符号化される。ピクチャのインターコーディング(PまたはB)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに対する空間予測、または他の参照ピクチャ内の参照サンプルに対する時間予測を使用してもよい。

10

【 0 0 0 5 】

空間または時間予測は、コーディングされるべきブロックのための予測ブロックをもたらす。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を表す。インターコーディングブロックは、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、コーディングブロックと予測ブロックとの間の差を示す残差データとに従って符号化される。イントラコーディングブロックは、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、ピクセル領域から変換領域に変換されてもよく、次いで量子化されてもよい残差変換係数をもたらす。

20

【 先行技術文献 】

【 非特許文献 】

【 0 0 0 6 】

【 非特許文献 1 】 Rapakaら、 「 On parallel processing capability of intra block copy」、 Document: JCTVC-S0220、 JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11、 第19回ミーティング: ストラスブール、 フランス、 2014年10月17 ~ 24日

【 非特許文献 2 】 ITUの電気通信標準化部門ITU-T、 シリーズH: オーディオビジュアルおよびマルチメディアシステム、 オーディオビジュアルサービスのインフラストラクチャ-動画像のコーディング: 高効率ビデオコーディング、 H.265、 2015年4月

30

【 非特許文献 3 】 高効率ビデオコーディング(HEVC)範囲拡張テキスト仕様: ドラフト7、 ITU-T SG 16WP3およびISO/IEC JTC1/SC29/WG11のビデオコーディングに関する共同のコラボレーションチーム(JCT-VC)、 第17回ミーティング: バレンシア、 スペイン、 2014年3月27日 ~ 4月4日、 Document: JCTVC-Q1005_v4

【 発明の概要 】

【 課題を解決するための手段 】

【 0 0 0 7 】

全体的には、本開示は、予測子ブロックが予測子ブロックを識別する動きベクトルの解像度に基づいて選択される探索範囲のサイズを調整するための、ビデオコーディングプロセスにおける技法について説明する。

40

【 0 0 0 8 】

一例では、ビデオデータを符号化するための方法は、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルに使用される解像度を決定するステップと、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合に現在のブロックのための探索範囲のサイズがより小さくなるように、決定された解像度に基づいて探索範囲を決定するステップと、探索範囲内から現在のブロックのための予測子ブロックを選択するステップと、現在のブロックのための選択された予測子ブロックを識別する動きベクトルを決定するステップと、コーディングビデオビットストリーム内に、動きベクトルの表現を符号化するステップと

50

を含む。

【 0 0 0 9 】

別の例では、ビデオデータを符号化するためのデバイスは、ビデオデータの一部を記憶するように構成されたメモリと、1つまたは複数のプロセッサとを含む。この例では、1つまたは複数のプロセッサは、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルに使用される解像度を決定し、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合に現在のブロックのための探索範囲のサイズがより小さくなるように、決定された解像度に基づいて探索範囲を決定し、探索範囲内から現在のブロックのための予測子ブロックを選択し、現在のブロックのための選択された予測子ブロックを識別する動きベクトルを決定し、コーディングビデオビットストリーム内に、動きベクトルの表現を符号化するように構成される。

10

【 0 0 1 0 】

別の例では、ビデオデータを符号化するための装置は、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルに使用される解像度を決定するための手段と、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合に現在のブロックのための探索範囲のサイズがより小さくなるように、決定された解像度に基づいて探索範囲を決定するための手段と、探索範囲内から現在のブロックのための予測子ブロックを選択するための手段と、現在のブロックのための選択された予測子ブロックを識別する動きベクトルを決定するための手段と、コーディングビデオビットストリーム内に、動きベクトルの表現を符号化するための手段とを含む。

20

【 0 0 1 1 】

別の例では、コンピュータ可読記憶媒体は、実行されたとき、ビデオ符号化デバイスの1つまたは複数のプロセッサに、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルに使用される解像度を決定させ、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合に現在のブロックのための探索範囲のサイズがより小さくなるように、決定された解像度に基づいて探索範囲を決定させ、探索範囲内から現在のブロックのための予測子ブロックを選択させ、現在のブロックのための選択された予測子ブロックを識別する動きベクトルを決定させ、コーディングビデオビットストリーム内に、動きベクトルの表現を符号化させる命令を記憶する。

30

【 0 0 1 2 】

別の例では、ビデオデータを復号するための方法は、コーディングビデオビットストリームから、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルの表現を取得するステップと、動きベクトルが分数ピクセル解像度または整数ピクセル解像度のどちらであるのかを決定するステップと、動きベクトルに基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲内から現在のブロックのための予測子ベクトルを決定するステップと、動きベクトルが分数ピクセル解像度を有すると決定したことに応答して、分数ピクセル補間を使用し、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定するステップと、予測子ブロックのピクセル値に基づいて現在のブロックを再構成するステップとを含む。

40

【 0 0 1 3 】

別の例では、ビデオデータを復号するためのデバイスは、ビデオデータの一部を記憶するように構成されたメモリと、1つまたは複数のプロセッサとを含む。この例では、1つまたは複数のプロセッサは、コーディングビデオビットストリームから、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルの表現を取得し、動きベクトルが分数ピクセル解像度または

50

整数ピクセル解像度のどちらであるのかを決定し、動きベクトルに基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲内から現在のブロックのための予測子ベクトルを決定し、動きベクトルが分数ピクセル解像度を有すると決定したことに応答して、分数ピクセル解像度を使用し、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定し、予測子ブロックのピクセル値に基づいて現在のブロックを再構成するように構成される。

【 0 0 1 4 】

別の例では、ビデオデータを復号するための装置は、コーディングビデオビットストリームから、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルの表現を取得するための手段と、動きベクトルが分数ピクセル解像度または整数ピクセル解像度のどちらであるのかを決定するための手段と、動きベクトルに基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲内から現在のブロックのための予測子ベクトルを決定するための手段と、動きベクトルが分数ピクセル解像度を有すると決定したことに応答して、分数ピクセル解像度を使用し、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定するための手段と、予測子ブロックのピクセル値に基づいて現在のブロックを再構成するための手段とを含む。

【 0 0 1 5 】

別の例では、コンピュータ可読記憶媒体は、実行されたとき、ビデオ復号デバイスの1つまたは複数のプロセッサに、コーディングビデオビットストリームから、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルの表現を取得させ、動きベクトルが分数ピクセル解像度または整数ピクセル解像度のどちらであるのかを決定させ、動きベクトルに基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲内から現在のブロックのための予測子ベクトルを決定させ、動きベクトルが分数ピクセル解像度を有すると決定したことに応答して、分数ピクセル補間を使用し、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定させ、予測子ブロックのピクセル値に基づいて現在のブロックを再構成させる命令を記憶する。

【 0 0 1 6 】

本開示の1つまたは複数の態様の詳細を添付の図面および以下の説明に記載する。本開示で説明する技法の他の特徴、目的、および利点は、これらの説明および図面、ならびに特許請求の範囲から明らかになる。

【図面の簡単な説明】

【 0 0 1 7 】

【図 1】本開示の技法を実装することができ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である。

【図 2】本開示の1つまたは複数の技法による、ピクチャの例示的なビデオシーケンスを示す概念図である。

【図 3】本開示に記載のイントラブロックコピーのための技法を使用することができるビデオエンコーダの一例を示すブロック図である。

【図 4】本開示に記載の技法を実装することができるビデオデコーダの一例を示すブロック図である。

【図 5】本開示の1つまたは複数の技法による、イントラブロックコピーイングプロセスの一例を示す図である。

【図 6 A】本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。

【図 6 B】本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的

10

20

30

40

50

な探索範囲を示す図である。

【図7】本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。

【図8】本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。

【図9】本開示の1つまたは複数の技法による、ピクチャの外部に位置する領域をパディングするための例示的な技法を示す図である。

【図10】本開示の1つまたは複数の技法による、クリッピング動作のいくつかの例示的なシナリオを示す図である。

【図11】本開示の1つまたは複数の技法による、クリッピング動作の追加の例示的なシナリオを示す図である。

【図12】本開示の1つまたは複数の技法による、ビデオデータのブロックのための探索範囲を決定するための例示的なプロセスを示すフローチャートである。

【図13】本開示の1つまたは複数の技法による、ビデオデータのブロックを復号するための例示的なプロセスを示すフローチャートである。

【発明を実施するための形態】

【0018】

ビデオシーケンスは、一般に、ピクチャのシーケンスとして表される。典型的には、ブロックベースのコーディング技法は、個々のピクチャの各々をコーディングするために使用される。すなわち、各ピクチャは、ブロックに分割され、各々のブロックは、個別にコーディングされる。ビデオデータのブロックをコーディングすることは、一般に、ブロック内のピクセルのための予測値を形成することと、残差値をコーディングすることを含む。予測値は、1つまたは複数の予測ブロック内のピクセルサンプルを使用して形成される。残差値は、元のブロックのピクセルと予測ピクセル値との間の差を表す。具体的には、ビデオデータの元のブロックは、ピクセル値のアレイを含み、予測ブロックは、予測ピクセル値のアレイを含む。残差値は、元のブロックのピクセル値と予測ピクセル値との間のピクセルごとの差を表す。

【0019】

ビデオデータのブロックのための予測技法は、一般に、イントラ予測およびインター予測に分類される。イントラ予測または空間予測は、任意の参照ピクチャからの予測を含まない。代わりに、ブロックは、隣接する以前にコーディングされたブロックのピクセル値から予測される。インター予測または時間予測は、一般に、1つまたは複数の参照ピクチャリスト(RPL)から選択された1つまたは複数の以前にコーディングされた参照ピクチャ(たとえば、フレームまたはスライス)のピクセル値からブロックを予測することを含む。ビデオコードは、RPLに含まれるピクチャを記憶するように構成された1つまたは複数の参照ピクチャバッファを含んでもよい。

【0020】

リモートデスクトップ、リモートゲーム、ワイヤレスディスプレイ、車載インフォテインメント、クラウドコンピューティングなどの多くの用途は、日々の生活の中で日常的になってきている。これらの用途でのビデオコンテンツは、通常、自然なコンテンツ、テキスト、人工グラフィックスなどの組合せである。テキストおよび人工グラフィックス領域では、繰り返されるパターン(文字、アイコン、シンボルなど)がしばしば存在する。イントラブロックコピーイング(イントラBC)は、ビデオコードがそのような冗長性を除去し、イントラピクチャコーディング効率を改善することを可能にし得る技法である。いくつかの例では、イントラBCは、代替的に、イントラ動き補償(MC:motion compensation)と呼ばれることもある。

【0021】

いくつかのイントラBC技法によれば、ビデオコードは、現在のブロックのピクセルの予測のためにビデオデータの現在のブロックと同じピクチャ内にある以前にコーディングされたビデオデータのブロック内の再構成されたピクセルを使用してもよい。いくつかの例

10

20

30

40

50

では、以前にコーディングされたビデオデータのブロックは、予測子ブロックまたは予測ブロックと呼ばれることもある。ビデオコードは、予測子ブロックを識別するために動きベクトルを使用してもよい。いくつかの例では、動きベクトルはまた、ブロックベクトル、オフセットベクトル、または変位ベクトルと呼ばれることもある。いくつかの例では、ビデオコードは、予測子ブロックを識別するために1次元動きベクトルを使用してもよい。したがって、いくつかのビデオコードは、x値の同じセット(すなわち、現在のブロックと垂直方向に並んだ)またはy値の同じセットのみを共有する(すなわち、現在のブロックと水平方向に並んだ)、以前にコーディングされたビデオデータのブロックに基づいて、ビデオデータの現在のブロックを予測することができる。他の例では、ビデオコードは、予測子ブロックを識別するために2次元動きベクトルを使用してもよい。たとえば、ビデオコードは、水平変位成分と垂直変位成分とを有する2次元動きベクトルを使用してもよく、各々の成分は、ゼロまたは非ゼロであり得る。水平変位成分は、ビデオデータの予測子ブロックと、ビデオデータの現在のブロックとの間の水平変位を表すことができ、垂直変位成分は、ビデオデータの予測子ブロックと、ビデオデータの現在のブロックとの間の垂直変位を表すことができる。

【0022】

イントラBCに関して、予測子ブロックのピクセルは、コーディングされているブロック(すなわち、現在のブロック)内の対応するピクセルのための予測サンプルとして使用され得る。ビデオコードは、加えて、ビデオデータの現在のブロックおよび予測ブロックに基づいて、ビデオデータの残差ブロックを決定することができ、2次元動きベクトルとビデオデータの残差ブロックとをコーディングすることができる。

【0023】

いくつかの例では、イントラBCは、特に画面コンテンツコーディングのための効率的なコーディングツールである。たとえば、いくつかの例では、イントラBCを使用してブロックをコーディングすることは、インターまたはイントラコーディングを使用してブロックをコーディングすることによって生成されるビットストリームよりも小さいビットストリームをもたらすことができる。上記で説明したように、イントラBCは、インターに似たコーディングツール(ピクチャのピクセル値がピクチャ内の他のピクセル値から予測されることを意味する)であるが、コーディングされるブロックと同じピクチャからの参照データを使用する。いくつかの例では、実用的な設計では好ましくない可能性がある、イントラBCに適用される1つまたは複数の制約のため、イントラBCを従来のイントラピクチャに統合することは困難である可能性がある。いくつかの例示的な制約は、限定はしないが、予測ブロックが、コーディングされるべき現在のブロックとして、同じスライスまたはタイル内になければならないこと、予測子ブロックがコーディングされるべき現在のブロックと重複してはならないこと、予測子ブロック内のすべてのピクセルが再構成されなければならないこと、(たとえば、Rapakaら、「On parallel processing capability of intra block copy」、Document: JCTVC-S0220, JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11、第19回ミーティング: ストラスブール、フランス、2014年10月17~24日(以下「JCTVC-S0220」)に記載のように並列化実装に関する考慮事項のために)予測子ブロックが特定の領域内にあること、および、制約されたイントラ予測が有効になっているとき、予測子ブロックが従来のインターモードを使用してコーディングされたいかなるピクセルをも含んではならないことを含む。加えて、いくつかの例では、従来のイントラおよびインターフレームのためのハードウェアアーキテクチャは、(たとえば、イントラBCがピクチャ内部のブロックコピーをもたらすため)変更なしでイントラBSに再利用することはできない可能性がある。そのように、イントラBCに現在適用されている制約の一部またはすべてを維持しながら、ハードウェアアーキテクチャへの(重大な)変更なしに、ビデオコードがイントラBCによって提供される効率を増すことができるようにすることが望ましい可能性がある。

【0024】

いくつかの例では、従来のイントラ予測技法を使用して現在のピクチャ内のサンプルに

10

20

30

40

50

基づいて現在のピクチャのブロックを予測することとは対照的に、ビデオコードは、従来のインター予測と同様の技法を使用して現在のピクチャ内のサンプルに基づいて現在のピクチャ内のブロックを予測するためにイントラBCを実行することができる。たとえば、ビデオコードは、現在のピクチャを予測するために使用される参照ピクチャリスト(RPL)内に現在のピクチャを含み、参照ピクチャバッファ内に現在のピクチャのバージョン(または、再構成されている現在のピクチャの少なくとも一部)を記憶し、参照ピクチャバッファ内に記憶された現在のピクチャのバージョン内に含まれるビデオデータの予測子ブロックに基づいて、現在のピクチャ内の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。このようにして、ビデオコードは、イントラBCに現在適用されている制約の一部またはすべてを維持しながら、イントラBCによって提供される効率を増すことができる。また、このようにして、ビデオコードは、重大な変更なしに、イントラBCのために従来のイントラおよびインターフレームのためのハードウェアアーキテクチャを再利用することができる。

10

【0025】

上記で説明したように、ビデオエンコーダは、同じピクチャ内からビデオデータの現在のブロックのための予測子ブロックを選択してもよい。いくつかの例では、ビデオエンコーダは、いくつかの候補予測子ブロックを評価し、ピクセル差の点から現在のブロックに密接に一致する候補予測子ブロックを選択してもよく、これは、絶対差合計(SAD)、自乗差合計(SSD)、または他の差分メトリックによって決定されてもよい。

20

【0026】

いくつかの例では、現在のピクチャ内の予測子ブロックを識別するために使用される動きベクトルは、整数ピクセル解像度を有してもよい。たとえば、動きベクトルは、単一のピクセルの増分内に現在のブロックと予測子ブロックとの間の変位を表す1つまたは複数の整数を含んでもよい。一例として、整数ピクセル解像度を有する動きベクトルは、現在のブロックと予測子ブロックとの間の水平変位を表す第1の整数(たとえば、3)と、現在のブロックと予測子ブロックとの間の垂直変位を表す第2の整数(たとえば、2)とを含んでもよい。

【0027】

いくつかの例では、現在のピクチャ内の予測子ブロックを識別するために使用される動きベクトルは、分数ピクセル解像度を有してもよい。たとえば、動きベクトルは、単一のピクセル未満の増分内に現在のブロックと予測子ブロックとの間の変位を表す1つまたは複数の値を含んでもよい。分数ピクセル動きベクトルが有し得るいくつかの例示的な解像度は、ハーフピクセル解像度(たとえば、 $1/2$ pel解像度)、 $1/4$ ピクセル解像度(たとえば、 $1/4$ pel解像度)、および $1/8$ ピクセル解像度(たとえば、 $1/8$ pel解像度)などを含んでもよいが、必ずしもこれらに限定されない。一例として、 $1/4$ ピクセル解像度を有する動きベクトルは、現在のブロックと予測子ブロックとの間の水平変位を表す第1の値(たとえば、2.75)と、現在のブロックと予測子ブロックとの間の垂直変位を表す第2の値(たとえば、2.5)とを含んでもよい。

30

【0028】

動きベクトルが整数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にある可能性があり、したがって、ビデオコードは、補間なしで前記サンプルピクセル値にアクセスすることができる。ビデオコードは、補間なしでサンプルピクセルにアクセスすることができるので、ビデオコードは、動きベクトルが整数ピクセル解像度を有する場合、現在のブロックを予測するために予測子ブロック内に位置するサンプルピクセル値のみを使用することができる。動きベクトルが分数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にないことがあり、したがって、ビデオコードは、サンプルピクセルを構成するために補間を実行する必要がある可能性がある。いくつかの例では、サンプルピクセル値を構成するために補間を実行するために、ビデオコードは、現在のブロックを予測するために、予測子ブロックの内部と

40

50

外部の両方に位置するサンプルピクセル値を使用する必要がある可能性がある。しかしながら、いくつかの例では、現在のブロックを予測するために、ビデオコードが予測子ブロックの外部に位置するサンプルピクセル値を使用することは望ましくないことがある。たとえば、予測子ブロックおよび現在のブロックが現在のピクチャ内に位置するとき、予測子ブロックの外部に位置するサンプルは、利用可能ではない可能性がある(すなわち、そのようなサンプルは、現在のピクチャのすでに再構成された範囲内に位置しない可能性がある)ので、ビデオデコーダが予測子ブロックの外部に位置するサンプルピクセル値を使用することができない可能性がある。

【0029】

本開示の1つまたは複数の技法によれば、ビデオコードは、予測子ブロックを識別する動きベクトルのために使用されるべき解像度に基づいて決定された探索範囲内から現在のブロックのための予測子ブロックを選択してもよい。たとえば、ビデオコードは、動きベクトルのために使用されるべき解像度が整数ピクセルであるときよりも、動きベクトルのために使用されるべき解像度が分数ピクセルであるときにより小さい探索範囲を使用してもよい。一例として、動きベクトルのために使用されるべき解像度が整数ピクセルであるとき、ビデオコードは、現在のピクチャの再構成された範囲を含む初期探索範囲内から予測子ブロックを選択してもよい。別の例として、動きベクトルのために使用されるべき解像度が分数ピクセルであるとき、ビデオコードは、初期探索範囲の右側境界および下側境界からMサンプルだけ初期探索範囲のサイズを縮小し、初期探索範囲の上側境界および左側境界からNサンプルだけ初期探索範囲のサイズを縮小することによって決定される縮小探索範囲内から予測子ブロックを選択してもよい。このようにして、ビデオコードは、予測子ブロックの外部に位置するサンプルピクセル値を含む、予測子ブロックを構成するために必要なすべてのサンプルピクセル値が、予測子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ときに使用するために利用可能であることを保証することができる。

【0030】

本開示は、現在のピクチャの一部を予測するとき、参照ピクチャとして現在のピクチャを利用することに関連する例示的な技法について説明する。理解を支援するために、例示的な技法は、4:4:4、4:2:2、4:2:0、4:0:0などを含む、できる限り高いビット深度の(たとえば、8ビットよりも高い)異なる彩度サンプリングフォーマットのサポートを含む、高効率ビデオコーディング(HEVC)ビデオコーディング規格への範囲拡張(RExt: range extension)に関して説明される。技法はまた、画面コンテンツコーディングのために適用可能であり得る。技法は、範囲拡張または画面コンテンツコーディングに限定されず、一般的に、規格ベースまたは非規格ベースのビデオコーディングを含むビデオコーディング技法に適用可能であり得ることが理解されるべきである。また、本開示に記載の技法は、将来的に開発される規格の一部になり得る。言い換えれば、本開示に記載の技法は、以前に開発されたビデオコーディング規格、現在開発中のビデオコーディング規格、および今後のビデオコーディング規格に適用可能であり得る。

【0031】

近年、新しいビデオコーディング規格の設計、すなわち、高効率ビデオコーディング(HEVC)が、ITU-Tビデオコーディングエキスパートグループ(VCEG)およびISO/IECモーションピクチャエキスパートグループ(MPEG)のビデオコーディングに関する共同のコラボレーションチーム(JCT-VC: Joint Collaboration Team on Video Coding)によって完成された。以後、HEVCバージョン1と呼ぶ、完成されたHEVC仕様は、ITUの電気通信標準化部門ITU-T、シリーズH: オーディオビジュアルおよびマルチメディアシステム、オーディオビジュアルサービスのインフラストラクチャ-動画像のコーディング: 高効率ビデオコーディング、H.265、2015年4月と題し、<http://www.itu.int/rec/T-REC-H.265-201504-1>から利用可能である。HEVCへの範囲拡張(RExt: range extension)、すなわち、HEVC RExtはまた、JCT-VCによって開発されている。以後、「RExt WD7」と呼ぶ、高効率ビデオコーディング(HEVC)範囲拡張テキスト仕様: ドラフト7、ITU-T SG 16WP3およびISO/IEC JTC1/SC29/WG11のビデオコーディングに関する共同のコラボレーションチーム(JCT-VC)、第17回ミーティング

: バレンシア、スペイン、2014年3月27日～4月4日、Document: JCTVC-Q1005_v4と題する範囲拡張の最近の作業草案(WD: Working Draft)は、http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zipから利用可能である。

【 0 0 3 2 】

範囲拡張仕様は、HEVC仕様のバージョン2となる可能性がある。しかしながら、大きな範囲で、提案された技法が関係する限り、たとえば、動きベクトル(MV)予測、HEVCバージョン1、および範囲拡張仕様は、技術的に類似する。したがって、変更がHEVCバージョン1に基づくと呼ばれるときはいつも、同じ変更は、範囲拡張使用に適合することができ、HEVCバージョン1モジュールが説明されるときはいつも、説明はまた、(同じ従属節(sub-clause)を有する)HEVC範囲拡張モジュールに適合可能であり得る。

10

【 0 0 3 3 】

近年、動きを有するテキストおよびグラフィックスなどの画面コンテンツ材料のための新しいコーディングツールの調査が要求され、画面コンテンツのコーディング効率を改善する技術が提案されている。新規な専用コーディングツールを用いて画面コンテンツの特徴を活用することによってコーディング効率の大幅な改善が取得され得るという証拠があるので、画面コンテンツコーディング(SCC: screen content coding)のための特定のツールを含む高効率ビデオコーディング(HEVC)標準の将来の拡張をできる限り開発することを目標に、発表募集(CfP: Call for Proposals)が発行されている。以後、「SCC WD6」と呼ぶ、SCC仕様の最近の作業草案(WD)は、http://phenix.it-sudparis.eu/jct/doc_end_user/documents/22_Geneva/wg11/JCTVC-W1005-v1.zipから利用可能である。

20

【 0 0 3 4 】

図1は、本開示の技法を実装することができる例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示すように、システム10は、宛先デバイス14によって後で復号されるべき符号化ビデオデータを提供するソースデバイス12を含む。具体的には、ソースデバイス12は、コンピュータ可読媒体16を介して宛先デバイス14にビデオデータを提供する。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、表示デバイス、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、広い範囲のデバイスのうちのいずれかを含んでもよい。いくつかの場合には、ソースデバイス12および宛先デバイス14は、ワイヤレス通信のために装備されてもよい。

30

【 0 0 3 5 】

宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化ビデオデータを受信してもよい。コンピュータ可読媒体16は、ソースデバイス12から宛先デバイス14に符号化ビデオデータを移動させることができる任意のタイプの媒体またはデバイスを備えてもよい。一例では、コンピュータ可読媒体16は、ソースデバイス12がリアルタイムで宛先デバイス14に符号化ビデオデータを直接送信することができるようにする通信媒体を備えてもよい。符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス14に送信されてもよい。通信媒体は、無線周波数(RF)スペクトルなどの任意のワイヤレスもしくはワイヤード通信媒体、または、1つもしくは複数の物理的伝送線を備えてもよい。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどの、パケットベースのネットワークの一部を形成してもよい。通信媒体は、ルータ、スイッチ、基地局、または、ソースデバイス12から宛先デバイス14への通信を容易にするために有用であってもよい任意の他の機器を含んでもよい。

40

【 0 0 3 6 】

いくつかの例では、符号化データは、ソースデバイス12の出力インターフェース22から記憶デバイス32に出力されてもよい。同様に、符号化データは、宛先デバイス14の入力インターフェース28によって記憶デバイス32からアクセスされてもよい。記憶デバイス32は

50

、ハードドライブ、Blu-ray(登録商標)ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性もしくは不揮発性メモリ、または符号化ビデオデータを記憶するための任意の他の適切なデジタル記憶媒体などの、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んでもよい。さらなる例では、記憶デバイス32は、ソースデバイス12によって生成された符号化ビデオを記憶することができるファイルサーバまたは別の中間記憶デバイスに対応してもよい。

【0037】

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記憶デバイス32から記憶されたビデオデータにアクセスしてもよい。ファイルサーバは、符号化ビデオデータを記憶し、宛先デバイス14にその符号化ビデオデータを送信することが可能な任意のタイプのサーバであってもよい。例示的なファイルサーバは、(たとえば、ウェブサイトのための)ウェブサーバ、FTPサーバ、ネットワーク接続ストレージ(NAS:network attached storage)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介して符号化ビデオデータにアクセスしてもよい。これは、ファイルサーバに記憶された符号化ビデオデータにアクセスするのに適した、ワイヤレスチャネル(たとえば、Wi-Fi接続)、ワイヤード接続(たとえば、DSL、ケーブルモデムなど)、または両方の組合せを含んでもよい。記憶デバイスからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組合せであってもよい。

【0038】

本開示の技法は、必ずしもワイヤレス用途または設定に限定されない。技法は、無線(over-the-air)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、インターネットストリーミングビデオ送信、動的適応型HTTPストリーミング(DASH: dynamic adaptive streaming over HTTP)、データ記憶媒体上で符号化されたデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の用途などの、様々なマルチメディア用途のいずれかを支持してビデオコーディングに適用されてもよい。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、ビデオ放送、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ビデオ送信をサポートするように構成されてもよい。

【0039】

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、表示デバイス31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、ビデオコーディングにおける変換を実行する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、他の構成要素または配置を含んでもよい。たとえば、ソースデバイス12は、外部カメラなどの外部ビデオソース18からビデオデータを受信してもよい。同様に、宛先デバイス14は、一体化された表示デバイスを含むのではなく、外部表示デバイスとインターフェースしてもよい。

【0040】

図1の図示のシステム10は、単なる一例である。ビデオコーディングにおける改善されたイントラブロックコピー合図(signaling)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てもよい。一般に、本開示の技法は、ビデオ符号化または復号デバイスによって実行されるが、技法はまた、組み合わされたビデオコーデックによって実行されてもよい。さらに、本開示の技法はまた、ビデオプリプロセッサによって実行されてもよい。ソースデバイス12および宛先デバイス14は、単に、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そのようなコーディングデバイスの例である。いくつかの例では、デバイス12、14は、デバイス12、14の各々がビデオ符号化および復号構成要素を含むように実質的に対称的な方法で動作してもよい。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、ビデオ放送、またはビデオ電話のための、ビデオデバイス12、14間の一

方向または双方向ビデオ送信をサポートしてもよい。

【0041】

ビデオデバイス12のビデオソース18は、ビデオカメラ、以前にキャプチャされたビデオを含むビデオアーカイブ、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ビデオ供給インターフェースなどの、ビデオキャプチャデバイスを含んでもよい。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コンピュータグラフィックスベースのデータを、または、ライブビデオ、アーカイブされたビデオ、およびコンピュータ生成ビデオの組合せを生成してもよい。いくつかの場合では、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、いわゆるカメラ電話またはビデオ電話を形成してもよい。上述したように、しかしながら、本開示に記載の技法は、一般に、ビデオコーディングに適用可能であり得、ワイヤレスおよび/またはワイヤード用途に適用されてもよい。各々の場合で、キャプチャされた、事前にキャプチャされた、またはコンピュータ生成されたビデオは、ビデオエンコーダ20によって符号化されてもよい。符号化ビデオ情報は、次いで、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上に出力されてもよい。

10

【0042】

コンピュータ可読媒体16は、ワイヤレス放送もしくはワイヤードネットワーク送信などの一時的媒体、または、ハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray(登録商標)ディスク、もしくは他のコンピュータ可読媒体などの記憶媒体(すなわち、非一時的記憶媒体)を含んでもよい。いくつかの例では、ネットワークサーバ(図示せず)は、ソースデバイス12から符号化ビデオデータを受信してもよく、たとえば、ネットワーク送信を介して、宛先デバイス14に符号化ビデオデータを提供してもよい。同様に、ディスクスタンプ設備などの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から符号化ビデオデータを受信してもよく、符号化ビデオデータを含むディスクを製造してもよい。したがって、コンピュータ可読媒体16は、様々な例において、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てもよい。

20

【0043】

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または記憶デバイス32から情報を受信してもよい。コンピュータ可読媒体16または記憶デバイス32の情報は、ビデオデコーダ30によっても使用される、ビデオエンコーダ20によって定義された構文情報を含んでもよく、構文情報は、ブロックおよび他のコーディング単位、たとえばGOPの特性および/または処理を記述する構文要素を含む。表示デバイス31は、復号ビデオデータをユーザに表示し、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表示デバイスなどの、様々な表示デバイスのいずれかを備えてもよい。

30

【0044】

ビデオエンコーダ20およびビデオデコーダ30は、該当する場合は、各々、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)、ディスクリート論理回路、ソフトウェア、ハードウェア、ファームウェア、またはそれらの組合せなどの、様々な適切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てもよい。技法が部分的にソフトウェアにおいて実装されているとき、デバイスは、適切な非一時的コンピュータ可読媒体内にソフトウェアのための命令を記憶してもよく、本開示の技法を実行するために、1つまたは複数のプロセッサを使用してハードウェアにおいて命令を実行してもよい。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内に含まれてもよく、それらのいずれかは、組み合わされたビデオエンコーダ/デコーダ(コーデック)の一部として組み込まれてもよい。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、セルラー電話などのワイヤレス通信デバイスを備えてもよい。

40

50

【 0 0 4 5 】

図1には示していないが、いくつかの態様では、ビデオエンコーダ20およびビデオデコーダ30は、各々、オーディオエンコーダおよびデコーダと一体化されてもよく、共通データストリームまたは別々のデータストリームにおけるオーディオとビデオの両方の符号化を処理するために、適切なMUX-DEMUXユニットまたは他のハードウェアおよびソフトウェアを含んでもよい。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:user datagram protocol)などの他のプロトコルに準拠してもよい。

【 0 0 4 6 】

本開示は、一般に、ビデオデコーダ30などの別のデバイスに特定の情報を「合図する」ビデオエンコーダ20を参照することがある。しかしながら、ビデオエンコーダ20は、特定の構文要素をビデオデータの様々な符号化部分と関連付けることによって、情報を合図してもよいことが理解されるべきである。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部分のヘッダに特定の構文要素を格納することによって、データを「合図」してもよい。いくつかの場合では、そのような構文要素は、ビデオデコーダ30によって受信および復号される前に、符号化および記憶(たとえば、記憶デバイス32に記憶)されてもよい。したがって、「合図する」という用語は、一般に、そのような通信がリアルタイムもしくはほぼリアルタイムで、またはある時間の期間にわたって生じるかにかかわらず、符号化の時点で、媒体に構文要素を記憶するときに生じてもよいような、圧縮されたビデオデータを復号するための構文または他のデータの通信を指してもよく、構文要素は、次いで、この媒体に記憶された後の任意の時間に、復号デバイスによって取得されてもよい。

【 0 0 4 7 】

ビデオエンコーダ20およびビデオデコーダ30は、HEVC規格などのビデオ圧縮規格に従って動作してもよい。本開示の技法は、どのような特定のコーディング規格にも限定されないが、技法は、HEVC規格と、特に、SCC拡張などのHEVC規格の拡張とに関連してもよい。

【 0 0 4 8 】

一般に、HEVCは、ビデオピクチャが輝度サンプルと彩度サンプルの両方を含むツリーブロックまたは最大コーディング単位(LCU)のシーケンスに分割されてもよいことを記載している。ビットストリーム内の構文データは、ピクセル数の点から最大コーディング単位であるLCUのサイズを定義してもよい。スライスは、連続したコーディングツリー単位(CTU: coding tree unit)の数を含む。CTUの各々は、輝度サンプルのコーディングツリーブロック(CTB)と、彩度サンプルの2つの対応するコーディングツリーブロックと、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構文構造とを備えてもよい。白黒ピクチャ、または3つの別個の色平面を有するピクチャでは、CTUは、単一のコーディングツリーブロックと、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構文構造とを備えてもよい。

【 0 0 4 9 】

ビデオピクチャは、1つまたは複数のスライスに分割されてもよい。各ツリーブロックは、4分木に従ってコーディング単位(CU: coding unit)に分割されてもよい。一般に、4分木データ構造は、CUごとに1つのノードを含み、ルートノードは、ツリーブロックに対応する。CUが4つのサブCUに分割されたとき、CUに対応するノードは、4つのリーフノードを含み、各々のリーフノードは、サブCUのうちの1つに対応する。CUは、輝度サンプルアレイと、CbサンプルアレイおよびCrサンプルアレイと、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構文構造とを有するピクチャの輝度サンプルのコーディングブロックと、彩度サンプルの2つの対応するコーディングブロックとを備えてもよい。白黒ピクチャ、または3つの別個の色平面を有するピクチャでは、CUは、単一のコーディングブロックと、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構文構造とを備えてもよい。コーディングブロックは、サンプルのN×Nブロックである。

【0050】

4分木データ構造の各ノードは、対応するCUのための構文データを提供してもよい。たとえば、4分木内のノードは、ノードに対応するCUがサブCUに分割されているかどうかを示す分割フラグを含んでもよい。CUのための構文要素は、再帰的に定義されてもよく、CUがサブCUに分割されているかどうかに依存してもよい。CUがさらに分割されていない場合、それは、リーフCUと呼ばれる。本開示では、リーフCUの4つのサブCUはまた、元のリーフCUの明示的な分割が存在しない場合でも、リーフCUと呼ばれることになる。たとえば、 16×16 サイズのCUがさらに分割されていない場合、 16×16 CUは、決して分割されていないが、4つの 8×8 サブCUはまた、リーフCUと呼ばれることになる。

【0051】

CUは、CUがサイズの区別を持たないことを除いて、H.264規格のマクロブロックと同様の目的を有する。たとえば、ツリーブロックは、4つの子ノード(サブCUとも呼ばれる)に分割されてもよく、各子ノードは、今度は、親ノードであってもよく、別の4つの子ノードに分割されてもよい。4分木のリーフノードと呼ばれる最後の分割されていない子ノードは、リーフCUとも呼ばれるコーディングノードを備える。コーディングされたビットストリームに関連付けられた構文データは、最大CU深度と呼ばれる、ツリーブロックが分割され得る最大回数を定義してもよく、また、コーディングノードの最小サイズを定義してもよい。したがって、ビットストリームはまた、最小コーディング単位(SCU)を定義してもよい。本開示は、HEVCの文脈におけるCU、PU、もしくはTUのいずれか、または、他の規格の文脈における同様のデータ構造(たとえば、H.264/AVCにおけるマクロブロックおよびそのサブブロック)を指すために、「ブロック」という用語を使用する。

【0052】

CUは、コーディングノードと、コーディングノードに関連付けられた予測単位(PU: prediction unit)および変換単位(TU: transform unit)とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状において正方形でなければならない。CUのサイズは、 8×8 ピクセルから、最大 64×64 ピクセルまたはそれ以上のツリーブロックのサイズまでの範囲であってもよい。各CUは、1つまたは複数のPUと1つまたは複数のTUとを含んでもよい。

【0053】

一般に、PUは、対応するCUのすべてまたは一部に対応する空間領域を表し、PUのための参照サンプルを取得するためのデータを含んでもよい。さらに、PUは、予測に関連するデータを含む。たとえば、PUがイントラモード符号化されているとき、PUのためのデータは、残差4分木(RQT: residual quadtree)内に含まれてもよく、PUに対応するTUのためのイントラ予測モードを記述するデータを含んでもよい。別の例として、PUがインターモード符号化されているとき、PUは、PUのための1つまたは複数の動きベクトルを定義するデータを含んでもよい。予測ブロックは、同じ予測が適用されるサンプルの長方形(すなわち、正方形または非正方形)ブロックであってもよい。CUのPUは、ピクチャの輝度サンプルの予測ブロックと、彩度サンプルの2つの対応する予測ブロックと、予測ブロックサンプルを予測するために使用される構文構造とを備えてもよい。白黒ピクチャ、または3つの別個の色平面を有するピクチャでは、PUは、単一の予測ブロックと、予測ブロックサンプルを予測するために使用される構文構造とを備えてもよい。

【0054】

TUは、変換、たとえば、離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または残差ビデオデータへの概念的に類似の変換の適用に続く変換領域における係数を含んでもよい。残差データは、符号化されていないピクチャのピクセルと、PUに対応する予測値との間のピクセル差に対応してもよい。ビデオエンコーダ20は、CUのための残差データを含むTUを形成し、次いで、CUのための変換係数を生成するためにTUを変換してもよい。変換ブロックは、同じ変換が適用されるサンプルの長方形ブロックであってもよい。CUの変換単位(TU: transform unit)は、輝度サンプルの変換ブロックと、彩度サンプルの2つの対応する変換ブロックと、変換ブロックサンプルを変換するために使用される構文構造と

を備えてもよい。白黒ピクチャ、または3つの別個の色平面を有するピクチャでは、TUは、単一の変換ブロックと、変換ブロックサンプルを変換するために使用される構文構造とを備えてもよい。

【0055】

変換に続いて、ビデオエンコーダ20は、変換係数の量子化を実行してもよい。量子化は、一般に、係数を表すために使用されるデータの量をできる限り低減するために変換係数が量子化され、さらなる圧縮を提供するプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減してもよい。たとえば、nビット値は、量子化の間にmビット値に切り捨てられてもよく、ここでnは、mよりも大きい。

【0056】

ビデオエンコーダ20は、量子化変換係数を含む2次元マトリクスから1次元ベクトルを生成して、変換係数を走査してもよい。走査は、アレイの前方により高いエネルギー(したがって、より低い周波数)係数を配置し、アレイの後方により低いエネルギー(したがって、より高い周波数)係数を配置するように設計されてもよい。いくつかの例では、ビデオエンコーダ20は、エントロピー符号化されることが可能なシリアル化ベクトルを生成するために、量子化変換係数を走査するために事前定義された走査順を利用してもよい。他の例では、ビデオエンコーダ20は、適応走査を実行してもよい。

【0057】

1次元ベクトルを形成するために、量子化変換係数を走査した後、ビデオエンコーダ20は、たとえば、コンテキスト適応型可変長コーディング(CAVLC: context-adaptive variable length coding)、コンテキスト適合型二値算術コーディング(CABAC: context-adaptive binary arithmetic coding)、構文ベースのコンテキスト適合型二値算術コーディング(SBAC: syntax-based context-adaptive binary arithmetic coding)、確率間隔区分エントロピー(PIPE: Probability Interval Partitioning Entropy)コーディング、または別のエントロピー符号化方法に従って、1次元ベクトルをエントロピー符号化してもよい。ビデオエンコーダ20はまた、ビデオデータを復号する上でビデオデコーダ30によって使用するための符号化ビデオデータに関連付けられた構文要素をエントロピー符号化してもよい。

【0058】

ビデオエンコーダ20は、さらに、ブロックベースの構文データ、ピクチャベースの構文データ、およびピクチャのグループ(GOP: group of pictures)ベースの構文データなどの構文データを、たとえば、ピクチャヘッダ、ブロックヘッダ、スライスヘッダ、またはGOPヘッダ内で、ビデオデコーダ30に送信してもよい。GOP構文データは、それぞれのGOP内のピクチャの数を記述してもよく、ピクチャ構文データは、対応するピクチャを符号化するために使用される符号化/予測モードを示してもよい。

【0059】

ビデオデコーダ30は、コーディングされたビデオデータを取得すると、ビデオエンコーダ20に関して説明した符号化パスと全体的に相互的な復号パスを実行してもよい。たとえば、ビデオデコーダ30は、ビデオエンコーダ20から、符号化ビデオスライスのビデオブロックと関連する構文要素とを表す符号化ビデオビットストリームを取得してもよい。ビデオデコーダ30は、ビットストリーム内に含まれたデータを使用して、元の符号化されていないビデオシーケンスを再構成してもよい。

【0060】

ビデオエンコーダ20およびビデオデコーダ30は、ビデオスライス内のビデオブロックのイントラおよびインターコーディングを実行してもよい。イントラコーディングは、所与のビデオピクチャ内のビデオにおける空間的冗長性を低減または除去するために、空間予測に依存する。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内のビデオにおける時間的冗長性を低減もしくは除去するために、または、他のビューにおけるビデオで冗長性を低減もしくは除去するために、時間予測またはインタービュー予測に依存する。イントラモード(Iモード)は、いくつかの空間ベースの圧縮モードのうちのいずれかを指してもよい。片方向予測(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

10

20

30

40

50

、いくつかの時間ベースの圧縮モードのうちのいずれかを指してもよい。

【0061】

画面コンテンツをコーディングするときなどのいくつかの例では、ビデオエンコーダ20および/またはビデオデコーダ30は、従来のインター予測と同様の技法を使用してイントラBCを実行してもよい。たとえば、ビデオデータの現在のピクチャの現在のブロックを符号化するために、ビデオエンコーダ20は、参照ピクチャバッファ内に記憶された現在のピクチャのバージョン内に含まれるビデオデータの予測子ブロックを選択し、現在のピクチャ内の現在のブロックに対する予測子ブロックの位置を識別する動きベクトルを符号化し、ビデオデータの現在のブロックと予測子ブロックとの間の差を表すビデオデータの残差ブロックを符号化してもよい。

10

【0062】

いくつかの例では、ビデオエンコーダ20は、動きベクトルに対して整数ピクセル解像度を使用してもよい。いくつかの例ではビデオエンコーダ20は、動きベクトルに対して分数ピクセル解像度を使用してもよい。

【0063】

動きベクトルが整数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にある可能性があり、したがって、ビデオエンコーダ20および/またはビデオデコーダ30は、補間なしで前記サンプルピクセル値にアクセスすることができる。ビデオエンコーダ20および/またはビデオデコーダ30は、補間なしでサンプルピクセルにアクセスすることができるので、ビデオエンコーダ20および/またはビデオデコーダ30は、動きベクトルが整数ピクセル解像度を有する場合、現在のブロックを予測するために予測子ブロック内に位置するサンプルピクセル値のみを使用することができる。動きベクトルが分数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にないことがあり、したがって、ビデオエンコーダ20および/またはビデオデコーダ30は、前記サンプルピクセル値を構成するために補間を実行する必要がある可能性がある。いくつかの例では、サンプルピクセルを構成するために補間を実行するために、ビデオエンコーダ20および/またはビデオデコーダ30は、現在のブロックを予測するために、予測子ブロックの内部と外部の両方に位置するサンプルピクセル値を使用する必要がある可能性がある。しかしながら、いくつかの例では、現在のブロックを予測するために、ビデオエンコーダ20および/またはビデオデコーダ30が予測子ブロックの外部に位置するサンプルピクセル値を使用することは望ましくないことがある。たとえば、予測子ブロックおよび現在のブロックが現在のピクチャ内に位置するとき、予測子ブロックの外部に位置するサンプルは、利用可能ではない可能性がある(すなわち、現在のピクチャの再構成された範囲内に位置しない可能性がある)ので、ビデオデコーダ30が予測子ブロックの外部に位置するサンプルピクセル値を使用することができない可能性がある。

20

30

【0064】

本開示の1つまたは複数の技法によれば、ビデオエンコーダ20は、予測子ブロックを識別する動きベクトルのために使用されるべき解像度に基づいて決定された探索範囲内から現在のブロックのための予測子ブロックを選択してもよい。たとえば、ビデオエンコーダ20は、動きベクトルのために使用されるべき解像度が整数ピクセル精度であるときよりも、動きベクトルのために使用されるべき解像度が分数ピクセル精度であるときにより小さい探索範囲を使用してもよい。一例として、動きベクトルのために使用されるべき解像度が整数ピクセルであるとき、ビデオエンコーダ20は、現在のピクチャの再構成された範囲を含む初期探索範囲内から予測子ブロックを選択してもよい。別の例として、動きベクトルのために使用されるべき解像度が分数ピクセルであるとき、ビデオエンコーダ20は、初期探索範囲の右側境界および下側境界からMサンプルだけ初期探索範囲のサイズを縮小し、初期探索範囲の上側境界および左側境界からNサンプルだけ初期探索範囲のサイズを縮小することによって決定される縮小探索範囲内から予測子ブロックを選択してもよい。このようにして、ビデオエンコーダ20は、予測子ブロックの外部に位置するサンプルピクセル

40

50

ル値を含む、予測子ブロックを構成するために必要なすべてのサンプルピクセル値が、予測子ブロックに基づいて現在のブロックを復号するときにはビデオデコーダ30によって使用するために利用可能であることを保証することができる。そのように、ビデオエンコーダ20は、エンコーダ/デコーダのミスマッチを回避することができる。

【0065】

図2は、表示順で、ピクチャ34、35A、36A、38A、35B、36B、38B、および35Cを含む例示的なビデオシーケンス33を示す概念図である。これらのピクチャの1つまたは複数、Pスライス、Bスライス、またはIスライスを含んでもよい。いくつかの場合では、ビデオシーケンス33は、ピクチャのグループ(GOP)と呼ばれることがある。ピクチャ39は、ビデオシーケンス33の後に発生するシーケンスの表示順で第1のピクチャである。図2は、全体的に、ビデオシーケンスのための例示的な予測構造を表し、異なるインター予測されたスライスタイプを符号化するために使用されるピクチャ参照を説明することのみを意図している。実際のビデオシーケンスは、異なるスライスタイプを異なる表示順で含むより多いまたはより少ないビデオピクチャを含んでもよい。

【0066】

ブロックベースのビデオコーディングのために、ビデオシーケンス33内に含まれるビデオピクチャの各々は、ビデオブロックまたはコーディング単位(CU)に分割され得る。ビデオピクチャの各CUは、1つまたは複数の予測単位(PU)を含んでもよい。いくつかの例では、ピクチャ内のPUを予測するために利用可能な予測方法は、ピクチャタイプに依存し得る。一例として、イントラ予測されたピクチャ(Iピクチャ)のスライス内のビデオブロックまたはPUは、イントラ予測モード(すなわち、同じピクチャ内の隣接ブロックに対する空間予測)を使用して予測されてもよい。別の例として、インター予測されたピクチャ(BピクチャまたはPピクチャ)のスライス内のビデオブロックまたはPUは、インターまたはイントラ予測モード(すなわち、同じピクチャ内の隣接ブロックに対する空間予測、または他の参照ピクチャに対する時間予測)を使用して予測されてもよい。言い換えれば、Iピクチャは、Iスライスを含んでもよく、Pピクチャは、IスライスとPスライスの両方を含んでもよく、Bピクチャは、Iスライスと、Pスライスと、Bスライスとを含んでもよい。

【0067】

Pスライスのビデオブロックは、参照ピクチャリスト内で特定された参照ピクチャから一方向予測コーディングを使用して符号化されてもよい。Bスライスのビデオブロックは、複数の参照ピクチャリスト内で特定された複数の参照ピクチャから双方向予測コーディングを用いて符号化されてもよい。

【0068】

図2の例では、第1のピクチャ34は、Iピクチャとしてイントラモードコーディングのために指定される。他の例では、第1のピクチャ34は、たとえば、前のシーケンスの第1のピクチャに対してPピクチャまたはBピクチャとしてインターモードコーディングでコーディングされてもよい。ビデオピクチャ35A~35C(総称して「ビデオピクチャ35」)は、過去のピクチャおよび将来のピクチャに対する双予測を使用してBピクチャとしてコーディングするために指定される。図2の例に示すように、ピクチャ35Aは、ピクチャ34およびピクチャ36Aからビデオピクチャ35Aへの矢印によって示されているように、第1のピクチャ34およびピクチャ36Aに対してBピクチャとして符号化されてもよい。図2の例では、第1のピクチャ34およびピクチャ36Aは、ピクチャ35Aのブロックの予測の際に使用される参照ピクチャリスト内に含まれてもよい。ピクチャ35Bおよび35Cは、同様に符号化される。

【0069】

ビデオピクチャ36A~36B(総称して「ビデオピクチャ36」)は、過去のピクチャに対する一方向予測を使用してPピクチャまたはBピクチャとしてコーディングするために指定されてもよい。図2の例に示すように、ピクチャ36Aは、ピクチャ34からビデオピクチャ36Aへの矢印によって示されているように、第1のピクチャ34に対してPピクチャまたはBピクチャとして符号化される。ピクチャ36Bは、ピクチャ38Aからビデオピクチャ36Bへの矢印によって示されているように、ピクチャ38Aに対してPピクチャまたはBピクチャとして同様

に符号化される。

【 0 0 7 0 】

ビデオピクチャ38A~38B(総称して「ビデオピクチャ38」)は、同じ過去のピクチャに対する一方向予測を使用してPピクチャまたはBピクチャとしてコーディングするために指定されてもよい。図2の例に示すように、ピクチャ38Aは、ピクチャ36Aからビデオピクチャ38Aへの2本の矢印によって示されているように、ピクチャ36Aへの2つの参照を用いて符号化される。ピクチャ38Bは、同様に符号化される。

【 0 0 7 1 】

いくつかの例では、ピクチャの各々は、ピクチャが出力されるべき順序を示す一意の値(すなわち、特定のビデオシーケンス、たとえば、復号順序で瞬間デコーダリフレッシュ(IDR: instantaneous decoder refresh)ピクチャに続くピクチャのシーケンスに対して一意である値)を割り当てられてもよい。この一意の値は、ピクチャ順序カウンタ(POC: picture order count)と呼ばれることがある。いくつかの例では、ピクチャが出力されるべき順序は、ピクチャがコーディングされる順序と異なってもよい。たとえば、ピクチャ35Aは、ピクチャ36Aの前に出力されてもよいが、ピクチャ36Aは、ピクチャ35Aの前にコーディングされてもよい。

【 0 0 7 2 】

いくつかの例では、ビデオコーダ(たとえば、ビデオエンコーダ20またはビデオデコーダ30)は、現在のピクチャ内のブロックを予測するために使用される参照ピクチャリスト(RPL)内に現在のピクチャを挿入することによってイントラBCを実行してもよい。たとえば、図2の例では、ビデオコーダは、ピクチャ35A内のブロックを予測するために使用されるRPL内に、ピクチャ34およびピクチャ36Aの指標とともに、ピクチャ35Aの指標を挿入してもよい。ビデオコーダは、次いで、ピクチャ35Aのブロックをコーディングするとき、参照ピクチャとしてピクチャ35Aを使用してもよい。

【 0 0 7 3 】

本開示の1つまたは複数の技法によれば、ビデオコーダは、予測子ブロックを識別する動きベクトルのために使用されるべき解像度に基づいて決定された探索範囲内から現在のピクチャ内の現在のブロックのための現在のピクチャ内の予測子ブロックを選択してもよい。ビデオエンコーダ20は、決定された解像度に基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合に現在のブロックのための探索範囲のサイズがより小さくなるように、探索範囲を決定してもよい。

【 0 0 7 4 】

図3は、本開示に記載のイントラブロックコピーのための技法を使用することができるビデオエンコーダ20の一例を示すブロック図である。ビデオエンコーダ20を、他のコーディング規格に関する本開示の限定なしで、説明の目的のためのHEVCコーディングの文脈で説明する。さらに、ビデオエンコーダ20は、HEVCの範囲拡張に従って技法を実装するように構成されてもよい。

【 0 0 7 5 】

ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターコーディングとを実行してもよい。イントラコーディングは、所与のビデオピクチャ内のビデオにおける空間的冗長性を低減または除去するために、空間予測に依存する。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内のビデオにおける時間的冗長性を低減もしくは除去するために、または、他のビューにおけるビデオで冗長性を低減もしくは除去するために、時間予測またはインタビュー予測に依存する。

【 0 0 7 6 】

図3の例では、ビデオエンコーダ20は、ビデオデータメモリ40と、予測処理ユニット42と、参照ピクチャメモリ64と、加算器50と、変換処理ユニット52と、量子化処理ユニット54と、エントロピー処理ユニット56とを含んでもよい。予測処理ユニット42は、次に、動き推定ユニット44と、動き補償ユニット46と、イントラ予測ユニット48とを含む。ブロック再構成のために、ビデオエンコーダ20はまた、逆量子化処理ユニット58と、逆変換処理

ユニット60と、加算器62とを含む。デブロッキングフィルタ(図3に示さず)はまた、再構成されたビデオからブロックネスアーティファクトを除去するためにブロック境界をフィルタリングするために含まれてもよい。所望ならば、デブロッキングフィルタは、典型的には、加算器62の出力をフィルタリングすることになる。追加のループフィルタ(ループ内またはループ後)はまた、デブロッキングフィルタに加えて使用されてもよい。

【0077】

ビデオデータメモリ40は、ビデオエンコーダ20の構成要素によって符号化されるべきビデオデータを記憶してもよい。ビデオデータメモリ40内に記憶されるビデオデータは、たとえば、ビデオソース18から取得されてもよい。参照ピクチャメモリ64は、(たとえば、イントラまたはインター予測コーディングモードとも呼ばれるイントラまたはインターコーディングモードで)ビデオエンコーダ20によってビデオデータを符号化する際に使用するための参照ビデオデータを記憶する復号ピクチャバッファ(DPB:decoding picture buffer)の一例である。ビデオデータメモリ40および参照ピクチャメモリ64は、シンクロナスDRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、磁気抵抗性RAM(MRAM)、抵抗性RAM(RRAM(登録商標))、または他のタイプのメモリデバイスなどの、様々なメモリデバイスのいずれかによって形成されてもよい。ビデオデータメモリ40および参照ピクチャメモリ64は、同じメモリデバイスまたは別個のメモリデバイスによって提供されてもよい。様々な例では、ビデオデータメモリ40は、ビデオエンコーダ20の他の構成要素とともにオンチップであってもよく、または、これらの構成要素に対してオフチップであってもよい。

【0078】

符号化プロセス中、ビデオエンコーダ20は、コーディングされるべきビデオ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てもよい。動き推定ユニット44および動き補償ユニット46は、時間的圧縮を提供するため、またはインタービュー圧縮を提供するために、1つまたは複数の参照ピクチャ内の1つまたは複数のブロックに対する受信ビデオブロックのインター予測コーディングを実行する。イントラ予測ユニット48は、代替的には、空間的圧縮を提供するために、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内の1つまたは複数の隣接ブロックに対して受信ビデオブロックのイントラ予測コーディングを実行してもよい。ビデオエンコーダ20は、(たとえば、ビデオデータの各ブロックのための適切なコーディングモードを選択するために)複数のコーディングパスを実行してもよい。

【0079】

さらに、分割ユニット(図示せず)は、以前のコーディングプロセスにおける以前に分割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分割してもよい。たとえば、分割ユニットは、最初にスライスまたはピクチャをLCUに分割し、レート-歪み分析(たとえば、レート-歪み最適化)に基づいて、LCUの各々をサブCUに分割してもよい。予測処理ユニット42は、さらに、サブCUへのLCUの分割を示す4分木データ構造を生成してもよい。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とを含んでもよい。

【0080】

予測処理ユニット42は、たとえば、エラー結果に基づいて、コーディングモードの1つ、イントラまたはインターを選択し、得られたイントラまたはインターコーディングされたブロックを、残差ブロックデータを生成するために加算器50に、参照ピクチャとして使用するための符号化ブロックを再構成するために加算器62に提供してもよい。予測処理ユニット42はまた、動きベクトル、イントラモードインジケータ、分割情報などの構文要素、および他のそのような構文情報を、エン트로ピー符号化ユニット56に提供する。

【0081】

動き推定ユニット44および動き補償ユニット46は、高度に統合されてもよいが、概念的な目的のために別々に示されている。動き推定ユニット44によって実行される動き推定は、ビデオブロックに関する動きを推定する動きベクトルを生成するプロセスである。動き

ベクトルは、たとえば、現在のピクチャ(または他のコーディング単位)内のコーディングされた現在のブロックに対する参照ピクチャ(または他のコーディング単位)内の予測ブロックに対する現在のビデオピクチャ内のビデオブロックのPUの変位を示してもよい。予測ブロックは、絶対差合計(SAD:sum of absolute difference)、自乗差合計(SSD:sum of square difference)、または他の差分メトリックによって決定され得るピクセル差の観点から、コーディングされるべきブロックに密接に一致することがビデオエンコーダ20によって見出されたブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ64内に記憶された参照ピクチャのサブ整数ピクセル位置の値を計算してもよい。たとえば、ビデオエンコーダ20は、参照ピクチャの4分の1ピクセル位置の値、8分の1ピクセル位置の値、または他の分数ピクセル位置の値を補間してもよい。したがって、動き推定ユニット44は、フルピクセル位置および分数ピクセル位置に対する動き探索を実行し、整数ピクセル精度または分数ピクセル精度のいずれかで動きベクトルを出力してもよい。

10

【0082】

動き推定ユニット44は、参照ピクチャの予測ブロックの位置とPUの位置とを比較することによって、インターコーディングスライス内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。参照ピクチャは、参照ピクチャメモリ64内に記憶された1つまたは複数の参照ピクチャを識別する1つまたは複数の参照ピクチャリスト(RPL)から選択されてもよい。動き推定ユニット44は、エントロピー符号化ユニット56および動き補償ユニット46に計算された動きベクトルを送信する。いくつかの例では、動き推定ユニット44は、選択された参照ピクチャの指標をエントロピー符号化ユニット56に送ってもよい。

20

【0083】

上記で説明したように、動き推定ユニット44は、選択された参照ピクチャの指標をエントロピー符号化ユニット56に送ってもよい。いくつかの例では、動き推定ユニット44は、RPL内の選択された参照ピクチャのインデックス値を送ることによって指標を送ってもよい。

【0084】

いくつかの例では、参照ピクチャとして他のピクチャを使用するためにインター予測を制限することとは対照的に、動き推定ユニット44は、現在のピクチャ内に含まれるビデオデータのブロックを予測するために、参照ピクチャとして現在のピクチャを使用してもよい。たとえば、動き推定ユニット44は、参照ピクチャメモリ64内に現在のピクチャのバージョンを記憶してもよい。いくつかの例では、動き推定ユニット44は、固定された値に初期化されたピクセル値を有する現在のピクチャの初期化されたバージョンを記憶してもよい。いくつかの例では、固定された値は、現在のピクチャのサンプルのビット深度に基づいてもよい。たとえば、固定された値は、 $1 < (\text{bitDepth} - 1)$ であってもよい。いくつかの例では、動き推定ユニット44は、現在のピクチャの任意のブロックを符号化する前に、現在のブロックの初期化バージョンを記憶してもよい。現在のピクチャの初期化バージョンを記憶することによって、動き推定ユニット44は、予測ブロックの探索(すなわち探索範囲)をすでに再構成されたブロックに限定することを必要とされなくてもよい。対照的に、動き推定ユニット44が現在のピクチャの初期化バージョンを記憶しない場合、たとえば、予測ブロックの探索は、デコーダ/エンコーダのミスマッチを回避するために、すでに再構成されたブロックに限定されてもよい。

30

40

【0085】

予測処理ユニット42は、現在のピクチャのための1つまたは複数のRPLを生成してもよい。たとえば、予測処理ユニット42は、現在のピクチャのためのRPLに現在のピクチャを含めてもよい。

【0086】

上記で説明したように、ビデオデータの現在のピクチャのビデオデータのブロックを符号化するとき、動き推定ユニット44は、現在のブロックに密接に一致する予測ブロックを選択してもよい。いくつかの例では、他のピクチャのブロックを探索するのとは対照的に(または、それに加えて)、動き推定ユニット44は、現在のピクチャの現在のブロックのた

50

めの予測ブロックとして使用するために現在のピクチャ内に位置するブロックを選択してもよい。たとえば、動き推定ユニット44は、現在のピクチャを含む1つまたは複数の参照ピクチャを含むピクチャに対して探索を実行してもよい。各ピクチャについて、動き推定ユニット44は、たとえば、ピクセルごとの絶対差合計(SAD)、自乗差合計(SSD)、平均絶対差(MAD:mean absolute difference)、平均自乗差(MSD:mean squared difference)、などを使用して、予測ブロックが現在のブロックに一致する程度を反映する探索結果を計算してもよい。次いで、動き推定ユニット44は、現在のブロックに最も一致したピクチャ内のブロックを識別し、ブロックおよびピクチャ(現在のピクチャであってもよい)の位置を予測処理ユニット42に示してもよい。このようにして、動き推定ユニット44は、たとえば、動き推定ユニット44が、予測子ブロックが現在のピクチャ、すなわち、予測されている現在のブロックと同じピクチャ内に含まれていると決定したとき、イントラBCを実行してもよい。

【0087】

いくつかの例では、イントラBCのための予測子ブロックが有効であることを確実にするために、RExt WD7は、予測処理ユニット42が1つまたは複数のビットストリーム適合制約を実施することを規定している。一例として、予測処理ユニット42は、予測子ブロックが現在のCUと同じスライス/タイル内にあることを保証してもよい。別の例として、予測処理ユニット42は、参照ブロックが現在のCUと重複することができないことを保証してもよい。別の例として、予測処理ユニット42は、予測子ブロックの内部のすべてのピクセルが再構成されるべきであることを保証してもよい。別の例として、http://phenix.it-sudparis.eu/jct/doc_end_user/documents/19_Strasbourg/wg11/JCTVC-S0220-v2.zipにおいて利用可能なRapakaら、「On parallel processing capability of intra block copy」、ITU-T SG 16WP3およびISO/IEC JTC1/SC29/WG11のビデオコーディングに関する共同のコラボレーションチーム(JCT-VC)、第19回ミーティング:ストラスブール、フランス、2014年10月17~24日、Document:JCTVC-S0220に記載のように、予測処理ユニット42は並列化実装に関する考慮事項のために予測子ブロックが特定の領域内にあることを保証してもよい。別の例として、制約されたイントラ予測が有効になっているとき、予測処理ユニット42は、予測子ブロックが従来のインターモードを使用してコーディングされたいかなるピクセルをも含まないことを保証してもよい。別の例として、予測処理ユニット42は、予測に使用される動きが補間を避けるために整数ピクセル精度を有することを保証してもよい。

【0088】

いくつかの例では、常に整数ピクセル精度を使用するのとは対照的に、予測処理ユニット42は、異なるレベルの精度で動きベクトルを決定してもよい。たとえば、予測処理ユニット42は、整数精度、デフォルトの精度、または最も詳細な動き精度(たとえば、HEVCにおける1/4ピクセル(「pel」)精度)で動きベクトルを決定してもよい。いくつかの例では、予測処理ユニット42は、たとえば、現在のピクチャによって参照されるSPSまたはVPS内で、コーディングされたイントラBC動きベクトルの精度を示す構文要素を符号化してもよい。いくつかの例では、イントラBC動きベクトルの精度は、ピクチャレベルで適応されてもよく、予測処理ユニット42は、たとえば、現在のブロックによって参照されるPPSまたはスライスにおいて、コーディングされたイントラBC動きベクトルの精度を示す構文要素を、エンタロピー符号化ユニット56に符号化させてもよい。たとえば、SCCのためのいくつかの提案では、適応型MV解像度(AMVR:adaptive MV resolution)が使用される。たとえば、http://phenix.it-sudparis.eu/jct/doc_end_user/documents/19_Strasbourg/wg11/JCTVC-S0085-v3.zipにおいて利用可能な、Liら、「Adaptive motion vector resolution for screen content」、ITU-T SG 16WP3およびISO/IEC JTC1/SC29/WG11のビデオコーディングに関する共同のコラボレーションチーム(JCT-VC)、第19回ミーティング:ストラスブール、フランス、2014年10月17~24日、Document:JCTVC-S0085に記載のように、各スライスについて、MVは、整数ピクセル解像度または1/4ピクセル解像度のいずれかで表され、コーディングされてもよく、どの動きベクトル解像度が使用されたかを示すために、フラグuse_integer_mv_flagがスライスヘッダ内で通知される。

【0089】

動きベクトルが整数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にある可能性があり、したがって、予測処理ユニット42は、補間なしでサンプルピクセル値にアクセスすることができる。予測処理ユニット42は、補間なしでサンプルピクセルにアクセスすることができるので、予測処理ユニット42は、動きベクトルが整数ピクセル解像度を有する場合、現在のブロックを予測するために予測子ブロック内に位置する標本ピクセル値のみを使用することができる。動きベクトルが分数ピクセル解像度を有する場合などのいくつかの例では、動きベクトルによって識別されるサンプルピクセル値は、整数ピクセル位置にないことがあり、したがって、予測処理ユニット42は、前記サンプルピクセルを構成するために補間を実行する必要がある可能性がある。いくつかの例では、サンプルピクセル値を構成するために補間を実行するために、予測処理ユニット42は、現在のブロックを予測するために、予測子ブロックの内部と外部の両方に位置するサンプルピクセル値を使用する必要がある可能性がある。しかしながら、いくつかの例では、現在のブロックを予測するために、予測処理ユニット42が予測子ブロックの外部に位置するサンプルピクセル値を使用することは望ましくないことがある。たとえば、予測子ブロックおよび現在のブロックが現在のピクチャ内に位置するとき、予測子ブロックの外部に位置するサンプルは、参照ピクチャメモリ64内で利用可能ではない可能性がある(すなわち、現在のピクチャの再構成された範囲内に位置しない可能性がある)ので、予測処理ユニット42が予測子ブロックの外部に位置するサンプルピクセル値を使用することができない可能性がある。

【0090】

本開示の1つまたは複数の技法によれば、予測処理ユニット42は、予測子ブロック内に含まれたサンプルと予測子ブロック内に含まれないサンプルとを含む、補間プロセスのために使用される任意のサンプルが、上記で説明したビットストリーム適合制約を満たすように、予測子ブロックを決定してもよい。たとえば、動き推定ユニット44は、予測子ブロックを識別する動きベクトルのために使用されるべき解像度に基づいて決定された探索範囲内から現在のブロックのための予測子ブロックを選択してもよい。たとえば、動き推定ユニット44は、動きベクトルのために使用されるべき解像度が整数ピクセル精度であるときよりも、動きベクトルのために使用されるべき解像度が分数ピクセル精度であるときにより小さい探索範囲を使用してもよい。一例として、動きベクトルのために使用されるべき解像度が整数ピクセル精度であるとき、動き推定ユニット44は、現在のピクチャの再構成された範囲を含む初期探索範囲内から予測子ブロックを選択してもよい。別の例として、動きベクトルのために使用されるべき解像度が分数ピクセルであるとき、動き推定ユニット44は、縮小探索範囲内から予測子ブロックを選択してもよい。このようにして、動き推定ユニット44は、予測子ブロックの外部に位置するサンプルピクセル値を含む、予測子ブロックを構成するために必要なすべてのサンプルピクセル値が、予測子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ときに使用するために利用可能であることを保証することができる。

【0091】

上記で説明したように、CUは、輝度サンプルアレイと、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輝度サンプルのコーディングブロックと、彩度サンプルの2つの対応するコーディングブロックとを含んでもよい。いくつかの例では、輝度サンプル、Cbサンプル、およびCrサンプルに対して異なるサンプリング比が使用されてもよい。一般に、サンプリング比は、3つの部分比A:B:Cとして表されてもよく、ここで、Aは、水平サンプリング基準(通常は4である、概念領域の幅)を表し、Bは、Aピクセルの第1の行内のクロミナンス(Cr、Cb)サンプルの数を表し、Cは、Aピクセルの第1の行と第2の行との間のクロミナンスサンプル(Cr、Cb)の変化の数を表す。ビデオエンコーダ20によって使用され得るいくつかの例示的なサンプリング比は、4:4:4、4:2:2、4:2:0、および4:0:0を含むが、必ずしもこれらに限定されない。4:4:4のサンプリング比を使用するとき、ビデオエンコーダ20は、成分の各々を同じレートでサンプリングしてもよい。4:2:2のサンプリ

ング比を使用するとき、ビデオエンコーダ20は、水平彩度解像度が半分になるように、輝度成分のサンプルレートの半分で彩度成分(すなわち、CbおよびCr)をサンプリングしてもよい。4:2:0のサンプリング比を使用するとき、ビデオエンコーダ20は、水平彩度解像度と垂直彩度解像度の両方が半分になるように、交互のライン上の彩度成分(すなわち、CbおよびCr)を輝度成分のサンプルレートの半分でサンプリングしてもよい。いくつかの例では、ビデオエンコーダ20は、サンプリング比を示す構文要素(たとえば、chroma_format_idc)を符号化してもよい。

【0092】

サンプリング比が4:4:4である場合などのいくつかの例では、ビデオエンコーダ20は、輝度成分と彩度成分とを一緒に符号化してもよい。いくつかの例では、ビデオエンコーダ20は、輝度成分と彩度成分とを別々に符号化してもよい。いくつかの例では、ビデオエンコーダ20は、輝度成分および彩度成分が一緒に符号化されるのか、または別々に符号化されるのかを示す構文要素(たとえば、separate_colour_plane_flag)を符号化してもよい。

【0093】

輝度成分および彩度成分が別々に符号化される場合などのいくつかの例では、ビデオエンコーダ20は、輝度成分および彩度成分に対して異なる動きベクトルを使用してもよい。たとえば、ビデオエンコーダ20は、輝度サンプルの予測ブロックを示すために輝度動きベクトルを使用し、彩度サンプルの予測ブロックを示すために彩度動きベクトルを使用してもよい。いくつかの例では、ビデオエンコーダ20は、輝度動きベクトルと彩度動きベクトルの両方を符号化してもよい。他の例では、ビデオエンコーダ20は、輝度動きベクトルを符号化してもよく、彩度動きベクトルは、輝度動きベクトルおよびサンプリング比に基づいて決定されてもよい。たとえば、彩度動きベクトルは、以下の式(1)および(2)に従って輝度動きベクトルに基づいて決定されてもよく、ここで、 $MVC_{Horizontal}$ および $MVC_{Vertical}$ は、彩度動きベクトルの水平成分および垂直成分であり、 $MV_{Horizontal}$ および $MV_{Vertical}$ は、輝度動きベクトルの水平成分および垂直成分であり、SubWidthCおよびSubHeightCは、サンプリング比に基づいてTable (1)(表1)を使用して決定される。

$$MVC_{Horizontal} = MV_{Horizontal} * 2 / SubWidthC \quad (1)$$

$$MVC_{Vertical} = MV_{Vertical} * 2 / SubHeightC \quad (2)$$

【0094】

【表1】

サンプリング比	<u>SubWidthC</u>	<u>SubHeightC</u>
4:2:0	2	2
4:2:2	2	1
4:4:4	1	1

Table (1)

【0095】

いくつかの例では、輝度動きベクトルからの彩度動きベクトルの導出により、輝度動きベクトルのために使用される解像度は、彩度動きベクトルのために使用される解像度と異なることがある。たとえば、いくつかの例では、輝度動きベクトルは、整数ピクセル解像度を有することがあり、彩度動きベクトルは、分数ピクセル解像度を有することがある。そのように、本開示の1つまたは複数の技法によれば、動き推定ユニット44は、彩度動きベクトルのために使用されるべき解像度に基づいて探索範囲のサイズを決定してもよい。たとえば、彩度動きベクトルのために使用されるべき解像度が整数ピクセルであるとき、動き推定ユニット44は、現在のピクチャの再構成された範囲を含む初期探索範囲内から予

測子ブロックを選択してもよい。別の例として、彩度動きベクトルのために使用されるべき解像度が分数ピクセルであるとき、動き推定ユニット44は、彩度動きベクトルの解像度がピクセル間であるときに使用される初期探索範囲よりも小さい縮小探索範囲内から予測子ブロックを選択してもよい。このようにして、動き推定ユニット44は、予測子ブロックの外部に位置するサンプル彩度ピクセル値を含む、予測子ブロックを構成するために必要なすべてのサンプルピクセル値が、予測子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ときに使用するために利用可能であることを保証することができる。

【0096】

いくつかの例では、動き推定ユニット44は、すべてのサンプリング比に対して縮小探索範囲を使用してもよい。他の例では、動き推定ユニット44は、輝度サンプルに対する彩度サンプリングの比が1よりも大きい(たとえば、4:2:0、4:2:2)とき、縮小探索範囲を使用してもよい。いくつかの例では、動き推定ユニット44は、白黒(すなわち、4:0:0のサンプリング比)に対して縮小探索範囲を使用しなくてもよい。いくつかの例では、動き推定ユニット44は、白黒以外のすべてのサンプリング比に対して縮小探索範囲を使用してもよい。

10

【0097】

上記で説明したように、いくつかの例では、動き推定ユニット44は、縮小探索範囲から予測子ブロックを選択してもよい。いくつかの例では、動き推定ユニット44は、初期探索範囲の右側境界および下側境界からM(たとえば、1、2、3、4、5、6、7、8)サンプルだけ初期探索範囲のサイズを縮小し、初期探索範囲の上側境界および左側境界からN(たとえば、1、2、3、4、5、6、7、8)サンプルだけ初期探索範囲のサイズを縮小することによって縮小探索範囲を決定してもよい。動き推定ユニット44がどのように縮小探索範囲を決定し得るのかの追加の詳細および他の例について、図6～図8を参照して以下に説明する。

20

【0098】

動き補償ユニット46によって実行される動き補償は、動き推定ユニット44によって決定された動きベクトルに基づいて、予測ブロックをフェッチまたは生成することを含んでもよい。再び、動き推定ユニット44および動き補償ユニット46は、いくつかの例では、機能的に統合されてもよい。現在のブロックのPUのための動きベクトルを受信すると、動き補償ユニット46は、動きベクトルが参照ピクチャリスト(RPL)のうちの1つにおいて指し示す予測ブロックの位置を突き止めてもよい。加算器50は、以下に説明するように、コーディングされている現在の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除算することによって残差ビデオブロックを形成し、ピクセル差分値を形成する。一般に、動き推定ユニット44は、輝度成分に対する動き推定を実行し、動き補償ユニット46は、彩度成分と輝度成分の両方のための輝度成分に基づいて計算された動きベクトルを使用する。予測処理ユニット42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ド30によって使用するためのビデオブロックおよびビデオスライスに関連付けられた構文要素を生成してもよい。

30

【0099】

イントラ予測ユニット48は、上記で説明したように、動き推定ユニット44および動き補償ユニット46によって実行されるインター予測の代替として、現在のブロックをイントラ予測してもよい。具体的には、イントラ予測ユニット48は、現在のブロックを符号化するために使用するイントラ予測モードを決定してもよい。いくつかの例では、イントラ予測ユニット48は、たとえば、別個の符号化パスの間、様々なイントラ予測モードを使用してブロックを符号化してもよく、イントラ予測ユニット48は、複数のイントラ予測モードから使用するための適切なイントラ予測モードを選択してもよい。

40

【0100】

たとえば、イントラ予測ユニット48は、様々な試験されたイントラ予測モードに対してレート-歪み分析を使用してレート-歪み値を計算し、試験されたモードの中から最良のレート-歪み特性を有するイントラ予測モードを選択してもよい。レート-歪み分析は、一般に、符号化ブロックと、符号化ブロックを生成するために符号化された元の未符号化ブロ

50

ックとの間の歪み(または誤差)の量、ならびに、符号化ブロックを生成するために使用されたビットレート(すなわち、ビット数)を決定する。イントラ予測ユニット48は、どのイントラ予測モードがブロックのための最良のレート-歪み値を示すのかを決定するために、様々な符号化ブロックに関する歪みおよびレートから比を計算してもよい。

【0101】

いくつかの例では、イントラ予測ユニット48によって使用するために利用可能な複数のイントラ予測モードは、平面予測モードと、DC予測モードと、1つまたは複数の角度予測モードとを含んでもよい。選択されたモードに関係なく、イントラ予測ユニット48は、常に、現在のブロックに隣接する再構成されたブロックに基づいて現在のブロックを予測してもよい。一例として、平面予測モードを使用するとき、イントラ予測ユニット48は、水平および垂直予測を平均することによって現在のブロックを予測してもよい。いくつかの例では、イントラ予測ユニット48は、(現在のブロックを予測するとき、右隣接ブロックのサンプルが再構成されない可能性があるとき)左隣接ブロックおよび右上隣接ブロックに基づいて水平予測を決定してもよく、(現在のブロックを予測するとき、下隣接ブロックのサンプルが再構成されない可能性があるとき)上隣接ブロックおよび左下隣接ブロックに基づいて水平予測を決定してもよい。

10

【0102】

別の例として、DC予測モードを使用するとき、イントラ予測ユニット48は、一定値で現在のブロックのサンプルを予測してもよい。いくつかの例では、一定値は、左隣接ブロック内のサンプルと上隣接ブロック内のサンプルの平均を表してもよい。別の例として、1つまたは複数の角度予測モードのうちの1つを使用するとき、イントラ予測ユニット48は、予測方向によって示される隣接ブロックからのサンプルに基づいて、現在のブロックのサンプルを予測してもよい。

20

【0103】

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、予測処理ユニット42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つまたは複数の構成要素を表す。

【0104】

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、残差変換係数値を含むビデオブロックを生成する。変換処理ユニット52は、DCTと概念的に同様の他の変換を実行してもよい。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はまた、使用され得る。いずれの場合にも、変換処理ユニット52は、残差ブロックに変換を適用し、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してもよい。

30

【0105】

変換処理ユニット52は、得られた変換係数を量子化処理ユニット54に送ってもよい。量子化処理ユニット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に関連するビット深度を低減してもよい。量子化の程度は、量子化パラメータを調整することによって変更されてもよい。いくつかの例では、量子化処理ユニット54は、次いで、量子化変換係数を含む行列の走査を実行してもよい。代替的には、エントロピー符号化ユニット56は、走査を実行してもよい。

40

【0106】

量子化に続いて、エントロピー符号化ユニット56は、量子化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、コンテキスト適合型二値算術コーディング(CABAC)、コンテキスト適応型可変長コーディング(CAVLC)、構文ベースのコンテキスト適合型二値算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピー符号化技法を実行してもよい。コンテキストベースのエントロピーコーディングの場合には、コンテキストは、隣接ブロックに基づいてもよい。エントロピー符号化ユニット56によるエントロピーコーディングに続いて、符号化ビ

50

ットストリームは、別のデバイス(たとえば、ビデオデコーダ30)に送信されてもよく、または後の送信もしくは検索のためにアーカイブされてもよい。

【0107】

逆量子化処理ユニット58および逆変換処理ユニット60は、たとえば、参照ブロックとして後で使用するために、ピクセル領域における残差ブロックを再構成するために、それぞれ、逆量子化および逆変換を適用する。

【0108】

動き補償ユニット46はまた、動き推定の際に使用するためのサブ整数ピクセル値を計算するために、参照ブロックに1つまたは複数の補間フィルタを適用してもよい。加算器62は、参照ピクチャメモリ64内に記憶するための再構成されたビデオブロックを生成するために、動き補償ユニット46によって生成された動き補償された予測ブロックに、再構成された残差ブロックを加算する。再構成されたビデオブロックは、後続のビデオピクチャ内のブロックをインターコーディングするために、参照ブロックとして動き推定ユニット44および動き補償ユニット46によって使用されてもよい。現在のピクチャが現在のピクチャを予測するために参照ピクチャとして使用される場合などのいくつかの例では、動き補償ユニット46および/または加算器62は、現在のピクチャをコーディングしながら、規則的な間隔で、参照ピクチャ64によって記憶された現在のピクチャのバージョンを更新してもよい。一例として、動き補償ユニット46および/または加算器62は、現在のピクチャの各ブロックをコーディングした後に、参照ピクチャメモリ64によって記憶された現在のピクチャのバージョンを更新してもよい。たとえば、現在のブロックのサンプルが初期化値として参照ピクチャメモリ64内に記憶される場合、動き補償ユニット46および/または加算器62は、現在のブロックのための再構成されたサンプルで、参照ピクチャメモリ64によって記憶された現在のピクチャの現在のサンプルを更新してもよい。

【0109】

フィルタリングユニット(図示せず)は、様々なフィルタリングプロセスを実行してもよい。たとえば、フィルタリングユニットは、デブロッキングを実行してもよい。すなわち、フィルタリングユニットは、再構成されたビデオのスライスまたはフレームを形成する複数の再構成されたビデオブロックを受信し、スライスまたはフレームからブロックネスアーティファクトを除去するために、ブロック境界をフィルタリングしてもよい。一例では、フィルタリングユニットは、ビデオブロックのいわゆる「境界強度」を評価する。ビデオブロックの境界強度に基づいて、ビデオブロックのエッジピクセルは、1つのビデオブロックからの遷移が、観察者が知覚するのがより困難になるように、隣接ビデオブロックのエッジピクセルに対してフィルタリングされてもよい。

【0110】

いくつかの例では、動き補償ユニット46および/または加算器62は、フィルタリングがサンプルへのフィルタリング(たとえば、デブロッキング、適応ループフィルタリング(ALF)および/またはサンプル適応オフセット(SAO))を実行する前に、参照ピクチャメモリ64によって記憶された現在のピクチャのバージョンを更新してもよい。たとえば、フィルタリングユニットは、フィルタリングを適用する前に、ピクチャ全体がコーディングされるまで待機してもよい。このようにして、動き推定ユニット44は、フィルタリングを適用する前に、参照として現在のピクチャを使用してもよい。いくつかの例では、フィルタリングユニットは、参照ピクチャメモリ64によって記憶された現在のピクチャのバージョンが更新されたとき、フィルタリングを実行してもよい。たとえば、フィルタリングユニットは、各ブロックが更新されたとき、フィルタリングを適用してもよい。このようにして、動き推定ユニット44は、フィルタリングを適用した後、参照として現在のピクチャを使用してもよい。

【0111】

技法のいくつかの異なる態様および例が本開示で説明されているが、技法の様々な態様および例は、互いに一緒にまたは別々に実行されてもよい。言い換えれば、技法は、上記で説明した様々な態様および例に厳密に限定されるべきではないが、組み合わせて使用さ

10

20

30

40

50

れても、または一緒におよび/もしくは別々に実行されてもよい。加えて、特定の技法は、(イントラ予測ユニット48、動き補償ユニット46、またはエントロピー符号化ユニット56などの)ビデオエンコーダ20の特定のユニットに起因することがあるが、ビデオエンコーダ20の1つまたは複数の他のユニットはまた、そのような技法を実行することを担当してもよいことが理解されるべきである。

【0112】

このようにして、ビデオエンコーダ20は、本開示に記載の1つまたは複数の例示的な技法を実装するように構成されてもよい。たとえば、ビデオエンコーダ20は、現在のピクチャ内、すなわち、同じピクチャ内に含まれる予測子ブロックを使用して、現在のピクチャ内の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てもよい。ビデオエンコーダ20は、さらに、たとえば、イントラBCを使用して現在のピクチャの1つまたは複数のブロックをコーディングする目的のため、VPS/SPS/PPSを参照するピクチャがピクチャ自体の参照ピクチャリスト内に存在し得るかどうかを示す構文要素を含むビットストリームを出力するように構成されてもよい。すなわち、ブロックがイントラBCモードを使用してコーディングされるとき、ビデオエンコーダ20は、(構文要素が、現在のピクチャがそれ自体のための参照ピクチャリスト内に含まれ得ることを示すと仮定して)たとえば、インデックス値がピクチャ自体に対応するように参照ピクチャリスト内へのインデックス値を使用して、ブロックのための参照ピクチャがブロックを含むピクチャであることを合図してもよい。ビデオエンコーダ20は、イントラBCモードを使用してコーディングされるブロックの動き情報内にこのインデックス値を含めてもよい。いくつかの例では、ビデオエンコーダ20のハードウェアアーキテクチャは、現在のピクチャの現在のブロックを予測するために、参照ピクチャとして現在のピクチャを使用するために特に適合されてもよく、適合されなくてもよい。

【0113】

図4は、本開示に記載の技法を実装することができるビデオデコーダ30の一例を示すブロック図である。再び、ビデオデコーダ30を、他のコーディング規格に関する本開示の限定なしで、説明の目的のためのHEVCコーディングの文脈で説明する。さらに、ビデオデコーダ30は、範囲拡張に従って技法を実装するように構成されてもよい。

【0114】

図4の例では、ビデオデコーダ30は、ビデオデータメモリ69と、エントロピー復号ユニット70と、予測処理ユニット71と、逆量子化処理ユニット76と、逆変換処理ユニット78と、加算器80と、参照ピクチャメモリ82とを含んでもよい。予測処理ユニット71は、動き補償ユニット72と、イントラ予測ユニット74とを含む。ビデオデコーダ30は、いくつかの例では、図3からのビデオエンコーダ20に関連して説明した符号化パスと全体的に相互的な復号パスを実行してもよい。

【0115】

ビデオデータメモリ69は、ビデオデコーダ30の構成要素によって復号されるべき、符号化ビデオビットストリームなどのビデオデータを記憶してもよい。ビデオデータメモリ69内に記憶されたビデオデータは、たとえば、記憶デバイス34から、カメラなどのローカルビデオソースから、ビデオデータのワイヤードもしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって取得されてもよい。ビデオデータメモリ69は、符号化ビデオビットストリームからの符号化ビデオデータを記憶するコーディングピクチャバッファ(CPB:coded picture buffer)を形成してもよい。

【0116】

参照ピクチャメモリ82は、(たとえば、イントラまたはインターコーディングモードにおいて)ビデオデコーダ30によってビデオデータを復号する際に使用するための参照ビデオデータを記憶する復号ピクチャバッファ(DPB:decoded picture buffer)の一例である。ビデオデータメモリ69および参照ピクチャメモリ82は、シンクロナスDRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、磁気抵抗性RAM(MRAM)、抵抗性RAM(RRAM(登録商標))、または他のタイプのメモリデバイスなどの、様々なメモリデバイスのいずれか

によって形成されてもよい。ビデオデータメモリ69および参照ピクチャメモリ82は、同じメモリデバイスまたは別個のメモリデバイスによって提供されてもよい。様々な例では、ビデオデータメモリ69は、ビデオデコーダ30の他の構成要素とともにオンチップであってもよく、または、これらの構成要素に対してオフチップであってもよい。

【0117】

復号プロセスの間、ビデオデコーダ30は、ビデオエンコーダ20から、符号化ビデオスライスのビデオブロックと関連する構文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70は、量子化係数、動きベクトルまたはイントラ予測モードインジケータ、および他の構文要素を生成するために、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70は、動きベクトルおよび他の構文要素を動き補償ユニット72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において構文要素を受信してもよい。

10

【0118】

いくつかの例では、ビデオスライスがイントラコーディング(I)スライスとしてコーディングされているとき、イントラ予測ユニット74は、合図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のための予測データを生成してもよい。いくつかの例では、ビデオピクチャがインターコーディング(すなわち、BまたはP)スライスとしてコーディングされているとき、動き補償ユニット72は、エントロピー復号ユニット70から受信した動きベクトルおよび他の構文要素に基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリスト(RPL)のうちの1つの中の参照ピクチャのうちの1つから生成されてもよい。予測処理ユニット71は、参照ピクチャメモリ82内に記憶された参照ピクチャに基づいて、構成技法を使用して、RPL、たとえば、List0およびList1を構成してもよい。

20

【0119】

いくつかの例では、参照ピクチャとして他のピクチャを使用することにインター予測を限定するのは対照的に、ビデオデコーダ30は、現在のピクチャ内に含まれたビデオデータのブロックを予測するために、参照ピクチャとして現在のピクチャを使用してもよい。たとえば、予測処理ユニット71は、予測処理ユニット71内に現在のピクチャのバージョンを記憶してもよい。いくつかの例では、予測処理ユニット71は、固定値に初期化されたピクセル値を有する現在のフレームの初期化バージョンを記憶してもよい。いくつかの例では、固定値は、現在のピクチャのサンプルのビット深度に基づいてもよい。たとえば、固定値は、 $1 \ll (\text{bitDepth} - 1)$ であってもよい。いくつかの例では、予測処理ユニット71は、現在のピクチャの任意のブロックを符号化する前に、現在のピクチャの初期化バージョンを記憶してもよい。現在のピクチャの初期化バージョンを記憶することによって、予測処理ユニット71は、まだ再構成されていない予測ブロックを使用してもよい。対照的に、予測処理ユニット71が現在のピクチャの初期化バージョンを記憶していない場合、すでに再構成されたブロックのみが、(すなわち、デコーダ/エンコーダのミスマッチを回避するために)予測子ブロックとして使用されてもよい。

30

【0120】

上記で説明したように、予測処理ユニット71は、現在のピクチャのための1つまたは複数のRPLを生成してもよい。たとえば、予測処理ユニット71は、現在のピクチャのためのRPLに現在のピクチャを含めてもよい。

40

【0121】

上記で説明したように、ビデオデコーダ30は、予測ブロックに基づいて、ビデオデータの現在のピクチャのビデオデータのブロックを復号してもよい。いくつかの例では、動き補償ユニット72は、現在のピクチャの現在のブロックのための予測ブロックとして使用するために、現在のピクチャ内に位置するブロックを選択してもよい。具体的には、予測処理ユニット71は、現在のブロックのために、現在のピクチャを含むRPLを構成してもよく、動き補償ユニット72は、RPL内のインデックスを示す現在のブロックのための動きパラ

50

メータを受信してもよい。いくつかの例では、インデックスは、RPL内の現在のピクチャを識別してもよい。これが生じたとき、動き補償ユニット72は、現在のブロックに対する動きベクトルによって識別された位置において現在のピクチャ自体から予測子ブロックを抽出するために、動きパラメータ内に含まれる動きベクトルを使用してもよい。このようにして、動き補償ユニット72は、イントラBCを実行してもよい。

【0122】

予測処理ユニット71は、ビデオデータの現在のブロックとビデオデータの予測子ブロックとの間の変位を表す動きベクトルを決定してもよい。いくつかの例では、予測処理ユニット71は、符号化ビデオビットストリームにおいて受信された1つまたは複数の構文要素に基づいて動きベクトルを決定してもよい。いくつかの例では、予測処理ユニット71は、整数精度の動きベクトルを決定してもよい。現在のピクチャが長期参照ピクチャとしてマークされている場合などの、そのような例では、予測処理ユニット71は、現在のピクチャを予測するために通常の長期参照ピクチャ(すなわち、現在のピクチャではない長期参照ピクチャ)を使用しなくてもよい。

【0123】

いくつかの例では、予測処理ユニット71は、異なるレベルの精度で動きベクトルを決定してもよい。たとえば、予測処理ユニット71は、整数精度、デフォルトの精度、または最も詳細な動き精度(たとえば、HEVCにおける1/4pel精度)で動きベクトルを決定してもよい。いくつかの例では、予測処理ユニット71は、たとえば、現在のピクチャによって参照されるSPSまたはVPS内で、コーディングされたイントラBC動きベクトルの精度を示す構文要素を受信してもよい。いくつかの例では、イントラBC動きベクトルの精度は、ピクチャレベルで適応されてもよく、予測処理ユニット71は、たとえば、現在のブロックによって参照されるPPSまたはスライスにおいて、コーディングされたイントラBC動きベクトルの精度を示す構文要素を受信してもよい。

【0124】

動き補償ユニット72は、動きベクトルおよび他の構文要素を構文解析することによって、現在のビデオスライスのビデオブロックのための予測情報を決定し、復号されている現在のブロックのための予測ブロックを生成するために予測情報を使用する。たとえば、動き補償ユニット72は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード(たとえば、イントラまたはインター予測)と、インター予測スライスタイプ(たとえば、BスライスまたはPスライス)と、スライスのための1つまたは複数の参照ピクチャリストのための構成情報と、スライスの各インター符号化ビデオブロックのための動きベクトルと、スライスの各インターコーディングビデオブロックのためのインター予測状態と、現在のビデオスライス内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、受信した構文要素のうちのいくつかを使用する。

【0125】

動き補償ユニット72はまた、補間フィルタに基づいて補間を実行してもよい。動き補償ユニット72は、参照ブロックのサブ整数ピクセルに関する補間値を計算するために、ビデオブロックの符号化の間にビデオエンコーダ20によって使用されるように補間フィルタを使用してもよい。この場合には、動き補償ユニット72は、受信した構文要素からビデオエンコーダ20によって使用された補間フィルタを決定し、予測ブロックを生成するために補間フィルタを使用してもよい。一例として、予測子ブロックを示す動きベクトルが分数ピクセル解像度を有する場合、動き補償ユニット72は、予測子ブロックのサブ整数ピクセルのための補間値を計算するために分数ピクセル補間を実行してもよい。いくつかの場合には、予測子ブロックのサブ整数ピクセルのための補間値を計算するために分数ピクセル補間を実行するために、動き補償ユニット72は、予測子ブロック内および予測子ブロックの外部からのサンプルピクセル値を参照する必要がある可能性がある。しかしながら、いくつかの例では、動き補償ユニット72が予測子ブロックの外部からのサンプルピクセル値を参照することは不可能であることがある。

【0126】

本開示の1つまたは複数の技法によれば、動き補償ユニット72は、予測子ブロックが選択され得る探索範囲は、動きベクトルの解像度に基づいて変化し得るので、予測子ブロックの外部からのサンプルピクセル値を参照することができる可能性がある。一例として、動きベクトルが整数ピクセル解像度を有するとき、現在のブロックのための探索範囲は、初期探索範囲であってもよい。別の例として、動きベクトルが分数ピクセル解像度を有する場合、現在のブロックのための探索範囲は、縮小探索範囲であってもよい。いくつかの例では、縮小探索範囲は、初期探索範囲の右側境界および下側境界からMサンプルだけ初期探索範囲を縮小し、初期探索範囲の上側境界および左側境界からNサンプルだけ初期探索範囲を縮小することによって決定されてもよい。いくつかの例では、予測子ブロックのピクセル値を決定するために分数ピクセル補間を実行するとき、動き補償ユニット72は、予測子ブロック内からのサンプルピクセル値と、縮小探索範囲の外部だが、初期探索範囲内からのサンプルピクセル値とを参照してもよい。しかしながら、サンプルピクセル値は、依然として初期探索範囲内に含まれているので、動き補償ユニット72は、依然として前記サンプルピクセル値を参照することができる可能性がある。

10

【0127】

逆量子化処理ユニット76は、ビットストリーム内で提供され、エントロピー復号ユニット70によって復号された量子化変換係数を逆量子化する(inverse quantize)、すなわち逆量子化する(de-quantize)。逆量子化プロセスは、量子化の程度と、同様に、適用されるべき逆量子化の程度とを決定するために、ビデオスライス内の各ビデオブロックのための、ビデオデコーダ30によって計算された量子化パラメータ QP_V の使用を含んでもよい。

20

【0128】

逆変換処理ユニット78は、ピクセル領域における残差ブロックを生成するために、変換係数に逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換プロセスを適用する。ビデオデコーダ30は、逆変換プロセスユニット78からの残差ブロックを動き補償ユニット72によって生成された対応する予測ブロックと加算することによって、復号ビデオブロックを形成する。加算器80は、この加算演算を実行する1つまたは複数の成分を表す。

【0129】

ビデオデコーダ30は、フィルタリングユニットを含んでもよく、フィルタリングユニットは、いくつかの例では、上記で説明したビデオエンコーダ20のフィルタリングユニットと同様に構成されてもよい。たとえば、フィルタリングユニットは、符号化ビットストリームからのビデオデータを復号し、再構成するとき、デブロッキング、SAO、または他のフィルタリング動作を実行するように構成されてもよい。

30

【0130】

技法のいくつかの異なる態様および例が本開示で説明されているが、技法の様々な態様および例は、互いに一緒にまたは別々に実行されてもよい。言い換えれば、技法は、上記で説明した様々な態様および例に厳密に限定されるべきではないが、組み合わせて使用されてもよいし、または一緒におよび/もしくは別々に実行されてもよい。加えて、特定の技法は、ビデオデコーダ30の特定のユニットに起因することがあるが、ビデオデコーダ30の1つまたは複数の他のユニットはまた、そのような技法を実行することを担当してもよいことが理解されるべきである。

40

【0131】

このようにして、ビデオデコーダ30は、本開示に記載の1つまたは複数の例示的な技法を実装するように構成されてもよい。たとえば、ビデオデコーダ30は、たとえば、イントラBCモードを使用して現在のピクチャの1つまたは複数のブロックをコーディングする目的のため、PPSを参照するピクチャがピクチャ自体の参照ピクチャリスト内に存在し得るかどうかを示す構文要素を含むビットストリームを受信するように構成されてもよい。すなわち、ビデオデコーダ30は、現在のピクチャがそれ自体のための参照ピクチャリスト内に含まれ得ることを示す構文要素のための値を復号してもよい。したがって、ブロックがイントラBCモードを使用してコーディングされるとき、ビデオデコーダ30は、たとえば、

50

インデックス値がピクチャ自体に対応するように参照ピクチャリスト内へのインデックス値を使用して、ブロックのための参照ピクチャがブロックを含むピクチャであることを決定してもよい。ビデオデコーダ30は、イントラBCモードを使用してコーディングされるブロックの動き情報内からこのインデックス値を復号してもよい。いくつかの例では、ビデオデコーダ30のハードウェアアーキテクチャは、現在のピクチャの現在のブロックを予測するために、参照ピクチャとして現在のピクチャを使用するために特に適合されなくてもよい。

【0132】

図5は、本開示の1つまたは複数の技法による、イントラBCプロセスの一例を示す図である。1つの例示的なイントラ予測プロセスによれば、ビデオエンコーダ20は、たとえば、ピクチャ内のコーディングされるべき現在のブロックについて、たとえば、同じピクチャ内の以前にコーディングされ、再構成されたビデオデータのブロックのセットから予測子ビデオブロックを選択してもよい。図5の例では、再構成された領域108は、以前にコーディングされ、再構成されたビデオブロックのセットを含む。再構成された領域108内のブロックは、ビデオデコーダ30によって復号され、再構成され、再構成領域メモリ92内に記憶されているブロック、または、ビデオエンコーダ20の再構成ループ内で復号および再構成され、再構成メモリ領域64内に記憶されているブロックを表してもよい。現在のブロック102は、コーディングされるべきビデオデータの現在のブロックを表す。予測子ブロック104は、現在のブロック102のイントラBC予測のために使用される、現在のブロック102と同じピクチャ内の再構成されたビデオブロックを表す。

【0133】

例示的なイントラ予測プロセスでは、ビデオエンコーダ20は、探索範囲内から予測子ブロック104を選択してもよい。上記で説明したように、本開示の1つまたは複数の技法によれば、ビデオエンコーダ20は、予測子ブロック104を示す動きベクトルのために使用されるべき解像度(すなわち、動きベクトル106のために使用される解像度)に基づいて探索範囲を決定してもよい。図5の例では、整数ピクセル解像度が動きベクトル106のために使用されることを決定することに基づいて、ビデオエンコーダ20は、探索範囲が再構成された範囲108からなることを決定し、再構成された範囲108から予測子ブロック104を選択してもよい。次いで、ビデオエンコーダ20は、動きベクトル106を決定し、符号化してもよく、動きベクトル106は、残差信号と一緒に、現在のブロック102に対する予測子ブロック104の位置を示す。たとえば、図5によって示すように、動きベクトル106は、現在のブロック102の左上隅に対する予測子ブロック104の左上隅の位置を示してもよい。上記で説明したように、動きベクトル106はまた、オフセットベクトル、変位ベクトル、またはブロックベクトル(BV)と呼ばれることもある。ビデオデコーダ30は、現在のブロックを復号するための符号化された情報を利用してもよい。

【0134】

ビデオデコーダ30は、RPLに基づいて、現在のピクチャ内のビデオデータのブロックを復号してもよい。具体的には、ビデオデコーダ30は、参照ピクチャメモリ82内に記憶された現在のピクチャのバージョン内に含まれる予測子ブロックに基づいて、ビデオデータのブロックを復号してもよい。言い換えれば、現在のピクチャのブロックを復号するとき、ビデオデコーダ30は、現在のピクチャ、すなわち、参照インデックスIdxCur(ListX内)による参照からブロックを予測してもよい。ビデオデコーダ30は、(たとえば、ビデオデコーダがブロックの復号を終了した後)初期化された値を置換するために、現在のピクチャのバッファ(たとえば、参照ピクチャメモリ82)にブロックの再構成されたサンプルを書き込んでもよい。この例では、ビデオデコーダ30は、ブロックを復号した後、再構成されたサンプルにデブロッキング、SAO、またはなにか他のフィルタリング動作を適用しないことに留意されたい。言い換えれば、ビデオデコーダ30は、デブロッキングおよびSAOを適用する前に、参照として現在のピクチャを使用してもよい。ピクチャ全体をコーディングした後、ビデオデコーダ30は、デブロッキング、SAO、および、HEVCバージョン1に記載のものと同様の方法における参照ピクチャマーキングなどの他の動作を適用してもよい。

【0135】

図6Aおよび図6Bは、本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。図6に示すように、ビデオエンコーダ20は、現在のSCCドラフト(たとえば、原因の(causal)または再構成された範囲)におけるイントラBCのための利用可能な探索範囲を表す探索範囲600を決定してもよい。本開示の技法によれば、ビデオエンコーダ20は、縮小探索範囲を決定してもよい。たとえば、予測子ブロックを識別するために使用される動きベクトルが分数ピクセル解像度を有することになることを決定することに応答して、ビデオエンコーダ20は、縮小探索範囲602(たとえば、図6A中の縮小探索範囲602Aまたは図6B中の縮小探索範囲602B)を決定してもよい。図6Aの例では、ビデオエンコーダ20は、水平方向に距離604、垂直方向に距離606だけ探索範囲600を縮小することによって、縮小探索範囲602Aを決定してもよい。図6Bの例では、ビデオエンコーダ20は、水平方向に距離604だけ探索範囲600を縮小することによって、縮小探索範囲602Bを決定してもよい。距離604は、Nサンプルであってもよく、Nは、輝度/彩度補間のために使用されるフィルタタップの数(たとえば、分数ピクセル補間プロセス中にエンコーダ20および/またはビデオデコーダ30によって使用されるタップの数)に依存してもよい。距離606は、Mサンプルであってもよく、Mは、輝度/彩度補間のために使用されるフィルタタップの数に依存してもよい。いくつかの例では、Nは、Mに等しくてもよい。他の例では、NおよびMは、異なってもよい。

10

【0136】

いくつかの例では、ビデオエンコーダ20は、サンプリング比に基づいて縮小探索範囲を決定してもよい。一例として、ビデオエンコーダ20は、すべての彩度サンプリング比に対して縮小探索範囲602Aを使用してもよい。別の例として、ビデオエンコーダ20は、4:0:0および4:4:4を除くすべての彩度サンプリング比に対して縮小探索範囲602Aを使用してもよい。別の例として、ビデオエンコーダ20は、4:2:0彩度サンプリング比に対して縮小探索範囲602Aを使用し、4:2:2彩度サンプリング比に対して縮小探索範囲602Bを使用してもよい。図6Aおよび図6Bに見られるように、縮小探索範囲602Bは、水平方向にのみ縮小されるという点で602Aと異なる。

20

【0137】

図7は、本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。図7に示すように、ビデオエンコーダ20は、現在のSCC(たとえば、原因のまたは再構成された範囲)におけるイントラBCのための利用可能な探索範囲を表す(垂直線で網掛けされた)探索範囲702を決定してもよい。本開示の技法によれば、ビデオエンコーダ20は、初期探索範囲の1つまたは複数の境界をサンプルの数だけオフセットすることによって縮小探索範囲を決定してもよい。たとえば、予測子ブロックを識別するために使用される動きベクトルが分数ピクセル解像度を有することになることを決定することに応答して、ビデオエンコーダ20は、探索範囲702の下側境界および右側境界を距離706だけ縮小することによって、(水平線で網掛けされた)縮小探索範囲704を決定してもよい。距離706は、Nサンプルであってもよく、Nは、輝度/彩度補間のために使用されるフィルタタップの数に依存してもよい。

30

【0138】

いくつかの例では、初期探索範囲(すなわち、探索範囲702)の下側境界および右側境界をオフセットすることに加えて、ビデオエンコーダ20は、スライス境界、タイル境界を含む、初期探索範囲の任意の境界において初期探索範囲をオフセットしてもよい。

40

【0139】

いずれの場合にも、ビデオエンコーダ20は、決定された探索範囲から現在のブロック700のための予測子ブロックを選択してもよい。一例として、予測子ブロックを示す動きベクトルが整数ピクセル解像度を有することになる場合、ビデオエンコーダ20は、探索範囲702から現在のブロック700のための予測子ブロックを選択してもよい。別の例として、予測子ブロックを示す動きベクトルが分数ピクセル解像度を有することになる場合、ビデオエンコーダ20は、縮小探索範囲704から現在のブロック700のための予測子ブロックを選択

50

してもよい。

【0140】

図8は、本開示の1つまたは複数の技法による、予測子ブロックが選択され得る例示的な探索範囲を示す図である。図8に示すように、ビデオエンコーダ20は、現在のSCC(たとえば、原因のまたは再構成された範囲)におけるイントラBCのための利用可能な探索範囲を表す(太線で囲まれた)初期探索範囲802を決定してもよい。本開示の技法によれば、ビデオエンコーダ20は、(たとえば、初期探索範囲の指定されたマージン内のブロックのピクセルを探索における考慮から除去することによって)初期探索範囲の1つまたは複数の境界をオフセットすることによって縮小探索範囲を決定してもよい。たとえば、ビデオエンコーダ20は、探索範囲802の上側境界および左側境界を距離806だけ縮小し、探索範囲802の下側境界および右側境界を距離808だけ縮小することによって、(破線で囲まれた)縮小探索範囲804を決定してもよい。距離806は、Nサンプル(たとえば、1、2、3、4、5、6、7、8)であってもよく、Nは、輝度/彩度補間のために使用されるフィルタタップの数に依存してもよい。距離808は、Mサンプル(たとえば、1、2、3、4、5、6、7、8)であってもよく、Mは、輝度/彩度補間のために使用されるフィルタタップの数に依存してもよい。たとえば、Mは、フィルタタップの数とともに対応して増加してもよい。いくつかの例では、Nは、Mと等しくてもよい。他の例では、NおよびMは、異なってもよい。1つの特定の例では、NおよびMは、2つのサンプルであってもよい。

10

【0141】

いくつかの例では、初期探索範囲(すなわち、探索範囲802)の上側境界、下側境界、左側境界、および右側境界をオフセットすることに加えて、ビデオエンコーダ20は、スライス境界、タイル境界を含む、初期探索範囲の任意の境界において初期探索範囲をオフセットしてもよい。

20

【0142】

いずれの場合にも、ビデオエンコーダ20は、決定された探索範囲から現在のブロック800のための予測子ブロックを選択してもよい。一例として、予測子ブロックを示す動きベクトルが整数ピクセル解像度を有することになる(すなわち、彩度補間が必要とされることになる)場合、ビデオエンコーダ20は、探索範囲802から現在のブロック800のための予測子ブロックを選択してもよい。別の例として、予測子ブロックを示す動きベクトルが分数ピクセル解像度を有することになる場合、ビデオエンコーダ20は、縮小探索範囲804から現在のブロック800のための予測子ブロックを選択してもよい。

30

【0143】

以下は、図8の技法の例示的な実装形態である。以前の例示的な実装形態に対する追加は、下線付きで示される。以前の例示的な実装形態に対する削除は、斜体で示される。

【0144】

一般的な例

【0145】

8.5.3.2.1一般

【0146】

IBC参照範囲の上側境界および左側境界に対する制約

40

【0147】

参照ピクチャが現在のピクチャであるとき、輝度動きベクトルmvLXが以下の制約に従わなければならないことが、ビットストリーム適合の要件である。

- 変数xRefおよびyRefは、以下のように導出される。
- ChromaArrayTypeが0に等しくなく、mvCLX[0]&0x7が0に等しくないとき、
- $xRef = (xPb + (mvLX[0] \gg 2) - IntSamplesX)$ であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[0] \gg 2))$ である。

- ChromaArrayTypeが0に等しくなく、mvCLX[1]&0x7が0に等しくなく、ここで、IntSample

50

sXが値0、1、2、3、4...をとることができるとき、

$xRef = (xPb + (mvLX[1] \gg 2) - IntSamplesY)$ であり、ここで、IntSamplesYは、値0、1、2...Nをとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[1] \gg 2))$ である。

- SCC WD6の6.4.1項において指定されたzスキャンオーダブロック利用可能性の導出プロセスが、(xCb,yCb)に等しく設定された(xCurr,yCurr)と、 $(xPb + (mvLX[0] \gg 2), (yPb + mvLX[1] \gg 2))$ (xRef,yRef)に等しく設定された隣接輝度位置(xNbY,yNbY)とを入力として用いて呼び出されたとき、出力は、TRUEに等しいものとする。

【0148】

10

IBC参照範囲の下側境界および右側境界に対する制約

- 変数xRefおよびyRefは、以下のように導出される。

- ChromaArrayTypeが0に等しくなく、 $mvCLX[0] \& 0x7$ が0に等しくないとき、

- $xRef = (xPb + (mvLX[0] \gg 2) + nPbW - 1 + IntSamplesX)$ であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[0] \gg 2) + nPbW - 1)$ である。

- ChromaArrayTypeが0に等しくなく、 $mvCLX[1] \& 0x7$ がゼロに等しくなく、ここで、IntSamplesXが値0、1、2、3、4...をとることができるとき、

$xRef = (xPb + (mvLX[1] \gg 2) + nPbH - 1 + IntSamplesY)$ であり、ここで、IntSamplesYは、値0、1、2...Nをとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[1] \gg 2) + nPbH - 1)$ である。

- 6.4.1項において指定されたzスキャンオーダブロック利用可能性の導出プロセスが、(xCb,yCb)に等しく設定された(xCurr,yCurr)と、 $(xPb + (mvLX[0] \gg 2) + nPbW - 1, yPb + (mvLX[1] \gg 2) + nPbH - 1)$ (xRef,yRef)に等しくされた隣接輝度位置(xNbY,yNbY)とを入力として用いて呼び出されたとき、出力は、TRUEに等しいものとする。

- 以下の条件の一方または両方は、真であるものとする。

- ChromaArrayTypeが0に等しくなく、 $mvCLX[0] \& 0x7$ が0に等しくないとき、

- $(mvLX[0] \gg 2) + nPbW + IntSamplesX + xB1$ の値は、0以下であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

- $(mvLX[0] \gg 2) + nPbW + xB1$ の値が0以下である。

ChromaArrayTypeが0に等しくなく、 $mvCLX[0] \& 0x7$ が0に等しくないとき、

そうでなければ、

- $(mvLX[1] \gg 2) + nPbH + IntSamplesY + yB1$ の値は、0以下であり、ここで、IntSamplesYは、値0、1、2、3...をとることができる。

- $(mvLX[0] \gg 2) + nPbH + xB1$ の値は、0以下である。

- 以下の条件は、真であるものとする。

$(xPb + (mvLX[0] \gg 2) + nPbSw - 1)xRef / CtbSizeY - xCurr / CtbSizeY \leq yCurr / CtbSizeY - (yPb + (mvLX[1] \gg 2) + nPbSh - 1)yRef / CtbSizeY$

【0149】

以下は、図8の技法の別の例示的な実装形態である。SCM4.0ソフトウェアに対する追加は、下線付きで示される。SCM4.0ソフトウェアに対する削除は、斜体で示される。

【0150】

8.5.3.2.1一般

【0151】

IBC参照範囲の上側境界および左側境界に対する制約

【0152】

参照ピクチャが現在のピクチャであるとき、輝度動きベクトルmvLXが以下の制約に従わ

50

なければならないことが、ビットストリーム適合の要件である。

- 変数xRefおよびyRefは、以下のように導出される。
- ChromaArrayTypeが0に等しくなく、mvCLX[0]&0x7が0に等しくないとき、
- $xRef = (xPb + (mvLX[0] \gg 2) - 2)$ であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[0] \gg 2))$ である。

- ChromaArrayTypeが0に等しくなく、mvCLX[1]&0x7が0に等しくなく、ここで、IntSamplesXが値0、1、2、3、4...をとることができるとき、

$xRef = (xPb + (mvLX[1] \gg 2) - 2)$ であり、ここで、IntSamplesYは、値0、1、2...Nをとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[1] \gg 2))$ である。

- 6.4.1項において指定されたzスキャンオーダブロック利用可能性の導出プロセスが、(xCb, yCb)に等しく設定された(xCurr, yCurr)と、 $(xPb + (mvLX[0] \gg 2), (yPb + mvLX[1] \gg 2))$ (xRef, yRef)に等しく設定された隣接輝度位置(xNbY, yNbY)とを入力として用いて呼び出されたとき、出力は、TRUEに等しいものとする。

【 0 1 5 3 】

IBC参照範囲の下側境界および右側境界に対する制約

- 変数xRefおよびyRefは、以下のように導出される。
- ChromaArrayTypeが0に等しくなく、mvCLX[0]&0x7が0に等しくないとき、
- $xRef = (xPb + (mvLX[0] \gg 2) + nPbW - 1 + 2)$ であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[0] \gg 2) + nPbW - 1)$ である。

- ChromaArrayTypeが0に等しくなく、mvCLX[1]&0x7がゼロに等しくなく、ここで、IntSamplesXが値0、1、2、3、4...をとることができるとき、

$xRef = (xPb + (mvLX[1] \gg 2) + nPbH - 1 + 2)$ であり、ここで、IntSamplesYは、値0、1、2...Nをとることができる。

そうでなければ、

$xRef = (xPb + (mvLX[1] \gg 2) + nPbH - 1)$ である。

- 6.4.1項において指定されたzスキャンオーダブロック利用可能性の導出プロセスが、(xCb, yCb)に等しく設定された(xCurr, yCurr)と、 $(xPb + (mvLX[0] \gg 2) + nPbW - 1, yPb + (mvLX[1] \gg 2) + nPbH - 1)$ (xRef, yRef)に等しくされた隣接輝度位置(xNbY, yNbY)とを入力として用いて呼び出されたとき、出力は、TRUEに等しいものとする。

- 以下の条件の一方または両方は、真であるものとする。
- ChromaArrayTypeが0に等しくなく、mvCLX[0]&0x7が0に等しくないとき、
- $(mvLX[0] \gg 2) + nPbW + 2 + xB1$ の値は、0以下であり、ここで、IntSamplesXは、値0、1、2、3...をとることができる。

そうでなければ、

- $(mvLX[0] \gg 2) + nPbW + xB1$ の値が0以下である。

ChromaArrayTypeが0に等しくなく、mvCLX[0]&0x7が0に等しくないとき、

- $(mvLX[1] \gg 2) + nPbH + 2 + yB1$ の値は、0以下であり、ここで、IntSamplesYは、値0、1、2、3...をとることができる。

そうでなければ、

- $(mvLX[0] \gg 2) + nPbH + xB1$ の値は、0以下である。

- 以下の条件は、真であるものとする。

$(xPb + (mvLX[0] \gg 2) + nPbSw - 1) \cdot xRef / CtbSizeY - xCurr / CtbSizeY \leq yCurr / CtbSizeY - (yPb + (mvLX[1] \gg 2) + nPbSh - 1) \cdot yRef / CtbSizeY$

【 0 1 5 4 】

10

20

30

40

50

上記で説明したように、本開示の技法は、輝度MV、または彩度MV、または輝度MVと彩度MVの両方が分数ピクセル解像度(たとえば、1/4ピクセル解像度)で表されるときの問題に対処する。上記で説明したように、MVが分数ピクセル位置を指し示すとき、予測ピクセルは、輝度/彩度用の補間フィルタを使用して取得される。補間フィルタは、予測ブロックの外部の追加の隣接サンプルを必要とすることがある。そのように、補間プロセスのために使用されるこれらのピクセル/サンプルは、上記で説明したように参照ブロックに対するイントラBC制約を満たす必要がある。上記の制約を実現するために、または本明細書で説明する他の技術と組み合わせて別々にまたは共同して適用され得るいくつかの技術を以下に提案する。

【0155】

一例として、参照範囲は、現在のブロックMVに基づいて制限されてもよい。言い換えれば、補間サンプルを有する参照ブロックが上記で説明したようにイントラBC制約を満たさないとき、特定の動きベクトル(輝度/彩度)が許可されなくてもよい。別の例として、補間サンプル(各側に4サンプル)を有する参照ブロックが上記説明したようにイントラBC制約を満たさないとき、特定の動きベクトル(輝度/彩度)が許可されなくてもよい。別の例として、ブロックに対応する動きベクトルが彩度補間を使用するとき、現在のIBC探索範囲は、(有効なIBC領域の4つの側すべてにおいて)いくつかのサンプル、たとえば、2サンプルだけ縮小されてもよい。別の例として、ブロックに対応する動きベクトルが彩度補間を使用するとき、現在のIBC探索範囲は、左側ピクチャ境界および上側ピクチャ境界に対して1サンプル、右側IBC参照範囲および下側IBC参照範囲に対して2サンプルだけ縮小されてもよい。別の例として、ブロックに対応する動きベクトルが彩度補間を使用するとき、現在のIBC参照範囲は、左側ピクチャ境界および上側ピクチャ境界に対して2サンプル、右側IBC参照範囲および下側IBC参照範囲に対して4サンプルだけ縮小されてもよい。一般に、図8を参照して上記で説明したように、上記で説明した制約を満たす有効なIBC参照範囲は、上側境界および左側境界に対してNサンプル、右側境界および下側境界に対してMサンプルだけ縮小されてもよい。いくつかの例では、上記で説明した制約は、白黒プロファイルに対して適用されない。

【0156】

図9は、本開示の1つまたは複数の技法による、ピクチャの外部に位置する領域をパディングするための例示的な技法を示す図である。RExt WD7では、イントラBC参照範囲は、現在のピクチャの完全に内部に制限される。この制約のため、上記で説明したオフセット(たとえば、Nサンプルだけの参照範囲の縮小)は、ピクチャ境界を越えるのを避けるために適用され得る。

【0157】

しかしながら、いくつかの例では、探索範囲(すなわち、イントラBC参照範囲)は、通常のインターモードのために行われるものと同様にピクチャ境界の外部に位置するピクセルを含んでもよく、ここで、ピクチャ境界内の最も近いピクセルを必要な外部のサンプルにコピーすること(パディングプロセス)によってピクチャの外部の探索範囲内のサンプル値が導出される。

【0158】

同様に、ピクチャの内部に位置するピクチャ境界の次のブロックがすでに再構成されているとき、ビデオコード(たとえば、ビデオエンコード20および/またはビデオデコード30)は、図9において矢印で示すように、境界ピクセルを外側空間に拡張するために(すなわち、ピクチャ境界の外部に位置するピクセルのためのピクセル値を生成するために)、パディングプロセスを適用してもよい。

【0159】

たとえば、外部または部分的に外部のブロックがイントラBC参照として利用可能であるかどうかをチェックするために、ビデオコードは、左上隅、右上隅、左下隅、および/または右下隅などの参照ブロックの隅の座標を、ピクチャ境界にクリップしてもよい(たとえば、水平座標は、0とピクチャ幅マイナス1の範囲内にクリップされてもよく、垂直座標

10

20

30

40

50

は、0とピクチャ高さマイナス1の範囲内にクリップされてもよい)。ビデオコードは、破線境界ブロックとして図9に示すそのようなクリップされたブロックに対して有効性チェックを実行してもよく、または、参照ブロックのクリップされた隅座標がすでに再構成されているかどうか、すなわち、ピクチャの外部に位置するもしくは部分的にピクチャの外部に位置する参照ブロックのためのIBC予測として使用され得るかどうかをチェックしてもよい。

【0160】

たとえば、ビデオコードは、上述したように、左上隅の水平座標および垂直座標をピクチャ境界にクリップしてもよく、すなわち、参照ブロックは、クリップされた点において左上隅を有するように移動される。クリッピングの後、2つの可能性が存在し、すなわち、参照ブロックは、ピクチャの内部に位置するか、または、依然としてピクチャの外部もしくは部分的にピクチャの外部に位置する。参照ブロックがピクチャの外部に位置する場合、ビデオコードは、ブロックの右下隅の水平座標がピクチャの外部に延びている距離だけブロックをピクチャ内に水平に移動してもよい。このプロセスは、概略的に、最初に基準ブロックを垂直方向においてピクチャ内に移動し、水平方向におけるブロック並進移動を続けるものと見ることができる。代替的には、プロセスは、逆であってもよく、ビデオコードは、最初に右下隅を水平に移動し、垂直方向における移動を続けてもよい。

【0161】

参照ブロックが現在のピクチャの外部または部分的に現在のピクチャの外部に位置する場合などのいくつかの例では、ビデオコードは、ピクチャ境界ピクセルが再構成され、イントラBC予測のために使用されるために利用可能であるかどうかを決定するために、クリップされた参照ブロックの隅と、参照ブロックの幅および高さを考慮してもよい。

【0162】

いくつかの例では、ビデオコードは、補間を可能にするために、上記で説明したように、拡張された参照範囲にオフセットを適用してもよい。たとえば、現在のブロックの真上の外部空間内に位置する参照ブロックは、まだ利用可能ではない。結果として、そのような領域内に位置するピクセルは、たとえば、説明した利用不可能な部分の近くに位置する参照ブロックのための補間のために使用することができない。したがって、ビデオコードは、補間を可能にするために、利用可能な範囲の内部に縮小するためにNピクセルのオフセットを適用してもよい。代替的には、オフセットを適用するのは対照的に、ビデオコードは、最も近い利用可能なピクセルをパディングしてもよい。

【0163】

RExt WD7では、イントラBC参照範囲は、(ブロック700を含まない)図7の斜線範囲702によって示すような範囲内に制限される。RExt WD7により、図7中の補間サンプルは、RExt WD7ではゼロであり、Nは、ブロック幅に等しい。

【0164】

RExt WD7では、IBC参照範囲の一部は、以下のときに利用不可能である。

- a) 参照範囲がピクチャ/スライス/タイル境界の外部にある。
- b) 参照範囲が、以下に定義する並列処理サンプル境界の外部にある。

$$(xPb+(mvLX[0]>>2)+nPbSw-1)/CtbSizeY-xCurr/CtbSizeY<=yCurr/CtbSizeY-(yPb+(mvLX[1]>>2)+nPbSh-1)/CtbSizeY$$
- c) 参照範囲が現在のコーディングユニットと重複する。
- d) 参照範囲が、参照のために利用可能ではない範囲と重複する(たとえば、参照範囲が制約されたイントラ予測範囲内に入る)。ブロック利用可能性の導出は、RExt WD7内の6.4.1項で詳述される。

【0165】

上記の制約は、RExt WD7においてエンコード適合性制約の形式で規定され、すなわち、SCCビットストリームを生成することができるエンコードは、上記の参照範囲制約に従わなければならない。

【0166】

しかしながら、RExt WD7において規定された技法は、望ましくない可能性がある1つまたは複数の問題を生じる可能性がある。一例として、RExt WD7において規定されたエンコーダ側の制約は、潜在的に危険である可能性がある。たとえば、非適格なエンコーダがこれらの制約に従わないとき、デコーダの挙動は、未定義である可能性がある。別の例として、RExt WD7において規定されたエンコーダ制約は、ビットストリームが境界外の範囲を参照するとき、制約する代わりに一時的な参照範囲がデコーダにおいて有効な領域にクリップされるインター復号プロセスと整合しない可能性がある。別の例として、参照範囲を制約することは、いくつかのシナリオにおいてコーディング効率の低下を生じる可能性がある。たとえば、予測ブロックの大部分が有効範囲内に入るシナリオでは、予測ブロック全体を制約する代わりに、残りのブロックを有効範囲内にクリップすることが有益であることがある。

10

【0167】

本開示は、上記の問題に対処し得るいくつかの技法を提案する。以下の技法は、本開示の他の技法と組み合わせて別々にまたは共同して適用され得る。以下で使用される参照範囲という用語は、予測ブロック内のサンプルまたはサンプルのセットに対応することができる。

【0168】

本開示の1つまたは複数の技法によれば、ビットストリームが境界外の範囲を参照するとき、IBC参照範囲は、デコーダにおいて有効な領域にクリップされてもよい。いくつかの例では、クリッピングは、以下のシナリオの1つまたは複数に適用されてもよい。一例として、ビデオコード(すなわち、ビデオエンコーダまたはビデオデコーダ)は、参照範囲がピクチャ境界の外部にあるとき、クリッピングを適用してもよい。別の例として、ビデオコードは、参照範囲が現在のコーディングユニットと重複するとき、クリッピングを適用してもよい。別の例として、ビデオコードは、参照領域が以下の式で定義される並列処理サンプル境界の外部にあるとき、クリッピングを適用してもよい。

20

$$(xPb+(mvLX[0]>>2)+nPbSw-1)/CtbSizeY-xCurr/CtbSizeY<=$$

$$yCurr/CtbSizeY-(yPb+(mvLX[1]>>2)+nPbSh-1)/CtbSizeY$$

【0169】

いくつかの例では、パディングは、以下のシナリオの1つまたは複数に適用されてもよい。一例として、ビデオコードは、参照範囲がスライス境界の外部にあるとき、パディングを適用してもよい。別の例として、ビデオコードは、参照範囲がタイル境界の外部にあるとき、パディングを適用してもよい。別の例として、ビデオコードは、参照範囲が、参照のために利用可能ではない領域と重複する(たとえば、参照範囲が、制約されたイントラ予測範囲内に入る)とき、パディングを適用してもよい。ビデオコードは、RExt WD7内の6.4.1項に記載の技法を使用して、領域が利用可能であるかどうかを決定してもよい。

30

【0170】

いくつかの例では、パディングプロセスは、以下のように適用されてもよい。ビデオコードは、予測ブロックのためのサンプルアレイ S_L 内の各要素の値を $1<(\text{BitDepth}_Y-1)$ に等しく設定してもよい。ChromaArrayTypeが0に等しくないとき、ビデオコードは、ピクチャのためのサンプルアレイ S_{Cb} および S_{Cr} 内の各要素の値を $1<(\text{BitDepth}_C-1)$ に等しく設定してもよい。ビデオコードは、予測ブロック内の各要素について、予測モードCuPredMode[x][y]をMODE_INTRAに等しく設定してもよい。

40

【0171】

上記で提案した技法は、異なる参照予測ブロックをもたらし得る異なる粒度レベル(たとえば、サンプルレベルまたは予測ブロックレベル)において適用されてもよい。

【0172】

本開示の1つまたは複数の技法によれば、ビデオコードは、予測ブロック内のサンプルのすべてが有効参照範囲内に入るように、予測ブロックをクリップしてもよい。たとえば、ビデオコードは、予測ブロック内のすべてのサンプルがピクチャ境界内にあり、予測ブロック内のすべてのサンプルが現在のコーディングユニットと重複せず、予測ブロック内

50

のすべてのサンプルが、以下の式によって定義される並列処理サンプル境界内にあるように、予測ブロックをクリップしてもよい。

$$(xPb+(mvLX[0]>>2)+nPbSw-1)/CtbSizeY-xCurr/CtbSizeY<=yCurr/CtbSizeY-(yPb+(mvLX[1]>>2)+nPbSh-1)/CtbSizeY$$

【 0 1 7 3 】

本開示の技法は、いくつかの方法で実施されてもよい。以下は、提案した技法の1つまたは複数の1つの例示的な実装形態に過ぎない。

【 0 1 7 4 】

この例では、位置 $(xA_{i,j}, yA_{i,j})$ は、輝度サンプルの $refPicLX_L=CurrPic$ であるときの所与の2次元アレイの内部の全サンプル位置における現在のピクチャの左上輝度サンプルに対する現在の輝度予測ブロックの右下サンプル位置を表すものとする。

$yA_{i,j}=Clip3(samples_{inter}, (CtbY_{curr}+1)<<CtbLog2SizeY-samples_{inter}, yInt_L+j)$
 $CtbY_{curr}=CtbY_{pred}$ && $CtbX_{curr} \geq CtbX_{pred}$ (すなわち、同じCTU行内)であるとき、

$xA_{i,j}=Clip3(samples_{inter}, (xA_{i,j}-(nPbW<<1)-xB1-1-samples_{inter}, xInt_L+i))$ である。
 そうでなければ、

$xA_{i,j}=Clip3(samples_{inter}, ((CtbY_{curr}-CtbY_{pred}+1))<<CtbLog2SizeY)+xCurr_{ctb}-nPbW-xB1-1-samples_{inter}, xInt_L+i))$ である。

【 0 1 7 5 】

上記の式において、 $nPbW$ および $nPbH$ は、それぞれ、輝度予測ブロックの幅および高さを指定し、 $CtbY_{curr}=yCurr>>CtbLog2SizeY$ 、 $CtbY_{pred}=(yA_{i,j})>>CtbLog2SizeY$ 、 $CtbX_{curr}=xCurr>>CtbLog2SizeY$ 、 $xCurr_{ctb}=(CtbX_{curr})<<CtbLog2SizeY$ 、 $xB1$ は、現在の輝度コーディングブロック左上サンプルに対する現在の輝度予測ブロックの左上サンプルの輝度位置であり、 $samples_{inter}$ は、輝度サンプルまたは彩度サンプルのために使用される最大補間サンプルに対応する。

【 0 1 7 6 】

図10は、本開示の1つまたは複数の技法による、上記のクリッピング動作のいくつかの例示的なシナリオを示す。図10のシナリオ/ケースでは、予測ブロックは、実線の境界線で示され、クリップされた予測ブロックは、破線の境界線で示されている。図10に示すように、参照ブロック全体(すなわち、予測ブロック)が有効参照範囲内にクリップされ得、これらのシナリオではパディングが必要でないことが観察され得る。

【 0 1 7 7 】

本開示の1つまたは複数の技法によれば、ビデオコードは、各予測サンプルが有効参照範囲内に入るように、各参照サンプルをクリップしてもよい。たとえば、ビデオコードは、参照ブロック内のサンプルが有効参照範囲内に入るように、参照サンプルをクリップしてもよい。たとえば、ビデオコードは、予測ブロック内のサンプルがピクチャ境界内にあり、予測ブロック内のサンプルが現在のコーディングユニットと重複せず、予測ブロック内のサンプルが、以下の式によって定義される並列処理サンプル境界内にあるように、予測ブロックをクリップしてもよい。

$$(xPb+(mvLX[0]>>2)+nPbSw-1)/CtbSizeY-xCurr/CtbSizeY<=yCurr/CtbSizeY-(yPb+(mvLX[1]>>2)+nPbSh-1)/CtbSizeY$$

【 0 1 7 8 】

図11は、上記のクリッピング動作のいくつかの例示的なシナリオを示す。図11のシナリオ/ケースでは、予測ブロックは、実線の境界線で示され、クリップされた予測ブロックは、破線の境界線で示されている。図11に示すように、特定の予測サンプルが有効参照範囲内にクリップされ、いくつかのシナリオではパディングが必要であり得ることが観察され得る。

【 0 1 7 9 】

この例では、位置 $(xA_{i,j}, yA_{i,j})$ は、輝度サンプルの $refPicLX_L=CurrPic$ であるときの所与の2次元アレイの内部の全サンプル位置における輝度参照サンプルを表すものとする。

$$yA_{i,j}=Clip3(samples_{inter}, (CtbY_{curr}+1)<<CtbLog2SizeY-samples_{inter}, yInt_L+j)$$

$CtbY_{curr} = CtbY_{pred} \&\& CtbX_{curr} \geq CtbX_{pred}$ (すなわち、同じCTU行内)であるとき、
 $xA_{i,j} = Clip3(samples_{inter}, (xA_{i,j} - (nPbW < 1) - xB1 - 1 - samples_{inter}, xInt_L + i))$ である。
 そうでなければ、

$xA_{i,j} = Clip3(samples_{inter}, ((CtbY_{curr} - CtbY_{pred} + 1)) < CtbLog2SizeY) + xCurr_{ctb} - nPbW - xB1 - 1 - samples_{inter}, xInt_L + i)$ である。

【0180】

上記の式において、nPbWおよびnPbHは、それぞれ、輝度予測ブロックの幅および高さを指定し、 $CtbY_{curr} = yCurr >> CtbLog2SizeY$ 、 $CtbY_{pred} = (yA_{i,j}) >> CtbLog2SizeY$ 、 $CtbX_{curr} = xCurr >> CtbLog2SizeY$ 、 $xCurr_{ctb} = (CtbX_{curr}) << CtbLog2SizeY$ 、xB1は、現在の輝度コーディングブロック左上サンプルに対する現在の輝度予測ブロックの左上サンプルの輝度位置であり、 $samples_{inter}$ は、輝度サンプルまたは彩度サンプルのために必要とされる最大補間サンプルに対応する。

10

【0181】

以後、「SCC WD3」と呼ぶ、SCC仕様の以前の作業草案(WD)は、http://phenix.it-sudparis.eu/jct/doc_end_user/documents/22_Geneva/wg11/JCTVC-T1005-v2.zipから利用可能である。SCC WD3では、輝度動きベクトルおよび彩度動きベクトルは、 $use_integer_mv_flag$ が1に等しいとき、動き補償の直前にスケーリング($<<2$)され、クリップされる。

【0182】

これは、以下に示すようにSCC WD3において規定されている。

【0183】

20

$use_integer_mv_flag$ が1に等しく、参照インデックス $refIdxLX$ が $currPic$ に等しくないとき、 $mvLX$ および $mvCLX$ (X が0または1の場合)は、以下のように修正される。

$mvLX = Clip3(-2^{15}, 2^{15} - 1, mvLX < < 2)$ (8-203)

$mvCLX = Clip3(-2^{15}, 2^{15} - 1, mvCLX < < 2)$ (8-204)

【0184】

いくつかの例では、スケーリングされた動きベクトル(輝度および彩度)を $2^{15} - 1$ にクリップすることは、補間を実行する必要性をもたらすことがある。補間は、他の場合には必要とされないことがある。この特殊な場合は、ビデオデコードに不必要な複雑さを加えることがあり、これは、望ましくない可能性がある。

【0185】

30

本開示の1つまたは複数の技法によれば、ビデオデコードは、動きベクトルを最も近い値にクリップしてもよく、これは、補間をもたらさない。

【0186】

$use_integer_mv_flag$ が1に等しく、参照インデックス $refIdxLX$ が $currPic$ に等しくないとき、 $mvLX$ および $mvCLX$ (X が0または1の場合)は、以下のように修正される(SCC WDに対する追加は、下線付きで示され、SCC WD3に対する削除は、斜体で示される)。

$mvLX = Clip3(-2^{15}, 2^{15} - 14, mvLX < < 2)$ (8-203)

$mvCLX = Clip3(-2^{15}, 2^{15} - 14, mvCLX < < 2)$ (8-204)

【0187】

いくつかの例では、提案した修正は、輝度動きベクトル($mvLX$)のみに適用されてもよい。いくつかの例では、提案した修正は、彩度動きベクトル($mvCLX$)のみに適用されてもよい。いくつかの例では、提案した修正は、輝度動きベクトル($mvLX$)と彩度動きベクトル($mvCLX$)の両方に適用されてもよい。

40

【0188】

いくつかの例では、ビデオデコードは、 $chroma_format_idc$ などのサンプリング比を示す構文要素(たとえば、 $chroma_format_idc$)の値に基づいて、提案した修正を実行してもよい。たとえば、ビデオコードは、提案した修正を4:4:4サンプリング比に対してのみ実行してもよい。

【0189】

図12は、本開示の1つまたは複数の技法による、ビデオデータのブロックのための探索

50

範囲内を決定するための例示的なプロセスを示すフローチャートである。図12の技法は、図1および図3に示すビデオエンコーダ20などの、ビデオエンコーダによって実行されてもよい。ビデオエンコーダ20の構成とは異なる構成を有するビデオエンコーダは、図12の技法を実行してもよいが、例示の目的のため、図12の技法は、図1および図3のビデオエンコーダ20の文脈内で説明される。

【0190】

本開示の1つまたは複数の技法によれば、ビデオエンコーダ20の1つまたは複数のプロセッサは、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルのために使用されることになる解像度(たとえば、整数または分数)を決定してもよい(1202)。たとえば、ビデオエンコーダ20の動き推定ユニット44は、現在のピクチャ内の彩度予測子ブロックを識別する彩度動きベクトルが整数ピクセル解像度または分数ピクセル解像度のどちらを有することになるのかを決定してもよい。いくつかの例では、動き推定ユニット44は、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きい場合、1/4ピクセル解像度などの分数ピクセル解像度が動きベクトルのために使用されることになることを決定してもよい。いくつかの例では、動き推定ユニット44は、現在のブロックのフォーマットが4:2:2または4:2:0である場合、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きいことを決定してもよい。いくつかの例では、動き推定ユニット44は、エントロピー符号化ユニット56に、動きベクトルのために使用されることになる解像度を示す構文要素を符号化させてもよい。たとえば、動き推定ユニット44は、動きベクトルが整数ピクセル解像度または分数ピクセル解像度のどちらを有することになるのかを示すために、エントロピー符号化ユニット56にuse_integer_mv_flagを符号化させてもよい。

【0191】

ビデオエンコーダ20の1つまたは複数のプロセッサは、決定された解像度に基づいて、解像度が整数ピクセルである場合の現在のブロックのための探索範囲のサイズよりも解像度が分数ピクセルである場合に探索範囲のサイズがより小さくなるように、探索範囲を決定してもよい(1204)。たとえば、動き推定ユニット44は、いくつかの例では、現在のピクチャの再構成された範囲を含んでもよい初期探索範囲を決定してもよい。たとえば、動き推定ユニット44は、図8の現在のブロック800のための初期探索範囲802を決定してもよい。動きベクトルのために使用されることになる解像度が整数ピクセルである場合、動き推定ユニット44は、初期探索範囲が現在のブロックのための探索範囲であることを決定してもよい。しかしながら、動きベクトルのために使用されることになる解像度が分数ピクセルである場合、動き推定ユニット44は、少なくとも初期探索範囲の右側境界および下側境界からMサンプルだけ初期探索範囲を縮小することによって、ならびに/または初期探索範囲の上側境界および左側境界からNサンプルだけ初期探索範囲を縮小することによって探索範囲を決定してもよい。たとえば、動き推定ユニット44は、図8の現在のブロック800のための縮小探索範囲804を生成するために、初期探索範囲802を右側境界および下側境界からMサンプル(たとえば、距離808)だけ縮小し、初期探索範囲802を上側境界および左側境界からNサンプル(たとえば、距離806)だけ縮小してもよい。

【0192】

ビデオエンコーダ20の1つまたは複数のプロセッサは、探索範囲内から、現在のブロックのための予測子ブロックを選択してもよい(1206)。一例として、動きベクトルのために使用されることになる解像度が整数ピクセルである場合、動き推定ユニット44は、初期探索範囲(たとえば、図8の初期探索範囲802)から現在のブロックのための予測子ブロックを選択してもよい。別の例として、動きベクトルのために使用されることになる解像度が分数ピクセルである場合、動き推定ユニット44は、縮小探索範囲(たとえば、図8の縮小探索範囲804)から現在のブロックのための予測子ブロックを選択してもよい。上記で説明したように、動き推定ユニット44は、決定された探索範囲からいくつかの候補予測子ブロックを識別し、ピクセル差の点から現在のブロックに密接に一致する候補予測子ブロックを選

折してもよく、これは、絶対差合計(SAD)、自乗差合計(SSD)、または他の差分メトリックによって決定されてもよい。

【0193】

ビデオエンコーダ20の1つまたは複数のプロセッサは、現在のブロックのための選択された予測子ブロックを識別する動きベクトルを決定してもよい(1208)。たとえば、図5の例では、動き推定ユニット44は、現在のブロック102と選択された予測子ブロック104との間の変位を表すベクトル106を決定してもよい。

【0194】

ビデオエンコーダ20の1つまたは複数のプロセッサは、コーディングビデオストリーム内に動きベクトルの表現を符号化してもよい(1210)。たとえば、動き推定ユニット44は、エントロピー符号化ユニット56に、決定された動きベクトルの表現を符号化させてもよい。いくつかの例では、表現は、決定された動きベクトルと、決定された動きベクトルのための予測子との間の差(すなわち、動きベクトル差(MVD))であってもよい。いくつかの例では、動き推定ユニット44は、現在のブロックのための輝度サンプルの予測子ブロックを識別する輝度動きベクトルの表現をエントロピー符号化ユニット56に符号化させてもよく、彩度動きベクトルは、(たとえば、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比に基づいて)輝度動きベクトルから導出可能であってもよい。

【0195】

動きベクトルが分数ピクセル解像度を有する場合などのいくつかの例では、ビデオエンコーダ20は、分数ピクセル補間を使用し、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定してもよい。たとえば、動きベクトルが分数ピクセル解像度を有する彩度動きベクトルである場合、動き補償ユニット46は、予測子ブロックの外部であるが、初期探索範囲内のサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのサンプルピクセル値を構成するために分数ピクセル補間を実行してもよい。

【0196】

ビデオエンコーダ20は、予測子ブロックのピクセル値に基づいて現在のブロックを再構成してもよい。たとえば、ビデオエンコーダ20は、現在のブロックのピクセル値を再構成するために、予測子ブロックのピクセル値を残差値に加算してもよい。

【0197】

図13は、本開示の1つまたは複数の技法による、ビデオデータのブロックを復号するための例示的なプロセスを示すフローチャートである。図13の技法は、図1および図4に示すビデオデコーダ30などの、ビデオデコーダによって実行されてもよい。ビデオデコーダ30の構成とは異なる構成を有するビデオデコーダは、図13の技法を実行してもよいが、例示の目的のため、図13の技法は、図1および図4のビデオデコーダ30の文脈内で説明される。

【0198】

本開示の1つまたは複数の技法によれば、ビデオデコーダ30の1つまたは複数のプロセッサは、コーディングビデオビットストリームから、ビデオデータの現在のピクチャ内の現在のブロックのためのビデオデータの現在のピクチャ内の予測子ブロックを識別する動きベクトルの表現を取得してもよい(1302)。たとえば、ビデオデコーダ30の動き補償ユニット72は、エントロピー復号ユニット70から、動きベクトルの表現を受信してもよい。いくつかの例では、表現は、決定された動きベクトルと、決定された動きベクトルのための予測子との間の差(すなわち、動きベクトル差(MVD))であってもよい。いくつかの例では、動きベクトルは、現在のブロックのための輝度サンプルの予測子ブロックを識別する輝度動きベクトルであってもよく、動き補償ユニット72は、(たとえば、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比に基づいて)輝度動きベクトルから現在のブロックのための彩度サンプルの予測子ブロックを識別する彩度動きベクトルを導出してもよい。

【0199】

ビデオデコーダ30の1つまたは複数のプロセッサは、動きベクトルが分数ピクセル解像

10

20

30

40

50

度または整数ピクセル解像度のどちらであるのかを決定してもよい(1304)。たとえば、動き補償ユニット72は、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きい場合、分数ピクセル解像度が動きベクトルのために使用されることを決定してもよい。いくつかの例では、動き補償ユニット72は、現在のブロックのフォーマットが4:2:2または4:2:0である場合、現在のブロックのための輝度サンプリングに対する彩度サンプリングの比が1よりも大きいことを決定してもよい。いくつかの例では、動き補償ユニット72は、エントロピー復号ユニット70から、動きベクトルのために使用されることになる解像度を示す構文要素を受信してもよい。たとえば、動き補償ユニット72は、動きベクトルが整数ピクセル解像度または分数ピクセル解像度のどちらを有することになるのかを示すuse_integer_mv_flagを受信してもよい。

10

【0200】

ビデオデコーダ30の1つまたは複数のプロセッサは、動きベクトルに基づいて、解像度が整数ピクセルである場合よりも解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲内から現在のブロックのための予測子ブロックを決定してもよい(1306)。たとえば、解像度が整数ピクセルである場合、現在のブロックのための探索範囲は、いくつかの例では、現在のピクチャの再構成された範囲を含み得る初期探索範囲であってもよい。たとえば、解像度が整数ピクセルである場合、現在のブロックのための探索範囲は、図8の現在のブロック800のための初期探索範囲802であってもよい。別の例として、動きベクトルのために使用されることになる解像度が分数ピクセルである場合、現在のブロックのための探索範囲は、初期探索範囲の右側境界および下側境界からMサンプルだけ初期探索範囲を縮小し、初期探索範囲の上側境界および左側境界からNサンプルだけ初期探索範囲を縮小することによって決定された縮小探索範囲であってもよい。たとえば、解像度が整数ピクセルである場合、現在のブロックのための探索範囲は、図8の現在のブロック800のための縮小探索範囲804であってもよい。

20

【0201】

ビデオデコーダ30の1つまたは複数のプロセッサは、動きベクトルが分数ピクセル解像度を有することを決定することに応答して、分数ピクセル補間を使用して、予測子ブロックの外部からのサンプルピクセル値に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定してもよい(1308)。たとえば、動き補償ユニット72は、初期探索範囲の外部であるが、縮小探索範囲内からのサンプルピクセル値(たとえば、初期探索範囲802内に含まれるが、図8の縮小探索範囲804内に含まれないピクセル値)に少なくとも部分的に基づいて、予測子ブロックのピクセル値を決定するために分数ピクセル補間を実行してもよい。

30

【0202】

ビデオデコーダ30の1つまたは複数のプロセッサは、予測子ブロックのピクセル値に基づいて現在のブロックを再構成してもよい(1310)。たとえば、ビデオデコーダ30の加算器80は、現在のブロックのピクセル値を再構成するために、予測子ブロックのピクセル値を残差値に加算してもよい。

【0203】

以下の番号付きの節は、本開示の1つまたは複数の技法を説明し得る。

40

【0204】

節1. ビデオデータを符号化または復号する方法であって、方法が、ビデオデータの現在のピクチャの現在のブロックのための予測子ブロックを識別する動きベクトルの解像度に基づいてイントラブロックコピー(イントラBC)探索範囲を決定するステップを備える、方法。

【0205】

節2. 動きベクトルの解像度が整数ピクセルであるときよりも動きベクトルの解像度が分数ピクセルである場合のほうがより小さいサイズを有する探索範囲を決定するステップをさらに備える、節1の方法。

【0206】

50

節3.動きベクトルの解像度が分数ピクセルであるときに決定される探索範囲のサイズと、動きベクトルの解像度が整数ピクセルであるときに決定される探索範囲との間の差が、予測子ブロック内のサンプルの補間のために使用されるフィルタタップの数に基づく、節1～2の任意の組合せの方法。

【0207】

節4.動きベクトルの解像度が分数ピクセルであるときに決定される探索範囲が、動きベクトルの解像度が整数ピクセルであるときに決定される探索範囲よりも水平方向において小さい、節1～3の任意の組合せの方法。

【0208】

節5.現在のブロックのフォーマットが4:2:2である場合、動きベクトルの解像度が分数ピクセルであるときに決定される探索範囲が、動きベクトルの解像度が整数ピクセルであるときに決定される探索範囲よりも水平方向において小さい、節1～4の任意の組合せの方法。

10

【0209】

節6.動きベクトルの解像度が分数ピクセルであるときに決定される探索範囲が、動きベクトルの解像度が整数ピクセルであるときに決定される探索範囲よりも水平方向および垂直方向において小さい、節1～5の任意の組合せの方法。

【0210】

節7.ビデオデータを符号化または復号する方法であって、方法が、ビデオコードによって、予測子ベクトルおよび差分ベクトルに基づいて、ビデオデータの現在のブロックのための予測子ブロックを識別するベクトルのための計算方法を示す構文要素をコーディングするステップと、ビデオコードによって、示された計算方法を使用してベクトルを決定するステップと、ビデオコードによって、ベクトルを記憶するステップとを備える、方法。

20

【0211】

節8.ベクトルが、ブロックベクトル(BV)または動きベクトル(MV)のいずれかである、節7の方法。

【0212】

節9.構文要素が、動きベクトルが記憶されるべき解像度を示し、動きベクトルを記憶するステップが、示された解像度で動きベクトルを記憶するステップを備える、節7～8の任意の組合せの方法。

30

【0213】

節10.構文要素の第1の値が、動きベクトルが分数ピクセル解像度で記憶されることを示し、構文要素の第2の値が、動きベクトルが整数ピクセル解像度で記憶されることを示す、節7～9の任意の組合せの方法。

【0214】

節11.構文要素が、差分ベクトルがコーディングされる解像度を示し、方法が、示された解像度で差分ベクトルをコーディングするステップをさらに備える、節7～10の任意の組合せの方法。

【0215】

節12.構文要素の第1の値が、差分ベクトルが分数ピクセル解像度でコーディングされることを示し、構文要素の第2の値が、差分ベクトルが整数ピクセル解像度でコーディングされることを示す、節11の方法。

40

【0216】

節13.構文要素が、動きベクトルが動き補償を実行するためにスケーリングされるかどうかを示し、方法が、動きベクトルが動き補償を実行するためにスケーリングされることを示す構文要素にตอบสนองして、スケーリングされた動きベクトルを生成するために動きベクトルをスケーリングするステップと、スケーリングされた動きベクトルを使用して動き補償を実行するステップとをさらに備える、節7～12の任意の組合せの方法。

【0217】

節14.構文要素によって示される計算方法が、現在のブロックがイントラブロックコピ

50

ー(イントラBC)モードを使用してコーディングされている場合、第1の式であり、現在のブロックがイントラモードを使用してコーディングされている場合、第2である、節7～13の任意の組合せの方法。

【0218】

節15. 第1の式が第2の式と同じである、節7～14の任意の組合せの方法。

【0219】

節16. 動きベクトルのための計算方法に加えて、構文要素が、動きベクトルが格納されるべき解像度、動きベクトルが動き補償を実行するためにスケーリングされるかどうか、および差分ベクトルがコーディングされる解像度のうちの1つまたは複数を示す、節7～15の任意の組合せの方法。

10

【0220】

節17. イントラブロックコピー予測に基づいて現在のピクチャ内の現在のビデオブロックを符号化または復号する方法であって、方法が、現在のビデオブロックのための予測子ブロックが現在のピクチャの有効範囲の外部に位置する少なくとも1つのピクセルを含むことを決定することに応答して、有効範囲の外部に位置する少なくとも1つのピクセルを含まないクリップされた予測子ブロックを生成するために予測子ブロックをクリップするステップと、クリップされた予測子ブロックに基づいて現在のビデオブロックを符号化または復号するステップとを備える、方法。

【0221】

節18. 有効範囲の外部に位置する少なくとも1つのピクセルが、現在のピクチャの境界の外部に位置する、節17の方法。

20

【0222】

節19. 有効範囲の外部に位置する少なくとも1つのピクセルが、並列処理サンプル境界の外部に位置する、節17～18の任意の組合せの方法。

【0223】

節20. 有効範囲の外部に位置する少なくとも1つのピクセルが、現在のビデオブロックの内部に位置する、節17～19の任意の組合せの方法。

【0224】

節21. 予測子ブロックが、ブロックベクトルによって識別され、予測子ブロックをクリップするステップが、ブロックベクトルをクリップするステップを備える、節17～20の任意の組合せの方法。

30

【0225】

節22. クリップされた予測子ブロックが、予測子ブロックと同じサイズである、節17～21の任意の組合せの方法。

【0226】

節23. イントラブロックコピー予測に基づいて現在のピクチャ内の現在のビデオブロックを符号化または復号する方法であって、方法が、現在のビデオブロックのための予測子ブロックの利用不可能なピクセルを識別するステップであって、利用不可能なピクセルが現在のピクチャの再構成された範囲の外部に位置する、ステップと、利用不可能なピクセルのための値を取得するためにパディングを実行するステップと、利用不可能なピクセルのための取得された値を含む予測子ブロックのバージョンに基づいて現在のビデオブロックを符号化または復号するステップとを備える、方法。

40

【0227】

節24. パディングを実行するステップが、現在のビデオブロックのピクセル値のビット深度に基づいて利用不可能なピクセルのための値を取得するステップを備える、節の方法。

【0228】

節25. 利用不可能なピクセルのための値が、以下の式、 $P_{\text{unavailable}} = 1 \ll (\text{BitDepth} - 1)$ に従って取得され、ここで、 $P_{\text{unavailable}}$ が、利用不可能なピクセルのための取得された値であり、BitDepthが、現在のビデオブロックのピクセル値のビット深度である、節23～

50

24の任意の組合せの方法。

【0229】

節26. イントラブロックコピー予測に基づいて現在のピクチャ内の現在のビデオブロックを符号化または復号する方法であって、方法が、予測子ブロックが現在のピクチャの有効範囲の外部に位置する少なくとも1つのピクセルを含むことを決定することに応答して、有効範囲の外部に位置する少なくとも1つのピクセルを含まないクリップされた予測子ブロックを生成するために予測子ブロックをクリップするステップと、クリップされた予測子ブロックの利用不可能なピクセルを識別するステップであって、利用不可能なピクセルが現在のピクチャの再構成された範囲の外部に位置する、ステップと、利用不可能なピクセルのための値を取得するためにパディングを実行するステップと、利用不可能なピクセルのための取得された値を含むクリップされた予測子ブロックのバージョンに基づいて現在のビデオブロックを符号化または復号するステップとを備える、方法。

10

【0230】

節27. 利用不可能なピクセルが、現在のビデオブロックを含むスライスのスライス境界の外部に位置する、節26の方法。

【0231】

節28. 利用不可能なピクセルが、現在のビデオブロックを含むタイルのタイル境界の外部に位置する、節26～27の任意の組合せの方法。

【0232】

節29. 利用不可能なピクセルが、まだ復号されていない現在のピクチャのブロックの一部を形成する、節26～28の任意の組合せの方法。

20

【0233】

節30. 有効範囲の外部に位置する少なくとも1つのピクセルが、現在のビデオブロックの内部に位置する、節26～29の任意の組合せの方法。

【0234】

節31. 有効範囲の外部に位置する少なくとも1つのピクセルが、現在のピクチャの境界の外部に位置する、節26～30の任意の組合せの方法。

【0235】

節32. 有効範囲の外部に位置する少なくとも1つのピクセルが、並列処理サンプル境界の外部に位置する、節26～31の任意の組合せの方法。

30

【0236】

節33. 有効範囲の外部に位置する少なくとも1つのピクセルが、現在のビデオブロックの内部に位置する、節26～32の任意の組合せの方法。

【0237】

節34. 予測子ブロックが、ブロックベクトルによって識別され、予測子ブロックをクリップするステップが、ブロックベクトルをクリップするステップを備える、節26～33の任意の組合せの方法。

【0238】

節35. クリップされた予測子ブロックが、予測子ブロックと同じサイズである、節26～34の任意の組合せの方法。

40

【0239】

節36. イントラブロックコピー予測に基づいて現在のピクチャ内の現在のビデオブロックを符号化または復号する方法であって、方法が、現在のビデオブロックのための予測子ブロックを識別するブロックベクトルを、補間を必要としない最も近い値にクリップするステップを備える、方法。

【0240】

節37. イントラブロックコピービデオコーディング技法を使用してビデオデータをコーディングする方法であって、方法が、ビデオデータの現在のブロックのための参照ブロックを識別するためのイントラブロックコピー探索範囲を決定するステップと、ビデオデータの現在のブロックに関連する動きベクトルが彩度補間を使用する場合、イントラブロッ

50

クコピー探索範囲の右側境界および下側境界からMサンプルだけイントラブロックコピー探索範囲を縮小するステップと、ビデオデータの現在のブロックに関連する動きベクトルが彩度補間を使用する場合、イントラブロックコピー探索範囲の上側境界および左側境界からNサンプルだけイントラブロックコピー探索範囲を縮小するステップと、イントラブロックコピー探索範囲を使用してビデオデータの現在のブロックをコーディングするステップとを備える、方法。

【0241】

節38.Mが2であり、Nが2である、節37の方法。

【0242】

節39.Mが2であり、Nが1である、節37の方法。

10

【0243】

節40.Mが4であり、Nが2である、節37の方法。

【0244】

節41.節1～40の任意の組合せを備える方法。

【0245】

本開示の特定の態様は、例示の目的のためにHEVC規格に関して説明されている。しかしながら、本開示に記載の技法は、現在開発中のH.266ビデオコーディング規格などの、まだ開発されていない他の規格または独自のビデオコーディングプロセスを含む、他のビデオコーディングプロセスに有用であり得る。

【0246】

20

ビデオコードは、本開示に記載のように、ビデオエンコードまたはビデオデコードを指してもよい。同様に、ビデオコーディングユニットは、ビデオエンコードまたはビデオデコードを指してもよい。同様に、ビデオコーディングは、該当すれば、ビデオ符号化またはビデオ復号を指してもよい。

【0247】

例に応じて、本明細書で説明される技法のいずれかのいくつかの行為またはイベントは異なる順序で実行されてもよく、一緒に追加され、統合され、または省略されてもよい(たとえば、説明される行為またはイベントのすべてが技法の実施のために必要とは限らない)ことを認識されたい。さらに、いくつかの例では、行為またはイベントは、順次的ではなく、たとえばマルチスレッド処理、割り込み処理またはマルチプロセッサを通じて同時に実行され得る。

30

【0248】

1つまたは複数の例では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せにおいて実装されてもよい。ソフトウェアにおいて実装された場合、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に記憶またはコンピュータ可読媒体を介して送信されてもよく、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もよい。コンピュータ可読媒体は、データ記憶媒体などの有形の媒体に対応するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従うある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含んでもよい。

40

【0249】

このように、コンピュータ可読媒体は、一般に、(1)非一時的である有形のコンピュータ可読記憶媒体、または(2)信号もしくは搬送波などの通信媒体に対応してもよい。データ記憶媒体は、本開示に記載の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得するために、1つもしくは複数のコンピュータ、または1つもしくは複数のプロセッサによってアクセスされ得る任意の利用可能な媒体であってもよい。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。

【0250】

例として、限定はしないが、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁

50

気記憶デバイス、フラッシュメモリ、または、命令もしくはデータ構造の形態で所望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る任意の他の媒体を備えることができる。また、任意の接続が、適切にコンピュータ可読媒体と呼ばれる。たとえば、命令が、同軸ケーブル、光ファイバケーブル、より対線、デジタル加入者線(DSL)、または、赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他の遠隔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、より対線、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は、媒体の定義内に含まれる。

【0251】

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ず、代わりに、非一時的な有形の記憶媒体に向けられていることが理解されるべきである。ディスク(disk)およびディスク(disc)は、本明細書で使用される場合、コンパクトディスク(CD)、レーザディスク、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびBlu-ray(登録商標)ディスクを含み、ディスク(disk)は通常、磁氣的にデータを再生し、ディスク(disc)は、レーザを用いて光学的にデータを再生する。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。

【0252】

命令は、1つもしくは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、または他の同等の集積もしくはディスクリート論理回路などの、1つまたは複数のプロセッサによって実行されてもよい。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、任意の前述の構造、または本明細書で説明する技法の実装に適した任意の他の構造のいずれかを指す場合がある。さらに、いくつかの態様では、本明細書で説明する機能は、符号化および復号のために構成された専用のハードウェアモジュールおよび/またはソフトウェアモジュール内に与えられてもよく、あるいは複合コーデックに組み込まれてもよい。また、技法は、1つまたは複数の回路または論理要素において完全に実装され得る。

【0253】

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てもよい。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、必ずしも異なるハードウェアユニットによる実現を必要とするとは限らない。そうではなくて、上で説明されたように、様々なユニットは、コーデックハードウェアユニットにおいて結合されてよく、または適切なソフトウェアおよび/もしくはファームウェアとともに、前述のような1つもしくは複数のプロセッサを含む、相互動作可能なハードウェアユニットの集合によって提供されてよい。

【0254】

様々な例が説明されている。これらおよび他の例は以下の特許請求の範囲内に入る。

【符号の説明】

【0255】

- 10 ビデオ符号化および復号システム
- 12 ソースデバイス
- 14 宛先デバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 20 ビデオエンコーダ
- 22 出力インターフェース
- 28 入力インターフェース

10

20

30

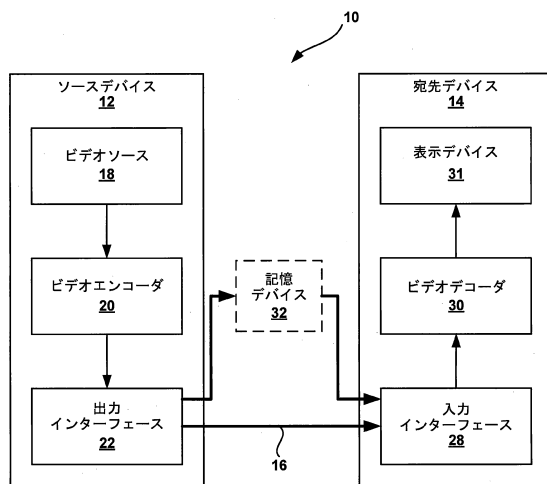
40

50

30	ビデオデコーダ	
31	表示デバイス	
32	記憶デバイス	
33	ビデオシーケンス	
34	ピクチャ	
35A	ピクチャ	
35B	ピクチャ	
35C	ピクチャ	
36A	ピクチャ	
36B	ピクチャ	10
38A	ピクチャ	
38B	ピクチャ	
39	ピクチャ	
40	ビデオデータメモリ	
42	予測処理ユニット	
44	動き推定ユニット	
46	動き補償ユニット	
48	イントラ予測ユニット	
50	加算器	
52	変換処理ユニット	20
54	量子化処理ユニット	
56	エントロピー符号化ユニット	
58	逆量子化処理ユニット	
60	逆変換処理ユニット	
62	加算器	
64	参照ピクチャメモリ	
69	ビデオデータメモリ	
70	エントロピー復号ユニット	
71	予測処理ユニット	
72	動き補償ユニット	30
74	イントラ予測ユニット	
76	逆量子化処理ユニット	
78	逆変換処理ユニット	
80	加算器	
82	参照ピクチャメモリ	
92	再構成領域メモリ	
102	現在のブロック	
104	予測子ブロック	
106	動きベクトル	
108	再構成された領域	40
600	探索範囲	
602、602A、602B	縮小探索範囲	
604	距離	
606	距離	
700	現在のブロック、ブロック	
702	探索範囲、斜線範囲	
704	縮小探索範囲	
706	距離	
800	現在のブロック	
802	初期探索範囲、探索範囲	50

804 縮小探索範囲
806 距離
808 距離

【図 1】



【図 2】

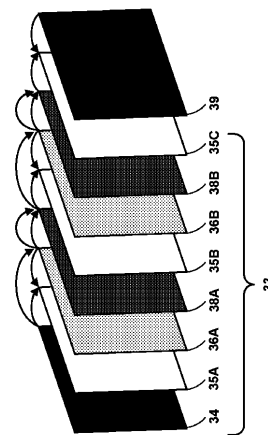
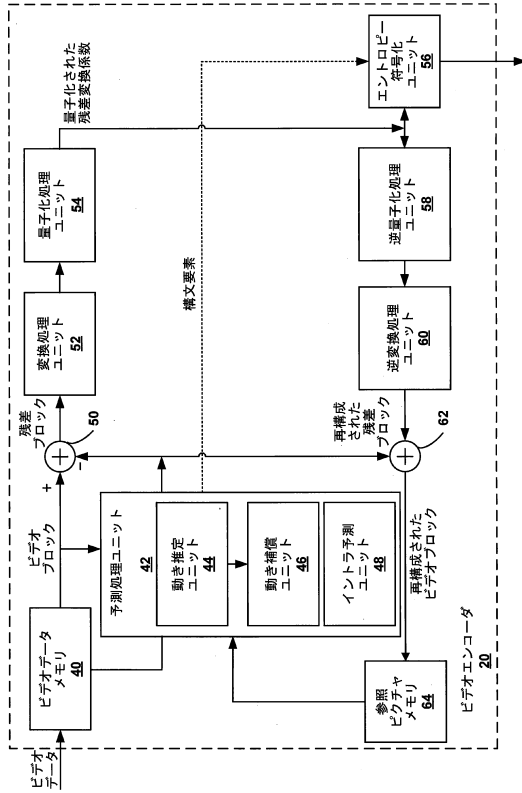
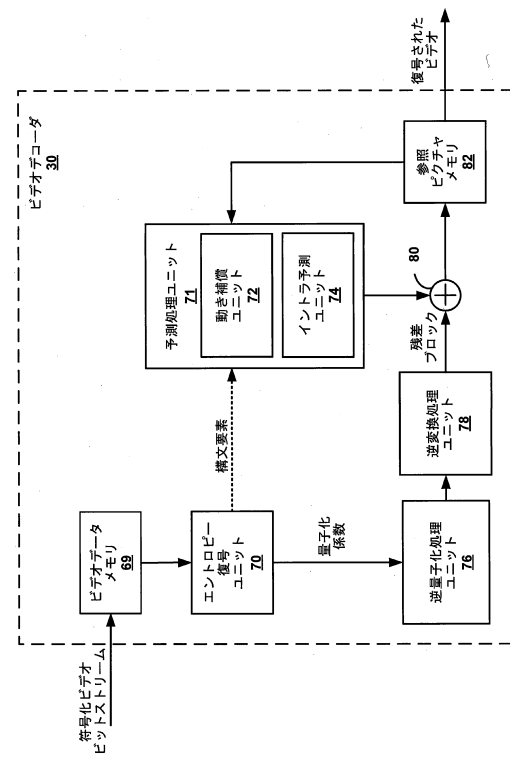


FIG. 2

【 図 3 】



【 図 4 】



【 図 5 】

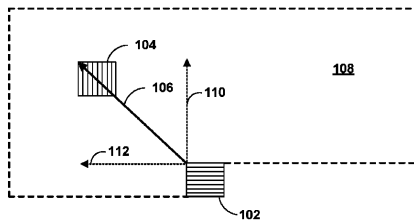


FIG. 5

【 図 6 A 】

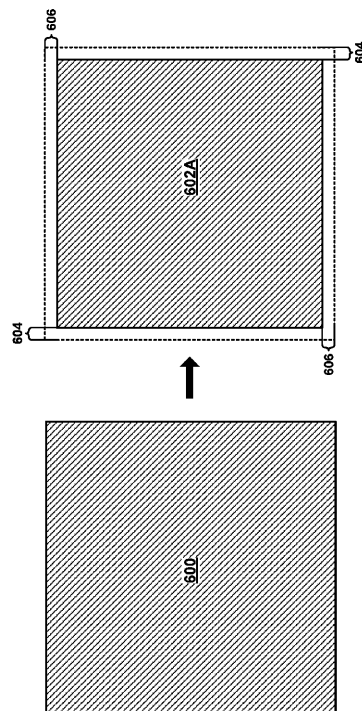
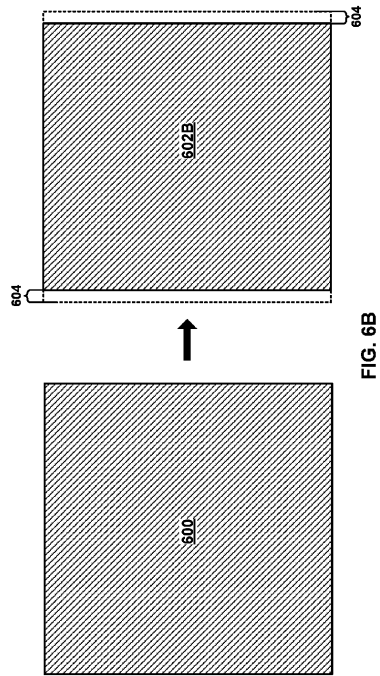
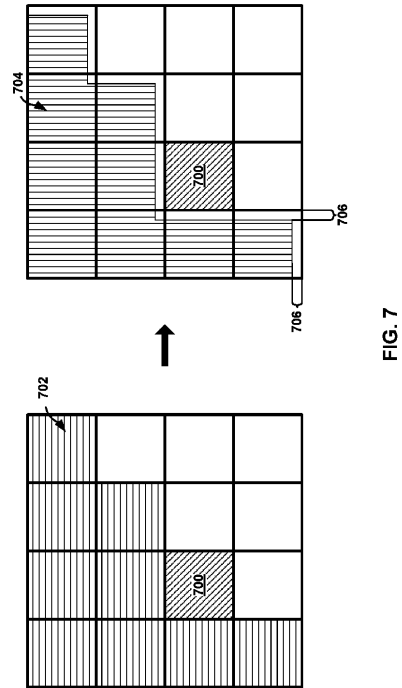


FIG. 6A

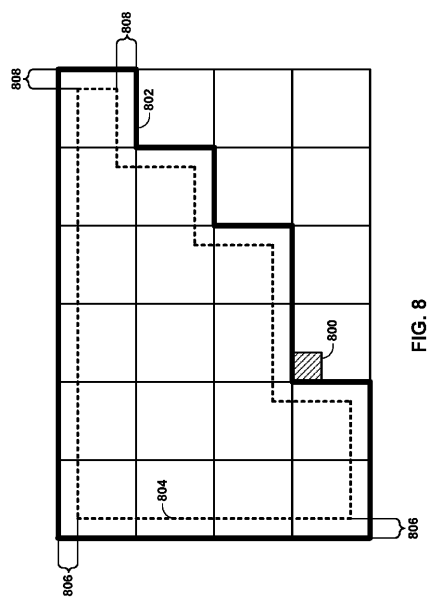
【図 6 B】



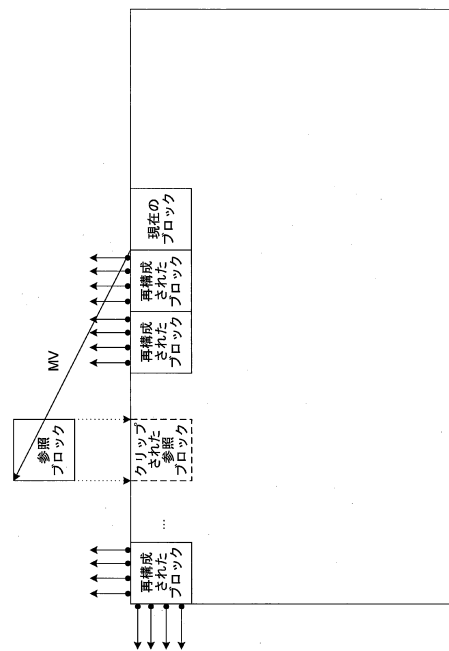
【図 7】



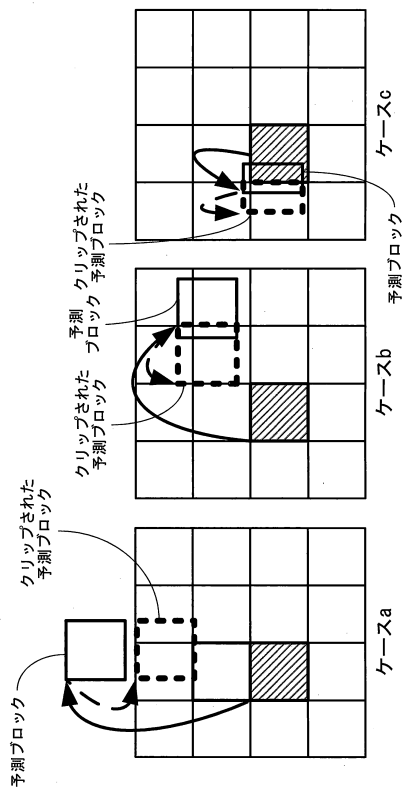
【図 8】



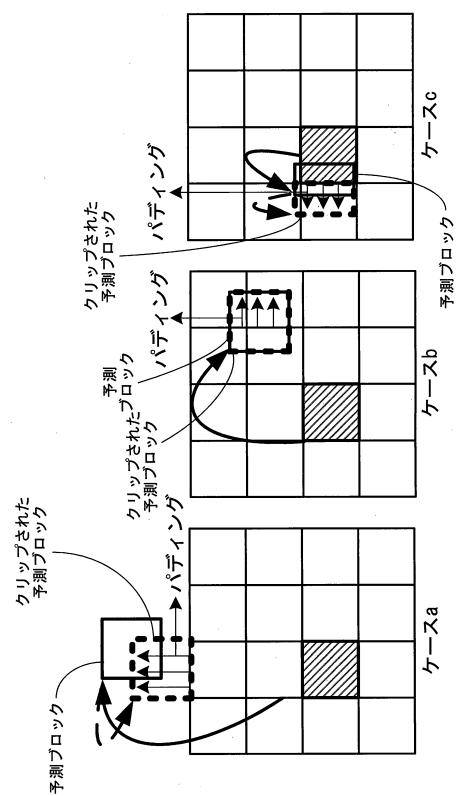
【図 9】



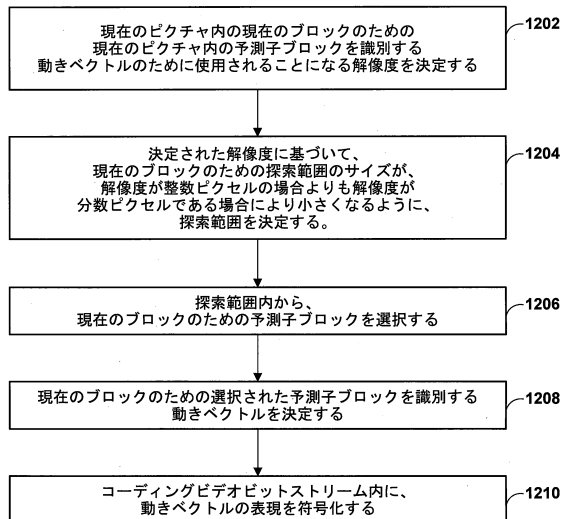
【図 10】



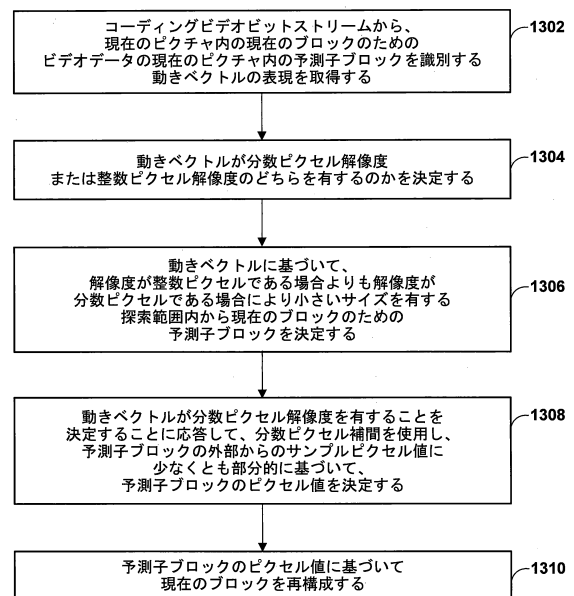
【図 11】



【図 12】



【図 13】



フロントページの続き

(31)優先権主張番号 62/175,179

(32)優先日 平成27年6月12日(2015.6.12)

(33)優先権主張国・地域又は機関
米国(US)

(31)優先権主張番号 15/151,416

(32)優先日 平成28年5月10日(2016.5.10)

(33)優先権主張国・地域又は機関
米国(US)

(72)発明者 クリシュナカンス・ラパカ

アメリカ合衆国・カリフォルニア・9 2 1 2 1 - 1 7 1 4・サン・ディエゴ・モアハウス・ドライヴ・5 7 7 5

(72)発明者 ヴァディム・セレギン

アメリカ合衆国・カリフォルニア・9 2 1 2 1 - 1 7 1 4・サン・ディエゴ・モアハウス・ドライヴ・5 7 7 5

(72)発明者 マータ・カルチェヴィッチ

アメリカ合衆国・カリフォルニア・5 5 1 2 5・サン・ディエゴ・モアハウス・ドライヴ・5 7 7 5

(72)発明者 ラジャン・ラックスマン・ジョシ

アメリカ合衆国・カリフォルニア・9 2 1 2 1 - 1 7 1 4・サン・ディエゴ・モアハウス・ドライヴ・5 7 7 5

審査官 鉢呂 健

(56)参考文献 国際公開第2 0 1 4 / 2 0 5 3 3 9 (WO , A 2)

特開2 0 1 5 - 8 2 8 3 9 (JP , A)

JOSHI, Rajan et al. , High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 2 , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP 3 and ISO/IEC/ J TC1/SC29/WG11 18th Meeting: Sapporo, JP, 30 June-9 July 2014, [JCTVC-S1005] , 2 0 1 4 年1 2 月1 0 日 , JCTVC-S1005 (version 1) , pp. 16-19, 155

RAPAKA, Krishna et al. , On Chroma MV derivation for intra block copy for non-444 , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 21st Meeting: Warsaw, PL, 19-26 June 2015, [JCTVC-U0080] , 2 0 1 5 年 6 月1 0 日 , JCTVC-U0080 (version 1) , pp. 1-2

(58)調査した分野(Int.Cl. , DB名)

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