

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6258288号
(P6258288)

(45) 発行日 平成30年1月17日(2018.1.17)

(24) 登録日 平成29年12月15日(2017.12.15)

(51) Int.Cl.	F 1
HO4N 19/52 (2014.01)	HO4N 19/52
HO4N 19/597 (2014.01)	HO4N 19/597
HO4N 19/39 (2014.01)	HO4N 19/39
HO4N 19/58 (2014.01)	HO4N 19/58
HO4N 19/70 (2014.01)	HO4N 19/70

請求項の数 47 (全 93 頁)

(21) 出願番号	特願2015-500629 (P2015-500629)
(86) (22) 出願日	平成25年3月14日 (2013.3.14)
(65) 公表番号	特表2015-511090 (P2015-511090A)
(43) 公表日	平成27年4月13日 (2015.4.13)
(86) 國際出願番号	PCT/US2013/031573
(87) 國際公開番号	W02013/138639
(87) 國際公開日	平成25年9月19日 (2013.9.19)
審査請求日	平成28年2月18日 (2016.2.18)
(31) 優先権主張番号	61/611,959
(32) 優先日	平成24年3月16日 (2012.3.16)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	61/624,990
(32) 優先日	平成24年4月16日 (2012.4.16)
(33) 優先権主張国	米国(US)

(73) 特許権者	595020643 クアアルコム・インコーポレイテッド QUALCOMM INCORPORATED アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
(74) 代理人	100108855 弁理士 蔵田 昌俊
(74) 代理人	100109830 弁理士 福原 淑弘
(74) 代理人	100103034 弁理士 野河 信久
(74) 代理人	100075672 弁理士 峰 隆司

最終頁に続く

(54) 【発明の名称】高効率ビデオコーディングのためのハイレベルシンタックスの拡張

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを復号する方法であって、
ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を参照する
第2のピクチャのデータを復号することと、
前記第1のピクチャについて第2次元ピクチャ識別子を参照する前記第2のピクチャの
データを復号することと、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを復号することと、

前記第2のピクチャの第1のブロックの第1の動きベクトルが短期参照ピクチャを参照
すると決定することと、

前記第2のピクチャの第2のブロックの第2の動きベクトルが長期参照ピクチャを参照
すると決定することと、

前記第1の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第2の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルと前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることと、

を備え、ここで、前記第1のブロックおよび前記第2のブロックは、前記第2のピクチ

ヤにおいて空間的に隣接するブロックであり、前記第1の動きベクトルと前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることは、前記第1の動きベクトルを予測するために前記第2の動きベクトルを使用することなしに前記第1の動きベクトルを復号することと、前記第2の動きベクトルを予測するために前記第1の動きベクトルを使用することなしに前記第2の動きベクトルを復号することと、を備える、方法。

【請求項2】

前記第2のピクチャを復号することは、

前記P O C値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別することと、

10

前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分を復号することと

、を備える、請求項1に記載の方法。

【請求項3】

前記第1のピクチャを識別することは、前記第2のピクチャのブロックについての動きベクトルの復号の間に前記第1のピクチャを識別することを備え、前記動きベクトルの復号は、発展型動きベクトル予測(A M V P)、時間的動きベクトル予測(T M V P)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルを復号することを備える、請求項2に記載の方法。

【請求項4】

前記ビデオデータの前記第1のピクチャが長期参照ピクチャを備えるかどうかを示す値を復号することをさらに備え、前記第1のピクチャが前記長期参照ピクチャを備えるかどうかを示す前記値がさらに、前記第1のピクチャがビュー間予測に使用されるかどうかを示す、請求項1に記載の方法。

20

【請求項5】

前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記P O C値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを復号することをさらに備える、請求項1に記載の方法。

【請求項6】

前記第2のピクチャを復号する前に、前記第1のピクチャを含む、すべてのビュー間参考ピクチャを、長期参照ピクチャとしてマークすることをさらに備える、請求項5に記載の方法。

30

【請求項7】

前記第2のピクチャについて前記ビュー間参考ピクチャの各々の状態を、前記ビュー間参考ピクチャを長期参照ピクチャとしてマークする前に記憶することと、ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つを備え、前記ビュー間参考ピクチャは、前記第1のピクチャを含み、

前記第2のピクチャを復号した後、前記記憶された状態に基づいて、前記ビュー間参考ピクチャの各々の新たな状態を設定することと、

をさらに備える、請求項6に記載の方法。

40

【請求項8】

前記ベースビデオコーディング仕様は高効率ビデオコーディング(H E V C)ベース仕様を備え、前記ベースビデオコーディング仕様への前記拡張は、前記H E V Cベース仕様に対するスケーラブルビデオコーディング(S V C)拡張と、前記H E V Cベース仕様に対するマルチビュービデオコーディング(M V C)拡張とのうちの1つを備える、請求項5に記載の方法。

【請求項9】

前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディン

50

グ(S V C)レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少なくとも1つを備える、請求項5に記載の方法。

【請求項10】

前記第2のピクチャを復号した後、各ビュー間参照ピクチャを、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つとしてマークすることをさらに備える、請求項5に記載の方法。

【請求項11】

ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たなPOC値を前記ビュー間参照ピクチャに割り当てること、

前記第2のピクチャを復号した後、前記ビュー間参照ピクチャについての元のPOC値をリストアすることと、10

をさらに備える、請求項10に記載の方法。

【請求項12】

前記元のPOC値は、前記第1のピクチャの前記POC値を備える、請求項11に記載の方法。10

【請求項13】

ビデオデータを符号化する方法であって、

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を参照する第2のピクチャのデータを符号化することと、

前記第1のピクチャについて第2次元ピクチャ識別子を参照する前記第2のピクチャのデータを符号化することと、20

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを符号化することと、

前記第2のピクチャの第1のブロックの第1の動きベクトルが短期参照ピクチャを参照すると決定することと、

前記第2のピクチャの第2のブロックの第2の動きベクトルが長期参照ピクチャを参照すると決定することと、

前記第1の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第2の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルと前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることと、30

を備え、ここで、前記第1のブロックおよび前記第2のブロックは、前記第2のピクチャにおいて空間的に隣接するブロックであり、前記第1の動きベクトルと前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることは、前記第1の動きベクトルを予測するために前記第2の動きベクトルを使用することなしに前記第1の動きベクトルを符号化することと、前記第2の動きベクトルを予測するために前記第1の動きベクトルを使用することなしに前記第2の動きベクトルを符号化することと、を備える、

方法。40

【請求項14】

前記POC値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別することと、

前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分をコーディングすることと、

をさらに備える、請求項13に記載の方法。

【請求項15】

前記第1のピクチャを識別することは、前記第2のピクチャのブロックについての前記動きベクトルの符号化の間に前記第1のピクチャを識別することを備え、前記動きベクトルの符号化は、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMV)50

P)、およびマージモードのうちの少なくとも 1 つに従って前記動きベクトルを符号化することを備える、請求項 1 4 に記載の方法。

【請求項 1 6】

前記第 2 のピクチャの第 1 の短期動きベクトルと前記第 2 のピクチャの第 2 の短期動きベクトルとの間の予測をイネーブルにすることと、

前記第 1 の短期動きベクトルによって参照される第 1 の短期参照ピクチャについての POC 値と、前記第 2 の短期動きベクトルによって参照される第 2 の短期参照ピクチャについての POC 値とに基づいて、前記第 1 の短期動きベクトルと前記第 2 の短期動きベクトルとのうちの少なくとも 1 つをスケーリングすることと、

をさらに備える、請求項 1 3 に記載の方法。

10

【請求項 1 7】

前記第 2 次元ピクチャ識別子は、前記第 1 のピクチャを含むビューのビュー識別子、前記第 1 のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第 1 のピクチャを含むスケーラブルビデオコーディング (SVC) レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少なくとも 1 つを備える、請求項 1 3 に記載の方法。

【請求項 1 8】

前記ビデオデータの前記第 1 のピクチャが長期参照ピクチャを備えるかどうかを示す値を符号化することをさらに備え、前記第 1 のピクチャが前記長期参照ピクチャを備えるかどうかを示す前記値がさらに、前記第 1 のピクチャがビュー間予測に使用されるかどうかを示す、請求項 1 3 に記載の方法。

20

【請求項 1 9】

前記ベースビデオコーディング仕様への拡張に従って、前記第 1 のピクチャの前記 POC 値と前記第 2 次元ピクチャ識別子とに少なくとも一部基づいて、前記第 2 のピクチャを符号化することをさらに備える、請求項 1 3 に記載の方法。

【請求項 2 0】

前記第 2 のピクチャを符号化する前に、すべてのビュー間参照ピクチャを、長期参照ピクチャとしてマークすることをさらに備える、請求項 1 9 に記載の方法。

【請求項 2 1】

前記ビュー間参照ピクチャの各々の状態を、前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前に記憶することと、ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの 1 つを備え、

30

前記第 2 のピクチャを符号化した後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々の新たな状態を設定することと、

をさらに備える、請求項 2 0 に記載の方法。

【請求項 2 2】

前記ベースビデオコーディング仕様は高効率ビデオコーディング (HEVC) ベース仕様を備え、前記ベースビデオコーディング仕様への前記拡張は、前記 HEVC ベース仕様に対するスケーラブルビデオコーディング (SVC) 拡張と、前記 HEVC ベース仕様に対するマルチビュービデオコーディング (MVC) 拡張とのうちの 1 つを備える、請求項 1 9 に記載の方法。

40

【請求項 2 3】

前記第 2 のピクチャを符号化した後、各ビュー間参照ピクチャを、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの 1 つとしてマークすることをさらに備える、請求項 1 9 に記載の方法。

【請求項 2 4】

ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たな POC 値を前記ビュー間参照ピクチャに割り当てることと、

前記第 2 のピクチャを符号化した後、前記ビュー間参照ピクチャについての元の POC 値をリストアすることと、

50

をさらに備える、請求項 2 3 に記載の方法。

【請求項 2 5】

前記元の P O C 値は、前記第 2 のピクチャの前記 P O C 値を備える、請求項 2 4 に記載の方法。

【請求項 2 6】

ビデオデータを復号するためのデバイスであって、

ビデオデータを記憶するように構成されたメモリと、

前記ビデオデータの第 1 のピクチャについてピクチャ順序カウント (P O C) 値を参照する第 2 のピクチャのデータを復号し、

前記第 1 のピクチャについて第 2 次元ピクチャ識別子を参照する前記第 2 のピクチャのデータを復号し,
10

ベースビデオコーディング仕様に従って、前記第 1 のピクチャの前記 P O C 値と前記第 2 次元ピクチャ識別子とに少なくとも一部基づいて前記第 2 のピクチャを復号し、

前記第 2 のピクチャの第 1 のブロックの第 1 の動きベクトルが短期参照ピクチャを参照すると決定し、

前記第 2 のピクチャの第 2 のブロックの第 2 の動きベクトルが長期参照ピクチャを参照すると決定し、

前記第 1 の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第 2 の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第 2 のピクチャの前記第 1 のブロックの前記第 1 の動きベクトルと前記第 2 のピクチャの前記第 2 のブロックの前記第 2 の動きベクトルとの間の動きベクトル予測をディセーブルにする

ように構成されたビデオデコーダと、

を備え、ここで、前記第 1 のブロックおよび前記第 2 のブロックは、前記第 2 のピクチャにおいて空間的に隣接するブロックであり、前記第 1 の動きベクトルと前記第 2 の動きベクトルとの間の動きベクトル予測をディセーブルにするために、前記ビデオデコーダは、前記第 1 の動きベクトルを予測するために前記第 2 の動きベクトルを使用することなしに前記第 1 の動きベクトルを復号し、前記第 2 の動きベクトルを予測するために前記第 1 の動きベクトルを使用することなしに前記第 2 の動きベクトルを復号するように構成される、

デバイス。

【請求項 2 7】

前記ビデオデコーダは、前記 P O C 値と前記第 2 次元ピクチャ識別子とを使用して前記第 1 のピクチャを識別し、前記第 1 のピクチャに対して前記第 2 のピクチャの少なくとも一部分を復号するように構成される、請求項 2 6 に記載のデバイス。

【請求項 2 8】

前記ビデオデコーダは、前記第 2 のピクチャのブロックについての動きベクトルの復号の間に前記第 1 のピクチャを識別するように構成され、前記ビデオデコーダは、発展型動きベクトル予測 (A M V P) 、時間的動きベクトル予測 (T M V P) 、およびマージモードのうちの少なくとも 1 つに従って前記動きベクトルを復号するように構成される、請求項 2 7 に記載のデバイス。

【請求項 2 9】

前記ビデオデコーダは、前記第 2 のピクチャの第 1 の短期動きベクトルと前記第 2 のピクチャの第 2 の短期動きベクトルとの間の予測をイネーブルにし、前記第 1 の短期動きベクトルによって参照される第 1 の短期参照ピクチャについての P O C 値と、前記第 2 の短期動きベクトルによって参照される第 2 の短期参照ピクチャについての P O C 値とに基づいて、前記第 1 の短期動きベクトルと前記第 2 の短期動きベクトルとのうちの少なくとも 1 つをスケーリングするように構成される、請求項 2 6 に記載のデバイス。

【請求項 3 0】

前記第 2 次元ピクチャ識別子は、前記第 1 のピクチャを含むビューのビュー識別子、前

10

20

30

40

50

記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング(SVC)レイヤのレイヤ識別子、および一般的なレイヤ識別子の少なくとも1つを備える、請求項26に記載のデバイス。

【請求項31】

前記ビデオデコーダは、前記ビデオデータの前記第1のピクチャが長期参照ピクチャを備えるかどうかを示す値を復号するように構成され、前記第1のピクチャが前記長期参照ピクチャを備えるかどうかを示す前記値がさらに、前記第1のピクチャがビュー間予測に使用されるかどうかを示す、請求項26に記載のデバイス。

【請求項32】

前記ビデオデコーダは、前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて、前記第2のピクチャを復号するようにさらに構成される、請求項26に記載のデバイス。

【請求項33】

前記ビデオデコーダは、

前記第1のピクチャを含む、前記第2のピクチャについてのすべてのビュー間参照ピクチャを、前記第2のピクチャを復号する前に、長期参照ピクチャとしてマークし、

前記ビュー間参照ピクチャの各々の状態を記憶し、ここで前記状態は、前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前の、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つを備え、

前記第2のピクチャを復号した後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々の新たな状態を設定する、

ように構成される、請求項32に記載のデバイス。

【請求項34】

前記ビデオデコーダは、

前記第1のピクチャを含む、前記第2のピクチャについての各ビュー間参照ピクチャを、前記第2のピクチャを復号した後に、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つとしてマークし、

ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たなPOC値を、前記ビュー間参照ピクチャの各々に割り当て、

前記第2のピクチャを復号した後、前記ビュー間参照ピクチャについての元のPOC値をリストアするようにさらに構成される、請求項32に記載のデバイス。

【請求項35】

集積回路と、

マイクロプロセッサと、

前記ビデオデコーダを含むワイヤレス通信デバイスと、

のうちの少なくとも1つを備える、請求項26に記載のデバイス。

【請求項36】

ビデオデータを符号化するためのデバイスであって、

ビデオデータを記憶するように構成されたメモリと、

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を参照する第2のピクチャのデータを符号化し、

前記第1のピクチャについて第2次元ピクチャ識別子を参照する前記第2のピクチャのデータを符号化し、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを符号化し、

前記第2のピクチャの第1のブロックの第1の動きベクトルが短期参照ピクチャを参照すると決定し、

前記第2のピクチャの第2のブロックの第2の動きベクトルが長期参照ピクチャを参

10

20

30

40

50

照すると決定し、

前記第1の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第2の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルと前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにする

ように構成されたビデオエンコーダと、

を備え、ここで、前記第1のブロックおよび前記第2のブロックは、前記第2のピクチャにおいて空間的に隣接するブロックであり、前記第1の動きベクトルと前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにするために、前記ビデオエンコーダは、前記第1の動きベクトルを予測するために前記第2の動きベクトルを使用することなしに前記第1の動きベクトルを符号化し、前記第2の動きベクトルを予測するために前記第1の動きベクトルを使用することなしに前記第2の動きベクトルを符号化するように構成される、

デバイス。

【請求項 3 7】

前記ビデオエンコーダは、前記 POC 値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別し、前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分を符号化するように構成される、請求項 3 6 に記載のデバイス。

【請求項 3 8】

前記ビデオエンコーダは、前記第2のピクチャのブロックについての動きベクトルの符号化の間に前記第1のピクチャを識別するように構成され、前記ビデオエンコーダは、発展型動きベクトル予測 (AMVP)、時間的動きベクトル予測 (TMVP)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルを符号化するように構成される、請求項 3 7 に記載のデバイス。

【請求項 3 9】

前記ビデオエンコーダは、前記第2のピクチャの第1の短期動きベクトルと前記第2のピクチャの第2の短期動きベクトルとの間の予測をイネーブルにし、前記第1の短期動きベクトルによって参照される第1の短期参照ピクチャについての POC 値と、前記第2の短期動きベクトルによって参照される第2の短期参照ピクチャについての POC 値とに基づいて、前記第1の短期動きベクトルと前記第2の短期動きベクトルとのうちの少なくとも1つをスケーリングするように構成される、請求項 3 6 に記載のデバイス。

【請求項 4 0】

前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング (SVC) レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少なくとも1つを備える、請求項 3 6 に記載のデバイス。

【請求項 4 1】

前記ビデオエンコーダは、前記ビデオデータの前記第1のピクチャが長期参照ピクチャを備えるかどうかを示す値を符号化するように構成され、ここで、前記第1のピクチャが前記長期参照ピクチャを備えるかどうかを示す前記値がさらに、前記第1のピクチャがビュー間予測に使用されるかどうかを示す、請求項 3 6 に記載のデバイス。

【請求項 4 2】

前記ビデオエンコーダはさらに、前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記 POC 値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて、第2のピクチャを符号化するように構成される、請求項 3 6 に記載のデバイス。

【請求項 4 3】

前記ビデオエンコーダは、

10

20

30

40

50

前記第2のピクチャを符号化する前に、前記第1のピクチャを含む前記第2のピクチャについてのすべてのビュー間参照ピクチャを長期参照ピクチャとしてマークし、

前記ビュー間参照ピクチャの各々の状態を記憶し、ここで前記状態は、前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前の、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つを備え、

前記第2のピクチャを符号化した後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々について新たな状態を設定するように構成される、請求項42に記載のデバイス。

【請求項44】

前記ビデオエンコーダはさらに、

10

前記第1のピクチャを含む前記第2のピクチャについての各ビュー間参照ピクチャを、前記第2のピクチャを符号化した後に、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つとしてマークし、

ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たなPOC値を前記ビュー間参照ピクチャの各々に割り当てる、

前記第2のピクチャを符号化した後、前記ビュー間参照ピクチャについての元のPOC値をリストアするように構成される、請求項42に記載のデバイス。

【請求項45】

ビデオデータを符号化するためのデバイスであって、

20

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を参照する第2のピクチャのデータを符号化する手段と、

前記第1のピクチャについて第2次元ピクチャ識別子を参照する前記第2のピクチャのデータを符号化する手段と、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを符号化する手段と、

前記第2のピクチャの第1のブロックの第1の動きベクトルが短期参照ピクチャを参照すると決定する手段と、

前記第2のピクチャの第2のブロックの第2の動きベクトルが長期参照ピクチャを参照すると決定する手段と、

30

前記第1の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第2の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルと前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにする手段と、

を備え、ここで、前記第1のブロックおよび前記第2のブロックは、前記第2のピクチャにおいて空間的に隣接するブロックであり、前記第1の動きベクトルと前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにする前記手段は、前記第1の動きベクトルを予測するために前記第2の動きベクトルを使用することなしに前記第1の動きベクトルを符号化する手段と、前記第2の動きベクトルを予測するために前記第1の動きベクトルを使用することなしに前記第2の動きベクトルを符号化する手段と、を備える、

40

デバイス。

【請求項46】

命令を記憶したコンピュータ可読記憶媒体であって、前記命令は、実行されると、プロセッサに、

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を参照する第2のピクチャのデータを復号することと、

前記第1のピクチャについて第2次元ピクチャ識別子を参照する前記第2のピクチャのデータを復号することと、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第

50

2次元ピクチャ識別子とに少なくとも一部基づいて前記第2のピクチャを復号することと

前記第2のピクチャの第1のブロックの第1の動きベクトルが短期参照ピクチャを参照すると決定することと、

前記第2のピクチャの第2のブロックの第2の動きベクトルが長期参照ピクチャを参照すると決定することと、

前記第1の動きベクトルが前記短期参照ピクチャを参照するとの前記決定と、前記第2の動きベクトルが前記長期参照ピクチャを参照するとの前記決定とに基づいて、前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルと前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることと、

を行わせ、ここで、前記第1のブロックおよび前記第2のブロックは、前記第2のピクチャにおいて空間的な隣接するブロックであり、

前記プロセッサに、前記第1の動きベクトルと前記第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることを行わせる前記命令は、前記プロセッサに、前記第1の動きベクトルを予測するために前記第2の動きベクトルを使用することなしに前記第1の動きベクトルを復号することと、前記第2の動きベクトルを予測するために前記第1の動きベクトルを使用することなしに前記第2の動きベクトルを復号することを行わせる命令を備える、

コンピュータ可読記憶媒体。

10

20

【請求項47】

前記第2のピクチャの前記第1のブロックの前記第1の動きベクトルが前記短期参照ピクチャを参照すると決定することは、前記第1の動きベクトルが、短期参照に使用されるとマークされた第1の参照ピクチャを参照すると決定することを備え、

前記第2のピクチャの前記第2のブロックの前記第2の動きベクトルが前記長期参照ピクチャを参照すると決定することは、前記第2の動きベクトルが、長期参照に使用されるとマークされた第2の参照ピクチャを参照すると決定することを備える、

請求項1に記載の方法。

【発明の詳細な説明】

【優先権の主張】

30

【0001】

本出願は、各々の内容全体が参照により本明細書に組み込まれる、2012年3月16日に出願された米国仮出願第61/611,959号、2012年4月16日に出願された米国仮出願第61/624,990号、2012年6月11日に出願された米国仮出願第61/658,344号、2012年6月22日に出願された米国仮出願第61/663,484号、および2012年12月27日に出願された米国仮出願第61/746,476号の利益を主張する。

【技術分野】

【0002】

本開示は、ビデオコーディングに関する。

40

【背景技術】

【0003】

[0003]デジタルビデオ機能は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップまたはデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤー、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」、ビデオ遠隔会議デバイス、ビデオストリーミングデバイスなどを含む、広範囲にわたるデバイスに組み込まれ得る。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4, Part 1

50

0 , Advanced Video Coding (AVC) 、現在開発中の高効率ビデオコーディング (HEVC : High Efficiency Video Coding) 規格によって定義された規格、およびそのような規格の拡張に記載されているビデオコーディング技法のような、ビデオコーディング技法を実装する。ビデオデバイスは、そのようなビデオコーディング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 / または記憶し得る。

【 0004 】

[0004] ビデオコーディング技法は、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の空間的 (イントラピクチャ) 予測および / または時間的 (インターピクチャ) 予測を含む。ブロックベースのビデオコーディングの場合、ビデオスライス (たとえば、ビデオフレームまたはビデオフレームの一部分) が、ツリーブロック、コーディングユニット (CU) および / またはコーディングノードと呼ばれることがあるビデオブロックに区分され得る。ピクチャのイントラコード化 (I) スライス中のビデオブロックは、同じピクチャ中の隣接ブロック中の参照サンプルに対する空間的予測を使用して符号化される。ピクチャのインターフォーム (P または B) スライス中のビデオブロックは、同じピクチャ中の隣接ブロック中の参照サンプルに対する空間的予測、または他の参照ピクチャ中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

【 0005 】

[0005] 空間的予測または時間的予測は、コーディングされるべきブロックの予測ブロックをもたらす。残差データは、コーディングされるべき元のブロックと予測ブロックとの間のピクセル差分を表す。インターフォーム (P または B) スライス中のビデオブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルと、コード化ブロックと予測ブロックとの間の差分を示す残差データとに従って符号化される。イントラコード化ブロックは、イントラコーディングモード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残差変換係数をもたらすことができ、その残差変換係数は、次いで量子化され得る。量子化変換係数は、最初に 2 次元アレイで構成され、変換係数の 1 次元ベクトルを生成するために走査されてよく、なお一層の圧縮を達成するためにエントロピーコーディングが適用されてよい。

【 発明の概要 】

【 0006 】

[0006] 一般に、本開示は、ハイレベルシンタックスの変更しか伴わずに、来たる高効率ビデオコーディング (HEVC) 規格のようなコーディング規格の拡張をサポートするための様々な技法を説明する。たとえば、本開示は、HEVCベース仕様と、ベースビューアがHEVCベース仕様に適合する、マルチビュービデオコーデックおよび / または 3 次元 (3D) ビデオコーデックのHEVC拡張との両方における技法を説明する。一般に、「ベースビデオコーディング仕様」は、2次元の単一レイヤのビデオデータをコーディングするために使用される、HEVCベース仕様のようなビデオコーディング仕様に相当し得る。ベースビデオコーディング仕様に対する拡張は、3Dビデオコーディングおよび / またはマルチレイヤビデオコーディングを可能にするように、ベースビデオコーディング仕様の機能を拡張し得る。HEVCベース仕様は、ベースビデオコーディング仕様の例を代表し、一方、HEVCベース仕様に対するMVCおよびSVC拡張は、ベースビデオコーディング仕様に対する拡張の例を代表する。

【 0007 】

[0007] 一例では、方法は、ビデオデータの第 1 のピクチャに対するピクチャ順序カウント (POC) 値を復号することと、第 1 のピクチャの第 2 次元ピクチャ識別子を復号することと、ベースビデオコーディング仕様に従って、第 1 のピクチャの POC 値および第 2 次元ピクチャ識別子に少なくとも一部に基づいて、第 2 のピクチャを復号することと、を含む。第 2 次元ピクチャ識別子はさらに、ピクチャのタイプ、たとえば、ピクチャが長期ピクチャであるか短期ピクチャであるか、または、ピクチャが参照ピクチャである場合、そ

10

20

30

40

50

のピクチャが、そのピクチャを参照するピクチャと同じピクチャ順序カウント(POC)値を有するかどうか、へと簡略化され得る。隣接ブロックから動きベクトル候補を生成するとき、候補が、予測されることになる動きベクトルとは異なる第2次元ピクチャ識別子を有する場合、その候補は利用不可能であると見なされ得、その第2次元ピクチャ識別子は、この動きベクトルが指し、ターゲット参照インデックスによって特定されるピクチャである。

【 0 0 0 8 】

[0008]別の例では、方法は、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を符号化することと、第1のピクチャの第2次元ピクチャ識別子を符号化することと、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化することとを含む。10

【 0 0 0 9 】

[0009]別の例では、デバイスは、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を復号し、第1のピクチャの第2次元ピクチャ識別子を復号し、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを復号するように構成されるビデオデコーダを含む。

【 0 0 1 0 】

[0010]別の例では、デバイスは、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を符号化し、第1のピクチャの第2次元ピクチャ識別子を符号化し、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化するように構成されるビデオエンコーダを含む。20

【 0 0 1 1 】

[0011]別の例では、デバイスは、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を復号する手段と、第1のピクチャの第2次元ピクチャ識別子を復号する手段と、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを復号する手段とを含む。30

【 0 0 1 2 】

[0012]別の例では、デバイスは、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を符号化する手段と、第1のピクチャの第2次元ピクチャ識別子を符号化する手段と、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化する手段とを含む。

【 0 0 1 3 】

[0013]別の例では、実行されるとプロセッサに、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を復号させ、第1のピクチャの第2次元ピクチャ識別子を復号させ、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを復号させる命令を記憶するコンピュータ可読記憶媒体。40

【 0 0 1 4 】

[0014]別の例では、実行されるとプロセッサに、ビデオデータの第1のピクチャに対するピクチャ順序カウント(POC)値を符号化させ、第1のピクチャの第2次元ピクチャ識別子を符号化させ、ベースビデオコーディング仕様に従って、第1のピクチャの POC 値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化させる命令を記憶するコンピュータ可読記憶媒体。

【 0 0 1 5 】

[0015]1つまたは複数の例の詳細は、添付の図面および以下の説明に記載されている。

50

他の特徴、目的、および利点は、その説明および図面、ならびに特許請求の範囲から明らかになろう。

【図面の簡単な説明】

【0016】

【図1】ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を利用し得る例示的なビデオ符号化および復号システムを示すブロック図。

【図2】ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を実装し得るビデオエンコーダの例を示すブロック図。
10

【図3】ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を実装し得るビデオデコーダの例を示すブロック図。

【図4】例示的なMVC予測パターンを示す概念図。

【図5】ハイレベルシンタックスのみのHEVC拡張を達成するために克服されるべき潜在的な問題を示す概念図。

【図6】ハイレベルシンタックスのみのHEVC拡張を達成するために克服されるべき潜在的な問題を示す概念図。

【図7】ハイレベルシンタックスのみのHEVC拡張を達成するために克服されるべき潜在的な問題を示す概念図。

【図8】ハイレベルシンタックスのみのHEVC拡張を達成するために克服されるべき潜在的な問題を示す概念図。
20

【図9】ハイレベルシンタックスのみのHEVC拡張を達成するために克服されるべき潜在的な問題を示す概念図。

【図10】動きベクトル予測において使用するための現在のブロックに対する隣接ブロックの例示的なセットを示す概念図。

【図11】本開示の技法による、ビデオデータを符号化するための例示的な方法を示すフローチャート。

【図12】本開示の技法による、ビデオデータを復号するための例示的な方法を示すフローチャート。

【発明を実施するための形態】

【0017】

[0024]一般に、本開示は、ハイレベルシンタックス(HLS)の変更しか伴わずに、来たる高効率ビデオコーディング(HEVC)規格のようなコーディング規格の拡張をサポートするための様々な技法を説明する。たとえば、本開示は、HEVCベース仕様と、ベースビューがHEVCベース仕様に適合する、マルチビュービデオコーディング(MVC)および/または3次元ビデオ(3DV)コーディングのHEVC拡張との両方における技法を説明する。
30

【0018】

[0025]本開示は、HEVC拡張仕様においてハイレベルシンタックスのみのプロファイルを可能にするための、いくつかの技法を説明する。MVC/3DVの文脈における「ビューア間」という用語は、スケーラブルビデオコーディング(SVC)の文脈では「レイヤ間」により置き換えられ得る。すなわち、これらの技法の説明は主に「ビューア間」予測を中心とするが、同じまたは同様の考えが、HEVCのHLSのみのSVC拡張に対する「レイヤ間」参照ピクチャに適用され得る。
40

【0019】

[0026]図1は、ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を利用し得る例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示されるように、システム10は、宛先デバイス14によって後で復号されるべき符号化ビデオデータを与えるソースデバイス12を含む。特に、ソースデバイス12は、コンピュータ可読媒体16を介してビデオデー
50

タを宛先デバイス14に与える。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック（すなわち、ラップトップ）コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、ビデオストリーミングデバイスなどを含む、広範囲にわたるデバイスのいずれかを備え得る。場合によっては、ソースデバイス12および宛先デバイス14は、ワイヤレス通信に対応し得る。

【0020】

[0027]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得る。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得る。一例では、コンピュータ可読媒体16は、ソースデバイス12が、符号化ビデオデータを宛先デバイス14にリアルタイムで直接送信することを可能にするための通信媒体を備え得る。符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス14に送信され得る。通信媒体は、高周波（RF）スペクトルあるいは1つまたは複数の物理伝送線路のような、任意のワイヤレスまたは有線通信媒体を備え得る。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークのような、パケットベースネットワークの一部を形成し得る。通信媒体は、ソースデバイス12から宛先デバイス14への通信を可能にするために有用であり得るルータ、スイッチ、基地局、または任意の他の機器を含み得る。

【0021】

[0028]いくつかの例では、符号化データは、出力インターフェース22からストレージデバイスに出力され得る。同様に、符号化データは、入力インターフェースによってストレージデバイスからアクセスされ得る。ストレージデバイスは、ハードドライブ、ブルーレイ（登録商標）ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性または不揮発性メモリ、あるいは、符号化された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のような、種々の分散されたまたはローカルにアクセスされるデータ記憶媒体のいずれかを含み得る。さらなる一例では、ストレージデバイスは、ソースデバイス12によって生成された符号化ビデオを記憶し得るファイルサーバまたは別の中間ストレージデバイスに対応し得る。宛先デバイス14は、ストリーミングまたはダウンロードを介して、ストレージデバイスから記憶されたビデオデータにアクセスし得る。ファイルサーバは、符号化ビデオデータを記憶し、その符号化ビデオデータを宛先デバイス14に送信することが可能な任意のタイプのサーバであり得る。例示的なファイルサーバは、（たとえば、ウェブサイトのための）ウェブサーバ、FTPサーバ、ネットワーク接続ストレージ（NAS）デバイス、またはローカルディスクドライブを含む。宛先デバイス14は、インターネット接続を含む、任意の標準的なデータ接続を通じて符号化ビデオデータにアクセスし得る。これは、ファイルサーバに記憶された符号化ビデオデータにアクセスするのに好適であるワイヤレスチャネル（たとえば、Wi-Fi（登録商標）接続）、有線接続（たとえば、DSL、ケーブルモデムなど）、または両方の組合せを含み得る。ストレージデバイスからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組合せであり得る。

【0022】

[0029]本開示の技法は、必ずしもワイヤレス適用例または設定に限定されると限らない。本技法は、オーバージエアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、dynamic adaptive streaming over HTTP（DASH）などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されたデジタルビデオ、データ記憶媒体に記憶されたデジタルビデオの復号、または他の適用例など、種々のマルチメディア適用例のいずれかをサポートするビデオコーディングに適用され得る。いくつかの例では、システム10は、ビデオストリーミング、

10

20

30

40

50

ビデオ再生、ビデオブロードキャスティング、および／またはビデオ電話などの適用例をサポートするために、一方向または双方向のビデオ送信をサポートするように構成され得る。

【0023】

[0030]図1の例では、ソースデバイス12は、ビデオソース18と、ビデオエンコーダ20と、出力インターフェース22とを含む。宛先デバイス14は、入力インターフェース28と、ビデオデコーダ30と、ディスプレイデバイス32とを含む。本開示によれば、ソースデバイス12のビデオエンコーダ20は、ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディング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コンポーネントまたは構成を含み得る。たとえば、ソースデバイス12は、外部カメラなどの外部ビデオソース18からビデオデータを受信し得る。同様に、宛先デバイス14は、内蔵ディスプレイデバイスを含むのではなく、外部ディスプレイデバイスとインターフェースし得る。10

【0024】

[0031]図1の示されるシステム10は一例にすぎない。ビデオコーディング規格のハイレベルシンタックスのみの拡張によるビデオデータをコーディングするための技法は、任意のデジタルビデオ符号化および／または復号デバイスによって実行され得る。一般に、本開示の技法はビデオ符号化デバイスによって実行されるが、本技法は、通常「コーデック」と呼ばれるビデオエンコーダ／デコーダによっても実行され得る。その上、本開示の技法はまた、ビデオプリプロセッサによって実行され得る。ソースデバイス12および宛先デバイス14は、ソースデバイス12が宛先デバイス14に送信するためのコード化ビデオデータを生成するような、コーディングデバイスの例にすぎない。いくつかの例では、デバイス12、14は、デバイス12、14の各々がビデオ符号化コンポーネントとビデオ復号コンポーネントとを含むように、実質的に対称的に動作し得る。したがって、システム10は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャストまたはビデオ電話のための、ビデオデバイス12とビデオデバイス14との間の一方向または双方向のビデオ送信をサポートすることができる。20

【0025】

[0032]ソースデバイス12のビデオソース18は、ビデオカメラなどのビデオキャプチャデバイス、以前にキャプチャされたビデオを含んでいるビデオアーカイブ、および／またはビデオコンテンツプロバイダからビデオを受信するためのビデオフィードインターフェースを含み得る。さらなる代替として、ビデオソース18は、ソースビデオとしてのコンピュータグラフィックスベースのデータ、またはライブビデオとアーカイブされたビデオとコンピュータにより生成されたビデオとの組合せを生成し得る。場合によっては、ビデオソース18がビデオカメラである場合、ソースデバイス12および宛先デバイス14は、いわゆるカメラ電話またはビデオ電話を形成し得る。しかしながら、上述のように、本開示で説明される技法は、全般にビデオコーディングに適用可能であってよく、ワイヤレスおよび／または有線の適用例に適用可能であってよい。各々の場合において、キャプチャされたビデオ、以前にキャプチャされたビデオ、またはコンピュータで生成されたビデオは、ビデオエンコーダ20によって符号化され得る。符号化ビデオ情報は、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上に出力され得る。3040

【0026】

[0033]コンピュータ可読媒体16は、ワイヤレスブロードキャストまたは有線ネットワーク送信などの一時媒体、あるいはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、ブルーレイディスク、または他のコンピュータ可読媒体などの記憶媒体（すなわち、非一時的記憶媒体）を含み得る。いくつかの例では、ネットワークサーバ（図示せず）は、たとえば、ネットワーク送信を介して、ソースデバイス12から符号化されたビデオデータを受信し、宛先デバイス14に符号化ビデオデータを与え得る。同様に、ディスクスタンピング設備などの媒体製造設備のコンピューティング50

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、その符号化ビデオデータを含んでいるディスクを製造することができる。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ことが理解されよう。

【0027】

[0034]宛先デバイス14の入力インターフェース28は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ビデオエンコーダ20によって定義され、またビデオデコーダ30によって使用される、ロックおよび他のコード化ユニット、たとえば、GOPの特性および/または処理を記述するシンタックス要素を含む、シンタックス情報を含み得る。ディスプレイデバイス32は、復号ビデオデータをユーザに対して表示し、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプのディスプレイデバイスのような、様々なディスプレイデバイスのいずれかを備え得る。10

【0028】

[0035]ビデオエンコーダ20およびビデオデコーダ30は、現在開発中の高効率ビデオコーディング(HEVC)規格などのビデオコーディング規格に従って動作することができ、HEVC Test Model(HM)に準拠することができる。「HEVC Working Draft 7」または「WD7」と呼ばれるHEVCの最近のドラフトは、文書JCTVC-I1003、Brossら、「High Efficiency Video Coding (HEVC) Text Specification Draft 7」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11のJoint Collaborative Team on Video Coding(JCT-VC)、第9回会合:スイス、ジュネーブ、2012年4月27日から2012年5月7日に記載されており、これは2012年6月22日の時点で、http://phenix.itt-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v3.zipからダウンロード可能である。上で述べられたように、本開示は、ハイレベルシンタックスを使用してHEVCを拡張するための技法を説明する。したがって、ビデオエンコーダ20およびビデオデコーダ30は、ハイレベルシンタックスを使用して拡張されるHEVCのバージョンに従って動作し得る。20

【0029】

[0036]代替的に、ビデオエンコーダ20およびビデオデコーダ30は、代替的にMPE G-4、Part 10、Advanced Video Coding(AVC)と呼ばれるITU-T H.264規格のような、他のプロプライエタリ規格または業界規格、あるいはそのような規格の拡張に従って動作し得る。やはり、これらの拡張は、ハイレベルシンタックスを使用して達成され得る。しかしながら、本開示の技法は、いかなる特定のコーディング規格にも限定されない。ビデオコーディング規格の他の例には、MPE G-2およびITU-T H.263がある。図1には示されていないが、いくつかの態様では、ビデオエンコーダ20およびビデオデコーダ30は各々、オーディオエンコーダおよびオーディオデコーダと統合されてよく、適切なMUX-DEMUXユニット、または他のハードウェアとソフトウェアとを含んで、共通のデータストリームまたは別個のデータストリーム中のオーディオとビデオの両方の符号化を処理することができる。適用可能な場合、MUX-DEMUXユニットは、ITU H.223マルチブレクサプロトコル、またはユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠し得る。40

【0030】

[0037]ITU-T H.264/MPEG-4(AVC)規格は、Joint Video Team(JVT)として知られる共同パートナーシップの成果として、ISO/IEC Moving Picture Experts Group(MPEG)とともにITU-T Video Coding Experts Group(VCEG)50

によって策定された。いくつかの態様では、本開示で説明される技法は、一般に H . 2 6 4 規格に準拠するデバイスに適用され得る。H . 2 6 4 規格は、I T U - T 研究グループによる 2 0 0 5 年 3 月付けの I T U - T 勧告 H . 2 6 4 、 Advanced Video Coding for generic audiovisual services に記載されており、本明細書では H . 2 6 4 規格または H . 2 6 4 仕様、あるいは H . 2 6 4 / A V C 規格または仕様と呼ばれ得る。Joint Video Team (J V T) は H . 2 6 4 / M P E G - 4 A V C への拡張に取り組み続けている。

【 0 0 3 1 】

[0038] ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 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 (D S P) 、特定用途向け集積回路 (A S I C) 、フィールドプログラマブルゲートアレイ (F P G A) 、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェアのような、種々の適切なエンコーダ回路のいずれか、またはそれらの任意の組合せとして実装され得る。本技法が部分的にソフトウェアで実装されるとき、デバイスは、好適な非一時的コンピュータ可読媒体にソフトウェアのための命令を記憶し、 1 つまたは複数のプロセッサを使用してその命令をハードウェアで実行して、本開示の技法を実行することができる。ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 の各々は 1 つまたは複数のエンコーダまたはデコーダ中に含まれてよく、そのいずれも、それぞれのデバイスにおいて複合エンコーダ / デコーダ (コーデック) の一部として統合されてよい。

【 0 0 3 2 】

[0039] J C T - V C は、 H E V C 規格の開発に取り組んでいる。 H E V C 規格化の取り組みは、 H E V C Test Model (H M) と呼ばれるビデオコーディングデバイスの発展的モデルに基づく。 H M は、たとえば、 I T U - T H . 2 6 4 / A V C に従う既存のデバイスに対してビデオコーディングデバイスのいくつかの追加の能力を仮定する。たとえば、 H . 2 6 4 は 9 つのイントラ予測符号化モードを提供するが、 H M は 3 3 個ものイントラ予測符号化モードを提供し得る。

【 0 0 3 3 】

[0040] 一般に、 H M の作業モデルは、ビデオフレームまたはピクチャが、ルーマとクロマの両方のサンプルを含む一連のツリープロックまたは最大コーディングユニット (L C U) に分割され得ることを記載する。ビットストリーム内のシンタックスデータが、ピクセルの数に関して最大コーディングユニットである L C U のサイズを定義し得る。スライスは、コーディング順序でいくつかの連続するツリープロックを含む。ビデオフレームまたはピクチャは、 1 つまたは複数のスライスに区分され得る。各ツリープロックは、 4 分木に従ってコーディングユニット (C U) に分割され得る。一般に、 4 分木データ構造は C U ごとに 1 つのノードを含み、ルートノードはツリープロックに対応する。 C U が 4 つのサブ C U に分割された場合、 C U に対応するノードは 4 つのリーフノードを含み、リーフノードの各々はサブ C U のうちの 1 つに対応する。

【 0 0 3 4 】

[0041] 4 分木データ構造の各ノードは、対応する C U のシンタックスデータを与え得る。たとえば、 4 分木のノードは、そのノードに対応する C U がサブ C U に分割されるかどうかを示す分割フラグを含み得る。 C U のシンタックス要素は、再帰的に定義されてよく、 C U がサブ C U に分割されるかどうかに依存し得る。 C U がさらに分割されない場合、その C U はリーフ C U と呼ばれる。本開示では、元のリーフ C U の明示的分割が存在しない場合でも、リーフ C U の 4 つのサブ C U もリーフ C U と呼ばれる。たとえば、 1 6 × 1 6 サイズの C U がこれ以上分割されない場合、この 1 6 × 1 6 C U がまったく分割されなくても、 4 つの 8 × 8 サブ C U もリーフ C U と呼ばれる。

【 0 0 3 5 】

[0042] C U は、 C U がサイズ差異を有さないことを除いて、 H . 2 6 4 規格のマクロブロックと同様の目的を有する。たとえば、ツリープロックは、 4 つの子ノード (サブ C U とも呼ばれる) に分割されてよく、各子ノードは、今度は親ノードとなり、別の 4 つの子

10

20

30

40

50

ノードに分割されてよい。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。コード化ビットストリームに関連するシンタックスデータは、最大CU深さと呼ばれる、ツリーブロックが分割され得る最大回数を定義することができ、また、コーディングノードの最小サイズを定義することもできる。それに応じて、ビットストリームは最小コーディングユニット(SCU)も定義することができる。本開示では、HEVCの文脈におけるCU、PU、またはTU、あるい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)のいずれかを指すために「ブロック」という用語を使用する。

【0036】

10

[0043] CUは、コーディングノードと、コーディングノードに関連する予測ユニット(PU)および変換ユニット(TU)とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状が方形でなければならない。CUのサイズは、 8×8 ピクセルから最大 64×64 以上のピクセルを有するツリーブロックのサイズまでに及び得る。各CUは、1つまたは複数のPUと、1つまたは複数のTUとを含み得る。CUに関連するシンタックスデータは、たとえば、CUを1つまたは複数のPUに区分することを記述し得る。区分モードは、CUが、スキップモード符号化またはダイレクトモード符号化されるか、イントラ予測モード符号化されるか、あるいはインター予測モード符号化されるかによって異なり得る。PUは、形状が非正方形になるように区分され得る。CUに関連するシンタックスデータは、たとえば、4分木に従って、CUを1つまたは複数のTUに区分することも記述し得る。TUは、形状が正方形または非正方形(たとえば、矩形)であり得る。

【0037】

20

[0044] HEVC規格は、CUごとに異なり得るTUに従った変換を可能にする。TUは、一般に、区分されたLCUについて定義された所与のCU内のPUのサイズに基づいてサイズ決定されるが、常にそうであるとは限らない。TUは通常、PUと同じサイズであるかまたはPUよりも小さい。いくつかの例では、CUに対応する残差サンプルは、「残差4分木」(RQT)として知られる4分木構造を使用して、より小さいユニットに再分割され得る。RQTのリーフノードは変換ユニット(TU)と呼ばれることがある。TUに関連するピクセル差分値は、量子化され得る変換係数を生成するために変換され得る。

30

【0038】

[0045] リーフCUは、1つまたは複数の予測ユニット(PU)を含み得る。一般に、PUは、対応するCUのすべてまたは一部分に対応する空間的エリアを表し、そのPUの参照サンプルを取り出すためのデータを含み得る。その上、PUは、予測に関するデータを含む。たとえば、PUがイントラモード符号化されるとき、PUのデータは、PUに対応するTUのイントラ予測モードを記述するデータを含み得る、残差4分木(RQT)中に含まれ得る。別の例として、PUがインターモード符号化されるとき、PUは、PUのための1つまたは複数の動きベクトルを定義するデータを含み得る。PUの動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度(たとえば、 $1/4$ ピクセル精度もしくは $1/8$ ピクセル精度)、動きベクトルが指す参照ピクチャ、および/または動きベクトルの参照ピクチャリスト(たとえば、リスト0、リスト1、もしくはリストC)を記述し得る。

40

【0039】

[0046] 1つまたは複数のPUを有するリーフCUはまた、1つまたは複数の変換ユニット(TU)を含み得る。変換ユニットは、上で論じられたように、(TU4分木構造とも呼ばれる)RQTを使用して指定され得る。たとえば、分割フラグは、リーフCUが4つの変換ユニットに分割されるかどうかを示し得る。次いで、各変換ユニットは、さらに、さらなるサブTUに分割され得る。TUがさらに分割されないとき、そのTUはリーフTUと呼ばれ得る。一般に、イントラコーディングの場合、リーフCUに属するすべてのリーフTUは同じイントラ予測モードを共有する。すなわち、一般に、リーフCUのすべて

50

の T U の予測値を計算するために同じイントラ予測モードが適用される。イントラコーディングの場合、ビデオエンコーダは、イントラ予測モードを使用して各リーフ T U の残差値を、 T U に対応する C U の一部と元のブロックとの間の差分として計算し得る。 T U は、必ずしも P U のサイズに制限されるとは限らない。したがって、 T U は P U よりも大きくまたは小さくなり得る。イントラコーディングの場合、 P U は、同じ C U について対応するリーフ T U と同じ位置にあり得る。いくつかの例では、リーフ T U の最大サイズは、対応するリーフ C U のサイズに対応し得る。

【 0 0 4 0 】

[0047] その上、リーフ C U の T U はまた、残差 4 分木 (R Q T) と呼ばれる、それぞれの 4 分木データ構造と関連付けられ得る。すなわち、リーフ C U は、リーフ C U がどのように T U に区分されるかを示す 4 分木を含み得る。 T U 4 分木のルートノードは一般にリーフ C U に対応し、 C U 4 分木のルートノードは一般にツリープロック (または L C U) に対応する。分割されない R Q T の T U はリーフ T U と呼ばれる。全般に、本開示では、別段明記されない限り、リーフ C U およびリーフ T U に言及するためにそれぞれ C U および T U という用語を使用する。10

【 0 0 4 1 】

[0048] ビデオシーケンスは通常、一連のビデオフレームまたはピクチャを含む。ピクチャグループ (G O P) は、一般に、ビデオピクチャのうちの一連の 1 つまたは複数を備える。 G O P は、 G O P 中に含まれるいくつかのピクチャを記述するシンタックスデータを、 G O P のヘッダ中、ピクチャのうちの 1 つまたは複数のヘッダ中、または他の場所に含み得る。ピクチャの各スライスは、それぞれのスライスの符号化モードを記述するスライスシンタックスデータを含み得る。ビデオエンコーダ 2 0 は通常、ビデオデータを符号化するために個々のビデオスライス内のビデオブロックに対して動作する。ビデオブロックは、 C U 内のコーディングノードに対応し得る。ビデオブロックは、固定のサイズまたは可変のサイズを有してよく、指定されるコーディング規格に応じてサイズが異なり得る。20

【 0 0 4 2 】

[0049] 一例として、 H M は、様々な P U サイズでの予測をサポートする。特定の C U のサイズが $2 N \times 2 N$ であると仮定すると、 H M は、 $2 N \times 2 N$ または $N \times N$ の P U サイズでのイントラ予測をサポートし、 $2 N \times 2 N$ 、 $2 N \times N$ 、 $N \times 2 N$ 、または $N \times N$ の対称的な P U サイズでのインター予測をサポートする。 H M はまた、 $2 N \times n U$ 、 $2 N \times n D$ 、 $n L \times 2 N$ 、および $n R \times 2 N$ の P U サイズでのインター予測のための非対称区分をサポートする。非対称区分では、 C U の一方向は区分されないが、他の方向は 25 % と 75 % とに区分される。 25 % の区分に対応する C U の部分は、「 n 」とその後ろに付く「 U p 」、「 D o w n 」、「 L e f t 」、または「 R i g h t 」という表示によって示される。したがって、たとえば、「 $2 N \times n U$ 」は、上部の $2 N \times 0 . 5 N$ P U と下部の $2 N \times 1 . 5 N$ P U へと水平方向に区分された $2 N \times 2 N$ C U を指す。30

【 0 0 4 3 】

[0050] 本開示では、「 $N \times N$ (NxN) 」および「 $N \times N$ (N by N) 」は、垂直寸法および水平寸法に関するビデオブロックのピクセル寸法、たとえば、 16×16 (16x16) ピクセルまたは 16×16 (16 by 16) ピクセルを指すために互換的に使用され得る。一般に、 16×16 ブロックは、垂直方向に 16 ピクセルを有し ($y = 16$) 、水平方向に 16 ピクセルを有する ($x = 16$) 。同様に、 $N \times N$ ブロックは、一般に、垂直方向に N ピクセルを有し、水平方向に N ピクセルを有し、ただし、 N は非負整数値を表す。ブロック内のピクセルは行と列で構成され得る。さらに、ブロックは、必ずしも、水平方向に垂直方向と同じ数のピクセルを有する必要はない。たとえば、ブロックは $N \times M$ ピクセルを備えてよく、ただし、 M は必ずしも N に等しいとは限らない。40

【 0 0 4 4 】

[0051] C U の P U を使用したイントラ予測コーディングまたはインター予測コーディングの後、ビデオエンコーダ 2 0 は、 C U の T U のための残差データを計算し得る。 P U は、（ピクセル領域とも呼ばれる）空間領域において予測ピクセルデータを生成する方法ま50

たはモードを記述するシンタックスデータを備えてよく、TUは、変換、たとえば、残差ビデオデータへの離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または概念的に同様の変換の適用後の、変換領域における係数を備え得る。残差データは、符号化されていないピクチャのピクセルと、PUに対応する予測値との間のピクセル差分に対応し得る。ビデオエンコーダ20は、CUのための残差データを含むTUを形成し、次いで、TUを変換して、CUの変換係数を生成し得る。

【0045】

[0052]変換係数を生成するための任意の変換の後に、ビデオエンコーダ20は、変換係数の量子化を実行し得る。量子化は、一般に、さらなる圧縮を提供する、係数を表すために使用されるデータの量ができるだけ低減するために変換係数が量子化されるプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連するビット深度を低減することができる。たとえば、量子化中にnビット値がmビット値に切り捨てられてよく、nはmよりも大きい。

【0046】

[0053]量子化の後に、ビデオエンコーダは、変換係数を走査して、量子化変換係数を含む2次元行列から1次元ベクトルを生成し得る。走査は、より高いエネルギー(したがってより低い周波数)の係数をアレイの前方に配置し、より低いエネルギー(したがってより高い周波数)の係数をアレイの後方に配置するように設計され得る。いくつかの例では、ビデオエンコーダ20は、あらかじめ定義された走査順序を利用して、量子化された変換係数を走査し、エントロピー符号化され得る直列化されたベクトルを生成し得る。他の例では、ビデオエンコーダ20は適応走査を実行し得る。量子化変換係数を走査して1次元ベクトルを形成した後に、ビデオエンコーダ20は、たとえば、コンテキスト適応可変長コーディング(CAVLC)、コンテキスト適応バイナリ算術コーディング(CABA C)、シンタックスペースコンテキスト適応バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピー符号化方法に従って、1次元ベクトルをエントロピー符号化し得る。ビデオエンコーダ20はまた、ビデオデータを復号する際にビデオデコーダ30が使用するための符号化ビデオデータに関連するシンタックス要素をエントロピー符号化することができる。

【0047】

[0054]CAVACを実行するために、ビデオエンコーダ20は、送信されるべきシンボルにコンテキストモデル内のコンテキストを割り当てることができる。コンテキストは、たとえば、シンボルの隣接値が0ではないかどうかに関係し得る。CAVLCを実行するために、ビデオエンコーダ20は、送信されるべきシンボルのための可変長コードを選択することができる。VLCにおけるコードワードは、比較的短いコードが優勢シンボルに対応し、より長いコードが劣勢シンボルに対応するように構成され得る。このようにして、VLCを使用すると、たとえば、送信されるべきシンボルごとに等長コードワードを使用するよりも、ビットの節約を実現することができる。確率の決定は、シンボルに割り当てられるコンテキストに基づき得る。

【0048】

[0055]ビデオエンコーダ20はさらに、ブロックベースのシンタックスデータ、フレームベースのシンタックスデータ、およびGOPベースのシンタックスデータなどのシンタックスデータを、たとえば、フレームヘッダ、ブロックヘッダ、ライスヘッダ、またはGOPヘッダ中でビデオデコーダ30に送り得る。GOPシンタックスデータは、それぞれのGOP中のいくつかのフレームを記述することができ、フレームシンタックスデータは、対応するフレームを符号化するために使用される符号化/予測モードを示すことができる。

【0049】

[0056]一般に、本開示は、HEVCのような、ビデオコーディング規格のハイレベルシンタックス(HLS)のみの拡張を可能にするための解決法の様々な例を説明する。たとえば、これらの技法は、MVCまたはSVCのようなHEVCのプロファイルに対するH

10

20

30

40

50

L Sのみの拡張を開発するために使用され得る。様々な例が以下で説明される。様々な例が別々に説明されるが、例のいずれかまたはすべての要素は任意の組合せで組み合わされ得ることを理解されたい。

【0050】

[0057]第1の例では、現在のHEVCベース仕様に対する変更はない。HEVC拡張では、ピクチャ（たとえば、ビュー成分）は、ピクチャ順序カウント（POC）値および第2次元ピクチャ識別子、たとえばview_id値（ピクチャが存在するビューを識別し得る）という、2つの特性によって識別され得る。ビデオエンコーダ20は、長期参照ピクチャとしてビュー間予測のために使用されるべきビュー成分を示すことが要求され得る。

10

【0051】

[0058]第2の例では、現在のHEVCベース仕様に対する変更はない。HEVC拡張では、次の変更が適用され得る。ピクチャ（たとえば、ビュー成分）は、POC値および第2次元ピクチャ識別子、たとえばview_idという2つの特性によって識別され得る。この第2の例では、すべてのビュー間参照ピクチャを長期参照ピクチャとしてマークするために、現在のビュー成分をコーディングする直前に、追加のピクチャマーキングプロセスが導入され得る。各ビュー間参照ピクチャを、現在のビュー成分がコーディングされる前の以前マークされた状態と同一である、長期、短期、または「参照に使用されない」のいずれかとしてマークするために、現在のビュー成分をコーディングした直後に、別のピクチャマーキングプロセスが導入され得る。

20

【0052】

[0059]第3の例では、第2の例の技法が使用され、次のように補足される。第2の例の技法に加えて、各ビュー間参照ピクチャに対して、長期参照ピクチャとしてマークされた後、ピクチャのPOC値が、任意の既存の参照ピクチャのPOC値と等しくない新たなPOC値にマッピングされる。現在のビュー成分を復号した後、各ビュー間参照ピクチャに対して、そのPOC値が、現在のビュー成分に等しい元のPOC値へと再びマッピングされる。たとえば、現在のビュー成分はビュー3に属し（ビュー識別子がビュー順序インデックスに等しいと仮定して）、5に等しいPOC値を有し得る。2つのビュー間参照ピクチャは、自身のPOC値（両方とも5である）を、たとえば1025および2053へと変換させ得る。現在のビュー成分を復号した後、ビュー間ピクチャのPOC値は再び5に変換され得る。

30

【0053】

[0060]第4の例では、第2の例と第3の例のいずれかの技法が使用され、次のように補足され得る。上で言及されたような第1の例または第2の例の技法に加えて、HEVCベース仕様において、特に発展型動きベクトル予測（AMVP：advanced motion vector prediction）の間に、短期ピクチャを参照する任意の動きベクトルと長期ピクチャを参照する別の動きベクトルとの間の予測をディセーブルにするために、追加のフック（hook）が使用され得る。

【0054】

40

[0061]第5の例では、HEVC拡張において、ピクチャは、POC値および第2次元ピクチャ識別情報、たとえばview_idという2つの特性によって識別され得る。HEVCベース仕様において、次のフックの1つまたは複数が（単独でまたは任意の組合せで）追加され得る。一例（例5.1と呼ばれる）では、AMVPおよびマージモードの間に参照ピクチャを識別するとき、第2次元ピクチャ識別情報、たとえば、ビュー順序インデックスが、POCと一緒に使用され得る。HEVCベース仕様における二次元2Dビデオ復号のコンテキストでは、第2次元ピクチャ識別情報は常に0に等しく設定され得る。

【0055】

[0062]別の例（例5.2）では、時間的動きベクトルとビュー間動きベクトルとの間の予測は、AMVP（時間的動きベクトル予測（TMVP）を含む）の間はディセーブルに

50

される。動きベクトルの性質が、参照ピクチャを識別する関連する参照インデックスによって決定され得るかどうか、および、たとえば、長期参照ピクチャ、短期参照ピクチャ、またはビュー間参照ピクチャのような、動きベクトルを含むピクチャによってどのように参照ピクチャが参照されているか。別の例（例 5 . 3）では、時間的短期動きベクトルと時間的長期動きベクトルとの間の予測が（たとえば、明示的にまたは暗黙的に）ディセーブルにされ得る。別の例（例 5 . 4）では、時間的短期動きベクトルと時間的長期動きベクトルとの間の予測が（たとえば、明示的にまたは暗黙的に）イネーブルにされ得る。

【 0 0 5 6 】

[0063]別の例（例 5 . 5）では、2つの異なるビュー間参照ピクチャを参照する動きベクトルの間の予測が（たとえば、明示的にまたは暗黙的に）ディセーブルにされ得る。2つのビュー間参照ピクチャは、それらの第2次元ピクチャ識別子の値が異なる場合、タイプが異なると見なされ得る。別の例（例 5 . 6）では、2つの異なるビュー間参照ピクチャを参照する動きベクトルの間の予測が（たとえば、明示的にまたは暗黙的に）イネーブルにされ得る。別の例（例 5 . 7）では、長期ピクチャを参照する動きベクトルとビュー間を参照する動きベクトルの間の予測が（たとえば、明示的にまたは暗黙的に）イネーブルにされ得る。別の例（例 5 . 8）では、長期ピクチャを参照する動きベクトルとビュー間を参照する動きベクトルの間の予測が（たとえば、明示的にまたは暗黙的に）ディセーブルにされ得る。

【 0 0 5 7 】

[0064]上の例のいずれにおいても、2つの異なる時間的短期参照ピクチャを参照する2つの動きベクトルの間の予測が常にイネーブルにされ、POC値に基づき一方から他方へのスケーリングがイネーブルにされ得る。加えて、または代替的に、上の例のいずれにおいても、2つの異なる長期ピクチャを参照する動きベクトルの間の予測はディセーブルにされ得る。上で説明された様々な例のいくつかの詳細が、以下でより詳細に論じられる。

【 0 0 5 8 】

[0065]一般に、本開示は、参照インデックス（すなわち、参照ピクチャへのポインタ）と、動きベクトル自体のx座標およびy座標とを含むものとして、「動きベクトル」または「動きベクトルデータ」に言及する。視差動きベクトルと時間的動きベクトルの両方が、一般に「動きベクトル」と呼ばれ得る。参照インデックスに対応する参照ピクチャは、動きベクトルが参照する参照ピクチャと呼ばれ得る。動きベクトルが同じビューの中の参照ピクチャを参照する場合、それは時間的動きベクトルと呼ばれる。動きベクトルが異なるビューの参照ピクチャを参照する場合、それは視差動きベクトルと呼ばれる。

【 0 0 5 9 】

[0066]時間的動きベクトルは、短期の時間的動きベクトル（「短期動きベクトル」）または長期の時間的動きベクトル（「長期動きベクトル」）であり得る。たとえば、動きベクトルが短期参照ピクチャを参照する場合、その動きベクトルは短期であり、動きベクトルが長期参照ピクチャを参照する場合、その動きベクトルは長期である。別段言及されない限り、視差動きベクトルおよび長期動きベクトルは一般に、たとえば、それぞれビュー間予測および時間的ビュー内予測のための、異なるカテゴリーの動きベクトルを表す。短期参照ピクチャおよび長期参照ピクチャは、時間的参照ピクチャの例を代表する。

【 0 0 6 0 】

[0067]ビデオエンコーダ20およびビデオデコーダ30は、参照ピクチャメモリとして実装され得る、復号ピクチャバッファ（DPB）から参照ピクチャを識別するように構成され得る。DPBから参照ピクチャを識別するプロセスは、本開示で説明される技法の例のいずれかにおいて使用され得る。DPBから参照ピクチャを識別するプロセスは、HEVC拡張仕様において、参照ピクチャセット構築、参照ピクチャリスト構築、および／または参照ピクチャマーキングの目的で使用され得る。

【 0 0 6 1 】

[0068]ビュー成分、テクスチャビュー成分、深度ビュー成分、またはスケーラブルレイヤ（たとえば、dependency_idとquality_idの固有の組合せを伴

10

20

30

40

50

う)が、ピクチャ順序カウント(POC)値および第2次元ピクチャ識別情報によって識別され得る。第2次元ピクチャ識別情報は、マルチビューのコンテキストにおけるビューID(view_id)、マルチビューのコンテキストにおけるビュー順序インデックス、3DV(深度を伴うマルチビュー)のコンテキストにおけるビュー順序インデックスとdepth_flag(現在のビュー成分がテクスチャか深度かを示す)との組合せ、たとえば、2にdepth_flagの値を足した数により乗算されるビュー順序インデックス、SVCのコンテキストにおけるレイヤID(スケーラブルコーディング環境では、たとえばAVCベースのSVCでは、レイヤIDは、16にquality_idを足した数により乗算されるdependency_idに等しくてよい)、または、一般的なレイヤID(layer_id)、たとえば、HEVCベース仕様において規定されるようなreserved_one_5bitから1を引いた値、のうちの1つまたは複数を含み得る。一般的なレイヤIDは、3DV(深度を伴うマルチビュー)シナリオとスケーラビリティシナリオとの混合に対して適用可能であり得ることに留意されたい。上で言及された例は、たとえば、ビューとして各レイヤを見なすことによって、スケーラブルビデオコーデックを含む、任意の複数レイヤコーデックに当てはまり得る。言い換えると、マルチビュービデオコーディングでは、様々なビューが、別個のレイヤとして見なされ得る。

【0062】

[0069]いくつかのシナリオでは、ベースレイヤまたは従属(dependent)ビューは、たとえば、異なるアップサンプリング/スマージングフィルタの使用が原因で、または、予測のためにビューが合成されたピクチャを使用するという事実が原因で、複数の表現を有することがあるので、1つのビュー位置において、使用の準備ができている2つのピクチャが存在することがあり、ここで一方のピクチャは通常の復元された従属ビューピクチャであり、他方のピクチャは合成されたビューピクチャであり、両方が同じview_idまたはビュー順序インデックスを伴う。この場合、第3次元のピクチャ識別情報が使用され得る。

【0063】

[0070]ビデオエンコーダ20およびビデオデコーダ30はまた、参照ピクチャリストから参照ピクチャを識別するように構成され得る。復号ピクチャバッファ(DPB)は、参照ピクチャリスト、たとえば、現在のピクチャのPOC値よりも小さいPOC値を有する潜在参照ピクチャを含むRefPicList0と、現在のピクチャのPOC値よりも大きいPOC値を有する潜在参照ピクチャを含むRefPicList1とを含む。参照ピクチャリストから参照ピクチャを識別するための技法は、現在のHEVCベース仕様書に対するフックとして使用される。定義された機能は、AMVPおよびマージモードの間に、ビデオエンコーダまたはビデオデコーダによって複数回呼び出され得る。

【0064】

[0071]ビュー成分、テクスチャビュー成分、深度ビュー成分、またはスケーラブルレイヤ(たとえば、dependency_idとquality_idの固有の組合せを伴う)は、POC値および第2次元ピクチャ識別情報によって識別されてよく、第2次元ピクチャ識別情報は、マルチビューと3DVのいずれかのコンテキストにおけるビュー順序インデックスの1つであり得る。関数viewOID(pic)は、「pic」として識別されるピクチャが属するビューのビュー順序インデックスを返す。この関数は、ベースビューの任意のビュー成分、テクスチャビュー成分、または深度ビュー成分に対しては0を返し、ビューID(view_id)を返し、3DVのコンテキストではビュー順序インデックスとdepth_flag(現在のビュー成分がテクスチャか深度かを示す)との組合せ、すなわち、2にdepth_flagの値を足した数により乗算されるビュー順序インデックスを返し、SVCのコンテキストではレイヤID(スケーラブルコーディング環境では、たとえばAVCベースのSVCでは、レイヤIDは、16にquality_idを足した数により乗算されるdependency_idに等しくてよい)を返し、または、一般的なレイヤID(layer_id)、たとえば、HEVCベース仕

10

20

30

40

50

様において規定されるような `reserved_one_5bit` から 1 を引いた値を返す。関数 `layerId(pic)` は、ピクチャ `pic` の `layer_id` を返す。`layerId(pic)` は、ベースビューの任意の（テクスチャ）ビュー成分に対しては 0 を返す。`layerId(pic)` は、SVC ベースレイヤの任意のピクチャ（またはレイヤ表現）に対しては 0 を返す。一般的なレイヤ ID は、3DV（深度を伴うマルチビュー）およびスケーラビリティシナリオの混合に対して適用可能であり得ることに留意されたい。

【0065】

[0072] いくつかの状況では、ベースレイヤまたは従属ビューは、たとえば、異なるアップサンプリング / 平滑フィルタの使用が原因で、または、予測のためにビューが合成されたピクチャを使用するという事実が原因で、複数の表現を有することがあるので、1つのビュー位置において、使用的準備ができている2つのピクチャが存在することがあり、一方のピクチャは通常の復元された従属ビューピクチャであり、他方のピクチャは合成されたビューピクチャであり、両方が同じ `view_id` またはビュー順序インデックスを伴う。この場合、第3次元のピクチャ識別情報が使用され得る。

【0066】

[0073] 上で言及された第2次元および / または第3次元ピクチャ識別情報の1つまたは複数は、関数 `AddPicId(pic)` を使用することによって定義され得る。

【0067】

[0074] ビデオエンコーダ 20 およびビデオデコーダ 30 はまた、参照ピクチャリスト中のエントリーのタイプを識別するように構成され得る。これは、現在の HEVC ベース仕様に対するフックとして使用され得る。以下で定義される関数のいずれかまたはすべてが、AMVP および / またはマージモードの間に、ビデオエンコーダ 20 および / またはビデオデコーダ 30 によって複数回呼び出され得る。以下の例示的な技法のいずれかまたはすべては、参照ピクチャリスト中のエントリーのタイプを識別するために使用され得る。一例では、関数「`RefPicType(pic)`」は、ピクチャ `pic` が時間的参照ピクチャである場合は 0 を返し、ピクチャ `pic` が時間的参照ピクチャではない場合は 1 を返す。別の例では、関数「`RefPicType(pic)`」は、ピクチャ `pic` が現在のピクチャと同じ POC を有する場合は 0 を返し、ピクチャ `pic` が現在のピクチャとは異なる POC を有する場合は 1 を返す。

【0068】

[0075] 別の例では、上で論じられた例の結果は、「`pic`」（関数の引数）の POC が現在のピクチャの POC に等しいかどうかを単に確認することによって、関数 `RefPicType(pic)` の使用を置き換えることで達成され得る。いくつかの例では、ビュー間参照ピクチャは、「参照に使用されない」としてマークされ得る。ビュー間参照ピクチャは、「参照に使用されない」としてマークされ得る。簡単にするために、そのようなピクチャは、HEVC ベース仕様では非参照ピクチャと呼ばれる。いくつかの例では、「長期参照に使用される」と「短期参照に使用される」のいずれかとしてマークされたピクチャは、HEVC ベース仕様において参照ピクチャと呼ばれ得る。いくつかの例では、関数 `RefPicType(pic)` は、ピクチャ `pic` が「長期参照に使用される」または「短期参照に使用される」としてマークされる場合は 0 を返し、ピクチャ `pic` が「参照に使用されない」としてマークされる場合は 1 を返す。加えて、いくつかの例では、HEVC 拡張において、復号の直後のビュー成分は、`nal_ref_flag` シンタクス要素の値とは無関係に、「参照に使用されない」としてマークされ得る。

【0069】

[0076] アクセスユニット全体がコーディングされた後、`nal_ref_flag` が真である場合、アクセスユニットのビュー成分は、「短期参照に使用される」または「長期参照に使用される」としてマークされ得る。あるいは、ビュー成分は、後続のビュー成分の参照ピクチャセット（RPS）が導出された直後に、ビュー成分が後続のビュー成分の RPS に、同じビューの中に復号の順序で含まれる場合にのみ、「短期参照に使用される

10

20

30

40

50

」または「長期参照に使用される」とマークされ得る。加えて、HEVCベース仕様では、現在のピクチャは、復号の直後に、「参照に使用されない」とマークされ得る。

【0070】

[0077]いくつかの例では、RefPicType(picX, refIdx, LX)は、picXが現在のピクチャであった時点でのRefPicType(pic)の値を返し、ここで、picは、ピクチャpicXの参照ピクチャリストLXからのインデックスrefIdxを伴う参照ピクチャである。

【0071】

[0078]「第4の例」として上で呼ばれる例に関して、ビデオエンコーダ20およびビデオデコーダ30は、AMVPおよびTMVPの間の、スケーリングを伴わない長期参照ピクチャの間の予測を可能にするように構成され得る。AMVPに関して、ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル予測子(MVP)の候補の修正された導出プロセスを実行するように構成され得る。このプロセスへの入力は、現在のピクチャの一番左上のサンプルに対する現在の予測ユニットの一番左上のルーマサンプルを規定するルーマ位置(xP, yP)と、ルーマ、nPSW、およびnPSHに対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックスrefIdxLX(Xは0または1である)とを含み得る。このプロセスの出力は、(図10の例で示されるように、NはAとBのいずれかで置き換えられ、Aは左側の隣接する候補に対応し、Bは上側の隣接する候補に対応する)隣接予測ユニットの動きベクトルmvLXNと、隣接予測ユニットの利用可能性フラグavailableFlagLXNとを含み得る。Xが0または1である変数issScalEdFlagLXは、0に等しく設定され得る。

【0072】

[0079]一例では、ビデオエンコーダ20およびビデオデコーダ30は、動きベクトルmvLXAと利用可能性フラグavailableFlagLXAとを次の順序付けられたステップで導出することができ、下線が引かれたテキストはHEVC WD7に対する変更を表す。

【0073】

1. 2つのサンプル位置のセットを()とし、k=0, 1であり、サンプル位置を、xA_k=xP-1, yA_0=yP+nPSH、およびyA_1=yA_0-MinPusizeにより規定する。サンプル位置のセット()は、左側の区分境界とその延長線のすぐ左側のサンプル位置を表す。

【0074】

2. 利用可能性フラグavailableFlagLXAが最初に0に等しく設定されるようにして、mvLXAの両方の成分が0に等しく設定される。

【0075】

3. 次の条件の1つまたは複数が真である場合、この例では、変数issScalEdFlagLXが1に等しく設定される。

【0076】

- ルーマ位置()をカバーする予測ユニットが利用可能であり[Ed.(BB):MinCbAddrZS[][], 最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではない。

【0077】

- ルーマ位置()をカバーする予測ユニットが利用可能であり[Ed.(BB):MinCbAddrZS[], 最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではない。

【0078】

4. yA_1=yA_0-MinPusizeである、(

10

20

30

40

50

)までの(x_{A_k}, y_{A_k})に対して、次のことが、availableFlagLXAが1に等しくなるまで繰り返し適用する。

【0079】

- ルーマ位置(x_{A_k}, y_{A_k})をカバーする予測ユニットが利用可能であり[Ed.(BB):MinCbAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX[x_{A_k}][y_{A_k}]が1に等しく、参照インデックスrefIdXLX[x_{A_k}][y_{A_k}]が現在の予測ユニットの参照インデックスrefIdxXLXに等しい場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLX[x_{A_k}][y_{A_k}]に等しく設定され、refIdxAがrefIdxLX[x_{A_k}][y_{A_k}]に等しく設定され、ListAがListXに等しく設定される。
10

【0080】

- そうではなく、ルーマ位置(x_{A_k}, y_{A_k})をカバーする予測ユニットが利用可能である場合[Ed.(BB):MinCbAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY[x_{A_k}][y_{A_k}](Y!=Xである)が1に等しく、PicOrderCnt(RefPicListY[refIdxLY[x_{A_k}][y_{A_k}]])がPicOrderCnt(RefPicListX[refIdxLX])に等しい場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLY[x_{A_k}][y_{A_k}]に等しく設定され、refIdxAがrefIdxLY[x_{A_k}][y_{A_k}]に等しく設定され、ListAがListYに等しく設定され、mvLXAがmvLXに等しく設定される。
20

【0081】

5. $y_{A_1} = y_{A_0} - \text{MinPusize}$ である、(x_{A_0}, y_{A_0})から(x_{A_1}, y_{A_1})までの(x_{A_k}, y_{A_k})に対して、availableFlagLXAが0に等しい場合、この例では、次のことが、availableFlagLXAが1に等しくなるまで繰り返し適用する。

【0082】

- ルーマ位置(x_{A_k}, y_{A_k})をカバーする予測ユニットが利用可能であり[Ed.(BB):MinCbAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX[x_{A_k}][y_{A_k}]が1に等しく、RefPicListX[refIdxXLX]とRefPicListX[refIdxLX[x_{A_k}][y_{A_k}]]が両方とも長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLX[x_{A_k}][y_{A_k}]に等しく設定され、refIdxAがrefIdxLX[x_{A_k}][y_{A_k}]に等しく設定され、ListAがListXに等しく設定される。
30

【0083】

- そうではなく、ルーマ位置(x_{A_k}, y_{A_k})をカバーする予測ユニットが利用可能であり[Ed.(BB):MinCbAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY[x_{A_k}][y_{A_k}](Y!=Xである)が1に等しく、RefPicListX[refIdxLY[x_{A_k}][y_{A_k}]]とRefPicListY[refIdxLY[x_{A_k}][y_{A_k}]]が両方とも長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLY[x_{A_k}][y_{A_k}]に等しく設定され、refIdxAがrefIdxLY[x_{A_k}][y_{A_k}]に等しく設定され、ListAがListYに等しく設定される。
40

【0084】

- a v a i l a b l e F l a g L X A が 1 に等しく、 R e f P i c L i s t A [r e f I d x A] と R e f P i c L i s t X [r e f I d x L X] の両方が短期参照ピクチャである場合、 m v L X A は以下で規定されるように導出される（ここで、 8 - # # # という表記は、 H E V C の現在のドラフト、すなわち W D 7 のセクションを指す）。

【 0 0 8 5 】

$tx = (16384 + (\text{Abs}(td) >> 1)) / td$ (8-126)

$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6)$ (8-127)

$\text{mvLXA} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA}) + 127) >> 8))$ (8-128)

ここで、 t d および t b は、次のように導出され得る。 10

【 0 0 8 6 】

$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[refIdxA]))$ (8-129)

$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX]))$ (8-130)

[0080] 一例では、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、動きベクトル m v L X B と利用可能性フラグ a v a i l a b l e F l a g L X B とを次の順序付けられたステップで導出するように構成されてよく、下線が引かれたテキストは H E V C W D 7 に対する変更を表す。 20

【 0 0 8 7 】

1 . 3 つのサンプル位置のセットを (x_{B_k}, y_{B_k}) とし、 $k = 0, 1, 2$ であり、 $x_{B_0} = x_P + n_P S W$ 、 $x_{B_1} = x_{B_0} - \text{MinPusize}$ 、 $x_{B_2} = x_P - \text{MinPusize}$ 、 および $y_{B_k} = y_P - 1$ によりサンプル位置を規定する。サンプル位置のセット (x_{B_k}, y_{B_k}) は、上側の区分境界とその延長線のすぐ上側のサンプル位置を表す。 [E d . (B B) : S P S 中で M i n P u S i z e を定義するが、導出は A M P フラグの使用に依存すべきである]

2 . $y_P - 1$ が $((y_C >> \log_2 C t b S i z e) << \log_2 C t b S i z e)$ より小さい場合、次のことが適用する。

【 0 0 8 8 】

$x_{B_0} = (x_{B_0} >> 3) << 3 + ((x_{B_0} >> 3) \& 1) * 7$ (8-131) 30

$x_{B_1} = (x_{B_1} >> 3) << 3 + ((x_{B_1} >> 3) \& 1) * 7$ (8-132)

$x_{B_2} = (x_{B_2} >> 3) << 3 + ((x_{B_2} >> 3) \& 1) * 7$ (8-133)

3 . 利用可能性フラグ a v a i l a b l e F l a g L X B が最初に 0 に等しく設定されるようにして、 m v L X B の両方の成分が 0 に等しく設定される。

【 0 0 8 9 】

4 . $x_{B_0} = x_P + n_P S W$ 、 $x_{B_1} = x_{B_0} - \text{MinPusize}$ 、 および $x_{B_2} = x_P - \text{MinPusize}$ である、 (x_{B_0}, y_{B_0}) から (x_{B_2}, y_{B_2}) までの (x_{B_k}, y_{B_k}) に対して、次のことが、 a v a i l a b l e F l a g L X B が 1 に等しくなるまで繰り返し適用する。

【 0 0 9 0 】

- ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [E d . (B B) : M i n C b A d d r Z S [] []] と最小コーディングブロックに対する利用可能なブロセスとを使用して書き換える]、 P r e d M o d e が M O D E _ I N T R A ではなく、 p r e d F l a g L X [x_{B_k}] [y_{B_k}] が 1 に等しく、 参照インデックス r e f I d x L X [x_{B_k}] [y_{B_k}] が現在の予測ユニットの参照インデックス r e f I d x L X に等しい場合、 a v a i l a b l e F l a g L X B が 1 に等しく設定され、 動きベクトル m v L X B が動きベクトル m v L X [x_{B_k}] [y_{B_k}] に等しく設定され、 r e f I d x B が r e f I d x L X [x_{B_k}] [y_{B_k}] に等しく設定され、 L i s t B が L i s t X に等しく設定される。 40

【 0 0 9 1 】

- そうではなく、ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [Ed . (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY [x_{B_k}] [y_{B_k}] (Y = !Xである) が1に等しく、PicOrderCnt (RefPicListY [refIdxLY [x_{B_k}] [y_{B_k}]]) がPicOrderCnt (RefPicListX [refIdxXL]) に等しい場合、availableFlagLXBが1に等しく設定され、動きベクトルmvLXBが動きベクトルmvLY [x_{B_k}] [y_{B_k}] に等しく設定され、refIdxBがrefIdxLY [x_{B_k}] [y_{B_k}] に等しく設定され、ListBがListYに等しく設定される。

10

【0092】

5. isScaledFlagLXが0に等しく、availableFlagLXBが1に等しい場合、mvLXAはmvLXBに等しく設定され、refIdxAはrefIdxBに等しく設定され、availableFlagLXAは1に等しく設定される。

【0093】

6. isScaledFlagLXが0に等しく、 $x_{B_0} = x_P + n_{PSW}$ 、 $x_{B_1} = x_{B_0} - \text{MinPusize}$ 、および $x_{B_2} = x_P - \text{MinPusize}$ である、(x_{B_0}, y_{B_0})から(x_{B_2}, y_{B_2})までの(x_{B_k}, y_{B_k})に対して、availableFlagLXBが0に等しく設定されている場合、次のことが、availableFlagLXBが1に等しくなるまで繰り返し適用する。

20

【0094】

- ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [Ed . (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX [x_{B_k}] [y_{B_k}] が1に等しく、RefPicListX [refIdxLX] と RefPicListX [refIdxLY [x_{B_k}] [y_{B_k}]] が両方とも長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLXBが1に等しく設定され、動きベクトルmvLXBが動きベクトルmvLX [x_{B_k}] [y_{B_k}] に等しく設定され、refIdxBがrefIdxLY [x_{B_k}] [y_{B_k}] に等しく設定され、ListBがListXに等しく設定される。

30

【0095】

- そうではなく、ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [Ed . (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY [x_{B_k}] [y_{B_k}] (Y = !Xである) が1に等しく、RefPicListX [refIdxLX] と RefPicListY [refIdxLY [x_{B_k}] [y_{B_k}]] が両方とも長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLXBが1に等しく設定され、動きベクトルmvLXBが動きベクトルmvLY [x_{B_k}] [y_{B_k}] に等しく設定され、refIdxBがrefIdxLY [x_{B_k}] [y_{B_k}] に等しく設定され、ListBがListYに等しく設定される。

40

【0096】

- availableFlagLXBが1に等しく、PicOrderCnt (RefPicListB [refIdxB]) がPicOrderCnt (RefPicListX [refIdxLX]) に等しくなく、RefPicListB [refIdxB] と RefPicListX [refIdxLX] の両方が短期参照ピクチャである場合、mvLXBは以下で規定されるように導出され得る(ここで、8 - # ##という表記は、HEVCの現在のドラフトのセクションを指す)。

【0097】

50

```

tx=(16384+(Abs(td)>>1))/td                                (8-134)
DistScaleFactor=Clip3( - 4096,4095,(tb * tx + 32) >>6)      (8-135)
mvLXB=Clip3( - 8192,8191.75,Sign(DistScaleFactor * mvLXA)*
    ((Abs(DistScaleFactor * mvLXA) + 127)>>8))           (8-136)

```

ここで、 t_d および t_b は、次のように導出され得る。

【 0 0 9 8 】

```

td=Clip3( - 128,127,PicOrderCntVal -
    PicOrderCnt(RefPicListB[refIdxB]))                         (8-137)
tb=Clip3( - 128,127,PicOrderCntVal -
    PicOrderCnt(RefPicListX[refIdxLX])) (8-138)

```

10

[0081] ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 はまた、輝度ブロックの動きベクトルをコーディングするための時間的動きベクトル予測 (TMVP) のための、修正された導出プロセスを実行するように構成され得る。一例では、このプロセスへの入力は、現在のピクチャの一一番左上のサンプルに対する現在の予測ユニットの一一番左上のルーマサンプルを規定するルーマ位置 (x_P 、 y_P) と、ルーマ、 n_{PSW} 、および n_{PSH} に対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックス $refIdx_{LX}$ (X は 0 または 1 である) を含む。このプロセスの出力は、動きベクトル予測 $mvLXCol$ と利用可能性フラグ $availableFlag_{LXCol}$ を含み得る。

【 0 0 9 9 】

20

[0082] 一例では、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、ピクチャ pic_X の参照ピクチャリスト LX からのインデックス $refIdx_{LX}$ を伴う参照ピクチャのピクチャ順序カウント $PicOrderCntVal$ を返す、関数 $RefPicOrderCnt(pic_X, refIdx_{LX}, LX)$ を実行し得る。この関数は次のように規定されてよく、(8-141) およびこの説明における同様の参照は、HEVC WD7 のセクションを指す。

【 0 1 0 0 】

```

RefPicOrderCnt( picX, refIdx, LX )
    = PicOrderCnt(RefPicListX[ refIdx ] of the picture picX) (8-141)

```

[0083] $slice_type$ 、 $collocated_from_10_flag$ 、および $collocated_ref_idx$ の値に応じて、コロケート区分を含むピクチャを規定する変数 $colPic$ は、次のように導出され得る。

30

【 0 1 0 1 】

- $slice_type$ が B に等しく、 $collocated_from_10_flag$ が 0 に等しい場合、変数 $colPic$ は、 $RefPicList1[collocated_ref_idx]$ によって規定されるようなコロケート区分を含むピクチャを規定する。

【 0 1 0 2 】

- そうではない場合 ($slice_type$ が B に等しく、 $collocated_from_10_flag$ が 1 に等しく、または $slice_type$ が P に等しい)、変数 $colPic$ は、 $RefPicList0[collocated_ref_idx]$ によって規定されるようなコロケート区分を含むピクチャを規定する。

40

【 0 1 0 3 】

[0084] 変数 $colPu$ およびその位置 ($xPCol$, $yPCol$) は、次の順序付けられたステップを使用して導出され得る。

【 0 1 0 4 】

1. 変数 $colPu$ は次のように導出され得る。

【 0 1 0 5 】

```

yPRb=yP+nPSH (8-139)

```

- ($yP >> \log_2 CtbSize$) が ($yPRb >> \log_2 CtbSize$) に等

50

しい場合、現在の予測ユニットの一一番右下のルーマ位置の水平成分は

$$xPRb=xP+nPSH \quad (8-140)$$

によって定義され、変数 `colPu` は `colPic` の内側の、`((xPRb>>4)<<4, (yPRb>>4)<<4)` によって与えられる修正された位置をカバーする予測ユニットとして設定される。

【0106】

- そうではない場合 (`(yP>>Log2CtbSize)` が `(yPRb>>Log2CtbSize)` と等しくない)、`colPu` は「利用不可能である」とマークされる。

【0107】

2. `colPu` がイントラ予測モードでコーディングされる場合、または `colPu` が「利用不可能」とマークされる場合、次のことが適用する。 10

【0108】

- 現在の予測ユニットの中心のルーマ位置は、

$$xPCtr=(xP+(nPsw>>1)) \quad (8-141)$$

$$yPCtr=(yP+(nPsw>>1)) \quad (8-142)$$

によって定義される。

【0109】

- 変数 `colPu` は、`colPic` の内側の、`((xPCtr>>4)<<4, (yPCtr>>4)<<4)` によって与えられる修正された位置を包含する予測ユニットとして設定される。 20

【0110】

`(xPCol, yPCol)` は、`colPic` の一番左上のルーマサンプルに対する `colPu` の一番左上のルーマサンプルに等しく設定される。

【0111】

[0085] 関数 `LongTermRefPic(picX, refIdx, LX)` は、次のように定義され得る。`picX` が現在のピクチャであった時点で、ピクチャ `picX` の参照ピクチャリスト `LX` からのインデックス `refIdx` を伴う参照ピクチャが「長期参照に使用される」とマークされた場合、`LongTermRefPic(picX, refIdx, LX)` は 1 を返し、それ以外の場合、`LongTermRefPic(pic, refIdx, LX)` は 0 を返す。 30

【0112】

[0086] 変数 `mvlXCol` および `availableFlagLXCol` は次のように導出されてよく、下線が引かれたテキストは H E V C W D 7 に対する変更を表す。

【0113】

- 次の条件の 1 つまたは複数が真である場合、`mvlXCol` の両方の成分は 0 に等しく設定され、`availableFlagLXCol` は 0 に等しく設定される。

【0114】

- `colPu` はイントラ予測モードでコーディングされる。

【0115】

- `colPu` は「利用不可能」とマークされる。 40

【0116】

- `pic_temporal_mv_penable_flag` は 0 に等しい。

【0117】

- そうではない場合、動きベクトル `mvlCol`、参照インデックス `refIdxCol`、および参照リスト識別子 `listCol` は次のように導出される。

【0118】

- `PredFlagL0[xPCol][yPCol]` が 0 に等しい場合、`mvlCol`、`refIdxCol`、および `listCol` はそれぞれ、`Mvl1[xPCol][yPCol]`、`RefIdxL1[xPCol][yPCol]`、および `L1` に等しく設定される。 50

【0119】

- そうではない場合 (`PredFlagL0[xPCol][yPCol]` が 1 に等しい)、次のことが適用する。

【0120】

- `PredFlagL1[xPCol][yPCol]` が 0 に等しい場合、`mvCol`、`refIdxCol`、および `listCol` はそれぞれ、`MvL0[xPCol][yPCol]`、`RefIdxL0[xPCol][yPCol]`、および `L0` に等しく設定される。

【0121】

- そうではない場合 (`PredFlagL1[xPCol][yPCol]` が 1 に等しい)、次の割当てが行われる。 10

【0122】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` 以下である場合、`mvCol`、`refIdxCol`、および `listCol` は、`MvLX[xPCol][yPCol]`、`RefIdxLX[xPCol][yPCol]`、および `LX` に等しく設定され、それ、`X` は、このプロセスが呼び出される `X` の値である。

【0123】

- そうではない場合 (少なくとも 1 つの参照ピクチャリスト中の少なくとも 1 つのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` より大きい)、`mvCol`、`refIdxCol`、および `listCol` は、`MvLN[xPCol][yPCol]`、`RefIdxLN[xPCol][yPCol]`、および `LN` に等しく設定され、それ、`N` は、`collocated_from_l0_flag` の値である。 20

【0124】

- 次の条件の 1 つが真である場合、変数 `availableFlagLXCol` は 0 に等しく設定される。

- `RefPicListX[refIdxLX]` は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は 0 に等しい。 30

- `RefPicListX[refIdxLX]` は短期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は 1 に等しい。

- そうではない場合、変数 `availableFlagLXCol` は 1 に等しく設定され、次のことが適用する。

【0125】

- `RefPicListX[refIdxLX]` が長期参照ピクチャである場合、または `LongTermRefPic(colPic, refIdxCol, listCol)` が 1 に等しい場合、または `PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)` が `PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])` に等しい場合、 40

`mvLXCol = mvCol` (8-143)

である。

【0126】

- そうではない場合、`mvLXCol` は、以下で規定されるような動きベクトル `mvCol` のスケーリングされたバージョンとして導出される。

【0127】

`tx=(16384+Abs(td)>>1))/td` (8-144)

`DistScaleFactor=Clip3(-4096, 4095, (tb * tx+ 32)>> 6)` (8-145)

```
mvLXCol = Clip3( -8192, 8191.75, Sign(DistScaleFactor * mvCol) *
    ((Abs(DistScaleFactor * mvCol)+127)>>8))
```

(8-146)

ここで、`t d` および `t b` は、次のように導出され得る。

【 0 1 2 8 】

```
td=Clip3( -128, 127, PicOrderCnt(colPic) -
    RefPicOrderCnt(colPic, refIdxCol, listCol))
```

(8-147)

```
tb=Clip3( -128, 127, PicOrderCntVal -
    PicOrderCnt(RefPicListX[refIdxLX]))
```

(8-148)

[0087]上の例では、`T M V P` の間に使用されるコロケート (`co-located`) ブロックの利用可能性は、コロケートブロックについての参照ピクチャのピクチャタイプ (たとえば、ピクチャが短期参照ピクチャか長期参照ピクチャか) にも依存し得る。すなわち、(`subclause`におけるステップ 1 の後で) `T M V P` についての一番右下のブロックが利用可能である場合であっても、一番右下のブロックはさらに、ブロック中の動きベクトルが、ターゲット参照ピクチャとは異なるピクチャタイプ (短期または長期) を指す場合、利用不可能となるように設定され得る。同様に、中心ブロックが `T M V P` のためにさらに使用され得る。

【 0 1 2 9 】

[0088]たとえば、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、次の詳細な例に従って、ルーマ動きベクトルのための動きベクトル予測子を導出するように構成され得る。ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 によって実施されるプロセスに対する入力は、

- 現在のピクチャの一番左上のサンプルに対する現在の予測ユニットの一番左上のルーマサンプルを規定するルーマ位置 (`x P`, `y P`) と、
- ルーマのための予測ユニットの幅と高さとを規定する変数、`n P S W` および `n P S H` と、
- 現在の予測ユニットの区分 `r e f I d x L X` (`X` は 0 または 1 である) の参照インデックスと、を含み得る。

【 0 1 3 0 】

[0089]プロセスからの出力は、

- 動きベクトル予測 `m v L X C o l` と、
- 利用可能性フラグ `a v a i l a b l e F l a g L X C o l` と、を含み得る。

【 0 1 3 1 】

[0090]関数 `R e f P i c O r d e r C n t (p i c X , r e f I d x , L X)` は、ビデオエンコーダ 2 0 および / またはビデオデコーダ 3 0 によって実行されると、ピクチャ `p i c X` の参照ピクチャリスト `L X` からのインデックス `r e f I d x` を伴う参照ピクチャのピクチャ順序カウント `P i c O r d e r C n t V a l` を返すことができる。この関数の例示的な実装が、次のように規定される。

【 0 1 3 2 】

```
RefPicOrderCnt( picX, refIdx, LX ) =
    PicOrderCnt(ピクチャpicXのRefPicListX[refIdx])
```

(8-141)

[0091] `s l i c e _ t y p e`、`c o l l o c a t e d _ f r o m _ l o _ f l a g`、および `c o l l o c a t e d _ r e f _ i d x` の値に応じて、コロケート (`collocated`) 区分を含むピクチャを規定する変数 `c o l P i c` は、次のように導出され得る。

【 0 1 3 3 】

- `s l i c e _ t y p e` が `B` に等しく、`c o l l o c a t e d _ f r o m _ l o _ f l a g` が 0 に等しい場合、変数 `c o l P i c` は、`R e f P i c L i s t 1 [c o l l o c a t e d _ r e f _ i d x]` によって規定されるようなコロケート区分を含むピクチャを規定する。

【 0 1 3 4 】

10

20

30

40

50

- そうではない場合 (slice_type が B に等しく、 collocated_from_10_flag が 1 に等しく、または slice_type が P に等しい) 、変数 colPic は、 RefPicList0 [collocated_ref_idx] によって規定されるようなコロケート区分を含むピクチャを規定する。

【0135】

[0092] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次の順序付けられたステップを使用して、変数 colPu とその位置 (x_PCol, y_PCol) とを導出することができる。

【0136】

1. ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のように変数 colPu を導出することができる。 10

【0137】

$$yPRb = yP + nPSH \quad (8-139)$$

- (yP >> Log2CtbSize) が (yPRb >> Log2CtbSize) に等しい場合、現在の予測ユニットの一番右下のルーマ位置の水平成分は

$$xPRb = xP + nPSW \quad (8-140)$$

によって定義され得、変数 colPu は colPic の内側の、 ((xPRb >> 4) < < 4 , (yPRb >> 4) < < 4) によって与えられる修正された位置をカバーする予測ユニットとして設定され得る。

【0138】

- そうではない場合 ((yP >> Log2CtbSize) が (yPRb >> Log2CtbSize) に等しくない) 、ビデオエンコーダ 20 およびビデオデコーダ 30 は、 colPu を「利用不可能」とマークすることができる。 20

【0139】

2. colPu がイントラ予測モードでコーディングされる場合、または colPu が「利用不可能」とマークされる場合、この例では、次のことが適応する。

【0140】

- 現在の予測ユニットの中心のルーマ位置は、

$$xPCtr = (xP + (nPSW >> 1) \quad (8-141)$$

$$yPCtr = (yP + (nPSH >> 1) \quad (8-142)$$

によって定義される。 30

【0141】

- 変数 colPu は、 colPic の内側の、 ((xPCtr >> 4) < < 4 , (yPCtr >> 4) < < 4) によって与えられる修正された位置をカバーする予測ユニットとして設定される。

【0142】

3. ビデオエンコーダ 20 およびビデオデコーダ 30 は、 (x_PCol, y_PCol) を、 colPic の一番左上のルーマサンプルに対する colPu の一番左上のルーマサンプルと等しく設定し得る。

【0143】

[0093] 関数 LongTermRefPic (picX, refIdx, LX) は、次のように定義され得る。 picX が現在のピクチャであった時点で、ピクチャ picX の参照ピクチャリスト LX からのインデックス refIdx を伴う参照ピクチャが「長期参照に使用される」とマークされた場合、 LongTermRefPic (picX, refIdx, LX) は 1 を返し、それ以外の場合、 LongTermRefPic (picX, refIdx, LX) は 0 を返す。 40

【0144】

[0094] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のように変数 mv_LX_Col と availableFlag_LX_Col を導出し得る。

【0145】

10

20

30

40

50

`a v a i l a b l e F l a g L X C o l` は 0 に設定され、`n u m T e s t B l o c k` は 0 に等しい。

【 0 1 4 6 】

`n u m T e s t B l o c k` が 2 未満であり `a v a i l a b l e F l a g L X C o l` が 0 に等しい間は、次のことが順番に実行される。

【 0 1 4 7 】

`xP C t r = (x P + (n P S W >> 1)`

`yP C t r = (y P + (n P S H >> 1)`

- `c o l P u` が `c o l P i c` の内側の $((x P C t r >> 4) << 4, (y P C t r >> 4) << 4)$ によって与えられる位置をカバーする場合、`n u m T e s t B l o c k` は 1 に設定される。 10

【 0 1 4 8 】

- そうではなく、`n u m T e s t B l o c k` が 1 に等しいとき、`c o l P u` は、`c o l P i c` の内側の $((x P C t r >> 4) << 4, (y P C t r >> 4) << 4)$ によって与えられる修正された位置をカバーする予測ユニットとして設定され、`(x P C o l, y P C o l)` は、`c o l P i c` の一番左上のルーマサンプルに対する `c o l P u` の一番左上のルーマサンプルに等しく設定される。

【 0 1 4 9 】

- `n u m T e s t B l o c k ++`

- 次の条件の 1つまたは複数が真である場合、`m v L X C o l` の両方の成分は 0 に等しく設定され、`a v a i l a b l e F l a g L X C o l` は 0 に等しく設定される。 20

【 0 1 5 0 】

- `c o l P u` はイントラ予測モードでコーディングされる。

【 0 1 5 1 】

- `c o l P u` は「利用不可能」とマークされる。

【 0 1 5 2 】

- `p i c _ t e m p o r a l _ m v p _ e n a b l e _ f l a g` は 0 に等しい。

【 0 1 5 3 】

- そうではない場合、動きベクトル `m v C o l`、参照インデックス `r e f I d x C o l`、および参照リスト識別子 `l i s t C o l` は次のように導出される。 30

【 0 1 5 4 】

- `P r e d F l a g L 0 [x P C o l] [y P C o l]` が 0 に等しい場合、`m v C o l`、`r e f I d x C o l`、および `l i s t C o l` はそれぞれ、`M v L 1 [x P C o l] [y P C o l]`、`R e f I d x L 1 [x P C o l] [y P C o l]`、および `L 1` に等しく設定される。

【 0 1 5 5 】

- そうではない場合 (`P r e d F l a g L 0 [x P C o l] [y P C o l]` が 1 に等しい)、次のことが適用する。

【 0 1 5 6 】

- `P r e d F l a g L 1 [x P C o l] [y P C o l]` が 0 に等しい場合、`m v C o l`、`r e f I d x C o l`、および `l i s t C o l` はそれぞれ、`M v L 0 [x P C o l] [y P C o l]`、`R e f I d x L 0 [x P C o l] [y P C o l]`、および `L 0` に等しく設定される。 40

【 0 1 5 7 】

- そうではない場合 (`P r e d F l a g L 1 [x P C o l] [y P C o l]` が 1 に等しい)、次の割当てが行われる。

【 0 1 5 8 】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ `p i c` の `P i c O r d e r C n t (p i c)` が `P i c O r d e r C n t V a l` 以下である場合、`m v C o l`、`r e f I d x C o l`、および `l i s t C o l` は、`M v L X [x P C o l] [y P C o l]`、 50

`RefIdxLX[xPicCol][yPicCol]`、および`LX`に等しく設定され、それぞれ、`X`は、このプロセスが呼び出される`X`の値である。

【0159】

- そうではない場合（少なくとも1つの参照ピクチャリスト中の少なくとも1つのピクチャ`pic`の`PicOrderCnt(pic)`が`PicOrderCntVal`より大きい）、`mvCol`、`refIdxCol`、および`listCol`は、`MvLN[xPicCol][yPicCol]`、`RefIdxLN[xPicCol][yPicCol]`、および`LN`に等しく設定され、それぞれ、`N`は、`collocated_from_10_flag`の値である。

【0160】

- 次の条件の1つが真である場合、変数`availableFlagLXCol`は0に等しく設定される。

【0161】

- `RefPicListX[refIdxLX]`は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)`は0に等しい。

【0162】

- `RefPicListX[refIdxLX]`は短期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)`は1に等しい。

【0163】

- そうではない場合、変数`availableFlagLXCol`は1に等しく設定され、次のことが適用する。

【0164】

- `RefPicListX[refIdxLX]`が長期参照ピクチャである場合、または`LongTermRefPic(colPic, refIdxCol, listCol)`が1に等しい場合、または`PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)`が`PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])`に等しい場合、

$$mvLXCol = mvCol \quad (8-143)$$

- そうではない場合、`mvLXCol`は、以下で規定されるような動きベクトル`mvCol`のスケーリングされたバージョンとして導出される。

【0165】

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-144)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-145)$$

$$mvLXCol = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvCol) * ((\text{Abs}(\text{DistScaleFactor} * mvCol) + 127) >> 8)) \quad (8-146)$$

)

ここで、`td`および`tb`は、次のように導出され得る。

【0166】

$$td = \text{Clip3}(-128, 127, \text{PicOrderCnt}(colPic) - \text{RefPicOrderCnt}(colPic, refIdxCol, listCol)) \quad (8-147)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX])) \quad (8-148)$$

[0095]代替的な例では、長期動きベクトルは、参照ピクチャの`POC`値が同じではない場合、別の長期動きベクトルからは決して予測されない。ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル予測子の候補を導出するための次のプロセスに従って

10

20

30

40

50

構成されてよく、下線が引かれたテキストはHEVC WD7に対する変更を表す。

【0167】

[0096] Xが0または1である変数`isScaledFlagLX`は0に等しく設定され得る。動きベクトル`mvLXA`および利用可能性フラグ`availableFlagLXA`は次の順序付けられたステップにおいて導出され、省略部分は、HEVCの現在のドラフトと同じテキストを表し、下線は、HEVCの現在のドラフトに対する変更を表す。

【0168】

1. . . .
2. . . .
3. . . .
4. . . .

5. $yA_1 = yA_0 - \text{MinPusize}$ である、 (xA_0, yA_0) から (xA_1, yA_1) までの (xA_k, yA_k) に対して、`availableFlagLXA`が0に等しい場合、次のことが、`availableFlagLXA`が1に等しくなるまで繰り返し適用する。

【0169】

- ルーマ位置 (xA_k, yA_k) をカバーする予測ユニットが利用可能であり[Ed.(BB) : `MinCbaaddrZS` [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode`が`MODE_INTRA`ではなく、`predFlagLX[xA_k][yA_k]`が1に等しく、`RefPicListX[refIdxLX]`と`RefPicListX[refIdxLX[xA_k][yA_k]]`が両方とも異なるPOC値を伴う長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、`availableFlagLXA`が1に等しく設定され、動きベクトル`mvLXA`が動きベクトル`mvLX[xA_k][yA_k]`に等しく設定され、`refIdxA`が`refIdxLX[xA_k][yA_k]`に等しく設定され、`ListA`が`ListX`に等しく設定される。

【0170】

- そうではなく、ルーマ位置 (xA_k, yA_k) をカバーする予測ユニットが利用可能であり[Ed.(BB) : `MinCbaaddrZS` [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode`が`MODE_INTRA`ではなく、`predFlagLY[xA_k][yA_k]`($Y = !X$ である)が1に等しく、`RefPicListX[refIdxLX]`と`RefPicListY[refIdxLY[xA_k][yA_k]]`が両方とも異なるPOC値を伴う長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、`availableFlagLXA`が1に等しく設定され、動きベクトル`mvLXA`が動きベクトル`mvLY[xA_k][yA_k]`に等しく設定され、`refIdxA`が`refIdxLY[xA_k][yA_k]`に等しく設定され、`ListA`が`ListY`に等しく設定される。

【0171】

- `availableFlagLXA`が1に等しく、`RefPicListA[refIdxA]`と`RefPicListX[refIdxLX]`の両方が短期参照ピクチャである場合、`mvLXA`は以下で規定されるように導出される。

【0172】

$$Tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-127)$$

$$\text{mvLXA} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) *$$

$$((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA}) + 127) >> 8)) \quad (8-128)$$

ここで、`td`および`tb`は、次のように導出され得る。

【0173】

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} -$$

10

20

30

40

50

PicOrderCnt(RefPicListA[refIdxA])

(8-129)

tb = Clip3(-128, 127, PicOrderCntVal -
PicOrderCnt(RefPicListX[refIdxLX]))

(8-130)

[0097] ビデオエンコーダ 20 およびビデオデコーダ 30 は、動きベクトル mvLB と利用可能性フラグ availableFlagLB を次の順序付けられたステップを使用して導出することができ、省略部分はやはり、HEVC WD 7 と同じテキストを表す。

【0174】

1.
2.
3.
4.
5.

6. $B_0 = xP + nPSW$ 、 $xB_1 = xB_0 - \text{MinPusize}$ 、および $xB_2 = xP - \text{MinPusize}$ である、 (xB_0, yB_0) から (xB_2, yB_2) までの (xB_k, yB_k) に対して、availableFlagLB が 0 に等しく設定され、isScaledBFlagLB が 0 に等しい場合、次のことと、availableFlagLB が 1 に等しくなるまで繰り返し適用する。

10

20

30

【0175】

- ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS[] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLX[xB_k][yB_k] が 1 に等しく、RefPicListX[refIdxLX] と RefPicListX[refIdxLX[xB_k][yB_k]] が両方とも異なる POC 値を伴う長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLB が 1 に等しく設定され、動きベクトル mvLB が動きベクトル mvLX[xB_k][yB_k] に等しく設定され、refIdxB が refIdxLX[xB_k][yB_k] に等しく設定され、listB が listX に等しく設定される。

【0176】

- そうではない場合、ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS[] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLY[xB_k][yB_k] ($Y = !X$ である) が 1 に等しく、RefPicListX[refIdxLX] と RefPicListY[refIdxLY[xB_k][yB_k]] が両方とも異なる POC 値を伴う長期参照ピクチャであるかまたは両方とも短期参照ピクチャである場合、availableFlagLB が 1 に等しく設定され、動きベクトル mvLB が動きベクトル mvLY[xB_k][yB_k] に等しく設定され、refIdxB が refIdxLY[xB_k][yB_k] に等しく設定され、listB が listY に等しく設定される。

40

【0177】

- availableFlagLB が 1 に等しく、PicOrderCnt(RefPicListB[refIdxLB]) が PicOrderCnt(RefPicListX[refIdxLX]) に等しくなく、RefPicListB[refIdxB] と RefPicListX[refIdxLX] の両方が短期参照ピクチャである場合、mvLB は以下で規定されるように導出される。

【0178】

tx = (16384 + (Abs(td) >> 1)) / td

(8-134)

50

`DistScaleFactor = Clip3(- 4096, 4095, (tb * tx + 32) >> 6)` (8-135)

`mvLXB =Clip3(- 8192, 8191.75, Sign(DistScaleFactor * mvLXA) *
((Abs(DistScaleFactor * mvLXA) + 127) >> 8))` (8-136)

ここで、`t d` および `t b` は、次のように導出され得る。

【0179】

`td = Clip3(- 128, 127, PicOrderCntVal -
PicOrderCnt(RefPicListB[refIdxB]))` (8-137)

`tb = Clip3(- 128, 127, PicOrderCntVal -
PicOrderCnt(RefPicListX[refIdxLX]))` (8-138)

10

[0098] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のプロセスに従って時間的ルーマ動きベクトル予測子を導出するように構成されてよく、下線が引かれたテキストは `H E V C_W D 7` に対する変更を表す。変数 `mvLXCol` および `availableFlagLXCol` は、次のように導出され得る。

【0180】

- 次の条件の 1つまたは複数が真である場合、`mvLXCol` の両方の成分は 0 に等しく設定され、`availableFlagLXCol` は 0 に等しく設定される。

【0181】

- `colPu` はイントラ予測モードでコーディングされる。

20

【0182】

- `colPu` は「利用不可能」とマークされる。

【0183】

- `pic_temporal_mvpenable_flag` は 0 に等しい。

【0184】

- そうではない場合、動きベクトル `mvCol`、参照インデックス `refIdxCOL`、および参照リスト識別子 `listCOL` は次のように導出される。

【0185】

- `PredFlagL0[xPCol][yPCol]` が 0 に等しい場合、`mvCol`、`refIdxCOL`、および `listCOL` はそれぞれ、`MvL1[xPCol][yPCol]`、`RefIdXL1[xPCol][yPCol]`、および `L1` に等しく設定される。

30

【0186】

- そうではない場合 (`PredFlagL0[xPCol][yPCol]` が 1 に等しい)、次のことが適用する。

【0187】

- `PredFlagL1[xPCol][yPCol]` が 0 に等しい場合、`mvCol`、`refIdxCOL`、および `listCOL` はそれぞれ、`MvL0[xPCol][yPCol]`、`RefIdXL0[xPCol][yPCol]`、および `L0` に等しく設定される。

40

【0188】

- そうではない場合 (`PredFlagL1[xPCol][yPCol]` が 1 に等しい)、次の割当てが行われる。

【0189】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` 以下である場合、`mvCol`、`refIdxCOL`、および `listCOL` は、`MvLX[xPCol][yPCol]`、`RefIdXLX[xPCol][yPCol]`、および `LX` に等しく設定され、それぞれ、`X` は、このプロセスが呼び出される `X` の値である。

【0190】

50

- そうではない場合（少なくとも1つの参照ピクチャリスト中の少なくとも1つのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` より大きい）、`mvCol`、`refIdxCol`、および `listCol` は、`MvLN[xPicCol][yPicCol]`、`RefIdxLN[xPicCol][yPicCol]`、および `LN` に等しく設定され、それぞれ、`N` は、`collocated_from_10_flag` の値である。

【0191】

- 次の条件の1つが真である場合、変数 `availableFlagLXCol` は0に等しく設定される。

- `RefPicListX[refIdxLX]` は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は0に等しい。

- `RefPicListX[refIdxLX]` は短期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は1に等しい。

- `RefPicListX[refIdxLX]` は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は1に等しく、`RefPicOrderCnt(colPic, refIdxCol, listCol)` は `PicOrderCnt(RefPicListX[refIdxLX])` に等しくない。

- そうではない場合、変数 `availableFlagLXCol` は1に等しく設定され、次のことが適用する。

【0192】

- `RefPicListX[refIdxLX]` が長期参照ピクチャである場合、または `LongTermRefPic(colPic, refIdxCol, listCol)` が1に等しい場合、または `PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)` が `PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])` に等しい場合、

`mvLXCol = mvCol` (8-143) 30

- そうではない場合、`mvLXCol` は、以下で規定されるような動きベクトル `mvCol` のスケーリングされたバージョンとして導出される。

【0193】

`tx = (16384 + (Abs(td) >> 1)) / td` (8-144)

`DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)` (8-145)

`mvLXCol = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) >> 8))` (8-146)

ここで、`td` および `tb` は、次のように導出され得る。

【0194】

`td = Clip3(-128, 127, PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol))` (8-147) 40

`tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]))` (8-148)

[0099]さらに別の例として、ビュー間参照ピクチャは、「参照に使用されない」としてマークされ得る。簡単にするために、そのようなピクチャは、HEVCベース仕様では非参照ピクチャと呼ばれることがあり、「長期参照に使用される」と「短期参照に使用される」のいずれかとしてマークされたピクチャは、HEVCベース仕様では参照ピクチャと呼ばれ得る。「参照ピクチャ」および「非参照ピクチャ」という用語は、「「参照に使用される」とマークされたピクチャ」および「「参照に使用されない」とマークされたピク

10

20

30

50

チャ」によって置き換えられ得る。

【0195】

[0100]関数UnusedRefPic(picX, refIdx, LX)は、次のように定義され得る。picXが現在のピクチャであった時点で、ピクチャpicXの参照ピクチャリストLXからのインデックスrefIdxをもつ参照ピクチャが「参照に使用されない」とマークされた場合、UnusedRefPic(picX, refIdx, LX)は1を返し、それ以外の場合、UnusedRefPic(pic, refIdx, LX)は0を返す。

【0196】

[0101]ビデオエンコーダ20およびビデオデコーダ30は、次のように動きベクトル予測子の候補に対する導出プロセスを実行するように構成されてよく、下線が引かれたテキストはHEVC WD7に対する変更を表し、省略部分はHEVC WD7と同じテキストを表す。
10

【0197】

Xが0または1である変数isScaledFlagLXは、0に等しく設定され得る。

【0198】

動きベクトルmvLXAおよび利用可能性フラグavailableFlagLXAは、次の順序付けられたステップで導出され得る。

【0199】

1. . . .
2. . . .
3. . . .
4. . . .
5. $y A_1 = y A_0 - \text{MinPusize}$ である、 $(x A_0, y A_0)$ から $(x A_1, y A_1)$ までの $(x A_k, y A_k)$ に対して、availableFlagLXAが0に等しい場合、次のことが、availableFlagLXAが1に等しくなるまで繰り返し適用する。
20

【0200】

- ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbaAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX $[x A_k][y A_k]$ が1に等しく、RefPicListX[refIdxLX]とRefPicListX[refIdxLX[x A_k][y A_k]]が両方とも参照ピクチャであるかまたは両方とも非参照ピクチャである場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLX $[x A_k][y A_k]$ に等しく設定され、refIdxAがrefIdxLX $[x A_k][y A_k]$ に等しく設定され、ListAがListXに等しく設定される。
30

【0201】

- そうではなく、ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbaAddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTERではなく、predFlagLY $[x A_k][y A_k]$ (Y = !Xである)が1に等しく、RefPicListX[refIdxLX]とRefPicListY[refIdxLY[x A_k][y A_k]]が両方とも参照ピクチャであるかまたは両方とも非参照ピクチャである場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLY $[x A_k][y A_k]$ に等しく設定され、refIdxAがrefIdxLY $[x A_k][y A_k]$ に等しく設定され、ListAがListYに等しく設定される。
40

【0202】

- availableFlagLXAが1に等しく、RefPicListA[ref

50

`I d x A]` と `R e f P i c L i s t X [r e f I d x L X]` の両方が短期参照ピクチャである場合、`m v L X A` は以下で規定されるように導出される。

【 0 2 0 3 】

$$T x = (16384 + (\text{Abs}(t d) >> 1)) / t d \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (t b * t x + 32) >> 6) \quad (8-127)$$

$$\begin{aligned} m v L X A &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * m v L X A) * \\ &\quad ((\text{Abs}(\text{DistScaleFactor} * m v L X A) + 127) >> 8)) \end{aligned} \quad (8-128)$$

ここで、`t d` および `t b` は、次のように導出され得る。

【 0 2 0 4 】

$$\begin{aligned} t d &= \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \\ &\quad \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}])) \end{aligned} \quad (8-129)$$

$$\begin{aligned} t b &= \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \\ &\quad \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \end{aligned} \quad (8-130)$$

[0102] ビデオエンコーダ 20 およびビデオデコーダ 30 は、動きベクトル `m v L X B` を利用可能なフラグ `a v a i l a b l e F l a g L X B` とを次の順序付けられたステップを使用して導出するように構成されてよく、下線が引かれたテキストは H E V C W D 7 に対する変更を表す。

【 0 2 0 5 】

1
2
3
4
5

6 . $B_0 = x P + n P S W$ 、 $x B_1 = x B_0 - \text{MinPusize}$ 、および $x B_2 = x P - \text{MinPusize}$ である、 $(x B_0, y B_0)$ から $(x B_2, y B_2)$ までの $(x B_k, y B_k)$ に対して、`i s S c a l e d F l a g L X` が 0 に等しく、`a v a i l a b l e F l a g L X B` が 0 に等しく設定される場合、次のことが、`a v a i l a b l e F l a g L X B` が 1 に等しくなるまで繰り返し適用する。

【 0 2 0 6 】

- ルーマ位置 $(x B_k, y B_k)$ をカバーする予測ユニットが利用可能であり [`E d . (B B) : M i n C b A d d r Z S [] []` と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`P r e d M o d e` が `M O D E _ I N T R A` ではなく、`p r e d F l a g L X [x B_k] [y B_k]` が 1 に等しく、`R e f P i c L i s t X [r e f I d x L X]` と `R e f P i c L i s t X [r e f I d x L X [x B_k] [y B_k]]` が両方とも参照ピクチャであるかまたは両方とも非参照ピクチャである場合、`a v a i l a b l e F l a g L X B` が 1 に等しく設定され、動きベクトル `m v L X B` が動きベクトル `m v L X [x B_k] [y B_k]` に等しく設定され、`r e f I d x B` が `r e f I d x L X [x B_k] [y B_k]` に等しく設定され、`L i s t B` が `L i s t X` に等しく設定される。

【 0 2 0 7 】

- そうではない場合、ルーマ位置 $(x B_k, y B_k)$ をカバーする予測ユニットが利用可能であり [`E d . (B B) : M i n C b A d d r Z S [] []` と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`P r e d M o d e` が `M O D E _ I N T R A` ではなく、`p r e d F l a g L Y [x B_k] [y B_k]` ($Y = ! X$ である) が 1 に等しく、`R e f P i c L i s t X [r e f I d x L X]` と `R e f P i c L i s t Y [r e f I d x L Y [x B_k] [y B_k]]` が両方とも参照ピクチャであるかまたは両方とも非参照ピクチャである場合、`a v a i l a b l e F l a g L X B` が 1 に等しく設定され、動きベクトル `m v L X B` が動きベクトル `m v L Y [x B_k] [y B_k]` に等しく設定され、`r e f I d x B` が `r e f I d x L Y [x B_k] [y B_k]` に等しく設定され、`L i s t B` が `L i s t Y` に等しく設定される。

【 0 2 0 8 】

10

20

30

40

50

- `availableFlagLXB` が 1 に等しく、`PicOrderCnt(RefPicListB[refIdxB])` が `PicOrderCnt(RefPicListX[refIdxLX])` に等しくなく、`RefPicListB[refIdxB]` と `RefPicListX[refIdxLX]` の両方が短期参照ピクチャである場合、`mvLXB` は以下で規定されるように導出される。

【0209】

`tx = (16384 + (Abs(td) >> 1)) / td` (8-134)

`DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)` (8-135)

`mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * (Abs(DistScaleFactor * mvLXA) + 127) >> 8))` (8-136)

ここで、`td` および `tb` は、次のように導出され得る。

【0210】

`td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListB[refIdxB]))` (8-137)

`tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]))` (8-138)

[0103] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のように時間的ルーマ動きベクトル予測子を導出するように構成されてよく、下線が引かれたテキストは H E V C W D 7 に対する変更を表す。

【0211】

変数 `mvLXC01` および `availableFlagLXC01` は、次のように導出され得る。

【0212】

- 次の条件の 1つまたは複数が真である場合、`mvLXC01` の両方の成分は 0 に等しく設定され、`availableFlagLXC01` は 0 に等しく設定される。

【0213】

- `colPu` はイントラ予測モードでコーディングされる。

【0214】

- `colPu` は「利用不可能」とマークされる。

【0215】

- `pic_temporal_mvpenable_flag` は 0 に等しい。

【0216】

- そうではない場合、動きベクトル `mvC01`、参照インデックス `refIdxC01`、および参照リスト識別子 `listC01` は次のように導出される。

【0217】

- `PredFlagL0[xPC01][yPC01]` が 0 に等しい場合、`mvC01`、`refIdxC01`、および `listC01` はそれぞれ、`MvL1[xPC01][yPC01]`、`RefIdxL1[xPC01][yPC01]`、および `L1` に等しく設定される。

【0218】

- そうではない場合 (`PredFlagL0[xPC01][yPC01]` が 1 に等しい)、次のことが適用する。

【0219】

- `PredFlagL1[xPC01][yPC01]` が 0 に等しい場合、`mvC01`、`refIdxC01`、および `listC01` はそれぞれ、`MvL0[xPC01][yPC01]`、`RefIdxL0[xPC01][yPC01]`、および `L0` に等しく設定される。

【0220】

- そうではない場合 (`PredFlagL1[xPC01][yPC01]` が 1 に等しい)、次の割当てが行われる。

10

20

30

40

50

【0221】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ `pic` の `PicOrd`
`erCnt(pic)` が `PicOrderCntVal` 以下である場合、`mvCol`、
`refIdxCol`、および `listCol` は、`MvLX[xPic][yPic]`、
`RefIdxLX[xPic][yPic]`、および `LX` に等しく設定され、それ
 れ、`X` は、このプロセスが呼び出される `X` の値である。

【0222】

- そうではない場合（少なくとも 1 つの参照ピクチャリスト中の少なくとも 1 つ
 のピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal`
 より大きい）、`mvCol`、`refIdxCol`、および `listCol` は、`MvLN[x`
`Pic][yPic]`、`RefIdxLN[xPic][yPic]`、および `LN` に等しく設定され、それ
 れ、`N` は、`collocated_from_10_flag` の値である。
10

【0223】

- 次の条件の 1 つが真である場合、変数 `availableFlagLXCol` は 0
 に等しく設定される。

- `RefPicListX[refIdxLX]` は非参照ピクチャであり、`UnusedRefPic(colPic, refIdxCol, listCol)` は 0 に等しい。

- `RefPicListX[refIdxLX]` は参照ピクチャであり、`UnusedRefPic(colPic, refIdxCol, listCol)` は 1 に等しい。
20

- そうではない場合、変数 `availableFlagLXCol` は 1 に等しく設定
 され、次のことが適用する。

【0224】

- `RefPicListX[refIdxLX]` が長期参照ピクチャである場合、
 または `LongTermRefPic(colPic, refIdxCol, listCol)` が 1 に等しい場合、または `PicOrderCnt(colPic) - RefPic`
`OrderCnt(colPic, refIdxCol, listCol)` が `PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])` に等しい場合、
30

$$mvLXCol = mvCol \quad (8-143)$$

である。

【0225】

- そうではない場合、`mvLXCol` は、以下で規定されるような動きベクトル
`mvCol` のスケーリングされたバージョンとして導出される。

【0226】

$$Tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-144)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * tx + 32) >> 6) \quad (8-145)$$

$$mvLXCol = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvCol) *$$

$$((\text{Abs}(\text{DistScaleFactor} * mvCol) + 127) >> 8)) \quad (8-146)$$

ここで、`td` および `tb` は、次のように導出される。

【0227】

$$td = \text{Clip3}(-128, 127, \text{PicOrderCnt}(colPic) - \text{RefPicOrderCnt}(colPic, refIdxCol, listCol)) \quad (8-147)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX])) \quad (8-148)$$

[0104] 「第 5 の例」として上で呼ばれる例に関して、ビデオエンコーダ 20 およびビデ
 オデコーダ 30 は、次の技法のいずれかまたはすべてに従って実行するように構成され得
50

る。この例では、異なる長期参照ピクチャを指す動きベクトルの間の予測はディセーブルにされてよく、異なるビュー間参照ピクチャを指す動きベクトルの間の予測はディセーブルにされてよく、ビュー間参照ピクチャを指す動きベクトルと長期参照ピクチャを指す動きベクトルとの間の予測はディセーブルにされてよい。

【0228】

[0105]この例では、関数 AddPicId(pic)は、ピクチャpicが属するビューまたはレイヤのビュー順序インデックスまたはレイヤIDを返す。したがって、ベースビューまたはベースレイヤに属する任意のピクチャ「pic」に対して、AddPicId(pic)は0を返す。HEVCベース仕様では、次のことが当てはまり得る。すなわち、関数AddPicId(pic)は、AddPicId(pic)が0(またはreserved_one_5ビット引く1)を返すというように定義され得る。この例では、AddPicId(pic)が0に等しくない場合、ピクチャpicは、時間的参照ピクチャではない(すなわち、短期参照ピクチャでも長期参照ピクチャでもない)。AddPicId(picX, refIdx, LX)はAddPicId(pic)を返すことができ、picは、ピクチャpicXの参照ピクチャリストLXからのインデックスrefIdxを伴う参照ピクチャである。

【0229】

[0106]ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル予測子の候補の導出プロセスを実行するように構成され得る。このプロセスへの入力は、現在のピクチャの一一番左上のサンプルに対する現在の予測ユニットの一一番左上のルーマサンプルを規定するルーマ位置(xP, yP)と、ルーマ、nPSW、およびnPSHに対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックスrefIdxLX(Xは0または1である)とを含み得る。このプロセスの出力は、隣接予測ユニットの動きベクトルmvLXNと、隣接予測ユニットの利用可能性フラグavailableFlagLXNとを含み得る(ここで、NはAまたはBにより置き換えられ得る)。

【0230】

[0107]Xが0または1である変数issScalEdFlagLXは、0に等しく設定され得る。ビデオエンコーダ20およびビデオデコーダ30は、動きベクトルmvLXAと利用可能性フラグavailableFlagLXNとを次の順序付けられたステップを使用して導出するように構成されてよく、下線が引かれたテキストはHEVC WD7に対する変更を表す。

【0231】

1. 2つのサンプル位置のセットを(x_{Ak}, y_{Ak})とし、 $k = 0, 1$ であり、 $x_{Ak} = x_P - 1$ 、 $y_{A_0} = y_P + nPSH$ 、および $y_{A_1} = y_{A_0} - \text{MinPusize}$ によりサンプル位置を規定する。サンプル位置のセット(x_{Ak}, y_{Ak})は、左側の区分境界とその延長線のすぐ左側のサンプル位置を表す。

【0232】

2. 利用可能性フラグavailableFlagLXNが最初に0に等しく設定されるようにして、mvLXAの両方の成分が0に等しく設定される。

【0233】

3. 次の条件の1つまたは複数が真である場合、変数issScalEdFlagLXが1に等しく設定される。

【0234】

- ルーマ位置(x_{A_0}, y_{A_0})をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbAddRZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではない。

【0235】

- ルーマ位置(x_{A_1}, y_{A_1})をカバーする予測ユニットが利用可能であり[Ed.(

10

20

30

40

50

B B) : M i n C b A d d r Z S [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、**P r e d M o d e**が**M O D E _ I N T R A**ではない。

【 0 2 3 6 】

4 . $y A_1 = y A_0 - \text{M i n P u S i z e}$ である、 $(x A_0, y A_0)$ から $(x A_1, y A_1)$ までの $(x A_k, y A_k)$ に対して、a v a i l a b l e F l a g L X Aが0に等しい場合、次のことが適用する。

【 0 2 3 7 】

- ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり [E d . (B B) : M i n C b A d d r Z S [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、**P r e d M o d e**が**M O D E _ I N T R A**ではなく、**p r e d F l a g L X [x A_k] [y A_k]** が1に等しく、参照インデックス **r e f I d x L X [x A_k] [y A_k]** が現在の予測ユニットの参照インデックス **r e f I d x L X** に等しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトル **m v L X A** が動きベクトル **m v L X [x A_k] [y A_k]** に等しく設定され、**r e f I d x A** が **r e f I d x L X [x A_k] [y A_k]** に等しく設定され、**L i s t A** が **L i s t X** に等しく設定される。

【 0 2 3 8 】

- そうではなく、ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり [E d . (B B) : M i n C b A d d r Z S [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、**P r e d M o d e**が**M O D E _ I N T R A**ではなく、**p r e d F l a g L Y [x A_k] [y A_k]** ($Y = ! X$ である) が1に等しく、A d d P i c I d (R e f P i c L i s t X [r e f I d x L X]) が A d d P i c I d (R e f P i c L i s t Y [r e f I d x L Y [x A_k] [y A_k]]) に等しく、P i c O r d e r C n t (R e f P i c L i s t Y [r e f I d x L Y [x A_k] [y A_k]]) が P i c O r d e r C n t (R e f P i c L i s t X [r e f I d x L X]) に等しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトル **m v L X A** が動きベクトル **m v L Y [x A_k] [y A_k]** に等しく設定され、**r e f I d x A** が **r e f I d x L Y [x A_k] [y A_k]** に等しく設定され、**L i s t A** が **L i s t Y** に等しく設定され、**m v L X A** が **m v L X A** に等しく設定される。

【 0 2 3 9 】

- a v a i l a b l e F l a g L X Aが1に等しい場合、次の1つまたは複数が真であれば、a v a i l a b l e F l a g L X Aは0に設定される。

- R e f P i c L i s t X [r e f I d x L X] と L i s t A [r e f I d x A] のうちの1つのみが長期参照ピクチャである。
- R e f P i c L i s t X [r e f I d x L X] と L i s t A [r e f I d x A] の両方が長期参照ピクチャであり、P i c O r d e r C n t (L i s t A [r e f I d x A]) が P i c O r d e r C n t (R e f P i c L i s t X [r e f I d x L X]) に等しくない。

5 . a v a i l a b l e F l a g L X Aが0に等しいとき、 $y A_1 = y A_0 - \text{M i n P u S i z e}$ である、 $(x A_0, y A_0)$ から $(x A_1, y A_1)$ までの $(x A_k, y A_k)$ に対して、a v a i l a b l e F l a g L X Aが0に等しい場合、次のことが適用する。

【 0 2 4 0 】

- ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり [E d . (B B) : M i n C b A d d r Z S [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、**P r e d M o d e**が**M O D E _ I N T R A**ではなく、**p r e d F l a g L X [x A_k] [y A_k]** が1に等しく、A d d P i c I d (R e f P i c L i s t L X [r e f I d x L X]) が A d d P i c I d (R e f P i c L i s t L X [r e f I d x L X [x A_k] [y A_k]]) に等しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトル **m v L X A** が動きベクトル **m v L X [x**

10

20

30

40

50

A_k [$y A_k$] に等しく設定され、 refIdxA が $\text{refIdxLX} [x A_k]$ [$y A_k$] に等しく設定され、 ListA が ListX に等しく設定される。

【0241】

- そうではなく、ルーマ位置 ($x A_k$, $y A_k$) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLY [$x A_k$] [$y A_k$] (Y = ! X である) が 1 に等しく、AddPicId(RefPicListLX[refIdxLY[$x A_k$] [$y A_k$]]) が AddPicId(RefPicListLY[refIdxLY[$x A_k$] [$y A_k$]]) に等しい場合、availableFlagLXA が 1 に等しく設定され、動きベクトル mvLXA が動きベクトル mvLY [$x A_k$] [$y A_k$] に等しく設定され、 refIdxA が $\text{refIdxLY} [x A_k] [y A_k]$ に等しく設定され、 ListA が ListY に等しく設定される。
10

【0242】

- availableFlagLXA が 1 に等しい場合、次の 1 つまたは複数が真であれば、availableFlagLXA は 0 に設定される。

- RefPicListX[refIdxLX] と ListA[refIdxA] のうちの 1 つのみが長期参照ピクチャである。

- RefPicListX[refIdxLX] と ListA[refIdxA] の両方が長期参照ピクチャであり、PicOrderCnt(ListA[refIdxA]) が PicOrderCnt(RefPicListX[refIdxLX]) に等しくない。
20

- availableFlagLXA が 1 に等しく、RefPicListA[refIdxA] と RefPicListX[refIdxLX] の両方が短期参照ピクチャである場合、mvLXA は以下で規定されるように導出される。

【0243】

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * tx + 32) >> 6) \quad (8-127)$$

$$\text{mvLXA} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor}) * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor}) * \text{mvLXA}) + 127) >> 8 \quad (8-128)$$

ここで、td および tb は、次のように導出され得る。

【0244】

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[refIdxA])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX])) \quad (8-130)$$

[0108] ビデオエンコーダ 20 およびビデオデコーダ 30 は、動きベクトル mvLXB と利用可能性フラグ availableFlagLXB を次の順序付けられたステップを使用して導出するように構成され得、下線が引かれたテキストは H E V C W D 7 に対する変更を表す。
40

【0245】

1. 3つのサンプル位置のセットを ($x B_k$, $y B_k$) とし、 $k = 0, 1, 2$ であり、 $x B_0 = x P + n P S W$ 、 $x B_1 = x B_0 - \text{MinPuSize}$ 、 $x B_2 = x P - \text{MinPuSize}$ 、および $y B_k = y P - 1$ によりサンプル位置を規定する。サンプル位置のセット ($x B_k$, $y B_k$) は、上側の区分境界とその延長線のすぐ上側のサンプル位置を表す。[Ed. (BB) : SPS 中で MinPuSize を定義するが、導出は AMP フラグの使用に依存すべきである]

2. $y P - 1$ が ($y C >> \log_2 C t b S i z e$) $<< \log_2 C t b S i z e$) より小さい場合、次のことが適用する。

【0246】

$xB_0 = (xB_0 >> 3) << 3) + ((xB_0 >> 3) \& 1) * 7$ (8-131)

$xB_1 = (xB_1 >> 3) << 3) + ((xB_1 >> 3) \& 1) * 7$ (8-132)

$xB_2 = (xB_2 >> 3) << 3) + ((xB_2 >> 3) \& 1) * 7$ (8-133)

3. 利用可能性フラグ a v a i l a b l e F l a g L X B が最初に 0 に等しく設定されるようにして、m v L X B の両方の成分が 0 に等しく設定される。

【0247】

4. $xB_0 = xP + nPSW$ 、 $xB_1 = xB_0 - \text{MinPuSize}$ 、および $xB_2 = xP - \text{MinPuSize}$ である、 (xB_0, yB_0) から (xB_2, yB_2) までの (xB_k, yB_k) に対して、a v a i l a b l e F l a g L X B が 0 に等しい場合、次のことが適用する。 10

【0248】

- ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLX [xB_k] [yB_k] が 1 に等しく、参照インデックス refIdxLX [xB_k] [yB_k] が現在の予測ユニットの参照インデックス refIdxLX に等しい場合、a v a i l a b l e F l a g L X B が 1 に等しく設定され、動きベクトル mvLXB が動きベクトル mvLX [xB_k] [yB_k] に等しく設定され、refIdxB が refIdxLX [xB_k] [yB_k] に等しく設定され、ListB が ListX に等しく設定される。 20

【0249】

- そうではなく、ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLY [xB_k] [yB_k] ($Y = !X$ である) が 1 に等しく、AddPicId (RefPicListX [refIdxLX]) が AddPicId (RefPicListLY [refIdxY [xB_k] [yB_k]]) に等しく、PicOrderCnt (RefPicListY [refIdxLY [xB_k] [yB_k]]) が PicOrderCnt (RefPicListX [refIdxLX]) に等しい場合、a v a i l a b l e F l a g L X B が 1 に等しく設定され、動きベクトル mvLXB が動きベクトル mvLY [xB_k] [yB_k] に等しく設定され、refIdxB が refIdxLY [xB_k] [yB_k] に等しく設定され、ListB が ListY に等しく設定される。 30

【0250】

- a v a i l a b l e F l a g L X A が 1 に等しい場合、次の 1 つまたは複数が真であれば、a v a i l a b l e F l a g L X A は 0 に設定される。

- RefPicListX [refIdxLX] と ListB [refIdxB] のうちの 1 つのみが長期参照ピクチャである。

- AddPicId (RefPicListX [refIdxLX]) が AddPicId (ListB [refIdxB]) に等しくない。 40

- RefPicListX [refIdxLX] と ListB [refIdxB] の両方が長期参照ピクチャであり、PicOrderCnt (ListB [refIdxB]) が PicOrderCnt (RefPicListX [refIdxLX]) に等しくない。

5. isScaledFlagLX が 0 に等しく、a v a i l a b l e F l a g L X B が 1 に等しい場合、m v L X A は m v L X B に等しく設定され、refIdxxA は refIdxxB に等しく設定され、a v a i l a b l e F l a g L X A は 1 に等しく設定される。

【0251】

6. isScaledFlagLX が 0 に等しく、a v a i l a b l e F l a g L X B 50

が0に等しく設定され、 $x_{B_0} = x_P + nPSW$ 、 $x_{B_1} = x_{B_0} - \text{MinPusize}$ 、および $x_{B_2} = x_P - \text{MinPusize}$ である、 (x_{B_0}, y_{B_0}) から (x_{B_2}, y_{B_2}) までの (x_{B_k}, y_{B_k}) に対して、availableFlagLXBが0に等しい場合、次のことが適用する。

【0252】

- ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbAddRZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX $[x_{B_k}] [y_{B_k}]$ が1に等しく、AddPicId(RefPicListX[refIdxLX])がAddPicId(RefPicListX[refIdxLX][y_{B_k}])に等しい場合、availableFlagLXBが1に等しく設定され、動きベクトルmvLXBが動きベクトルmvLX $[x_{B_k}] [y_{B_k}]$ に等しく設定され、refIdxBがrefIdxLX $[x_{B_k}] [y_{B_k}]$ に等しく設定され、ListBがListXに等しく設定される。

10

【0253】

- そうではなく、ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbAddRZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY $[x_{B_k}] [y_{B_k}]$ (Y = !Xである)が1に等しく、AddPicId(RefPicListY[refIdxLY][y_{B_k}])がAddPicId(RefPicListY[refIdxLY][y_{B_k}])に等しい場合、availableFlagLXBが1に等しく設定され、動きベクトルmvLXBが動きベクトルmvLY $[x_{B_k}] [y_{B_k}]$ に等しく設定され、refIdxBがrefIdxLY $[x_{B_k}] [y_{B_k}]$ に等しく設定され、ListBがListYに等しく設定される。

20

【0254】

- availableFlagLXAが1に等しい場合、次の1つまたは複数が真であれば、availableFlagLXAは0に設定される。

- RefPicListX[refIdxLX]とListB[refIdxB]のうちの1つのみが長期参照ピクチャである。

30

- RefPicListX[refIdxLX]とListB[refIdxB]の両方が長期参照ピクチャであり、PicOrderCnt(ListB[refIdxB])がPicOrderCnt(RefPicListX[refIdxLX])に等しくない。

- availableFlagLXBが1に等しく、PicOrderCnt(RefPicListB[refIdxB])がPicOrderCnt(RefPicListX[refIdxLX])に等しくなく、RefPicListB[refIdxB]とRefPicListX[refIdxLX]の両方が短期参照ピクチャである場合、mvLXBは以下で規定されるように導出される。

40

【0255】

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-135)$$

$$mvLXB = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) *$$

$$((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8)) \quad (8-136)$$

ここで、tdおよびtbは、次のように導出され得る。

【0256】

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[refIdxB])) \quad (8-137)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX])) \quad (8-138)$$

50

[0109] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のように時間的ルーマ動きベクトル予測子を導出するように構成され得る。このプロセスへの入力は、現在のピクチャの一一番左上のサンプルに対する現在の予測ユニットの一一番左上のルーマサンプルを規定するルーマ位置 (x_P 、 y_P) と、ルーマ、 $nPSW$ 、および $nPSH$ に対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックス $refIdx_{LX}$ (X は 0 または 1 である) を含み得る。このプロセスの出力は、動きベクトル予測 $mvLXCol$ と利用可能性フラグ $availableFlag_{LXCol}$ を含み得る。

【0257】

[0110] ピクチャ $picX$ の参照ピクチャリスト LX からのインデックス $refIdx$ を伴う参照ピクチャのピクチャ順序カウント $RefPicOrderCntVal$ を返し得る関数 $RefPicOrderCnt(picX, refIdx, LX)$ は、次のように規定され得る。

【0258】

$RefPicOrderCnt(picX, refIdx, LX) =$
 $PicOrderCnt(\text{ピクチャ } picX \text{ の } RefPicListX[\ refIdx \]) \quad (8-141)$

[0111] $slice_type$ 、 $collocated_from_10_flag$ 、および $collocated_ref_idx$ の値に応じて、ビデオエンコーダ 20 およびビデオデコーダ 30 は、コロケート区分を含むピクチャを規定する変数 $colPic$ を、次のように導出し得る。

【0259】

- $slice_type$ が B に等しく、 $collocated_from_10_flag$ が 0 に等しい場合、変数 $colPic$ は、 $RefPicList1[collocated_ref_idx]$ によって規定されるようなコロケート区分を含むピクチャを規定する。

【0260】

- そうではない場合 ($slice_type$ が B に等しく、かつ $collocated_from_10_flag$ が 1 に等しい、または $slice_type$ が P に等しい)、変数 $colPic$ は、 $RefPicList0[collocated_ref_idx]$ によって規定されるようなコロケート区分を含むピクチャを規定する。

【0261】

[0112] ビデオエンコーダ 20 およびビデオデコーダ 30 は、変数 $colPu$ とその位置 ($xPCol$ 、 $yPCol$) とを次の順序付けられたステップを使用して導出することができ、下線が引かれたテキストは H E V C WD 7 に対する変更を表す。

【0262】

1. 変数 $colPu$ は次のように導出され得る。

【0263】

$yPRb = yP + nPSH \quad (8-139)$

- ($yP >> Log2CtbSize$) が ($yPRb >> Log2CtbSize$) に等しい場合、現在の予測ユニットの一一番右下のルーマ位置の水平成分は

$yPRb = yP + nPSH \quad (8-139)$

によって定義され、変数 $colPu$ は $colPic$ の内の ($(xPRb >> 4) << 4$ 、 $(yPRb >> 4) << 4$) によって与えられる修正された位置をカバーする予測ユニットとして設定される。

【0264】

- そうではない場合 ($(yP >> Log2CtbSize)$ が ($yPRb >> Log2CtbSize$) と等しくない)、 $colPu$ は「利用不可能である」とマークされる。

【0265】

2. $colPu$ がイントラ予測モードでコーディングされる場合、または $colPu$ が「利用不可能」とマークされる場合、次のことが適用する。

10

20

30

40

50

【0266】

- 現在の予測ユニットの中心のルーマ位置は、

$$xPCtr = (xP + (nPSW \gg 1)) \quad (8-141)$$

$$yPCtr = (yP + (nPSH \gg 1)) \quad (8-142)$$

と定義される。

【0267】

- 変数 `colPu` は、 `colPic` 内の $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$ によって与えられる修正された位置をカバーする予測ユニットとして設定される。

【0268】

3. $(xPCol, yPCol)$ は、 `colPic` の一番左上のルーマサンプルに対する `colPu` の一番左上のルーマサンプルに等しく設定される。

【0269】

関数 `LongTermRefPic(picX, refIdx, LX)` は、次のように定義される。 `picX` が現在のピクチャであった時点で、ピクチャ `picX` の参照ピクチャリスト `LX` からのインデックス `refIdx` を伴う参照ピクチャが「長期参照に使用される」とマークされた場合、 `LongTermRefPic(picX, refIdx, LX)` は 1 を返し、それ以外の場合、 `LongTermRefPic(pic, refIdx, LX)` は 0 を返す。

【0270】

関数 `AddDPicId(picX, refIdx, LX)` は `AddPicId(pic)` を返し、 `pic` は、ピクチャ `picX` の参照ピクチャリスト `LX` からのインデックス `refIdx` を伴う参照ピクチャである。

変数 `mvLXCol` および `availableFlagLXCol` は、次のように導出される。

【0271】

- 次の条件の 1 つまたは複数が真である場合、 `mvLXCol` の両方の成分は 0 に等しく設定され、 `availableFlagLXCol` は 0 に等しく設定される。

【0272】

- `colPu` はイントラ予測モードでコーディングされる。

【0273】

- `colPu` は「利用不可能」とマークされる。

【0274】

- `pic_temporal_mv_enable_flag` は 0 に等しい。

【0275】

- そうではない場合、動きベクトル `mvCol`、参照インデックス `refIdxCol`、および参照リスト識別子 `listCol` は次のように導出される。

【0276】

- `PredFlagL0[xPCol][yPCol]` が 0 に等しい場合、 `mvCol`、`refIdxCol`、および `listCol` はそれぞれ、 `MvL1[xPCol][yPCol]`、`RefIdxL1[xPCol][yPCol]`、および `L1` に等しく設定される。

【0277】

- そうではない場合 (`PredFlagL0[xPCol][yPCol]` が 1 に等しい)、次のことが適用する。

【0278】

- `PredFlagL1[xPCol][yPCol]` が 0 に等しい場合、 `mvCol`、`refIdxCol`、および `listCol` はそれぞれ、 `MvL0[xPCol][yPCol]`、`RefIdxL0[xPCol][yPCol]`、および `L0` に等しく設定される。

10

20

30

40

50

【0279】

- そうではない場合 (`PredFlagL1[xPCol][yPCol]` が 1 に等しい)、次の割り当てが行われる。

【0280】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` 以下である場合、`mvCol`、`refIdxCol`、および `listCol` は、`MvLX[xPCol][yPCol]`、`RefIdxLX[xPCol][yPCol]`、および `LX` に等しく設定され、それぞれ、`X` は、このプロセスが呼び出される `X` の値である。

【0281】

- そうではない場合 (少なくとも 1 つの参照ピクチャリスト中の少なくとも 1 つのピクチャ `pic` の `PicOrderCnt(pic)` が `PicOrderCntVal` より大きい)、`mvCol`、`refIdxCol`、および `listCol` は、`MvLN[xPCol][yPCol]`、`RefIdxLN[xPCol][yPCol]`、および `LN` に等しく設定され、それぞれ、`N` は、`collocated_from_10_flag` の値である。

【0282】

- 次の条件の 1 つが真である場合、変数 `availableFlagLXCol` は 0 に等しく設定される。

- `AddPicId(RefPicListX[refIdxLX])` が `AddPicId(colPic, refIdxCol, listCol)` に等しくない。

- `RefPicListX[refIdxLX]` は短期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は 1 に等しい。

- `RefPicListX[refIdxLX]` は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は 0 に等しい。

- `RefPicListLX[refIdxLX]` は長期参照ピクチャであり、`LongTermRefPic(colPic, refIdxCol, listCol)` は 1 に等しく、`RefPicOrderCnt(colPic, refIdxCol, listCol)` は `PicOrderCnt(RefPicListLX[refIdxLX])` に等しくない。

- そうではない場合、変数 `availableFlagLXCol` は 1 に等しく設定され、次のことが適用する。

【0283】

- `RefPicListX[refIdxLX]` が長期参照ピクチャである、または `LongTermRefPic(colPic, refIdxCol, listCol)` が 1 に等しい、または `PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)` が `PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])` に等しい場合、

$$mvLXCol = mvCol \quad (8-143)$$

- そうではない場合、`mvLXCol` は、以下で規定されるような動きベクトル `mvCol` のスケーリングされたバージョンとして導出される。

【0284】

$$tx=(16384+(Abs(td)>>1))/td \quad (8-144)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-145)$$

$$mvLXCol = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) *$$

$$((Abs(DistScaleFactor * mvCol) + 127) >> 8)) \quad (8-146)$$

10

20

30

40

50

ここで、 t_d および t_b は、次のように導出され得る。

【0285】

```
td= Clip3( - 128, 127, PicOrderCnt(colPic)
    - RefPicOrderCnt( colPic, refIdxCol, listCol)) (8-147)
```

```
Tb = Clip3( - 128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX
    [refIdxLX] )) (8-148)
```

[0113] ビデオエンコーダ 20 およびビデオデコーダ 30 は、組み合わされた双方向予測マージ候補についての次の導出プロセスを実行するように構成され得る。このプロセスの入力は、マージ候補リスト mergeCandList と、mergeCandList の中にある各々の候補 N の参照インデックス refIdxL0N および refIdxL1N と、mergeCandList の中にある各々の候補 N の予測リスト利用フラグ predFlagL0N および predFlagL1N と、mergeCandList の中にある各々の候補 N の動きベクトル mvL0N および mvL1N と、mergeCandList 内の要素の数 numMergeCand と、空間的および時間的なマージ候補導出プロセスの後の、mergeCandList 内の要素の数 numOrigMergeCand とを含み得る。このプロセスの出力は、マージ候補リスト mergeCandList と、mergeCandList 内の要素の数 numMergeCand と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の参照インデックス refIdxL0combCandk および refIdxL1combCandk と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の予測リスト利用フラグ predFlagL0combCandk および predFlagL1combCandk と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の動きベクトル mvL0combCandk および mvL1combCandk を含み得る。
10

【0286】

[0114] numOrigMergeCand が 1 よりも大きく MaxNumMergeCand より小さい場合、変数 numInputMergeCand は numMergeCand に設定され、変数 combIdx および combCnt は 0 に設定され、変数 combStop は FALSE に設定され、combStop が TRUE に等しくなるまで次のステップが繰り返され得る（ここで、省略部分は HEVC WD7 で与えられるものと同じステップを表し、下線が引かれたテキストは HEVC WD7 に対する変更を表す）
30
。

【0287】

1. 変数 l0CandIdx および l1CandIdx は、表 8-8 において規定されるように combIdx を使用して導出される。

【0288】

2. 次の割当ては、マージ候補リスト mergeCandList 中の、位置 l0CandIdx における候補である l0Cand および位置 l1CandIdx における候補である l1Cand について行われる ($l0Cand = mergeCandList[l0CandIdx]$ 、 $l1Cand = mergeCandList[l1CandIdx]$)。
40

【0289】

3. 次の条件のすべてが真である場合、

- predFlagL0l0Cand == 1
- predFlagL1l1Cand == 1
- AddPicId(RefPicListL0[refIdxL0l0Cand]) !=
AddPicId(RefPicListL1[refIdxL1l1Cand]) || PicOrderCnt(RefPicList0[refIdxL0l0Cand])
!= icOrderCnt(RefPicList1[refIdxL1l1Cand]) || mvL0l0Cand != mvL1l1Cand

次のことが適用する。

50

【0290】

1
2
3

[0115]ある代替形態として、2つの長期参照ピクチャの間の予測は、スケーリングを伴わずにイネーブルされ、2つのビュー間参照ピクチャの間の予測は、スケーリングを伴わずにイネーブルされ得る。ビデオエンコーダ20およびビデオデコーダ30は、次のように動きベクトル予測子の候補に対する導出プロセスを実行するように構成され得、下線が引かれたテキストはHEVC_WD7に対する変更を表し、省略部分はHEVC_WD7と同じテキストを表す。
10

【0291】

Xが0または1である変数isScaledFlagLXは、0に等しく設定され得る。
。

【0292】

動きベクトルmvLXAおよび利用可能性フラグavailableFlagLXAは、次の順序付けられたステップで導出され得る。

【0293】

1
2
3
4

5 . $y A_1 = y A_0 - \text{MinPusize}$ である、 $(x A_0, y A_0)$ から $(x A_1, y A_1)$ までの $(x A_k, y A_k)$ に対して、availableFlagLXAが0に等しい場合、次のことが、availableFlagLXAが1に等しくなるまで繰り返し適用する。

【0294】

- ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbaaddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLX $[x A_k][y A_k]$ が1に等しく、RefPicType(RefPicListX[refIdxLX])がRefPicType(RefPicListX[refIdxLX][y A_k])に等しい場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLX $[x A_k][y A_k]$ に等しく設定され、refIdxAがrefIdxLX $[x A_k][y A_k]$ に等しく設定され、ListaがlistXに等しく設定される。
30

【0295】

- そうではなく、ルーマ位置 $(x A_k, y A_k)$ をカバーする予測ユニットが利用可能であり[Ed.(BB) : MinCbaaddrZS[][]と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、predFlagLY $[x A_k][y A_k]$ (Y= ! Xである)が1に等しく、RefPicType(RefPicListX[refIdxLX])がRefPicType(RefPicListY[refIdxLY][x A_k][y A_k])に等しい場合、availableFlagLXAが1に等しく設定され、動きベクトルmvLXAが動きベクトルmvLY $[x A_k][y A_k]$ に等しく設定され、refIdxAがrefIdxLY $[x A_k][y A_k]$ に等しく設定され、ListaがlistYに等しく設定される。
40

【0296】

- availableFlagLXAが1に等しく、RefPicListA[refIdxA]とRefPicListX[refIdxLX]の両方が短期参照ピクチャである場合、mvLXAは以下で規定されるように導出される。
50

【0297】

$tx=(16384+(\text{Abs}(td)>>1))/td$ (8-126)

$\text{DistScaleFactor}=\text{Clip3}(-4096,4095,(tb * tx + 32) >> 6)$ (8-127)

$\text{mvLXA}=\text{Clip3}(-8192,8191.75,\text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA})+127)>>8))$ (8-128)

ここで、 td および tb は、次のように導出され得る。

【0298】

$td=\text{Clip3}(-128,127,\text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[refIdxA]))$ (8-129)

$tb=\text{Clip3}(-128,127,\text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX]))$ (8-130)

[0116] ビデオエンコーダ 20 およびビデオデコーダ 30 は、動きベクトル $mvLB$ と利用可能性フラグ $availableFlagLagLB$ を次の順序付けられたステップを使用して導出し得、下線が引かれたテキストは $HEVC_WD7$ に対する変更を表し、省略部分は $HEVC_WD7$ と同じテキストを表す。

【0299】

1.

2.

3.

4.

5.

6. $xB_0 = xP + nPSW$ 、 $xB_1 = xB_0 - \text{MinPusize}$ 、および $xB_2 = xP - \text{MinPusize}$ である、 (xB_0, yB_0) から (xB_2, yB_2) までの (xB_k, yB_k) に対して、 $availableFlagLagLB$ が 0 に等しく設定され、 $isScalableFlagLX$ が 0 に等しい場合、次のことが、 $availableFlagLX$ が 1 に等しくなるまで繰り返し適用する。

【0300】

- ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、 $predFlagLX[xB_k][yB_k]$ が 1 に等しく、RefPicType(RefPicListX[refIdxLX]) が RefPicType(RefPicListX[refIdxLX[xB_k][yB_k]]) に等しい場合、availableFlagLX が 1 に等しく設定され、動きベクトル $mvLB$ が動きベクトル $mvLX$ $[xB_k][yB_k]$ に等しく設定され、 $refIdxB$ が $refIdxLX[xB_k][yB_k]$ に等しく設定され、 $ListB$ が $ListX$ に等しく設定される。

【0301】

- そうではなく、ルーマ位置 (xB_k, yB_k) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbAddrZS [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、 $predFlagLY[xB_k][yB_k]$ ($Y = !X$ である) が 1 に等しく、RefPicType(RefPicListX[refIdxLX]) が RefPicType(RefPicListY[refIdxLY[xB_k][yB_k]]) に等しい場合、availableFlagLB が 1 に等しく設定され、動きベクトル $mvLB$ が動きベクトル $mvLY[xB_k][yB_k]$ に等しく設定され、 $refIdxB$ が $refIdxLY[xB_k][yB_k]$ に等しく設定され、 $ListB$ が $ListY$ に等しく設定される。

【0302】

- availableFlagLB が 1 に等しく、PicOrderCnt(RefPicListB[refIdxB]) が PicOrderCnt(RefPicListY[refIdxLY[xB_k][yB_k]]) が PicOrderCnt(RefPicListX[refIdxLX[xB_k][yB_k]]) に等しい場合、availableFlagLY が 1 に等しく設定され、動きベクトル $mvLY$ $[xB_k][yB_k]$ に等しく設定され、 $refIdxB$ が $refIdxLY[xB_k][yB_k]$ に等しく設定され、 $ListB$ が $ListY$ に等しく設定される。

10

20

30

40

50

$tX [refIdxLX]$ に等しくなく、 $RefPicListB [refIdxB]$ と $RefPicListX [refIdxLX]$ の両方が短期参照ピクチャである場合、 $mvLB$ は以下で規定されるように導出される。

【0303】

$tx = (16384 + (\text{Abs}(td) >> 1)) / td$ (8-134)

$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6)$ (8-135)

$mvLXB = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8))$ (8-136)

ここで、 td および tb は、次のように導出され得る。

【0304】

$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[refIdxB]))$ (8-137)

$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[refIdxLX]))$ (8-138)

[0117] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次のように、時間的ルーマ動きベクトル予測の導出プロセスを実行するように構成され得る。

【0305】

変数 $mvLCOL$ および $availableFlagLCOL$ は、次のように導出され得る。

【0306】

- 次の条件の 1つまたは複数が真である場合、 $mvLCOL$ の両方の成分は 0 に等しく設定され、 $availableFlagLCOL$ は 0 に等しく設定される。

【0307】

- $COLPU$ はイントラ予測モードでコーディングされる。

【0308】

- $COLPU$ は「利用不可能」とマークされる。

【0309】

- $\text{pic_temporal_mvp_enable_flag}$ は 0 に等しい。

【0310】

- それ以外の場合、動きベクトル $mvCOL$ 、参照インデックス $refIdxCOL$ 、および参照リスト識別子 $listCOL$ は次のように導出される。

【0311】

- $\text{PredFlagL0} [xPCOL] [yPCOL]$ が 0 に等しい場合、 $mvCOL$ 、 $refIdxCOL$ 、および $listCOL$ はそれぞれ、 $MvL1 [xPCOL] [yPCOL]$ 、 $RefIdxL1 [xPCOL] [yPCOL]$ 、および $L1$ に等しく設定される。

【0312】

- そうではない場合 ($\text{PredFlagL0} [xPCOL] [yPCOL]$ が 1 に等しい)、次のことが適用する。

【0313】

- $\text{PredFlagL1} [xPCOL] [yPCOL]$ が 0 に等しい場合、 $mvCOL$ 、 $refIdxCOL$ 、および $listCOL$ はそれぞれ、 $MvL0 [xPCOL] [yPCOL]$ 、 $RefIdxL0 [xPCOL] [yPCOL]$ 、および $L0$ に等しく設定される。

【0314】

- そうではない場合 ($\text{PredFlagL1} [xPCOL] [yPCOL]$ が 1 に等しい)、次の割当てが行われる。

【0315】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャ pic の $PicOrderCnt (pic)$ が $PicOrderCntVal$ 以下である場合、 $mvCOL$ 、 $refIdxCOL$ は $listCOL$ に等しく設定される。

10

20

30

40

50

`f Id x Col`、および`list Col`は、`MvLX[x P Col][y P Col]`、`RefId x LX[x P Col][y P Col]`、および`LX`に等しく設定され、それぞれ、`X`は、このプロセスが呼び出される`X`の値である。

【0316】

- そうではない場合（少なくとも1つの参照ピクチャリスト中の少なくとも1つのピクチャ`pic`の`PicOrderCnt(pic)`が`PicOrderCntVal`より大きい）、`mvCol`、`refId x Col`、および`list Col`は、`MvLN[x P Col][y P Col]`、`RefId x LN[x P Col][y P Col]`、および`LN`に等しく設定され、それぞれ、`N`は、`collocated_from_10_flag`の値である。

10

【0317】

- `RefPicType(RefPicListX[refId x LX])`が`RefPicType(colPic, refIdxCol, listCol)`に等しくない場合、変数`availableFlagLXCol`は0に等しく設定される。

- そうではない場合、変数`availableFlagLXCol`は1に等しく設定され、次のことが適用する。

【0318】

- `RefPicListX[refId x LX]`が長期参照ピクチャである、または`LongTermRefPic(colPic, refIdxCol, listCol)`が1に等しい、または`PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)`が`PicOrderCntVal - PicOrderCnt(RefPicListX[refId x LX])`に等しい場合、`mvLXCol=mvCol`

(8-143)

- そうではない場合、`mvLXCol`は、以下で規定されるような動きベクトル`mvCol`のスケーリングされたバージョンとして導出される。

【0319】

$$tx=(16384+|Abs(td)|>>1))/td \quad (8-144)$$

$$DistScaleFactor=Clip3(-4096,4095,(tb*tx+32)>>6) \quad (8-145)$$

$$mvLXCol=Clip3(-8192,8191.75,Sign(DistScaleFactor*mvCol)*$$

$$((Abs(DistScaleFactor*mvCol)+127)>>8)) \quad (8-146)$$

20

ここで、`td`および`tb`は、次のように導出され得る。

【0320】

$$td=Clip3(-128,127,PicOrderCnt(colPic) - \\ RefPicOrderCnt(colPic, refIdxCol, listCol)) \quad (8-147)$$

$$tb=Clip3(-128,127,PicOrderCntVal - \\ PicOrderCnt(RefPicListX[refIdxLX])) \quad (8-148)$$

この例では、HEVC WD 7のセクション 8.5.2.1.3 は同じままであり得る。

【0321】

[0118]ある代替的な例では、異なる長期参照ピクチャを指す動きベクトルの間の予測はディセーブルにされ、ビュー間参照ピクチャを指す動きベクトルと長期参照ピクチャを指す動きベクトルとの間の予測はディセーブルにされ、異なるビュー間参照ピクチャを指す動きベクトルの間の予測はイネーブルにされ得る。この例では、以下で説明されるように、ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル予測子の候補の導出プロセスを実行するように構成され得る。このプロセスへの入力は、現在のピクチャの一番左上のサンプルに対する現在の予測ユニットの一番左上のルーマサンプルを規定するルーマ位置(`x P`、`y P`)と、ルーマ、`n PSW`、および`n PSN`に対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックス(`X`は0または1である)とを含み得る。このプロセスの出力は、隣接予測ユニットの動きベクトル`mvLXN`と、隣接予測ユニットの利用可能性フラグ`availableFlagLXN`とを含み得る(`N`はAまたはBにより置き換えられ得る)。

40

50

【0322】

[0119] Xが0または1である変数 `isScaledFlagX` は、0に等しく設定され得る。ビデオエンコーダ20およびビデオデコーダ30は、動きベクトル `mvLXA` を利用可能性フラグ `availableFlagXA` とを次の順序付けられたステップを使用して導出することができ、下線が引かれたテキストは H E V C WD 7 に対する変更を表す。

【0323】

1. 2つのサンプル位置のセットを (x_{A_k}, y_{A_k}) とし、 $k = 0, 1$ であり、 $x_{A_k} = x_P - 1$ 、 $y_{A_0} = y_P + nPSH$ 、および $y_{A_1} = y_{A_0} - \text{MinPusize}$ によりサンプル位置を規定する。サンプル位置のセット (x_{A_k}, y_{A_k}) は、左側の区分境界とその延長線のすぐ左側のサンプル位置を表す。
10

【0324】

2. 利用可能性フラグ `availableFlagXA` が最初に0に等しく設定されるようにして、`mvLXA` の両方の成分が0に等しく設定される。

【0325】

3. 次の条件の1つまたは複数が真である場合、変数 `isScaledFlagX` が1に等しく設定される。

【0326】

- ルーマ位置 (x_{A_0}, y_{A_0}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : `MinCbAddrZS` [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode` が `MODE_INTRA` ではない。
20

【0327】

- ルーマ位置 (x_{A_1}, y_{A_1}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : `MinCbAddrZS` [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode` が `MODE_INTRA` ではない。
。

【0328】

4. $y_{A_1} = y_{A_0} - \text{MinPusize}$ である、 (x_{A_0}, y_{A_0}) から (x_{A_1}, y_{A_1}) までの (x_{A_k}, y_{A_k}) に対して、`availableFlagXA` が0に等しい場合、次のことが適用する。
30

【0329】

- ルーマ位置 (x_{A_k}, y_{A_k}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : `MinCbAddrZS` [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode` が `MODE_INTRA` ではなく、`predFlagX[x_A_k][y_A_k]` が1に等しく、参照インデックス `refIdXLX[x_A_k][y_A_k]` が現在の予測ユニットの参照インデックス `refIdXLX` に等しい場合、`availableFlagXA` が1に等しく設定され、動きベクトル `mvLXA` が動きベクトル `mvLX[x_A_k][y_A_k]` に等しく設定され、`refIdXLX` が `refIdXLX[x_A_k][y_A_k]` に等しく設定され、`listA` が `listX` に等しく設定される。
40

【0330】

- そうではなく、ルーマ位置 (x_{A_k}, y_{A_k}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : `MinCbAddrZS` [] [] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、`PredMode` が `MODE_INTRA` ではなく、`predFlagY[x_A_k][y_A_k]` ($Y = !X$ である) が1に等しく、`AddPicId(RefPicListX[refIdXLX[x_A_k][y_A_k]])` が `AddPicId(RefPicListY[refIdXLX[x_A_k][y_A_k]])` に等しく、`PicOrderCnt(RefPicListY[refIdXLX[x_A_k][y_A_k]])` が `PicOrderCnt(RefPicListX[refIdXLX])` に等

しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトルm v L X Aが動きベクトルm v L Y [x A_k] [y A_k]に等しく設定され、r e f I d x Aがr e f I d x L Y [x A_k] [y A_k]に等しく設定され、L i s t AがL i s t Yに等しく設定され、m v L X Aがm v L X Aに等しく設定される。

【0331】

- a v a i l a b l e F l a g L X Aが1に等しい場合、次のことが真であれば、a v a i l a b l e F l a g L X Aは0に設定される。

- R e f P i c L i s t X [r e f I d x L X]とL i s t A [r e f I d x A]のうちの1つのみが長期参照ピクチャである。

5. a v a i l a b l e F l a g L X Aが0に等しいとき、y A₁=y A₀-M i n P u S i z eである、(x A₀, y A₀)から(x A₁, y A₁)までの(x A_k, y A_k)に対して、a v a i l a b l e F l a g L X Aが0に等しい場合、次のことが適用する。 10

【0332】

- ルーマ位置(x A_k, y A_k)をカバーする予測ユニットが利用可能であり[E d . (B B) : M i n C b A d d r Z S [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、P r e d M o d eがM O D E _ I N T R Aではなく、p r e d F l a g L X [x A_k] [y A_k]が1に等しく、A d d P i c I d (R e f P i c L i s t L X [r e f I d x L X])がA d d P i c I d (R e f P i c L i s t L X [r e f I d x L X [x A_k] [y A_k]])に等しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトルm v L X Aが動きベクトルm v L X [x A_k] [y A_k]に等しく設定され、r e f I d x Aがr e f I d x L Y [x A_k] [y A_k]に等しく設定され、L i s t AがL i s t Xに等しく設定される。 20

【0333】

- そうではなく、ルーマ位置(x A_k, y A_k)をカバーする予測ユニットが利用可能であり[E d . (B B) : M i n C b A d d r Z S [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、P r e d M o d eがM O D E _ I N T R Aではなく、p r e d F l a g L Y [x A_k] [y A_k] (Y= ! Xである)が1に等しく、A d d P i c I d (R e f P i c L i s t L Y [r e f I d x L Y [x A_k] [y A_k]])がA d d P i c I d (R e f P i c L i s t L Y [r e f I d x L Y [x A_k] [y A_k]])に等しい場合、a v a i l a b l e F l a g L X Aが1に等しく設定され、動きベクトルm v L X Aが動きベクトルm v L Y [x A_k] [y A_k]に等しく設定され、r e f I d x Aがr e f I d x L Y [x A_k] [y A_k]に等しく設定され、L i s t AがL i s t Yに等しく設定される。 30

【0334】

- a v a i l a b l e F l a g L X Aが1に等しい場合、次のことが真であれば、a v a i l a b l e F l a g L X Aは0に設定される。

- R e f P i c L i s t X [r e f I d x L X]とL i s t A [r e f I d x A]のうちの1つのみが長期参照ピクチャである。

- a v a i l a b l e F l a g L X Aが1に等しく、R e f P i c L i s t A [r e f I d x A]とR e f P i c L i s t X [r e f I d x L X]の両方が短期参照ピクチャである場合、m v L X Aは以下で規定されるように導出される。 40

【0335】

tx=(16384+(Abs(td)>>1))/td (8-126)

DistScaleFactor=Clip3(- 4096,4095,(tb*tx+32)>>6) (8-127)

mvLXA=Clip3(- 8192,8191.75,Sign(DistScaleFactor*mvLXA)*

((Abs(DistScaleFactor*mvLXA)+127)>>8)) (8-128)

ここで、t d およびt b は、次のように導出され得る。

【0336】

td=Clip3(- 128,127,PicOrderCntVal -
PicOrderCnt(RefPicListA[refIdxA])) (8-129)

```
tb=Clip3( - 128,127,PicOrderCntVal -
    PicOrderCnt( RefPicListX [refIdxLX]))
```

(8-130)

[0120] ビデオエンコーダ 20 およびビデオデコーダ 30 は、動きベクトル mvLXB と利用可能性フラグ availableFlagLXB を次の順序付けられたステップを使用して導出するように構成され得、下線が引かれたテキストは HEVC WD 7 に対する変更を表す。

【0337】

1. 3つのサンプル位置のセットを (x_{B_k}, y_{B_k}) とし、 $k = 0, 1, 2$ であり、 $x_{B_0} = xP + nPSW$ 、 $x_{B_1} = x_{B_0} - \text{MinPusize}$ 、 $x_{B_2} = xP - \text{MinPusize}$ 、および $y_{B_k} = yP - 1$ によりサンプル位置を規定する。サンプル位置のセット (x_{B_k}, y_{B_k}) は、上側の区分境界とその延長線のすぐ上側のサンプル位置を表す。
[Ed. (BB) : SPS 中で MinPusize を定義するが、導出は AMP フラグの使用に依存すべきである]

2. $yP - 1$ が ($yC >> \log_2 CtbSize$) << $\log_2 CtbSize$) より小さい場合、次のことが適用する。

【0338】

$$xB_0 = (xB_0 >> 3) << 3 + ((xB_0 >> 3) \& 1) * 7 \quad (8-131)$$

$$xB_1 = (xB_1 >> 3) << 3 + ((xB_1 >> 3) \& 1) * 7 \quad (8-132)$$

$$xB_2 = (xB_2 >> 3) << 3 + ((xB_2 >> 3) \& 1) * 7 \quad (8-133)$$

3. 利用可能性フラグ availableFlagLXB が最初に 0 に等しく設定されるようにして、mvLXB の両方の成分が 0 に等しく設定される。
[Ed. (BB) : MinCbadrzs[][] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLX[xBk][yBk] が 1 に等しく、参照インデックス refIdXLX[xBk][yBk] が現在の予測ユニットの参照インデックス refIdXLX に等しい場合、availableFlagLXB が 1 に等しく設定され、動きベクトル mvLXB が動きベクトル mvLX[xBk][yBk] に等しく設定され、refIdXLX が refIdXLX[xBk][yBk] に等しく設定され、listB が listX に等しく設定される。

【0340】

- ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbadrzs[][] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLX[xBk][yBk] が 1 に等しく、参照インデックス refIdXLX[xBk][yBk] が現在の予測ユニットの参照インデックス refIdXLX に等しい場合、availableFlagLXB が 1 に等しく設定され、動きベクトル mvLXB が動きベクトル mvLX[xBk][yBk] に等しく設定され、refIdXLX が refIdXLX[xBk][yBk] に等しく設定され、listB が listX に等しく設定される。

【0341】

- そうではなく、ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり [Ed. (BB) : MinCbadrzs[][] と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredMode が MODE_INTRA ではなく、predFlagLY[xBk][yBk] (Y = !X である) が 1 に等しく、AddPicId(RefPicListX[refIdXLX]) が AddPicId(RefPicListLY[refIdXLX[xBk][yBk]]) に等しく、PicOrderCnt(RefPicListY[refIdXLX[xBk][yBk]]) が PicOrderCnt(RefPicListX[refIdXLX]) に等しい場合、availableFlagLXB が 1 に等しく設定され、動きベクトル mvLXB が動きベクトル mvLY[xBk][yBk] に等しく設定され、refIdXLX が refIdLY[xBk][yBk] に等しく設定され、listB が listY に等しく設定される。

【0342】

10

20

30

40

50

- a v a i l a b l e F l a g L X Aが1に等しい場合、次のことが真であれば、a v a i l a b l e F l a g L X Aは0に設定される。

- R e f P i c L i s t X [r e f I d x L X]とL i s t B [r e f I d x B]のうちの1つのみが長期参照ピクチャである。

5. i s S c a l e d F l a g L Xが0に等しく、a v a i l a b l e F l a g L X Bが1に等しい場合、m v L X Aはm v L X Bに等しく設定され、r e f I d x Aはr e f I d x Bに等しく設定され、a v a i l a b l e F l a g L X Aは1に等しく設定される。

【0343】

6. i s S c a l e d F l a g L Xが0に等しく、a v a i l a b l e F l a g L X Bが0に等しく設定された場合、 $x_{B_0} = x_P + nPSW$ 、 $x_{B_1} = x_{B_0} - \text{MinPuSize}$ 、および $x_{B_2} = x_P - \text{MinPuSize}$ である、 (x_{B_0}, y_{B_0}) から (x_{B_2}, y_{B_2}) までの (x_{B_k}, y_{B_k}) に対して、a v a i l a b l e F l a g L X Bが0に等しく設定されるとき、次のことが適用する。

【0344】

- ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり[E d . (B B) : MinCbAddrZS [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、p r e d F l a g L X [x_{B_k}] [y_{B_k}]が1に等しく、A d d P i c I d (R e f P i c L i s t X [r e f I d x L X])がA d d P i c I d (R e f P i c L i s t X [r e f I d x L X [x_{B_k}] [y_{B_k}]])に等しい場合、a v a i l a b l e F l a g L X Bが1に等しく設定され、動きベクトルm v L X Bが動きベクトルm v L X [x_{B_k}] [y_{B_k}]に等しく設定され、r e f I d x Bがr e f I d x L X [x_{B_k}] [y_{B_k}]に等しく設定され、L i s t BがL i s t Xに等しく設定される。

【0345】

- そうではなく、ルーマ位置 (x_{B_k}, y_{B_k}) をカバーする予測ユニットが利用可能であり[E d . (B B) : MinCbAddrZS [] []と最小コーディングブロックに対する利用可能なプロセスとを使用して書き換える]、PredModeがMODE_INTRAではなく、p r e d F l a g L Y [x_{B_k}] [y_{B_k}] ($Y = !X$ である)が1に等しく、A d d P i c I d (R e f P i c L i s t X [r e f I d x L X])がA d d P i c I d (R e f P i c L i s t Y [r e f I d x L Y [x_{B_k}] [y_{B_k}]])に等しい場合、a v a i l a b l e F l a g L X Bが1に等しく設定され、動きベクトルm v L X Bが動きベクトルm v L Y [x_{B_k}] [y_{B_k}]に等しく設定され、r e f I d x Bがr e f I d x L Y [x_{B_k}] [y_{B_k}]に等しく設定され、L i s t BがL i s t Yに等しく設定される。

【0346】

- a v a i l a b l e F l a g L X Aが1に等しい場合、次のことが真であれば、a v a i l a b l e F l a g L X Aは0に設定される。

- R e f P i c L i s t X [r e f I d x L X]とL i s t B [r e f I d x B]のうちの1つのみが長期参照ピクチャである。

- a v a i l a b l e F l a g L X Bが1に等しく、P i c O r d e r C n t (R e f P i c L i s t B [r e f I d x B])がP i c O r d e r C n t (R e f P i c L i s t X [r e f I d x L X])に等しくなく、R e f P i c L i s t B [r e f I d x B]とR e f P i c L i s t X [r e f I d x L X]の両方が短期参照ピクチャである場合、m v L X Bは以下で規定されるように導出される。

【0347】

$$tx=(16384+(Abs(td)>>1))/td \quad (8-134)$$

$$\text{DistScaleFactor}=\text{Clip3}(-4096, 4095, (tb*tx+32)>>6) \quad (8-135)$$

$$mvLXB=\text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor}*mvLXA)*$$

$$((\text{Abs}(\text{DistScaleFactor}*mvLXA)+127)>>8)) \quad (8-136)$$

10

20

30

40

50

ここで、 t_d および t_b は、次のように導出され得る。

【0348】

```
td=Clip3( -128,127,PicOrderCntVal -
    PicOrderCnt(RefPicListB[refIdxB]))
```

(8-137)

```
tb=Clip3( -128,127,PicOrderCntVal -
    PicOrderCnt(RefPicListX[refIdxLX]))
```

(8-138)

[0121] ビデオエンコーダ 20 およびビデオデコーダ 30 は、以下で論じられるように、時間的ルーマ動きベクトル予測の導出プロセスを実施するように構成され得る。このプロセスへの入力は、現在のピクチャの一一番左上のサンプルに対する現在の予測ユニットの一一番左上のルーマサンプルを規定するルーマ位置 (x_P 、 y_P) と、ルーマ、nPSW、および nPSH に対する予測ユニットの幅と高さとを規定する変数と、現在の予測ユニットの区分の参照インデックス refIdxLX (X は 0 または 1 である) とを含み得る。このプロセスの出力は、動きベクトル予測 mvLXCol と利用可能性フラグ availab1eFlagLXCol を含み得る。

【0349】

[0122] 一例では、関数 RefPicOrderCnt(picX, refIdx, LX) は、ピクチャ picX の参照ピクチャリスト LX からのインデックス refIdx を伴う参照ピクチャのピクチャ順序カウント PicOrderCntVal を返し、次のように規定され得る。

【0350】

```
RefPicOrderCnt( picX, refIdx, LX ) =
    PicOrderCnt(ピクチャpicXのRefPicListX[ refIdx ])
```

(8-141)

[0123] slice_type、collocated_from_10_flag、および collocated_ref_idx の値に応じて、ビデオエンコーダ 20 およびビデオデコーダ 30 は、コロケート区分を含むピクチャを規定する変数 colPic を、次のように導出し得る。

【0351】

- slice_type が B に等しく、collocated_from_10_flag が 0 に等しい場合、変数 colPic は、RefPicList1[collocated_ref_idx] によって規定されるようなコロケート区分を含むピクチャを規定する。

【0352】

- そうではない場合 (slice_type が B に等しく collocated_from_10_flag が 1 に等しい、または slice_type が P に等しい)、変数 colPic は、RefPicList0[collocated_ref_idx] によって規定されるようなコロケート区分を含むピクチャを規定する。

【0353】

[0124] ビデオエンコーダ 20 およびビデオデコーダ 30 は、次の順序付けられたステップを使用して、変数 colPu とその位置 (x_{PCol} , y_{PCol}) とを導出することができる。

【0354】

1. 変数 colPu は次のように導出される。

【0355】

```
yPRb=yP+nPSH
```

(8-139)

- ($y_P >> \log_2 CtbSize$) が ($yPRb >> \log_2 CtbSize$) に等しい場合、現在の予測ユニットの一一番右下のルーマ位置の水平成分は

```
xPRb=xP+nPSW
```

(8-140)

によって定義され、変数 colPu は colPic 内の、($(xPRb >> 4) << 4$, $(yPRb >> 4) << 4$) によって与えられる修正された位置をカバーする予測ユニットとして設定される。

10

20

30

40

50

【0356】

- そうではない場合 ((yP>>Log2CtbSize) が (yPRb>>Log2CtbSize) と等しくない)、colPuは「利用不可能である」とマークされる。

【0357】

2. colPuがイントラ予測モードでコーディングされる場合、またはcolPuが「利用不可能」とマークされる場合、次のことが適用する。

【0358】

- 現在の予測ユニットの中心のルーマ位置は、

$$xPCtr = (xP + (nPSW >> 1)) \quad (8-141)$$

$$yPCtr = (yP + (nPSH >> 1)) \quad (8-142)$$

によって定義される。

10

【0359】

- 変数colPuは、colPic内の、((xPCtr >> 4) << 4, (yPCtr >> 4) << 4) によって与えられる修正された位置をカバーする予測ユニットとして設定される。

【0360】

3. (xPCol, yPCol) は、colPicの一番左上のルーマサンプルに対するcolPuの一番左上のルーマサンプルに等しく設定される。

【0361】

[0125] 関数 LongTermRefPic(picX, refIdx, LX) は次のように定義され得る。picXが現在のピクチャであった時点で、ピクチャpicXの参照ピクチャリストLXからのインデックスrefIdxを伴う参照ピクチャが「長期参照に使用される」とマークされた場合、LongTermRefPic(picX, refIdx, LX) は1を返し、それ以外の場合、LongTermRefPic(picX, refIdx, LX) は0を返す。

20

【0362】

[0126] ビデオエンコーダ20およびビデオデコーダ30は、HEVCの「AddPicId()」関数の修正されたバージョンを実装し得る。たとえば、ビデオエンコーダ20およびビデオデコーダ30は、この関数がAddPicId(pic)を返すようにAddPicId(picX, refIdx, LX)を実装することができ、「pic」は、ピクチャpicXの参照ピクチャリストLXからのインデックスrefIdxを伴う参照ピクチャである。

30

【0363】

[0127] ビデオエンコーダ20およびビデオデコーダ30は、次のように変数mvLXC01とavailableFlagLXC01とを導出することができ、下線が引かれたテキストはHEVC WD7に対する変更を表す。

【0364】

- 次の条件の1つまたは複数が真である場合、mvLXC01の両方の成分は0に等しく設定され、availableFlagLXC01は0に等しく設定される。

【0365】

- colPuはイントラ予測モードでコーディングされる。

40

【0366】

- colPuは「利用不可能」とマークされる。

【0367】

- pic_temporal_mv_p_enable_flagは0に等しい。

【0368】

- そうではない場合、動きベクトルmvC01、参照インデックスrefIdxC01、および参照リスト識別子listC01は次のように導出される。

【0369】

- PredFlagL0[xPCol][yPCol]が0に等しい場合、mvC01

50

、refIdxCol、およびlistColはそれぞれ、MvL1[xPCol][yPCol]、RefIdxL1[xPCol][yPCol]、およびL1に等しく設定される。

【0370】

- そうではない場合 (PredFlagL0[xPCol][yPCol]が1に等しい)、次のことが適用する。

【0371】

- PredFlagL1[xPCol][yPCol]が0に等しい場合、mvCol、refIdxCol、およびlistColはそれぞれ、MvL0[xPCol][yPCol]、RefIdxL0[xPCol][yPCol]、およびL0に等しく設定される。
10

【0372】

- そうではない場合 (PredFlagL1[xPCol][yPCol]が1に等しい)、次の割当てが行われる。

【0373】

- それぞれの参照ピクチャリスト中のそれぞれのピクチャpicのPicOrderCnt(pic)がPicOrderCntVal以下である場合、mvCol、refIdxCol、およびlistColは、MvLX[xPCol][yPCol]、RefIdxLX[xPCol][yPCol]、およびLXに等しく設定され、それ
れ、Xは、このプロセスが呼び出されるXの値である。
20

【0374】

- そうではない場合 (少なくとも1つの参照ピクチャリスト中の少なくとも1つのピクチャpicのPicOrderCnt(pic)がPicOrderCntValより大きい)、mvCol、refIdxCol、およびlistColは、MvLN[xPCol][yPCol]、RefIdxLN[xPCol][yPCol]、およびLNに等しく設定され、それ
れ、Nは、collocated_from_10_flagの値である。

【0375】

- 次の条件の1つが真である場合、変数availableFlagLXColは0に等しく設定される。
30

- AddPicId(RefPicListX[refIdxLX])がAddPicId(colPic, refIdxCol, listCol)に等しくない。

- RefPicListX[refIdxLX]は短期参照ピクチャであり、LongTermRefPic(colPic, refIdxCol, listCol)は1に等しい。

- RefPicListX[refIdxLX]は長期参照ピクチャであり、LongTermRefPic(colPic, refIdxCol, listCol)は0に等しい。

- そうではない場合、変数availableFlagLXColは1に等しく設定され、次のことが適用する。
40

【0376】

- RefPicListX[refIdxLX]が長期参照ピクチャである、またはLongTermRefPic(colPic, refIdxCol, listCol)が1に等しい、またはPicOrderCnt(colPic) - RefPicOrderCnt(pic, refIdxCol, listCol)がPicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])に等しい場合、
mvLXCol=mvCol
(8-143)

- そうではない場合、mvLXColは、以下で規定されるような動きベクトルmvColのスケーリングされたバージョンとして導出される。
50

【0377】

$tx=(16384+(\text{Abs}(td)>>1))/td$ (8-144)

$\text{DistScaleFactor}=\text{Clip3}(-4096,4095,(tb*tx+32)>>6)$ (8-145)

$\text{mvLXCol}=\text{Clip3}(-8192,8191.75,\text{Sign}(\text{DistScaleFactor}*\text{mvCol})*$

$((\text{Abs}(\text{DistScaleFactor}*\text{mvCol})+127)>>8))$ (8-146)

ここで、 td および tb は、次のように導出され得る。

【0378】

$td=\text{Clip3}(-128,127,\text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic},\text{refIdxCol},\text{listCol}))$ (8-147)

$tb=\text{Clip3}(-128,127,\text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])))$ (8-148)

[0128] ビデオエンコーダ 20 およびビデオデコーダ 30 は、組み合わされた双方向予測マージ候補の導出プロセスを実行するように構成され得る。このプロセスの入力は、マージ候補 mergeCandList と、 mergeCandList の中にある各々の候補 N の参照インデックス refIdxL0N および refIdxL1N と、 mergeCandList の中にある各々の候補 N の予測リスト利用フラグ predFlagL0N および predFlagL1N と、 mergeCandList の中にある各々の候補 N の動きベクトル mvL0N および mvL1N と、 mergeCandList 内の要素の数 numMergeCand と、空間的および時間的なマージ候補導出プロセスの後の、 mergeCandList 内の要素の数 numOrigMergeCand を含み得る。このプロセスの出力は、マージ候補リスト mergeCandList と、 mergeCandList 内の要素の数 numMergeCand と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の参照インデックス refIdxL0combCandk および refIdxL1combCandk と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の予測リスト利用フラグ $\text{predFlagL0combCandk}$ および $\text{predFlagL1combCandk}$ と、このプロセスの呼出しの間に mergeCandList に追加されている各々の新たな候補 combCandk の動きベクトル mvL0combCandk および mvL1combCandk を含み得る。

【0379】

[0129] このプロセスは次のように定義され得、下線が引かれたテキストは HEVC_WD_7 に対する変更を表し、省略部分は HEVC_WD_7 と同じテキストを表す。 numOrigMergeCand が 1 よりも大きく MaxNumMergeCand より小さい場合、変数 numInputMergeCand は numMergeCand に設定され、変数 combIdx および combCnt は 0 に設定され、変数 combStop は FALSE に設定され、 combStop が TRUE に等しくなるまで次のステップが繰り返され得る。

【0380】

1. 変数 $l0\text{CandIdx}$ および $l1\text{CandIdx}$ は、表 8-8 において規定されるように combIdx を使用して導出される。

【0381】

2. 次の割当ては、マージ候補リスト mergeCandList 中の、位置 $l0\text{CandIdx}$ における候補である $l0\text{Cand}$ 、および位置 $l1\text{CandIdx}$ における候補である $l1\text{Cand}$ について行われる ($l0\text{Cand}=\text{mergeCandList}[l0\text{CandIdx}]$ 、 $l1\text{Cand}=\text{mergeCandList}[l1\text{CandIdx}]$)。

【0382】

3. 次の条件のすべてが真である場合、

- $\text{predFlagL0l0Cand}==1$

10

20

30

40

50

- predFlagL111Cand==1
 - AddPicId(RefPicListL0[refIdxL010Cand]) != AddPicId(RefPicListL1[refIdxL111Cand]) || PicOrderCnt(RefPicList0 [refIdxL010Cand]) != PicOrderCnt(RefPicList1 [refIdxL111Cand]) || mvL010Cand != mvL111Cand

次のことが適用する。

【0383】

4

5

[0130]いくつかの例では、2つの異なる長期参照ピクチャを指す2つの動きベクトルの間の予測がディセーブルにされ得る。他の例では、2つの異なるビュー間参照ピクチャを指す2つの動きベクトルの間の予測がディセーブルにされ得る。10

【0384】

[0131]このようにして、ビデオエンコーダ20およびビデオデコーダ30は、ビデオデータの第1のピクチャについてのピクチャ順序カウント(POC)値をコーディングし、第1のピクチャの第2次元ピクチャ識別子をコーディングし、第1のピクチャのPOC値および第2次元ピクチャ識別子に基づいて、第2のピクチャをコーディングするように構成される、ビデオコーダの例を表す。第1のピクチャのPOC値と第2次元ピクチャ識別子に基づいて第2のピクチャをコーディングすることは、第1のピクチャのPOC値と第2次元ピクチャ識別子との両方を使用して第1のピクチャを識別することを含み得る。20

【0385】

[0132]その上、上で示されたように、第2のピクチャをコーディングすることは、第1のピクチャのPOC値および第2次元ピクチャ識別子と、候補動きベクトル予測子が参照する参照ピクチャのPOC値および第2次元ピクチャ識別子に基づいて、第1のピクチャを参照する動きベクトルについての動きベクトル予測をイネーブルまたはディセーブルにすることを含み得る。たとえば、第1のピクチャが短期ピクチャであることを第1のピクチャの第2次元ピクチャ識別子が示し、参照ピクチャが長期参照ピクチャであることを参照ピクチャの第2次元ピクチャ識別子が示す場合、ビデオエンコーダ20およびビデオデコーダ30は、第1のピクチャを参照する動きベクトルと参照ピクチャを参照する動きベクトルとの間の動きベクトル予測をディセーブルにし得る。30

【0386】

[0133]さらに、上でも示されたように、第2のピクチャをコーディングすることは、上で述べられたように、第1のピクチャを参照する第2のピクチャのロックの動きベクトルをコーディングすることを含み得る。そのようなコーディングは、動きベクトル予測子が異なるPOC値を有する参照ピクチャを参照する場合には、ビデオエンコーダ20およびビデオデコーダ30が、第1のピクチャと第2のピクチャとの間でのPOC値の差と、参照ピクチャと第2のピクチャとの間でのPOC値の差に基づいて動きベクトル予測子をスケーリングするように構成され得るという点で、第1のピクチャのPOC値に基づき得る。

【0387】

[0134]ビデオエンコーダ20およびビデオデコーダ30は、上で説明された例のいずれかまたはすべての技法を、単独で、または任意の組合せで実行するように構成され得る。ビデオエンコーダ20およびビデオデコーダ30は、適宜、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、ファームウェアなど、種々の好適なエンコーダまたはデコーダ回路のいずれか、あるいはそれらの任意の組合せとして各自実装され得る。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダ内に含まれてよく、それらのいずれも複合ビデオエンコーダ/デコーダ(コードック)の一部として統合されてよい。ビデオエンコーダ20および/またはビデオデコーダ30を含4050

むデバイスは、集積回路、マイクロプロセッサ、および／または、携帯電話もしくはタブレットコンピュータのようなワイヤレス通信デバイスを備え得る。

【0388】

[0135]図2は、ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を実装し得るビデオエンコーダ20の例を示すブロック図である。ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターフォーマンスの隣接フレームまたはピクチャ内でのビデオの空間的冗長性を低減または除去する。インターフォーマンスは、時間的予測を利用して、ビデオシーケンスの隣接フレームまたはピクチャ内でのビデオの時間的冗長性を低減または除去する。イントラモード(Iモード)は、いくつかの空間ベースのコーディングモードのいずれかを指し得る。単方向予測(Pモード)または双方向予測(Bモード)などのインターモードは、いくつかの時間ベースのコーディングモードのいずれかを指し得る。10

【0389】

[0136]図2に示されるように、ビデオエンコーダ20は、符号化されるべきビデオフレーム内の現在のビデオブロックを受信する。図2の例では、ビデオエンコーダ20は、モード選択ユニット40と、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、量子化ユニット54と、エントロピー符号化ユニット56とを含む。モード選択ユニット40は、今度は、動き補償ユニット44と、動き推定ユニット42と、イントラ予測ユニット46と、区分ユニット48とを含む。ビデオブロックの復元のために、ビデオエンコーダ20はまた、逆量子化ユニット58と、逆変換ユニット60と、加算器62とを含む。復元されたビデオからブロックキネスアーティファクトを除去するためにブロック境界をフィルタリングする、デブロックキングフィルタ(図2に図示せず)も含まれ得る。所望される場合、デブロックキングフィルタは一般に、加算器62の出力をフィルタリングすることになる。また、デブロックキングフィルタに加えて追加のフィルタ(ループ内またはループ後)が使用され得る。そのようなフィルタは、簡潔のために示されていないが、所望される場合、(ループ内フィルタとして)加算器50の出力をフィルタリングし得る。20

【0390】

[0137]符号化プロセス中に、ビデオエンコーダ20は、コーディングされるべきビデオフレームまたはスライスを受信する。フレームまたはスライスは、複数のビデオブロックに分割され得る。動き推定ユニット42および動き補償ユニット44は、時間的な予測を行うために、1つまたは複数の参照フレーム中の1つまたは複数のブロックに対する受信されたビデオブロックのインター予測コーディングを実行する。動き補償ユニット44は、たとえば、発展型動きベクトル予測(AMVP: advanced motion vector prediction)、時間的動きベクトル予測(TMVP: temporal motion vector prediction)、またはマージモードコーディングの間に、本開示の技法に従って動きベクトルをコーディングし得る。イントラ予測ユニット46は代替的に、空間的な予測を行うために、コーディングされるべきブロックと同じフレームまたはスライス中の1つまたは複数の隣接ブロックに対して受信されたビデオブロックのイントラ予測コーディングを実行し得る。ビデオエンコーダ20は、たとえば、ビデオデータのブロックごとに適切なコーディングモードを選択するために、複数のコーディングパスを実行し得る。3040

【0391】

[0138]その上、区分ユニット48は、前のコーディングパスにおける前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分し得る。たとえば、区分ユニット48は、初めにフレームまたはスライスをLCUに区分し、レートひずみ分析(たとえば、レートひずみ最適化)に基づいてLCUの各々をサブCUに区分し得る。モード選択ユニット40は、さらに、LCUをサブCUに区分することを示す4分木データ構造を生成し得る。4分木のリーフノードCUは、1つまたは複数のPUと、1つまたは複数のTUとを含み得る。50

【0392】

[0139]モード選択ユニット40は、たとえば、誤差結果に基づいて、コーディングモード、すなわち、イントラまたはインターのうちの1つを選択し得、残差ブロックデータを生成するために、得られたイントラコーディングされたブロックまたはインターコーディングされたブロックを加算器50に与え、参照フレームとして使用するための符号化ブロックを復元するために、得られたイントラコーディングされたブロックまたはインターコーディングされたブロックを加算器62に与える。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのようなシンタックス情報などのシンタックス要素をエントロピー符号化ユニット56に与える。

【0393】

[0140]動き推定ユニット42および動き補償ユニット44は高度に統合され得るが、概念的な目的のために別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、たとえば、現在のフレーム（または他のコード化ユニット）内でコーディングされている現在のブロックに対する参照フレーム（または他のコード化ユニット）内の予測ブロックに対する現在のビデオフレームまたはピクチャ内のビデオブロックのPUの変位を示し得る。予測ブロックは、絶対値差分和（SAD）、2乗差分和（SSD）、または他の差分尺度によって決定され得るピクセル差分に関して、コーディングされるブロックに精密に一致することがわかっているブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ64に記憶された参照ピクチャのサブ整数ピクセル位置の値を計算し得る。たとえば、ビデオエンコーダ20は、参照ピクチャの1/4ピクセル位置、1/8ピクセル位置、または他の分数ピクセル位置の値を補間し得る。したがって、動き推定ユニット42は、フルピクセル位置と分数ピクセル位置とに対する動き探索を実行し、分数ピクセル精度で動きベクトルを出力し得る。

【0394】

[0141]動き推定ユニット42は、PUの位置を参照ピクチャの予測ブロックの位置と比較することによって、インターコード化スライス中のビデオブロックのPUのための動きベクトルを計算する。参照ピクチャは、第1の参照ピクチャリスト（リスト0）または第2の参照ピクチャリスト（リスト1）から選択されてよく、それらの参照ピクチャリストの各々は、参照ピクチャメモリ64に記憶された1つまたは複数の参照ピクチャを識別する。動き推定ユニット42は、計算された動きベクトルをエントロピー符号化ユニット56と動き補償ユニット44とに送る。

【0395】

[0142]動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定された動きベクトルに基づいて予測ブロックをフェッチまたは生成することに関与し得る。この場合も、いくつかの例では、動き推定ユニット42と動き補償ユニット44とは機能的に統合され得る。現在のビデオブロックのPUのための動きベクトルを受信すると、動き補償ユニット44は、参照ピクチャリストのうちの1つにおいて動きベクトルが指す予測ブロックの位置を特定し得る。加算器50は、以下で説明されるように、コーディングされている現在ビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42はルーマ成分に対して動き推定を実行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方のためにルーマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40はまた、ビデオスライスのビデオブロックを復号する際にビデオデコーダ30が使用するためのビデオブロックとビデオスライスとに関連するシンタックス要素を生成し得る。

【0396】

[0143]イントラ予測ユニット46は、上で説明されたように、動き推定ユニット42と動き補償ユニット44とによって実行されるインター予測の代替として、現在のブロックをイントラ予測し得る。特に、イントラ予測ユニット46は、現在のブロックを符号化す

10

20

30

40

50

るために使用すべきイントラ予測モードを決定し得る。いくつかの例では、イントラ予測ユニット46は、たとえば、別々の符号化バスの間に、様々なイントラ予測モードを使用して、現在のブロックを符号化することができ、イントラ予測ユニット46（または、いくつかの例において、モード選択ユニット40）は、テストされたモードから使用するのに適切なイントラ予測モードを選択することができる。

【0397】

[0144]たとえば、イントラ予測ユニット46は、様々なテストされたイントラ予測モードのためのレートひずみ分析を使用してレートひずみ値を計算し、テストされたモードの中で最良のレートひずみ特性を有するイントラ予測モードを選択し得る。レートひずみ分析は、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符号化されていないブロックとの間のひずみ（または誤差）の量、ならびに符号化ブロックを生成するために使用されるビットレート（すなわち、ビット数）を決定する。イントラ予測ユニット46は、どのイントラ予測モードがブロックについて最良のレートひずみ値を呈するかを決定するために、様々な符号化ブロックのひずみおよびレートから比率を計算し得る。10

【0398】

[0145]ブロック用のイントラ予測モードを選択した後、イントラ予測ユニット46は、ブロック用に選択されたイントラ予測モードを示す情報を、エントロピー符号化ユニット56に提供することができる。エントロピー符号化ユニット56は、選択されたイントラ予測モードを示す情報を符号化することができる。ビデオエンコーダ20は、（コードワードマッピングテーブルとも呼ばれる）複数のイントラ予測モードインデックステーブルおよび複数の修正されたイントラ予測モードインデックステーブルと、様々なブロック用の符号化コンテキストの定義と、最確イントラ予測モードの指示とを含み得る送信されるビットストリーム構成データの中に、コンテキストの各々について使用する、イントラ予測モードインデックステーブルと修正されたイントラ予測モードインデックステーブルとを含めることができる。20

【0399】

[0146]ビデオエンコーダ20は、コーディングされている元のビデオブロックから、モード選択ユニット40からの予測データを減算することによって、残差ビデオブロックを形成する。加算器50は、この減算演算を実行する1つまたは複数のコンポーネントを表す。変換処理ユニット52は、離散コサイン変換（DCT）または概念的に同様の変換などの変換を残差ブロックに適用し、残差変換係数値を備えるビデオブロックを生成する。変換処理ユニット52は、DCTと概念的に同様である他の変換を実行し得る。ウェーブレット変換、整数変換、サブバンド変換または他のタイプの変換も使用され得る。いずれの場合も、変換処理ユニット52は、変換を残差ブロックに適用し、残差変換係数のブロックを生成する。変換は、残差情報をピクセル値領域から周波数領域などの変換領域に変換し得る。変換処理ユニット52は、得られた変換係数を量子化ユニット54に送り得る。量子化ユニット54は、ビットレートをさらに低減するために変換係数を量子化する。量子化プロセスは、係数の一部またはすべてに関連するビット深度を低減することができる。量子化の程度は、量子化パラメータを調整することによって修正され得る。いくつかの例では、量子化ユニット54は、次いで、量子化変換係数を含む行列の走査を実行し得る。代替的に、エントロピー符号化ユニット56が走査を実行し得る。3040

【0400】

[0147]量子化の後、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、コンテキスト適応型可変長コーディング（CAVL C）、コンテキスト適応型バイナリ算術コーディング（CABAC）、シンタックスペースコンテキスト適応型バイナリ算術コーディング（S B A C）、確率間隔区分エントロピー（P I P E）コーディングまたは別のエントロピーコーディング技法を実行し得る。コンテキストベースエントロピーコーディングの場合、コンテキストは隣接ブロックに基づき得る。エントロピー符号化ユニット56によるエント50

ロピーコーディングの後、符号化ビットストリームは、別のデバイス（たとえば、ビデオエンコーダ30）に送信されてよく、または後で送信するかもしくは取り出すために保管され得る。

【0401】

[0148]逆量子化ユニット58および逆変換ユニット60は、それぞれ逆量子化および逆変換を適用して、たとえば参照ブロックとして後で使用するために、ピクセル領域内で残差ブロックを復元する。動き補償ユニット44は、残差ブロックを参照ピクチャメモリ64のフレームのうちの1つの予測ブロックに加算することによって参照ブロックを計算し得る。動き補償ユニット44はまた、復元された残差ブロックに1つまたは複数の補間フィルタを適用して、動き推定において使用するサブ整数ピクセル値を計算し得る。加算器62は、復元された残差ブロックを、動き補償ユニット44によって生成された動き補償予測ブロックに加算して、参照ピクチャメモリ64に記憶するための復元されたビデオブロックを生成する。復元されたビデオブロックは、後続のビデオフレーム中のブロックをインターフェービングするための参照ブロックとして、動き推定ユニット42および動き補償ユニット44によって使用され得る。10

【0402】

[0149]ビデオエンコーダ20は、図1に関して論じられた様々な例示的な技法のいずれかまたはすべてを、単独で、または任意の組合せで実行するように構成され得る。たとえば、本開示の技法によれば、ビデオエンコーダ20は、参照ピクチャのピクチャ順序カウント（POC）値と参照ピクチャの第2次元の識別子とに基づいてピクチャを符号化することができる。すなわち、ビデオエンコーダ20は、第1のピクチャ（この例では参照ピクチャ）のPOC値とともに、第1のピクチャの第2次元ピクチャ識別子を符号化することができる。第2次元ピクチャ識別子は、たとえば、第1のピクチャを含むビューのビュー識別子、第1のピクチャを含むビューのビュー順序インデックス、ビュー順序インデックスと深度フラグの組合せ、第1のピクチャを含むスケーラブルビデオコーディング（SVC）レイヤのレイヤ識別子、および一般的なレイヤ識別子を備え得る。20

【0403】

[0150]加えて、または代替的に、第2次元の識別子は、第1のピクチャが長期参照ピクチャか短期参照ピクチャかを示す値を備え得る。あるいは、POC値および第2次元ピクチャ識別子に加えて、別個の値が、第1のピクチャが長期参照ピクチャか短期参照ピクチャかを示すことができる。いくつかの例では、参照ピクチャが長期か短期かを示すものは、参照ピクチャが時間的参照ピクチャかビュー間参照ピクチャかを示し得る。たとえば、長期参照ピクチャは時間的参照ピクチャ（すなわち、同じレイヤまたはビューの中の参照ピクチャ）に対応してよく、短期参照ピクチャは、ビュー間参照ピクチャに対応してよい。別の例として、長期参照ピクチャはビュー間参照ピクチャに対応してよく、短期参照ピクチャは時間的参照ピクチャに対応してよい。30

【0404】

[0151]同様に、ビデオエンコーダ20は、様々なタイプの動きベクトルの間の動きベクトル予測をディセーブルにし得る。動きベクトルの「タイプ」は、たとえば、時間的参照ピクチャ（すなわち、符号化されている現在のピクチャと同じビューの中のピクチャ）を指す時間的動きベクトル、および、ビュー間参照ピクチャ（すなわち、現在のピクチャを含むビューとは異なるビューの中のピクチャ）を指す視差動きベクトルを含み得る。通常、ビュー間参照ピクチャは、現在のピクチャと同じPOC値を有する。すなわち、通常、ビュー間参照ピクチャおよび現在のピクチャは、同じアクセスユニット中に現れる。ビデオエンコーダ20は、様々なタイプの動きベクトルの間の動きベクトル予測をディセーブルにし得る。すなわち、現在のピクチャの現在の動きベクトルが時間的動きベクトルであるとき、ビデオエンコーダ20は、視差動きベクトルに対する動きベクトル予測をディセーブルにし得る。同様に、現在の動きベクトルは視差動きベクトルであるとき、ビデオエンコーダ20は、時間的動きベクトルに対する動きベクトル予測をディセーブルにし得る。ビデオエンコーダ20は、そうでなければ、発展型動きベクトル予測（AMP）また4050

はマージモードのような動きベクトルコーディングプロセスを使用して、現在の動きベクトルを符号化することができる。

【0405】

[0152]いくつかの例では、ビデオエンコーダ20は、第1のピクチャ（たとえば、マルチビュービデオコーディングにおけるビュー成分）が長期参照ピクチャかどうかを示す値を、第1のピクチャがビュー間予測に使用されるかどうかに少なくとも一部基づいてコーディングするように構成され得る。たとえば、ビデオエンコーダ20は、第1のピクチャが、第1のピクチャを含むシーケンスに対応するシーケンスパラメータセット（S P S）において長期参照ピクチャか短期参照ピクチャかを示す、シンタックス要素を符号化することができる。

10

【0406】

[0153]加えて、または代替的に、ビデオエンコーダ20は、少なくとも一時的に、長期参照ピクチャとしてビュー間参照ピクチャをマークするように構成され得る。ビデオエンコーダ20はさらに、ビュー間参照ピクチャの現在の状態を記憶することができ、ここで、状態は、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つを備え得る。したがって、第1のピクチャがビュー間ピクチャを備える場合、ビデオエンコーダ20は、第1のピクチャを長期参照ピクチャとしてマークし得る。第1のピクチャに対して第2のピクチャをコーディングした後、ビデオエンコーダ20は、記憶された状態に基づいて、ビュー間参照ピクチャの状態を取り戻すことができる。

【0407】

20

[0154]加えて、または代替的に、ビデオエンコーダ20は一時的に、新たなP O C値をビュー間参照ピクチャに割り当てつつ、第2のピクチャを符号化することができる。たとえば、ビデオエンコーダ20は、現在の時間的参照ピクチャに対するP O C値のセットを決定し、使用されていないP O C値をビュー間参照ピクチャに割り当てることができる。ビデオエンコーダ20はまた、各々のビュー間参照ピクチャに対して、それぞれの現在のP O C値を記憶することができる。第2のピクチャを符号化した後、ビデオエンコーダ20は、ビュー間参照ピクチャに対する記憶された（すなわち、オリジナルの）P O C値をリストアし得る。ビュー間参照ピクチャは通常、第2のピクチャ（すなわち、現在符号化されているピクチャ）と同じアクセスユニット中にあるので、いくつかの例では、代わりに、ビデオエンコーダ20は単に、ビュー間参照ピクチャについてのP O C値を、第2のピクチャ、すなわち符号化されている現在のピクチャのP O C値と等しく設定し、その結果、P O C値の記憶が必須ではなくなる。

30

【0408】

[0155]このようにして、図2のビデオエンコーダ20は、ビデオデータの第1のピクチャに対するピクチャ順序カウント（P O C）値を符号化し、第1のピクチャの第2次元ピクチャ識別子を符号化し、ベースビデオコーディング仕様に従って、第1のピクチャのP O C値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化するように構成されるビデオエンコーダの例を表す。ベースビデオコーディング仕様はH E V Cを備え得る。加えて、ビデオエンコーダ20は、ベースビデオコーディング仕様の拡張、たとえば、H E V CのS V CまたはM V C拡張に従って、ピクチャを符号化するように構成され得る。したがって、ビデオエンコーダ20はまた、ビデオデータの第1のピクチャに対するピクチャ順序カウント（P O C）値を符号化し、第1のピクチャの第2次元ピクチャ識別子を符号化し、ベースビデオコーディング仕様の拡張に従って、第1のピクチャのP O C値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを符号化するように構成されるビデオエンコーダの例を表す。

40

【0409】

[0156]図3は、ビデオコーディング規格のハイレベルシンタックスのみの拡張による、ビデオデータをコーディングするための技法を実装し得るビデオデコーダ30の例を示すブロック図である。図3の例では、ビデオデコーダ30は、エントロピー復号ユニット70と、動き補償ユニット72と、イントラ予測ユニット74と、逆量子化ユニット76と

50

、逆変換ユニット78と、参照ピクチャメモリ82と、加算器80とを含む。ビデオデコーダ30は、いくつかの例では、ビデオエンコーダ20(図2)に関して説明された符号化パスとは全般に逆の復号パスを実行し得る。動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルに基づいて予測データを生成することができ、イントラ予測ユニット74は、エントロピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて予測データを生成することができる。

【0410】

[0157]復号プロセス中に、ビデオデコーダ30は、ビデオエンコーダ20から、符号化ビデオスライスのビデオブロックと、関連するシンタックス要素とを表す符号化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、量子化係数と、動きベクトルまたはイントラ予測モードインジケータと、他のシンタックス要素とを生成するためにビットストリームをエントロピー復号する。エントロピー復号ユニット70は、動きベクトルと他の予測シンタックス要素とを動き補償ユニット72に転送する。ビデオデコーダ30は、ビデオスライスレベルおよび/またはビデオブロックレベルでシンタックス要素を受信し得る。10

【0411】

[0158]ビデオスライスがイントラコード化(I)スライスとしてコーディングされるとき、イントラ予測ユニット74は、シグナリングされたイントラ予測モードと、現在のフレームまたはピクチャの、前に復号されたブロックからのデータとに基づいて、現在のビデオスライスのビデオブロックのための予測データを生成し得る。ビデオフレームがインターフード化(たとえば、B、P、またはGPB)スライスとしてコーディングされるとき、動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルと他のシンタックス要素とに基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリストの1つの中の参照ピクチャの1つから生成され得る。ビデオデコーダ30は、参照ピクチャメモリ82に記憶された参照ピクチャに基づいて、デフォルトの構成技法を使用して、参照フレームリスト、すなわち、リスト0とリスト1とを構築し得る。20

【0412】

[0159]本開示の技法によれば、エントロピー復号ユニット70は、現在のピクチャの現在のブロックに対する動き情報を表す、エントロピー符号化データを復号することができる。たとえば、AMPに従って、エントロピー復号ユニット70は、現在のブロックに対する動きベクトル差分(MVD)値を復号することができる。動き補償ユニット72(または、エントロピー復号ユニット70のようなビデオデコーダ30の別のユニット)は、MVD値のような、エントロピー復号された動き情報を使用して、現在のブロックに対する動きベクトルを復元することができる。たとえば、動き補償ユニット72は、たとえば、現在の動きベクトルが長期参照ピクチャまたは短期参照ピクチャ(または時間的参照ピクチャまたはビュー間参照ピクチャ)を参照するかどうか、および、候補参照ピクチャのセットも長期参照ピクチャまたは短期参照ピクチャ(または、時間的参照ピクチャまたはビュー間参照ピクチャ)を指すかどうかに基づいて、現在の動きベクトルに対する利用可能な動きベクトル予測子のセットを決定することができる。30

【0413】

[0160]上で論じられたように、動き補償ユニット72は、異なるタイプの候補動きベクトル予測子が現在の動きベクトルを予測するために使用できないと決定し得る。たとえば、現在の動きベクトルが時間的動きベクトルである場合、動き補償ユニット72は、視差動きベクトルが現在の動きベクトルに対する動きベクトル予測子として使用できないと決定し得る。同様に、現在の動きベクトルが視差動きベクトルである場合、動き補償ユニット72は、時間的動きベクトルが現在の動きベクトルに対する動きベクトル予測子として使用できないと決定し得る。いくつかの例では、動き補償ユニット72は、長期参照ピクチャと短期参照ピクチャの間の動きベクトル予測を、同様に、または代替的にディセーブルにし得る。4050

【0414】

[0161] 現在の動きベクトルが視差動きベクトルである場合、動き補償ユニット72はまた、動きベクトル予測子のスケーリングを避けることができる（このことは同様に、視差動きベクトルにも対応し得る）。加えて、または代替的に、動き補償ユニット72は、視差動きベクトル予測子が視差動きベクトルの動きベクトル予測の間に参照するビュー間参照ピクチャに、時間的POC値を割り当て得る。

【0415】

[0162] いずれの場合でも、動き補償ユニット72またはビデオデコーダ30の別の要素は、たとえば、AMPまたはマージモードを使用して、現在のブロックに対する動きベクトルを再現することができる。動き補償ユニット72は、動きベクトルと他のシンタックス要素とを解析することによって現在のビデオスライスのビデオブロックについての予測情報を決定し、予測情報を使用して、復号されている現在のビデオブロックのための予測ブロックを生成する。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）、インター予測スタイル（たとえば、Bスライス、Pスライス、またはGPBスライス）、スライスの参照ピクチャリストのうちの1つまたは複数に対する構成情報、スライスの各インター符号化ビデオブロックに対する動きベクトル、スライスの各インターフォード化ビデオブロックに対するインター予測ステータス、および現在のビデオスライス中のビデオブロックを復号するための他の情報を決定するために、受信されたシンタックス要素のいくつかを使用する。動き補償ユニット72は、たとえば、発展型動きベクトル予測（AMP）、時間的動きベクトル予測（T MVP）、またはマージモードコーディングの間に、本開示の技法に従って動きベクトルをコーディングし得る。

【0416】

[0163] 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得る。動き補償ユニット72は、ビデオブロックの符号化中にビデオエンコーダ20によって使用された補間フィルタを使用して、参照ブロックのサブ整数ピクセルの補間値を計算し得る。この場合、動き補償ユニット72は、受信されたシンタックス要素からビデオエンコーダ20によって使用された補間フィルタを決定し、その補間フィルタを使用して予測ブロックを生成し得る。

【0417】

[0164] 逆量子化ユニット76は、ピットストリーム中で与えられ、エントロピー復号ユニット70によって復号された量子化変換係数を逆量子化（inverse quantize）、すなわち、逆量子化（de-quantize）する。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するための、ビデオスライス中のビデオブロックごとにビデオエンコーダ30によって計算された量子化パラメータ Q_{PY} の使用を含み得る。逆変換ユニット78は、逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換処理を変換係数に適用して、ピクセル領域において残差ブロックを生成する。

【0418】

[0165] 動き補償ユニット72が、動きベクトルと他のシンタックス要素とに基づいて現在のビデオブロックのための予測ブロックを生成した後、ビデオデコーダ30は、逆変換ユニット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つまたは複数のコンポーネントを表す。所望される場合、プロックキネティクスアーティファクトを除去するために、復号されたブロックをフィルタリングするためのデブロッキングフィルタも適用され得る。ピクセル遷移を平滑化し、またはそうではなくビデオ品質を改善するために、（コーディングループ内またはコーディングループ後の）他のループフィルタも使用され得る。所与のフレームまたはピクチャの中の復号されたビデオブロックは、次いで、その後の動き補償のために使用される参照ピクチャを記憶する参照ピクチャメモリ82に記憶される。参照ピクチャメモリ82はまた、図1のディ

10

20

30

40

50

スプレイデバイス 3 2 のようなディスプレイデバイス上で後の表示のために、復号されたビデオを記憶する。

【 0 4 1 9 】

[0166]このようにして、図 3 のビデオデコーダ 3 0 は、ビデオデータの第 1 のピクチャに対するピクチャ順序カウント (P O C) 値を復号し、第 1 のピクチャの第 2 次元ピクチャ識別子を復号し、第 1 のピクチャの P O C 値および第 2 次元ピクチャ識別子に少なくとも一部基づいて、第 2 のピクチャを復号するように構成される、ビデオデコーダの例を表す。ベースビデオコーディング仕様は H E V C を備え得る。加えて、ビデオデコーダ 3 0 は、ベースビデオコーディング仕様の拡張、たとえば、H E V C の S V C または M V C 拡張に従って、ピクチャを符号化するように構成され得る。したがって、ビデオデコーダ 3 0 はまた、ビデオデータの第 1 のピクチャに対するピクチャ順序カウント (P O C) 値を復号し、第 1 のピクチャの第 2 次元ピクチャ識別子を復号し、ベースビデオコーディング仕様の拡張に従って、第 1 のピクチャの P O C 値および第 2 次元ピクチャ識別子に少なくとも一部基づいて、第 2 のピクチャを復号するように構成されるビデオデコーダの例を表す。10

【 0 4 2 0 】

[0167]図 4 は、例示的な M V C 予測パターンを示す概念図である。マルチビュービデオコーディング (M V C) は、 I T U - T H . 2 6 4 / A V C の拡張である。同様の技法が、 H E V C に適用され得る。図 4 の例では、(ビュー I D 「 S 0 」 ~ 「 S 7 」を有する) 8 つのビューが示され、各ビューについて 1 2 個の時間位置 (「 T 0 」 ~ 「 T 1 1 」) が示されている。すなわち、図 4 中の各行はビューに対応し、各列は時間位置を示す。20

【 0 4 2 1 】

[0168] M V C は H . 2 6 4 / A V C デコーダによって復号可能であるいわゆるベースビューを有し、ステレオビューペアは M V C によってサポートされ得るが、 M V C の 1 つの利点は、 M V C が、 3 D ビデオ入力として 3 つ以上のビューを使用し複数のビューによって表されるこの 3 D ビデオを復号する例を、サポートできることである。 M V C デコーダを有するクライアントのレンダラは、複数のビューを伴う 3 D ビデオコンテンツを予想することができる。

【 0 4 2 2 】

[0169]典型的な M V C の復号順序の構成は、時間優先コーディングと呼ばれる。アクセスユニットは、 1 つの出力時間インスタンスのためのすべてのビューのコード化ピクチャを含み得る。たとえば、時刻 T 0 のピクチャの各々は、共通のアクセスユニットに含まれてよく、時刻 T 1 のピクチャの各々は、第 2 の共通のアクセスユニットに含まれてよく、以下同様である。復号順序は、出力または表示順序と必ずしも同一とは限らない。30

【 0 4 2 3 】

[0170] 図 4 中のフレーム、すなわちピクチャは、対応するフレームがイントラコーディングされる (すなわち、 I フレームである) のか、一方向に (すなわち、 P フレームとして) インターコーディングされるのか、複数の方向に (すなわち、 B フレームとして) インターコーディングされるのかを指定する、文字を含む影付きブロックを使用して、図 4 中の各行と各列との交差部において示されている。一般に、予測は矢印によって示され、ここで矢印の終点のフレームは、予測参照のために矢印の始点のオブジェクトを使用する。たとえば、時間位置 T 0 にあるビュー S 2 の P フレームは、時間位置 T 0 にあるビュー S 0 の I フレームから予測される。40

【 0 4 2 4 】

[0171] シングルビュービデオ符号化の場合と同様に、マルチビュービデオコーディングのビデオシーケンスのフレームは、異なる時間位置におけるフレームに関して予測的に符号化され得る。たとえば、時間位置 T 1 におけるビュー S 0 の b フレームは、時間位置 T 0 におけるビュー S 0 の I フレームからその b フレームに向けられた矢印を有し、その矢印は、 b フレームが I フレームから予測されることを示す。しかしながら、加えて、マルチビュービデオ符号化のコンテキストでは、フレームは、ビュー間予測され得る。すなわ50

ち、ビュー成分は、参照のために他のビュー中のビュー成分を使用することができる。MVCでは、たとえば、別のビュー中のビュー成分がインター予測参照であるかのように、ビュー間予測が実現される。潜在的なビュー間参照は、シーケンスパラメータセット(SPS) MVC 拡張においてシグナリングされ、インター予測またはビュー間予測参照のフレキシブルな順序付けを可能にする参照ピクチャリスト構成プロセスによって変更され得る。

【0425】

[0172] H . 2 6 4 / AVC の MVC 拡張では、一例として、ビュー間予測は視差動き補償によってサポートされ、視差動き補償は、H . 2 6 4 / AVC 動き補償のシンタックスを使用するが、異なるビュー中のピクチャが参照ピクチャとして使用されることを可能にする。2つのビューのコーディングは、MVC によってサポートされ得るが、一般に、ステレオスコピックビューと呼ばれる。MVC の利点の1つは、MVC エンコーダが3Dビデオ入力として3つ以上のビューをとらえることができ、MVC デコーダがそのようなマルチビュー表現を復号できることである。したがって、MVC デコーダを有するレンダリングデバイスは、3つ以上のビューを伴う3Dビデオコンテンツを予測することができる。

10

【0426】

[0173] MVC では、ビュー間予測(IVP)は、同じアクセスユニット(すなわち、同じ時間インスタンスを有する)内の画像間で可能になる。アクセスユニットは、一般に、共通の時間インスタンスのすべてのビュー成分(たとえば、すべてのNALユニット)を含むデータユニットである。したがって、MVCにおいて、ビュー間予測は、同じアクセスユニットにおけるピクチャ間で許可される。非ベースビューの1つの中のピクチャをコーディングするとき、ピクチャが異なるビュー中にあるが同じ時間インスタンス内にある(たとえば、同じPOC値を有し、したがって同じアクセスユニットの中にある)場合、そのピクチャは参照ピクチャリストに追加され得る。ビュー間予測の参照ピクチャは、任意のインター予測の参照ピクチャと同様に、参照ピクチャリストの任意の位置に置かれ得る。

20

【0427】

[0174] 通常、Bピクチャの第1または第2の参照ピクチャリストに対する参照ピクチャリスト構成は、2つのステップ、すなわち参照ピクチャリスト初期化と、参照ピクチャリスト並べ替え(修正)とを含む。参照ピクチャリスト初期化は、参照ピクチャメモリ(復号ピクチャバッファとしても知られる)中の参照ピクチャを、POC(ピクチャの表示順でアラインされるPicture Order Count)値の順序に基づいてリストに入れる際に従う、明確なメカニズムである。

30

【0428】

[0175] ビデオコーダは、参照ピクチャリスト並べ替えメカニズムを使用して、参照ピクチャリスト初期化中にリストに入れられたピクチャの位置を任意の新しい位置へと修正し、または参照ピクチャメモリ中の任意の参照ピクチャを、そのピクチャが初期化リストに属さなくても、任意の位置に入れる。参照ピクチャリスト並べ替え(修正)後のいくつかのピクチャは、リスト中の離れた位置に入れられる場合がある。しかしながら、ピクチャの位置が、リストのアクティブ参照ピクチャの数を超える場合、ピクチャは、最終参照ピクチャリストのエントリーとは見なされない。アクティブな参照ピクチャの数は、リストごとにスライスヘッダでシグナリングされ得る。参照ピクチャリストが構築された後(たとえば、利用可能であれば、RefPicList0 および RefPicList1)、参照ピクチャリストへの参照インデックスは、参照ピクチャリストに含まれる任意の参照ピクチャを識別するために使用され得る。

40

【0429】

[0176] 時間的動きベクトル予測子(TMVP)を得るために、まず、コロケートピクチャが識別されることになる。現在のピクチャがBスライスである場合、collocated_from_flag は、コロケートピクチャがRefPicList0 から

50

のものか RefPicList1 からのものかを示すために、スライスヘッダにおいてシグナリングされる。参照ピクチャリストが識別された後、スライスヘッダでシグナリングされる colllocated_ref_idx が、リスト中のピクチャの中のピクチャを識別するために使用される。コロケート PU が次いで、コロケートピクチャをチェックすることによって識別される。この PU を含む CU の一番右下の PU の動きと、この PU を含む CU の中心 PU 内の一番右下の PU の動きのいずれかが、使用される。上記のプロセスによって識別される動きベクトルが、AMPV またはマージモードのための動き候補を生成するために使用される場合、動きベクトルは、時間的な位置 (POC に反映される) に基づいてスケーリングされる必要がある。

【0430】

10

[0177] HEVC では、シーケンスパラメータセット (SPS) は、フラグ spp_teemporal_mv_p_enable_flag を含み、スライスヘッダは、 spp_temporal_mv_p_enable_flag が 1 に等しい場合、フラグ pic_temporal_mv_p_enable_flag を含む。ある特定のピクチャに対して pic_temporal_mv_p_enable_flag と temporal_id の両方が 0 に等しい場合、復号順序においてその特定のピクチャの前にあるピクチャからの動きベクトルは、当該特定のピクチャまたは復号順序で当該特定のピクチャの後にあるピクチャの復号において、時間的動きベクトル予測子として使用されない。

【0431】

20

[0178] 現在、Moving Pictures Experts Group (MPEG) は、HEVC に基づいて 3DV 規格を開発しており、その 3DV 規格に対して、規格化の努力の一部は、HEVC に基づくマルチビュービデオコーデックの規格化も含む。同様に、HEVC ベースの 3DV では、異なるビューから復元されたビュー成分に基づくビュー間予測がイネーブルにされる。

【0432】

30

[0179] 拡張が実際に「HLS オンリー」(ハイレベルシンタックスのみ)という要件を満たすような方式で、AVC はマルチビュー拡張によって拡張された。「HLS オンリー」という要件は、AVC 中のマクロブロックレベルにおけるモジュールが再設計される必要がなく、完全にマルチビュービデオコーディング (MVC) に対して再使用され得るように、MVCにおいてハイレベルシンタックス (HLS) の変更しか存在しないことを保証する。「HLS オンリー」という要件は、HEVC の MVC / 3DV 拡張に対して、また、マルチループ復号が受け入れ可能であると考えられる場合には、HEVC のスケーラブルビデオコーディング (SVC) 拡張に対しても満たされ得ることが可能である。

【0433】

[0180] ビュー間予測を可能にするために、HLS の変更が、ピクチャの識別という目的のために行われてよく、ここで、参照ピクチャリストの構築およびマーキングは、特定のビューの中のピクチャを識別することが可能である必要がある。

【0434】

40

[0181] HLS の変更は、H.264 / MVC における「HLS オンリー」という要件を満たすには十分ではなく、それは、ローレベルコーディングモジュールが、たとえばゼロ動き関連スケーリングを処理する状況に決して遭遇しないように、他の制約、仮定が行われるからである。そのような制約、修正、および仮定は以下の通りである。

【0435】

- ・コロケートピクチャがビュー間（のみ）参照ピクチャである場合、時間ダイレクトモードをディセーブルにする
- ・短期ではないものとしてビュー間（のみ）参照ピクチャを見なす：空間ダイレクトに関する
- ・暗黙的加重予測をディセーブルにする

[0182] 「HLS オンリー」という要件を満たすために、拡張におけるそのような修正は、ハイレベルシンタックスのみにおいて行われなければならない。したがって、スライス

50

ヘッダのもとにあるシンタックス要素に対しては修正はあるべきではなく、拡張仕様に対する C U レベルの復号プロセスの変更はない。たとえば、HEVC 拡張仕様の動きベクトル予測は、HEVC ベース仕様における動きベクトル予測と厳密に同じであるべきである。HLS の変更は、拡張仕様の規範的なデコーダの変更であるが、ベース仕様の観点からは、そのような変更は、必ずしも既知である必要はなく、義務的ではなくてよい。

【0436】

[0183]効率的なビュー間予測のような機能を可能にするために、HEVC 拡張における修正とベース仕様の両方が実装され得る。通常の復号プロセスまたは基本 HEVC デコーダのコーディング効率に影響を与えないが、拡張仕様における機能をイネーブルにすることをターゲットとするベース仕様の変更は、フックと呼ばれる。多くの場合、「HLS オンリー」という要件は、ベース仕様におけるフックと拡張仕様における HLS の変更の両方により満たされる。ベース仕様におけるフックが良好に定義されなければ、ある所望の機能が、拡張仕様においてイネーブルにされないことがあり、または、拡張仕様において多くの修正を必要とすることがある。

【0437】

[0184]HLS オンリー SVC では、場合によってはアップサンプリングおよび / またはフィルタリングの後で、ベースレイヤの表現が、現在のレイヤの現在のピクチャの参照ピクチャリストへと入れられ得る。そのようなピクチャは、レイヤ間参照ピクチャと呼ばれる。

【0438】

[0185]ベース仕様と、HLS オンリー HEVC 修正の拡張仕様との両方において、様々な修正が行われ得る。ある所望の機能を考えると、ベース仕様と拡張仕様の両方の設計が修正され得る段階において、ベース仕様の修正と拡張仕様の修正との間でトレードオフの問題がある。

【0439】

[0186]図 5 ~ 図 9 は、HLS オンリー HEVC 拡張を達成するために克服されるべき潜在的な問題を示す概念図である。図 5 は、たとえば、現在のピクチャ 100 が、様々な予測技法を使用して予測された、ブロック 102 および 104 のようなブロックを含む例を示す。具体的には、現在のピクチャ 100 は非ベースビューのピクチャに対応するが、ビュー間参照ピクチャ 110 はベースビューのピクチャである。現在のピクチャ 100 のブロック 102 は、(視差動きベクトル 106 を使用して) ビュー間参照ピクチャ 110 に対してビュー間予測されるが、ブロック 104 は、(時間的動きベクトル 108 を使用して) 同じ非ベースビューの短期 (ST) 参照ピクチャ 112 に対するインター予測を使用して予測される。したがって、図 5 は、時間的動きベクトル (時間的動きベクトル 108) とビュー間動きベクトル (視差動きベクトル、すなわち視差動きベクトル 106 とも呼ばれる) の両方を伴う隣接ブロックを現在のピクチャが含む例を示す。

【0440】

[0187]本開示は、いくつかの例では、視差動きベクトルは時間的動きベクトルを予測するためにスケーリングされるべきではないことを認識する。加えて、本開示は、いくつかの例では、時間的動きベクトルは視差動きベクトルを予測するためにスケーリングされるべきではないことも認識する。本開示は、たとえば AMVP の間に、時間的短期動きベクトルから視差動きベクトルを予測するのをディセーブルにすることと、視差動きベクトルからの時間的動きベクトルの予測をディセーブルにすることとが可能であるべきであることも認識する。視差動きベクトルは通常、異なるビューにおける同じオブジェクトの局所的な視差に対応する。しかしながら、時間的動きベクトルは通常、オブジェクトの動きに対応する。3DV 参照ソフトウェアである HTM では、上記の 2 つのカテゴリの動きベクトル間の予測はディセーブルにされる。

【0441】

[0188]図 6 は、異なるビューのビュー間参照ピクチャを使用して予測されるブロックを現在のピクチャが含む例を示す。具体的には、この例では、ビュー間参照ピクチャ 120

10

20

30

40

50

はビュー0の中にある、ビュー間参照ピクチャ122はビュー1の中にある。現在のピクチャ124はビュー2の中にある。現在のピクチャ124は、ビュー0のビュー間参照ピクチャ120とビュー1のビュー間参照ピクチャ122の両方から、ビュー間予測を使用して予測される、ブロック126、128を含む。具体的には、この例では、ブロック126はビュー間参照ピクチャ122から予測され、ブロック128はビュー間参照ピクチャ120から予測される。

【0442】

[0189]ブロック126および128は、異なる視差動きベクトルを使用して予測される。すなわち、ブロック126は視差動きベクトル130を使用して予測され、視差動きベクトル130はビュー間参照ピクチャ122の一部分を参照し、一方、ブロック128は視差動きベクトル132を使用して予測され、視差動きベクトル132はビュー間参照ピクチャ120の一部分を参照する。したがって、図6は、異なるビューのビュー間参照ピクチャを参照するビュー間動きベクトルを伴う隣接ブロックを現在のピクチャが含む例を表す。

【0443】

[0190]本開示は、2つの視差動きベクトルが同じ参照ピクチャに対応するかどうかを識別することが可能であるべきであることを認識する。RefPicList0のエントリーおよびRefPicList1のエントリーが両方ともビュー間参照ピクチャである場合、AMPの間にこれらの2つの参照ピクチャが同じかどうかを識別することが可能であるべきである。RefPicListX（「X」は、たとえば0または1という値を表し得る）がビュー間参照ピクチャである2つのエントリーを含む場合、AMPの間にこれらの2つの参照ピクチャが同じかどうかを識別することが可能であるべきである。さらに、図6に示されるように、たとえば、2つのエントリーが異なるビューに対応する場合、同じPOC値を伴う2つのエントリーは同一ではないことがある。

【0444】

[0191]図7は、ベースビューの中のビュー間参照ピクチャに対するビュー間予測と、非ベースビューの中の長期(LT)参照ピクチャに対するインター予測との両方を使用して予測されるブロックを、非ベースビューの中の現在のピクチャが含む例を示す。すなわち、図7は、現在のピクチャ140が、時間的動きベクトル152(長期参照ピクチャ144を参照する)とビュー間動きベクトル150(ビュー間参照ピクチャ142を参照する)との両方を伴う隣接ブロック146、148を含む例を示す。ビュー間動きベクトル150はまた、「視差動きベクトル150」と呼ばれ得る。本開示は、時間的動きベクトル152のような時間的長期動きベクトルから視差動きベクトル150のような視差動きベクトルを予測することをディセーブルにし、視差動きベクトルから時間的長期動きベクトルを予測することをディセーブルにすることが可能であるべきであることを認識する。

【0445】

[0192]図8は、長期(LT)参照ピクチャと短期(ST)参照ピクチャの両方からの非ベースビューのインター予測を使用して予測されるブロックを非ベースビューの中の現在のピクチャが含む例を示す。すなわち、図8は、時間的長期動きベクトルと時間的短期動きベクトルの両方を伴う隣接ブロック166、168を現在のピクチャ160が含む例を示す。具体的には、ブロック166は時間的動きベクトル170を使用して予測され、時間的動きベクトル170は長期参照ピクチャ162を参照するが、ブロック168は時間的動きベクトル172を使用して予測され、時間的動きベクトル172は短期参照ピクチャ164を参照する。したがって、時間的動きベクトル170は長期動きベクトルまたは長期時間的動きベクトルと呼ばれ、一方時間的動きベクトル172は短期動きベクトルまたは短期時間的動きベクトルと呼ばれ得る。本開示は、たとえばAMPの間の、時間的短期動きベクトルと時間的長期動きベクトルとの間の予測をディセーブルにすることが可能であるべきであることを認識する。

【0446】

[0193]図9は、インター予測を使用して予測されるブロックを非ベースビューの中の現

10

20

30

40

50

在のピクチャが含み、非ベースビューの異なる長期(L T)参照ピクチャに対してブロックが予測される例を示す。すなわち、図9は、長期ピクチャ184、182をそれぞれ参照する時間的動きベクトル190、192を伴う隣接ブロック186、188を現在のピクチャ180が含む例を示す。具体的には、この例では、ブロック186は時間的動きベクトル190を使用して予測され、時間的動きベクトル190は長期参照ピクチャ184の一部分を参照し、一方、ブロック188は時間的動きベクトル192を使用して予測され、時間的動きベクトル192は長期参照ピクチャ182の一部分を指す。本開示は、AMVPの間の、時間的長期動きベクトルの予測をイネーブルおよび／またはディセーブルにすることが可能であるべきであることを認識する。

【0447】

10

[0194]図10は、現在のブロックに対する隣接ブロックの例示的なセットを示す概念図である。具体的には、この例では、現在のブロックは、A0およびA1とラベルされた左側の隣接ブロックと、上側の隣接ブロックB0、B1、およびB2とを有する。現在のブロックは、インター予測、たとえば時間的予測またはビュー間予測を使用してコーディングされ得る。したがって、ビデオエンコーダ20またはビデオデコーダ30のようなビデオコーダは、動きベクトルを使用して現在のブロックをコーディングし得る。その上、ビデオコーダは動きベクトルをコーディングし得る。様々な例において、ビデオコーダは、たとえば、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMVP)、またはマージモードのために、上で説明される技法を使用して現在のブロックに対する動きベクトルをコーディングし得る。TMVP予測子は、以前にコーディングされたピクチャの中の現在のブロックと同じ位置にある(co-located)ブロックに対する動きベクトルに対応し得る。

20

【0448】

[0195]隣接ブロックA0、A1、B0、B1、およびB2の1つまたは複数の動きベクトルは、現在のブロックをコーディングするために使用される動きベクトルとは異なるタイプであり得る。たとえば、現在のブロックは長期動きベクトルを使用してコーディングされてよく、一方ブロックA0、A1、B0、B1、およびB2の1つまたは複数は短期動きベクトルを使用してコーディングされてよい。別の例として、現在のブロックは短期動きベクトルを使用してコーディングされてよく、一方ブロックA0、A1、B0、B1、およびB2の1つまたは複数は長期動きベクトルを使用してコーディングされてよい。さらに別の例として、現在のブロックは視差動きベクトルを使用してコーディングされてよく、一方ブロックA0、A1、B0、B1、およびB2の1つまたは複数は時間的動きベクトルを使用してコーディングされてよい。さらに別の例として、現在のブロックは時間的動きベクトルを使用してコーディングされてよく、一方ブロックA0、A1、B0、B1、およびB2の1つまたは複数は視差動きベクトルを使用してコーディングされてよい。そのような場合、上で説明されたように、ビデオエンコーダ20またはビデオデコーダ30のようなビデオコーダは、異なるタイプの動きベクトルの間での動きベクトル予測をディセーブルにし得る。

30

【0449】

40

[0196]図10の例は、空間動きベクトル予測子の候補を示す。しかしながら、時間的動きベクトル予測(TMVP)のために時間的動きベクトル予測子の候補も考慮され得ることを理解されたい。そのようなTMVPの候補は、以前にコーディングされたピクチャ内のコロケートブロック、すなわち、図10で「現在のブロック」とラベルされたブロックと同じ位置にあるブロックに対する動き情報に対応し得る。加えて、本開示の技法によれば、TMVPの候補は、TMVPの候補の動き情報および現在のブロックの動きベクトルが異なるタイプのピクチャ、たとえば、短期参照ピクチャおよび長期参照ピクチャを指す場合、動きベクトル予測子として使用することが不可能であると見なされ得る。

【0450】

50

[0197]図11は、本開示の技法による、ビデオデータを符号化するための例示的な方法を示すフローチャートである。いくつかの例では、図11の例示的な方法のステップは、

代替的に、異なる順序で、または実質的に並列に実行され得る。同様に、いくつかのステップが省略されてよく、かつ／または他のステップが追加されてよい。ビデオエンコーダ20によって実行されるものとして説明されるが、他のビデオ符号化デバイスが実質的に同様の方法を実行するように構成され得ることを理解されたい。

【0451】

[0198]この例では、ビデオエンコーダ20は、現在のピクチャに対する参照ピクチャのピクチャ順序カウント（P O C）値を符号化する（200）。たとえば、ビデオエンコーダ20は、現在のピクチャを含むシーケンスに対するシーケンスパラメータセット（S P S）データ構造の中のいくつかの参照ピクチャに対するP O C値、またはP O C値を表すデータ（最下位ビット（L S B s）など）を符号化し得る。ビデオエンコーダ20はまた、加えて、または代替的に、現在のピクチャの現在のスライスのスライスヘッダ中の1つまたは複数の参照ピクチャに対するP O C値を符号化し得る。いくつかの例では、ビデオエンコーダ20は、S P S中の長期参照ピクチャのP O C値とスライスヘッダ中の短期参照ピクチャのP O C値とを表すデータを符号化し得る。ビデオエンコーダ20はまた、S P S中、スライスヘッダ中、または別の場所にあるビュー間参照ピクチャのP O C値を符号化し得る。一般に、ビュー間参照ピクチャのP O C値は、符号化されている現在のピクチャのP O C値と同じである。10

【0452】

[0199]ビデオエンコーダ20はまた、参照ピクチャの第2次元の識別子を符号化し得る（202）。第2次元の識別子は、参照ピクチャを含むビューのビュー識別子、参照ピクチャを含むビューのビュー順序インデックス、ビュー順序インデックスと深度フラグの組合せ、参照ピクチャを含むスケーラブルビデオコーディング（S V C）レイヤのレイヤ識別子、および／または一般的なレイヤ識別子のうちの1つまたは複数を含み得る。このようにして、参照ピクチャに対するP O C値と参照ピクチャの第2次元の識別子との組合せが、参照ピクチャを識別するために使用され得る。20

【0453】

[0200]ビデオエンコーダ20はさらに、現在のピクチャの現在のブロックに対する動き探索を実行し得る。すなわち、動き推定ユニット42は、現在のブロックに最も近く一致する参照ブロックに対する参照ピクチャを探索し得る。このことは、当該参照ブロックを参照し、また、当該参照ブロックが現れる参照ピクチャも参照する動きベクトルを含む動き情報をもたらし得る。したがって、ビデオエンコーダ20の動き補償ユニット44は、参照ピクチャの1つを指す動きベクトルを使用して、現在のブロックを予測し得る（204）。

【0454】

[0201]ビデオエンコーダ20はまた、たとえば、発展型動きベクトル予測（A M V P）、時間的動きベクトル予測（T M V P、）またはマージモードを使用して、動きベクトルを符号化し得る。具体的には、ビデオエンコーダ20は、利用可能な候補動きベクトル予測子のセットを決定し得る（206）。たとえば、図10を参照すると、ビデオエンコーダ20は、隣接ブロックA0、A1、B0、B1、およびB2に対する動きベクトルが利用可能かどうかを決定し得る。具体的には、本開示の技法によれば、ビデオエンコーダ20は、隣接ブロックの動きベクトルが現在のブロックに対する動きベクトルとは異なるタイプである場合、これらの隣接ブロックの1つの動きベクトルが利用可能ではないと決定し得る。同様に、ビデオエンコーダ20は、T M V P候補が現在のブロックの動きベクトルをコーディングするための予測子として使用されることが可能であるかどうかを決定する際、時間的動きベクトル予測子の候補についての動きベクトルが現在のブロックについての動きベクトルとは異なるタイプの参照ピクチャを参照しているかどうかを決定し得る。

【0455】

[0202]上で説明されたように、例示的な異なるタイプの動きベクトルは、長期動きベクトルと、短期動きベクトルと、時間的動きベクトルと、視差動きベクトルとを含む。した4050

がって、ビデオエンコーダ 20 は、現在のブロックの動きベクトルのタイプ、さらには、隣接ブロックの動きベクトルのタイプを決定し、現在のブロックの現在の動きベクトルのタイプとは異なるタイプの隣接ブロックの動きベクトルが現在の動きベクトルに対する動きベクトル予測子として使用されることは不可能であると決定し得る。タイプを決定するために、ビデオエンコーダ 20 は、候補動きベクトルが参照する参照ピクチャの POC 値、現在の動きベクトルが参照する参照ピクチャの POC 値、候補動きベクトルが参照する参照ピクチャの第 2 次元識別子、および / または、現在の動きベクトルが参照する参照ピクチャの第 2 次元識別子を参照し得る。

【 0 4 5 6 】

[0203] 続いて、ビデオエンコーダ 20 は、現在の動きベクトルに対する動きベクトル予測子として、隣接ブロック（これは、以前にコーディングされたピクチャ中のコロケートブロックおよび / または異なるビューのピクチャ中の対応するブロックを含み得る）から利用可能な候補動きベクトル予測子の 1 つを選択し得る（208）。ビデオエンコーダ 20 は次いで、選択された動きベクトル予測子を使用して現在の動きベクトルを符号化し得る（210）。

【 0 4 5 7 】

[0204] さらに、ビデオエンコーダ 20 は、現在のブロックに対する残差ブロックを計算することができる（212）。図 2 に関して説明されたように、加算器 50 は、元のコーディングされていないブロックと、動き補償ユニット 44 によって形成された予測ブロックとのピクセルごとの差分を計算し得る。変換処理ユニット 52、量子化ユニット 54、およびエントロピー符号化ユニット 56 は次いで、それぞれ、残差ブロックを変換し、量子化し、走査し得る（214）。具体的には、変換処理ユニット 52 は、残差ブロックを変換して変換係数のブロックを生成し、量子化ユニット 52 は変換係数を量子化し、エントロピー符号化ユニット 56 は量子化変換係数を走査し得る。エントロピー符号化ユニット 56 は次いで、量子化変換係数と、符号化動きベクトル情報とをエントロピー符号化し得る（216）。

【 0 4 5 8 】

[0205] このようにして、図 11 の方法は、ビデオデータの第 1 のピクチャについてのピクチャ順序カウント（POC）値を符号化することと、第 1 のピクチャの第 2 次元ピクチャ識別子を符号化することと、ベースビデオコーディング仕様（またはベースビデオコーディング仕様の拡張）に従って、第 1 のピクチャの POC 値および第 2 次元ピクチャ識別子に少なくとも一部基づいて、第 2 のピクチャを符号化することとを含む、方法の例を表す。加えて、方法は、

第 2 のピクチャの第 1 のブロックの第 1 の動きベクトルと（ここで前記第 1 の動きベクトルは短期参照ピクチャを参照し）、第 2 のピクチャの第 2 のブロックの第 2 の動きベクトル（ここで第 2 の動きベクトルは長期参照ピクチャを参照し）との間の動きベクトル予測をディセーブルにすることを含み得る。加えて、または代替的に、方法は、第 2 のピクチャの第 1 のブロックの第 1 の動きベクトル（ここで前記第 1 の動きベクトルはビュー間参照ピクチャを参照し）と、第 2 のピクチャの第 2 のブロックの第 2 の動きベクトル（ここで前記第 2 の動きベクトルは時間的参照ピクチャを参照し）との間の動きベクトル予測をディセーブルにすることを含み得る。

【 0 4 5 9 】

[0206] 図 12 は、本開示の技法による、ビデオデータを復号するための例示的な方法を示すフローチャートである。いくつかの例では、図 12 の例示的な方法のステップは、代替的に、異なる順序で、または実質的に並列に実行され得る。同様に、いくつかのステップが省略されてよく、かつ / または他のステップが追加されてよい。ビデオデコーダ 30 によって実行されるものとして説明されるが、他のビデオ復号デバイスが実質的に同様の方法を実行するように構成され得ることを理解されたい。

【 0 4 6 0 】

[0207] この例では、ビデオデコーダ 30 は、現在のピクチャに対する参照ピクチャの P

10

20

30

40

50

OC 値を復号する (230)。たとえば、ビデオデコーダ30は、現在のピクチャを含むシーケンスに対するシーケンスパラメータセット (SPS) データ構造の中のいくつかの参照ピクチャに対する POC 値、または POC 値を表すデータ (最下位ビット (LSBs) など) を復号し得る。ビデオデコーダ30は、たとえば以前に復号された完全な POC 値から導出されたそれぞれの MSB に LSB を付加することによって、POC 値に対する復号された LSB から当該 POC 値を復元し得る。ビデオデコーダ30はまた、加えて、または代替的に、現在のピクチャの現在のスライスのスライスヘッダ中の 1つまたは複数の参照ピクチャに対する POC 値を復号し得る。いくつかの例では、ビデオデコーダ30は、SPS 中の長期参照ピクチャの POC 値と、スライスヘッダ中の短期参照ピクチャの POC 値とを表すデータを復号し得る。ビデオデコーダ30はまた、SPS 中、スライスヘッダ中、または別の場所にあるビュー間参照ピクチャの POC 値を復号し得る。一般に、ビュー間参照ピクチャの POC 値は、符号化されている現在のピクチャの POC 値と同じである。

【0461】

[0208] ビデオエンコーダ30はまた、参照ピクチャの第2次元識別子を復号し得る (232)。第2次元識別子は、参照ピクチャを含むビューのビュー識別子、参照ピクチャを含むビューのビュー順序インデックス、ビュー順序インデックスと深度フラグの組合せ、参照ピクチャを含むスケーラブルビデオコーディング (SVC) レイヤのレイヤ識別子、および / または一般的なレイヤ識別子のうちの 1つまたは複数を含み得る。このようにして、参照ピクチャについての POC 値と参照ピクチャの第2次元識別子との組合せが、参照ピクチャを識別するために使用され得る。したがって、参照ピクチャを識別するために、動き情報は、参照ピクチャについての POC 値と第2次元識別子との両方を含み得る。

【0462】

[0209] ビデオエンコーダ30はまた、現在のピクチャの現在のブロックに対する動きベクトルを復号し得る。具体的には、ビデオデコーダ30は、利用可能な候補動きベクトル予測子のセットを決定し得る (234)。たとえば、図10を参照すると、ビデオデコーダ30は、隣接ブロック A0、A1、B0、B1、および B2 についての動きベクトルが利用可能かどうかを決定し得る。具体的には、本開示の技法によれば、ビデオデコーダ30は、隣接ブロックの動きベクトルが現在のブロックに対する動きベクトルとは異なるタイプである場合、これらの隣接ブロックの 1つの動きベクトルが利用可能ではないと決定し得る。同様に、ビデオデコーダ30は、TMVP 候補が現在のブロックの動きベクトルをコーディングするための予測子として使用されることが可能であるかどうかを決定する際、時間的動きベクトル予測子の候補についての動きベクトルが現在のブロックについての動きベクトルとは異なるタイプの参照ピクチャを参照しているかどうかを決定し得る。

【0463】

[0210] 上で説明されたように、例示的な異なるタイプの動きベクトルは、長期動きベクトルと、短期動きベクトルと、時間的動きベクトルと、視差動きベクトルとを含む。したがって、ビデオデコーダ30は、現在のブロックの動きベクトルのタイプ、さらには、隣接ブロックの動きベクトルのタイプを決定し、現在のブロックの動きベクトルのタイプとは異なるタイプの隣接ブロックの動きベクトルが現在の動きベクトルに対する動きベクトル予測子として使用されることは不可能であると決定し得る。タイプを決定するために、ビデオデコーダ30は、候補動きベクトルが参照する参照ピクチャの POC 値、現在の動きベクトルが参照する参照ピクチャの POC 値、候補動きベクトルが参照する参照ピクチャの第2次元識別子、および / または、現在の動きベクトルが参照する参照ピクチャの第2次元識別子を参照し得る。

【0464】

[0211] 続いて、ビデオデコーダ30は、現在の動きベクトルに対する動きベクトル予測子として、隣接ブロック (これは、以前にコーディングされたピクチャ中のコロケートブロックおよび / または異なるビューのピクチャ中の対応するブロックを含み得る) から利用可能な候補動きベクトル予測子の 1つを選択し得る (236)。ビデオデコーダ30は

10

20

30

40

50

次いで、選択された動きベクトル予測子を使用して現在の動きベクトルを復号し得る(238)。たとえば、AMVPを使用して、ビデオデコーダ30は、現在の動きベクトルについての動きベクトル差分(MVD)値を復号し、次いで、選択された動きベクトル予測子にMVD値を適用し得る。すなわち、ビデオデコーダ30は、MVD値のx成分を選択された動きベクトル予測子のx成分に加え、MVD値のy成分を選択された動きベクトル予測子のy成分に加え得る。

【0465】

[0212]ビデオデコーダ30の動き補償ユニット72は次いで、参照ピクチャの1つを指す動きベクトルを使用して、現在のブロックを予測し得る(240)。すなわち、動きベクトル自体に加えて、ビデオデコーダ30は、POC値および第2次元の識別値のような、動きベクトルが対応するブロックについての参照ピクチャ識別情報を復号し得る。このようにして、ビデオデコーダ30は、POC値と第2次元の識別値とを使用して、動きベクトルが指す参照ピクチャを決定し得る。したがって、動き補償ユニット72は、動きベクトルと、参照ピクチャ識別情報、すなわちPOC値および第2次元の識別値とを使用して、現在のブロックに対する予測ブロックを形成し得る。

【0466】

[0213]エントロピー復号ユニット70はさらに、現在のブロックに対応する残差ブロックについての量子化残差係数をエントロピー復号し得る(242)。エントロピー復号ユニット70、逆量子化ユニット76、および逆変換ユニット78はそれぞれ、量子化変換係数を逆走査、量子化、変換して、残差ブロックを再現し得る(244)。ビデオデコーダ30の加算器80は次いで、予測ブロックと残差ブロックとを組み合わせて(すなわち、ピクセルごとに加算して)、現在のブロックを再現し得る(246)。

【0467】

[0214]このようにして、図12の方法は、ビデオデータの第1のピクチャについてのピクチャ順序カウント(POC)値を復号することと、第1のピクチャの第2次元ピクチャ識別子を復号することと、ベースビデオコーディング仕様(またはベースビデオコーディング仕様の拡張)に従って、第1のピクチャのPOC値および第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを復号することとを含む、方法の例を表す。加えて、方法は、短期参照ピクチャを指す、第2のピクチャの第1のブロックの第1の動きベクトルと、長期参照ピクチャを指す、第2のピクチャの第2のブロックの第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることを含み得る。加えて、または代替的に、方法は、ビュー間参照ピクチャを指す、第2のピクチャの第1のブロックの第1の動きベクトルと、時間的参照ピクチャを指す、第2のピクチャの第2のブロックの第2の動きベクトルとの間の動きベクトル予測をディセーブルにすることを含み得る。

【0468】

[0215]例によっては、本明細書で説明された技法のうちいずれかの、いくつかの行為またはイベントは、異なる順番で実行される可能性があり、追加され、統合され、または完全に除外され得る(たとえば、すべての説明された行為またはイベントが、本技法の実施のために必要であるとは限らない)ことを認識されたい。さらに、いくつかの例では、行為またはイベントは、連続的にではなく、同時に、たとえば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて実行され得る。

【0469】

[0216]1つまたは複数の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、1つまたは複数の命令またはコードとしてコンピュータ可読媒体上に記憶されてよく、あるいは、コンピュータ可読媒体を介して送信され、ハードウェアベースの処理ユニットによって実行されてよい。コンピュータ可読媒体は、たとえば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を支援する、任意の媒体を含むデータ記憶媒体または通信媒体などの有形媒体に対応するコンピュータ可読記憶媒体を含み得る。このようにして、コンピュータ可読媒体は、一般に、(1)非一

10

20

30

40

50

時的である有形コンピュータ可読記憶媒体、あるいは(2)信号または搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明された技法の実装のための命令、コードおよび／またはデータ構造を取り出すために、1つまたは複数のコンピュータあるいは1つまたは複数のプロセッサによってアクセスされ得る任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【0470】

[0217]限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM(登録商標)、CD-ROMまたは他の光ディスクストレージ、磁気ディスクストレージ、または他の磁気ストレージデバイス、フラッシュメモリ、あるいは、命令またはデータ構造の形態の所望のプログラムコードを記憶するために使用されコンピュータによってアクセスされ得る、任意の他の媒体を備え得る。同様に、いかなる接続も適切にコンピュータ可読媒体と称される。たとえば、命令が、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的媒体を含まないが、代わりに非一時的有形記憶媒体を対象とすることを理解されたい。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)およびブルーレイディスク(disc)を含み、ディスク(disk)は、通常、データを磁気的に再生し、ディスク(disc)は、データをレーザで光学的に再生する。上記の組合せもコンピュータ可読媒体の範囲内に含まれるべきである。
10

【0471】

[0218]命令は、1つまたは複数のデジタル信号プロセッサ(DSP)などの1つまたは複数のプロセッサ、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、あるいは他の等価な集積回路またはディスクリート論理回路によって実行され得る。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造、または本明細書で説明される技法の実装に好適な他の構造のいずれかを指す。加えて、いくつかの態様では、本明細書で説明された機能は、符号化および復号のために構成された専用のハードウェアおよび／またはソフトウェアモジュール内で与えられてよく、あるいは複合コーデックに組み込まれてよい。また、本技法は、1つまたは複数の回路または論理要素中で完全に実装され得る。
20

【0472】

[0219]本開示の技法は、ワイヤレスハンドセット、集積回路(IC)、またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置において実装され得る。本開示では、開示される技法を実行するように構成されたデバイスの機能的態様を強調するために、様々なコンポーネント、モジュール、またはユニットが説明されたが、それらのコンポーネント、モジュール、またはユニットは、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。むしろ、上で説明されたように、様々なユニットが、好適なソフトウェアおよび／またはファームウェアとともに、上で説明された1つまたは複数のプロセッサを含めて、コーデックハードウェアユニットにおいて組み合わせられてよく、または相互動作ハードウェアユニットの集合によって与えられてよい。
40

【0473】

[0220]様々な例が説明されてきた。これらおよび他の例は以下の特許請求の範囲内に入る。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

[1] ビデオデータを復号する方法であって、

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を復号することと、

前記第1のピクチャについて第2次元ピクチャ識別子を復号することと、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第2のピクチャを復号することと、
を備える方法。

[2] 前記第2のピクチャの第1のブロックの第1の動きベクトルと、ここで前記第1の動きベクトルは短期参照ピクチャを参照し、

前記第2のピクチャの第2のブロックの第2の動きベクトルと、ここで前記第2の動きベクトルは長期参照ピクチャを参照し、

の間の動きベクトル予測をディセーブルにすることをさらに備える、[1]に記載の方法。

10

[3] 前記第2のピクチャをコーディングすることは、

前記POC値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別することと、

前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分を復号することと、
を備える、[1]に記載の方法。

[4] 前記第1のピクチャを識別することは、前記第2のピクチャのブロックについての動きベクトルの復号の間に前記第1のピクチャを識別することを備え、前記動きベクトルのコーディングは、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMVP)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルをコーディングすることを備える、[3]に記載の方法。

20

[5] 前記第2のピクチャの第1の短期動きベクトルと前記第2のピクチャの第2の短期動きベクトルとの間の予測をイネーブルにすることと、

前記第1の短期動きベクトルによって参照される第1の短期参照ピクチャについてのPOC値と、前記第2の短期動きベクトルによって参照される第2の短期参照ピクチャについてのPOC値とに基づいて、前記第1の短期動きベクトルと前記第2の短期動きベクトルとのうちの少なくとも1つをスケーリングすることと、

をさらに備える、[1]に記載の方法。

30

[6] 前記ビュー成分がビュー間予測に使用されるかどうかに少なくとも一部基づいて、前記ビデオデータのビュー成分が長期参照ピクチャを備えるかどうかを示す値を復号することをさらに備える、[1]に記載の方法。

[7] 前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第3のピクチャを復号することをさらに備える、[1]に記載の方法。

[8] 前記第3のピクチャを復号する前に、前記第1のピクチャを含むすべてのビュー間参照ピクチャを、長期参照ピクチャとしてマークすることをさらに備える、[7]に記載の方法。

[9] 前記第3のピクチャについて前記ビュー間参照ピクチャの各々についての状態を、前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前に記憶することと、
ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、
のうちの1つを備え、前記ビュー間参照ピクチャは前記第1のピクチャを含み、

40

前記第2のピクチャをコーディングした後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々の新たな状態を設定することと、

をさらに備える、[8]に記載の方法。

[10] 前記ベースビデオコーディング仕様は高効率ビデオコーディング(HEVC)ベース仕様を備え、前記ベースビデオコーディング仕様への前記拡張は、前記HEVCベース仕様に対するスケーラブルビデオコーディング(SVC)拡張と、前記HEVCベース仕様に対するマルチビュービデオコーディング(MVC)拡張とのうちの1つを備え

50

る、[7] に記載の方法。

[1 1] 前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング(SVC)レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少なくとも1つを備える、[7] に記載の方法。

[1 2] 前記第3のピクチャを復号した後、各ビュー間参照ピクチャを、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つとしてマークすることをさらに備える、[7] に記載の方法。

[1 3] ビュー間参照ピクチャを長期参照ピクチャとしてマークした後で、現在使用されていない新たなPOC値を前記ビュー間参照ピクチャに割り当てることと、
10

前記第2のピクチャを復号した後、前記ビュー間参照ピクチャについての元のPOC値をリストアすることと、

をさらに備える、[1 2] に記載の方法。

[1 4] 前記元のPOC値は、前記第1のピクチャの前記POC値を備える、[1 3] に記載の方法。

[1 5] ビデオデータを符号化する方法であって、
ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を符号化することと、

前記第1のピクチャについて第2次元ピクチャ識別子を符号化することと、
20

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第2のピクチャを符号化することと、
を備える方法。

[1 6] 前記第2のピクチャの第1のブロックの第1の動きベクトルと、ここで前記第1の動きベクトルは短期参照ピクチャを参照し、

前記第2のピクチャの第2のブロックの第2の動きベクトルと、ここで前記第2の動きベクトルは長期参照ピクチャを参照し、

の間の動きベクトル予測をディセーブルにすることをさらに備える、[1 5] に記載の方法。

[1 7] 前記POC値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別することと、
30

前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分をコーディングすることと、

をさらに備える、[1 5] に記載の方法。

[1 8] 前記第1のピクチャを識別することは、前記第2のピクチャのブロックについての動きベクトルのコーディングの間に前記第1のピクチャを識別することを備え、前記動きベクトルのコーディングは、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMVP)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルをコーディングすることを備える、[1 7] に記載の方法。

[1 9] 前記第2のピクチャの第1の短期動きベクトルと前記第2のピクチャの第2の短期動きベクトルとの間の予測をイネーブルにすることと、
40

前記第1の短期動きベクトルによって参照される第1の短期参照ピクチャについてのPOC値と、前記第2の短期動きベクトルによって参照される第2の短期参照ピクチャについてのPOC値とに基づいて、前記第1の短期動きベクトルと前記第2の短期動きベクトルとのうちの少なくとも1つをスケーリングすることと、

をさらに備える、[1 5] に記載の方法。

[2 0] 前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング(SVC)レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少
50

なくとも1つを備える、[15]に記載の方法。

[21] 前記ビュー成分がビュー間予測に使用されるかどうかに少なくとも一部基づいて、前記ビデオデータのビュー成分が長期参照ピクチャを備えるかどうかを示す値をコーディングすることをさらに備える、[15]に記載の方法。

[22] 前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第3のピクチャを符号化することをさらに備える、[15]に記載の方法。

[23] 前記第3のピクチャを符号化する前に、すべてのビュー間参照ピクチャを、長期参照ピクチャとしてマークすることをさらに備える、[22]に記載の方法。

[24] 前記ビュー間参照ピクチャの各々の状態を、前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前に記憶することと、ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つを備え、

前記第2のピクチャをコーディングした後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々の新たな状態を設定することと、

をさらに備える、[23]に記載の方法。

[25] 前記ベースビデオコーディング仕様は高効率ビデオコーディング(HEVC)ベース仕様を備え、前記ベースビデオコーディング仕様への前記拡張は、前記HEVCベース仕様に対するスケーラブルビデオコーディング(SVC)拡張と、前記HEVCベース仕様に対するマルチビュービデオコーディング(MVC)拡張との1つを備える、[22]に記載の方法。

[26] 前記第3のピクチャをコーディングした後、各ビュー間参照ピクチャを、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちの1つとしてマークすることをさらに備える、[15]に記載の方法。

[27] ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たなPOC値を前記ビュー間参照ピクチャに割り当てるのことと、

前記第2のピクチャをコーディングした後、前記ビュー間参照ピクチャについての元のPOC値をリストアすることと、

をさらに備える、[26]に記載の方法。

[28] 前記元のPOC値は、前記第2のピクチャの前記POC値を備える、[27]に記載の方法。

[29] ビデオデータを復号するためのデバイスであって、
ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を復号し、前記第1のピクチャについて第2次元ピクチャ識別子を復号し、ベースビデオコーディング仕様に従って、前記第1のピクチャの前記POC値および前記第2次元ピクチャ識別子に少なくとも一部基づいて、第2のピクチャを復号するように構成されるビデオデコーダを備える、デバイス。

[30] 前記ビデオデコーダは、

前記第2のピクチャの第1のブロックの第1の動きベクトルと、ここで前記第1の動きベクトルは短期参照ピクチャを参照し、

前記第2のピクチャの第2のブロックの第2の動きベクトルと、ここで前記第2の動きベクトルは長期参照ピクチャを参照し、

の間の動きベクトル予測をディセーブルにするように構成される、[29]に記載のデバイス。

[31] 前記ビデオデコーダは、前記POC値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別し、前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分を復号するように構成される、[29]に記載のデバイス。

[32] 前記ビデオデコーダは、前記第2のピクチャのブロックについての動きベクトルのコーディングの間に前記第1のピクチャを識別するように構成され、前記ビデオデコーダは、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMVP)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルを復号するよう

10

20

30

40

50

に構成される、[31] に記載のデバイス。

[33] 前記ビデオデコーダは、前記第2のピクチャの第1の短期動きベクトルと前記第2のピクチャの第2の短期動きベクトルとの間の予測をイネーブルにし、前記第1の短期動きベクトルによって参照される第1の短期参照ピクチャについての POC 値と、前記第2の短期動きベクトルによって参照される第2の短期参照ピクチャについての POC 値とに基づいて、前記第1の短期動きベクトルと前記第2の短期動きベクトルとのうちの少なくとも1つをスケーリングするように構成される、[29] に記載のデバイス。

[34] 前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング (SVC) レイヤのレイヤ識別子、および一般的なレイヤ識別子の少なくとも1つを備える、[29] に記載のデバイス。

[35] 前記ビデオデコーダは、前記ビュー成分がビュー間予測に使用されるかどうかに少なくとも一部基づいて、前記ビデオデータのビュー成分が長期参照ピクチャを備えるかどうかを示す値を復号するように構成される、[29] に記載のデバイス。

[36] 前記ビデオデコーダはさらに、前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記 POC 値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第3のピクチャを復号するように構成される、[29] に記載のデバイス。

[37] 前記ビデオデコーダは、

前記第3のピクチャを復号する前に、前記第1のピクチャを含む前記第3のピクチャについてのすべてのビュー間参照ピクチャを長期参照ピクチャとしてマークし、

前記ビュー間参照ピクチャの各々の状態を記憶し、ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前は参照に使用されない、のうちの1つを備え、

前記第3のピクチャを復号した後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々についての新たな状態を設定するように構成される、[36] に記載のデバイス。

[38] 前記ビデオデコーダはさらに、

前記第1のピクチャを含む前記第3のピクチャについての各ビュー間参照ピクチャを、長期参照ピクチャ、短期参照ピクチャ、および前記第3のピクチャを復号した後は参照に使用されない、のうちの1つとしてマークし、

ビュー間参照ピクチャを長期参照ピクチャとしてマークした後、現在使用されていない新たな POC 値を、前記ビュー間参照ピクチャの各々に割り当て、

前記第2のピクチャをコーディングした後、前記ビュー間参照ピクチャについての元の POC 値をリストアするように構成される、[36] に記載のデバイス。

[39] 集積回路と、

マイクロプロセッサと、

前記ビデオデコーダを含むワイヤレス通信デバイスと、

のうちの少なくとも1つを備える、[29] に記載のデバイス。

[40] ビデオデータを符号化するためのデバイスであって、

ビデオデータの第1のピクチャについてピクチャ順序カウント (POC) 値を符号化し、前記第1のピクチャについて第2次元ピクチャ識別子を符号化し、ベースビデオコーディング仕様に従って、前記第1のピクチャの前記 POC 値および前記第2次元ピクチャ識別子とに少なくとも一部基づいて、第2のピクチャを符号化するように構成されるビデオエンコーダを備える、デバイス。

[41] 前記ビデオエンコーダは、

前記第2のピクチャの第1のブロックの第1の動きベクトルと、ここで前記第1の動きベクトルは短期参照ピクチャを参照し、

前記第2のピクチャの第2のブロックの第2の動きベクトルと、ここで前記第2の動き

10

20

30

40

50

ベクトルは長期参照ピクチャを参照し、

の間の動きベクトル予測をディセーブルするように構成される、[40]に記載のデバイス。

[42] 前記ビデオエンコーダは、前記POC値と前記第2次元ピクチャ識別子とを使用して前記第1のピクチャを識別し、前記第1のピクチャに対して前記第2のピクチャの少なくとも一部分を符号化するように構成される、[40]に記載のデバイス。

[43] 前記ビデオエンコーダは、前記第2のピクチャのブロックについての動きベクトルのコーディングの間に前記第1のピクチャを識別するように構成され、前記ビデオエンコーダは、発展型動きベクトル予測(AMVP)、時間的動きベクトル予測(TMVP)、およびマージモードのうちの少なくとも1つに従って前記動きベクトルを符号化するように構成される、[42]に記載のデバイス。
10

[44] 前記ビデオエンコーダは、前記第2のピクチャの第1の短期動きベクトルと前記第2のピクチャの第2の短期動きベクトルとの間の予測をイネーブルにし、前記第1の短期動きベクトルによって参照される第1の短期参照ピクチャについてのPOC値と、前記第2の短期動きベクトルによって参照される第2の短期参照ピクチャについてのPOC値とに基づいて、前記第1の短期動きベクトルと前記第2の短期動きベクトルとのうちの少なくとも1つをスケーリングするように構成される、[40]に記載のデバイス。

[45] 前記第2次元ピクチャ識別子は、前記第1のピクチャを含むビューのビュー識別子、前記第1のピクチャを含む前記ビューのビュー順序インデックス、前記ビュー順序インデックスと深度フラグとの組合せ、前記第1のピクチャを含むスケーラブルビデオコーディング(SVC)レイヤのレイヤ識別子、および一般的なレイヤ識別子のうちの少なくとも1つを備える、[40]に記載のデバイス。
20

[46] 前記ビデオエンコーダは、前記ビュー成分がビュー間予測に使用されるかどうかに少なくとも一部基づいて、前記ビデオデータのビュー成分が長期参照ピクチャを備えるかどうかを示す値を符号化するように構成される、[40]に記載のデバイス。

[47] 前記ビデオエンコーダは、さらに、前記ベースビデオコーディング仕様への拡張に従って、前記第1のピクチャの前記POC値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第3のピクチャを符号化するように構成される、[40]に記載のデバイス。

[48] 前記ビデオエンコーダは、
30

前記第3のピクチャを符号化する前に、前記第1のピクチャを含む前記第3のピクチャについてのすべてのビュー間参照ピクチャを長期参照ピクチャとしてマークし、

前記ビュー間参照ピクチャの各々の状態を記憶し、ここで前記状態は、長期参照ピクチャ、短期参照ピクチャ、および前記ビュー間参照ピクチャを長期参照ピクチャとしてマークする前は参照に使用されない、のうちの1つを備え、

前記第3のピクチャを符号化した後、前記記憶された状態に基づいて、前記ビュー間参照ピクチャの各々について新たな状態を設定するように構成される、[47]に記載のデバイス。

[49] 前記ビデオエンコーダは、さらに、

前記第1のピクチャを含む前記第3のピクチャについての各ビュー間参照ピクチャを、长期参照ピクチャ、短期参照ピクチャ、および前記第3のピクチャを符号化した後は参照に使用されない、のうちの1つとしてマークし、
40

ビュー間参照ピクチャを長期参照ピクチャとして標識した後、現在使用されていない新たなPOC値を前記ビュー間参照ピクチャの各々に割り当て、

前記第2のピクチャをコーディングした後、前記ビュー間参照ピクチャについての元のPOC値をリストアするように構成される、[47]に記載のデバイス。

[50] ビデオデータを符号化するためのデバイスであって、

ビデオデータの第1のピクチャについてピクチャ順序カウント(POC)値を符号化する手段と、

前記第1のピクチャについて第2次元ピクチャ識別子を符号化する手段と、
50

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記P O C値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第2のピクチャを符号化する手段と、
を備えるデバイス。

【51】命令を記憶したコンピュータ可読記憶媒体であって、前記命令は、実行されると、プロセッサに、

ビデオデータの第1のピクチャについてピクチャ順序カウント(P O C)値を復号させ、

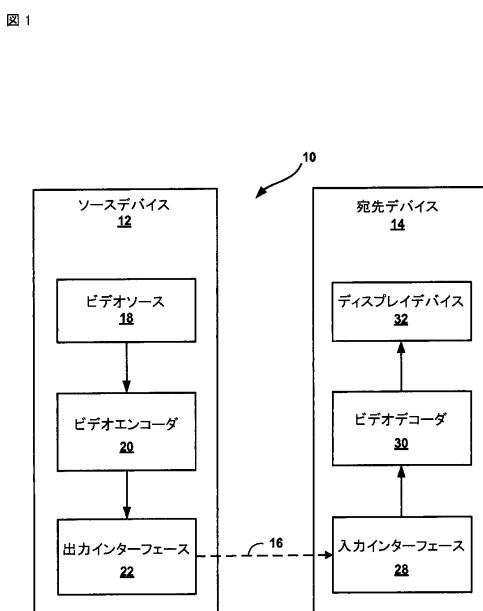
前記第1のピクチャについて第2次元ピクチャ識別子を復号させ、

ベースビデオコーディング仕様に従って、前記第1のピクチャの前記P O C値と前記第2次元ピクチャ識別子とに少なくとも一部基づいて第2のピクチャを復号させる、

コンピュータ可読記憶媒体。

10

【図1】



【図2】

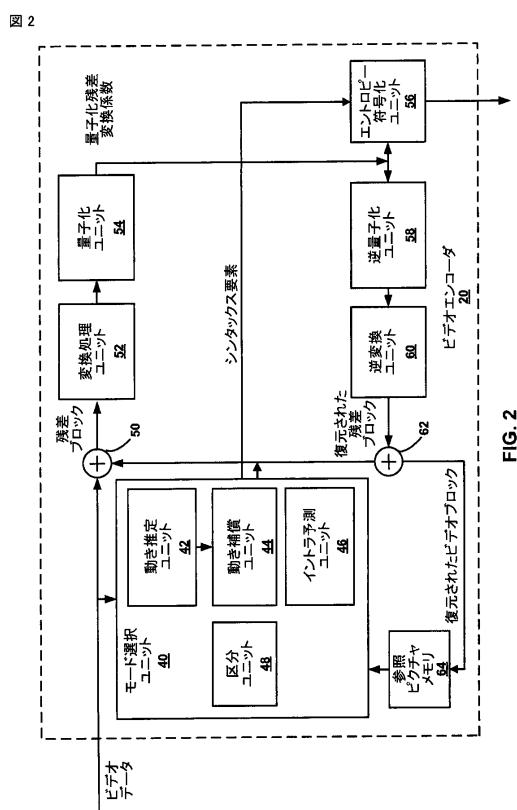
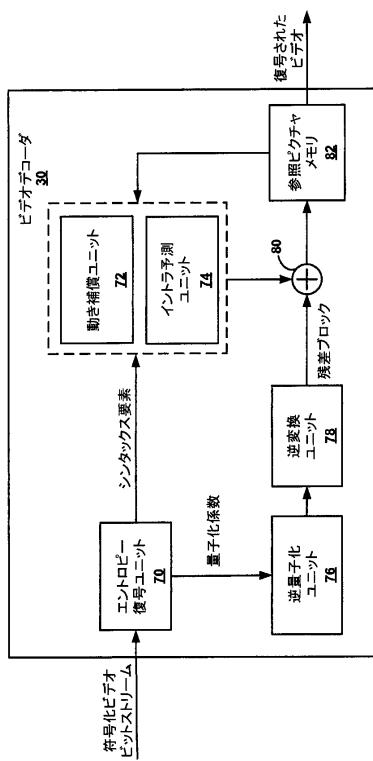


FIG. 2

FIG. 1

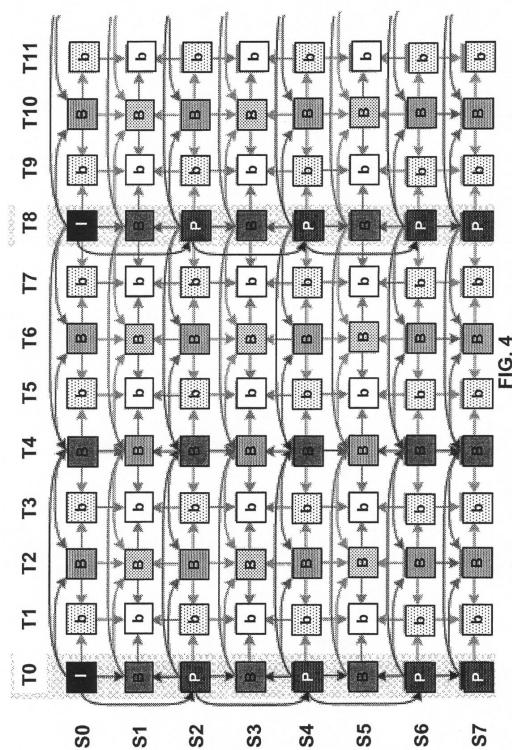
【図3】

図3



【図4】

図4



【図5】

図5

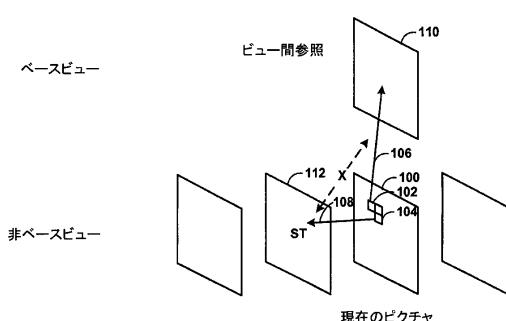


FIG. 5

【図6】

図6

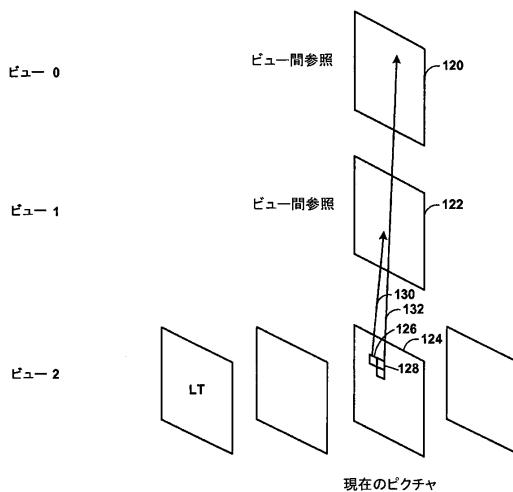


FIG. 6

【図7】

図7

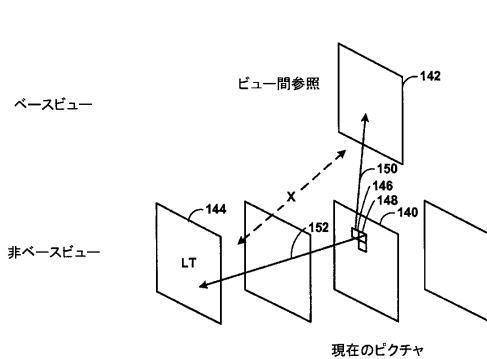


FIG. 7

【図8】

図8

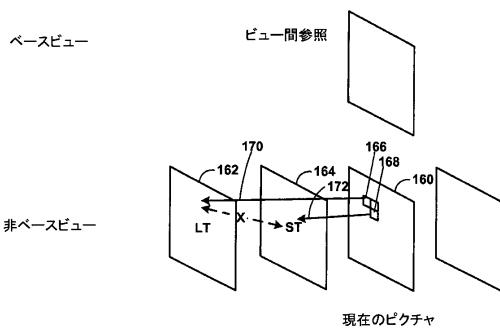


FIG. 8

【図9】

図9

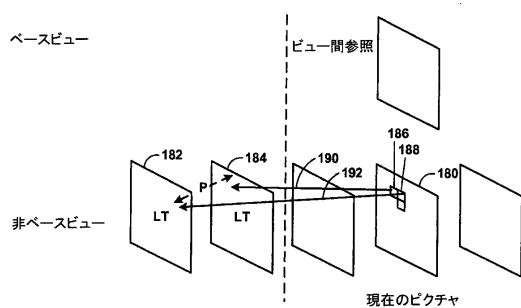


FIG. 9

【図10】

図10

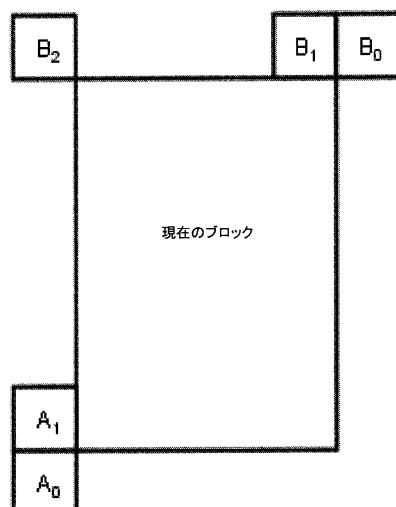


FIG. 10

【図11】

図11

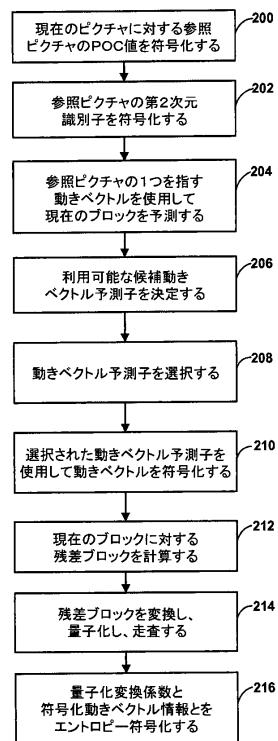


FIG. 11

【図12】

図12

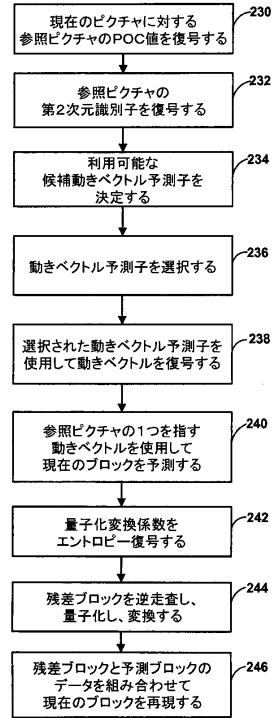


FIG. 12

フロントページの続き

(31)優先権主張番号 61/658,344
(32)優先日 平成24年6月11日(2012.6.11)
(33)優先権主張国 米国(US)
(31)優先権主張番号 61/663,484
(32)優先日 平成24年6月22日(2012.6.22)
(33)優先権主張国 米国(US)
(31)優先権主張番号 61/746,476
(32)優先日 平成24年12月27日(2012.12.27)
(33)優先権主張国 米国(US)
(31)優先権主張番号 13/801,731
(32)優先日 平成25年3月13日(2013.3.13)
(33)優先権主張国 米国(US)

(74)代理人 100153051
弁理士 河野 直樹
(74)代理人 100140176
弁理士 砂川 克
(74)代理人 100158805
弁理士 井関 守三
(74)代理人 100179062
弁理士 井上 正
(74)代理人 100124394
弁理士 佐藤 立志
(74)代理人 100112807
弁理士 岡田 貴志
(74)代理人 100111073
弁理士 堀内 美保子
(72)発明者 チェン、イン
アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 57
75
(72)発明者 ワン、イエ-クイ
アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 57
75
(72)発明者 ジャン、リ
アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 57
75

審査官 堀井 啓明

(56)参考文献 國際公開第2008/047303(WO,A2)
米国特許出願公開第2010/0202539(US,A1)
YOSHITOMO TAKAHASHI, DESCRIPTIONS OF 3D VIDEO CODING PROPOSAL (HEVC-COMPATIBLE CATEGOR
Y) BY SONY, 98. MPEG MEETING, 2011年11月22日, N M22566
YOSHITOMO TAKAHASHI, 3D-HEVC-CE3: RESULTS ON MOTION PARAMETER PREDICTION BY SONY, 99.
MPEG MEETING, 2012年 1月31日, N M23639
II-Koo Kim,et.al., Restriction on motion vector scaling for Merge and AMVP, Joint Coll
aborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7t
h Meeting:Geneva,CH,21-30 November,2011, 米国, JCTVC, 2011年11月22日, JCTVC-G5

51 , P.1-P.4 , U R L , <http://phenix.it-sudparis.eu/jct/index.php>

(58)調査した分野(Int.Cl. , DB名)

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