

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7577749号
(P7577749)

(45)発行日 令和6年11月5日(2024.11.5)

(24)登録日 令和6年10月25日(2024.10.25)

(51)国際特許分類

F I

H O 4 N 19/70 (2014.01)

H O 4 N 19/537 (2014.01)

H O 4 N 19/52 (2014.01)

H O 4 N 19/70

H O 4 N 19/537

H O 4 N 19/52

請求項の数 16 (全165頁)

(21)出願番号	特願2022-542912(P2022-542912)	(73)特許権者	503433420
(86)(22)出願日	令和3年1月13日(2021.1.13)		華為技術有限公司
(65)公表番号	特表2023-511276(P2023-511276 A)		HUAWEI TECHNOLOGIES CO., LTD.
(43)公表日	令和5年3月17日(2023.3.17)		中華人民共和国 5 1 8 1 2 9 広東省深
(86)国際出願番号	PCT/RU2021/050007		▲チェン▼市龍崗区坂田 華為總部▲ベ
(87)国際公開番号	WO2021/045659		ン▼公樓
(87)国際公開日	令和3年3月11日(2021.3.11)		Huawei Administrat
審査請求日	令和4年8月30日(2022.8.30)		ion Building, Banti
(31)優先権主張番号	62/961,159		an, Longgang Distri
(32)優先日	令和2年1月14日(2020.1.14)		ct, Shenzhen, Guang
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	dong 5 1 8 1 2 9, P. R. C
			hina
			100107766
			弁理士 伊東 忠重
			最終頁に続く

(54)【発明の名称】 マージ・モードのための候補の数をシグナリングする方法及び装置

(57)【特許請求の範囲】

【請求項1】

ビデオ復号のための幾何学的パーティショニング・マージ・モード候補の最大数を取得する方法であって、

ビデオ・シーケンスのためのビットストリームを取得することと、

前記ビットストリームに従って第1のインジケータの値を取得することであって、前記第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数を表す、取得することと、

前記ビットストリームに従って第2のインジケータの値を取得することであって、前記第2のインジケータは、前記ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得することと、

前記第1のインジケータの値が閾値より大きいとき、及び前記第2のインジケータの値がプリセット値に等しいときに、前記ビットストリームから第3のインジケータの値を解析することであって、前記第3のインジケータは、前記第1のインジケータの値から差し引かれた前記幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析することと、

前記第1のインジケータの前記値及び前記第3のインジケータの前記値に従って、前記幾何学的パーティショニング・マージ・モード候補の最大数を取得することと、

(i) 前記第1のインジケータの値が前記閾値に等しく、かつ(ii) 前記第2のインジケータの値が前記プリセット値に等しいときに、前記幾何学的パーティショニング・マ

ージ・モード候補の最大数の値を2にセットすることと、を含む、方法。

【請求項2】

前記閾値は2である、請求項1に記載の方法。

【請求項3】

前記プリセット値は1である、請求項1又は2に記載の方法。

【請求項4】

前記第2のインジケータの値を取得することは、前記第1のインジケータの値を取得することの後に実行される、請求項1～3のいずれか一項に記載の方法。

【請求項5】

前記第1のインジケータの値が前記閾値以上であるときに、前記第2のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セットSPSから解析される、請求項4に記載の方法。

10

【請求項6】

前記第2のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セットSPSから取得される、請求項1～5のいずれか一項に記載の方法。

【請求項7】

前記第3のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セットSPSから取得される、請求項1～6のいずれか一項に記載の方法。

【請求項8】

ビデオ復号装置であって、

20

ビデオ・シーケンスのためのビットストリームを取得するように構成されている受信モジュールと、

前記ビットストリームに従って第1のインジケータの値を取得するように構成されている取得モジュールであって、前記第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数を表し、

前記取得モジュールは、前記ビットストリームに従って第2のインジケータの値を取得するように構成されており、前記第2のインジケータは、前記ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得モジュールと、

前記第1のインジケータの値が閾値より大きいとき、及び前記第2のインジケータの値がプリセット値に等しいときに、前記ビットストリームから第3のインジケータの値を解析するように構成されている解析モジュールであって、前記第3のインジケータは、前記第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析モジュールと、を含み、

30

前記第1のインジケータの前記値及び前記第3のインジケータの前記値に従って、前記幾何学的パーティショニング・マージ・モード候補の最大数が取得され、

前記取得モジュールは、(i)前記第1のインジケータの値が前記閾値に等しく、かつ(ii)前記第2のインジケータの値が前記プリセット値に等しいときに、前記幾何学的パーティショニング・マージ・モード候補の最大数の値を2にセットするように構成されている、ビデオ復号装置。

40

【請求項9】

前記閾値は2である、請求項8に記載のビデオ復号装置。

【請求項10】

前記プリセット値は1である、請求項8又は9に記載のビデオ復号装置。

【請求項11】

前記第2のインジケータの値を取得することは、前記第1のインジケータの値を取得することの後に実行される、請求項8～10のいずれか一項に記載のビデオ復号装置。

【請求項12】

前記第1のインジケータの値が前記閾値以上であるときに、前記第2のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セットSPSから解析される、

50

請求項 1 1 に記載のビデオ復号装置。

【請求項 1 3】

前記第 2 のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セット S P S から取得される、請求項 8 ～ 1 2 のいずれか一項に記載のビデオ復号装置。

【請求項 1 4】

前記第 3 のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セット S P S から取得される、請求項 8 ～ 1 3 のいずれか一項に記載のビデオ復号装置。

【請求項 1 5】

コンピュータ又はプロセッサで実行されたときに、請求項 1 ～ 7 のいずれか一項に記載の方法を実行するためのプログラム・コードを含む、コンピュータ・プログラム。

【請求項 1 6】

復号器であって、

1 つ以上のプロセッサと、

前記 1 つ以上のプロセッサに結合され、前記 1 つ以上のプロセッサによる実行のためのプログラムを記憶する非一時的なコンピュータ可読記憶媒体であって、前記プログラムは、前記 1 つ以上のプロセッサによって実行される場合に、請求項 1 ～ 7 のいずれか一項に記載の方法を実施するように復号器を設定する、非一時的なコンピュータ可読記憶媒体と、を含む、復号器。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本特許出願は、2020 年 1 月 1 4 日に出願された U S 6 2 / 9 6 1 , 1 5 9 号に対する優先権を主張する。前述の特許出願の開示は、その全体が参照により本明細書に組み込まれる。

【0 0 0 2】

本出願の実施形態は、一般に、動画コーディングの分野に関係し、より詳細には、マージ・モードの候補の数をシグナリングすることに関係する。

【背景技術】

【0 0 0 3】

ビデオ・コーディング（ビデオ符号化及び復号）は、広範囲のデジタル・ビデオ・アプリケーション、例えば、放送デジタル TV、インターネット及びモバイル・ネットワークを介したビデオ伝送、ビデオ・チャット、ビデオ会議などのリアルタイムの会話アプリケーション、DVD 及びブルーレイ・ディスク、ビデオ・コンテンツの収集及び編集システム、並びにセキュリティ・アプリケーションのカムコーダに使用される。

【0 0 0 4】

比較的短いビデオでさえ描写するために必要とされるビデオ・データの量は、相当なものであり、これは、データがストリーム化されるか、又は他の方法で限定された帯域幅容量を有する通信ネットワークを介して通信されるときに、困難になることがある。したがって、ビデオ・データは、一般に、現代の電気通信ネットワークを介して通信される前に圧縮される。また、メモリ・リソースが制限されることがあるため、ビデオが記憶デバイスに記憶される場合に、ビデオのサイズも問題となる可能性がある。ビデオ圧縮デバイスは、しばしば、伝送又は記憶の前にビデオ・データをコーディングするためにソースにおいてソフトウェア及び／又はハードウェアを使用し、それによってデジタル・ビデオ画像を表すのに必要なデータ量を減少させる。次いで、圧縮されたデータは、ビデオ・データを復号するビデオ解凍デバイスによって宛先で受信される。ネットワークリソースが限られており、より高いビデオ品質の要求が絶えず増加しているため、ピクチャ品質にほとんど犠牲を払わずに圧縮比を改善する改善された圧縮及び解凍技術が望ましい。

【発明の概要】

【0 0 0 5】

本出願の実施形態は、独立請求項による符号化及び復号するための装置及び方法を提供

10

20

30

40

50

する。

前述の目的及び他の目的は、独立請求項の主題事項によって達成される。さらなる実装は、従属請求項、明細書及び図面から明らかである。

特定の実施形態は、添付の独立請求項において概説されており、他の実施形態は、従属クレームにおいて概説されている。

【0006】

本発明の第1の態様は、ビデオ復号のための幾何学的パーティショニング・マージ・モード候補の最大数を取得する方法を提供し、方法は、ビデオ・シーケンスのためのビットストリームを取得することと、ビットストリームに従って第1のインジケータの値を取得することであって、第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数を表す、取得することと、ビットストリームに従って第2のインジケータの値を取得することであって、第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得することと、第1のインジケータの値が閾値より大きいとき、及び第2のインジケータの値がプリセット値に等しいときに、ビットストリームから第3のインジケータの値を解析することであって、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析することと、を含む。

【0007】

本発明の実施形態によれば、マージ・モード候補の数のインジケータのシグナリング方式が開示される。幾何学的パーティショニング・マージ・モード候補の最大数は、条件付きでシグナリングされる。したがって、ビットストリーム利用及び復号効率が改善されている。

【0008】

一実装では、方法は、第1のインジケータの値が閾値に等しく、かつ第2のインジケータの値がプリセット値に等しいときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を2にセットすることをさらに含む。

【0009】

一実装では、方法は、第1のインジケータの値が閾値より小さいか、又は第2のインジケータの値がプリセット値に等しくないときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を0にセットすることをさらに含む。

【0010】

一実装では、閾値は2である。

【0011】

一実装では、プリセット値は1である。

【0012】

一実装では、第2のインジケータの値を取得することは、第1のインジケータの値を取得することの後に実行される。

【0013】

一実装では、第一のインジケータは、ビットストリームにおいてコーディングされた構文要素により取得される。

【0014】

一実装では、第1のインジケータの値が閾値以上であるときに、第2のインジケータの値は、前記ビットストリームのシーケンス・パラメータ・セットSPSから解析される。例えば、第2のインジケータの値を取得するために、シーケンス・パラメータ・セットSPSにおける構文要素を解析する。

【0015】

一実装では、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから取得される。例えば、第2のインジケータの値を取得するために、シーケンス・パラメータ・セットSPSにおける構文要素を解析する。一実装では、第3の

インジケータの値は、ビットストリームのシーケンス・パラメータ・セット S P S から取得される。例えば、第 3 のインジケータの値を取得するために、シーケンス・パラメータ・セット S P S における構文要素を解析する。

【 0 0 1 6 】

本発明の第 2 の態様は、ビデオ復号装置を提供し、ビデオ復号装置は、ビデオ・シーケンスのためのビットストリームを取得するように構成されている受信モジュールと、ビットストリームに従って第 1 のインジケータの値を取得するように構成されている取得モジュールであって、第 1 のインジケータは、モーション・ベクトル予測 M V P 候補をマージする最大数を表し、取得モジュールは、ビットストリームに従って第 2 のインジケータの値を取得するように構成されており、第 2 のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得モジュールと、第 1 のインジケータの値が閾値より大きいとき、及び第 2 のインジケータの値がプリセット値に等しいときに、ビットストリームから第 3 のインジケータの値を解析するように構成されている解析モジュールであって、第 3 のインジケータは、第 1 のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析モジュールと、を含む。

【 0 0 1 7 】

本発明の第 1 の態様による方法は、本発明の第 2 の態様によるデバイスによって実行され得る。本発明の第 1 の態様による方法のさらなる特徴及び実装は、本発明の第 2 の態様による装置の特徴及び実装形式に対応する。

【 0 0 1 8 】

一実装では、第 1 のインジケータの値が閾値に等しく、かつ第 2 のインジケータの値がプリセット値に等しいときに、取得モジュールは、幾何学的パーティショニング・マージ・モード候補の最大数の値を 2 にセットするように構成されている。

【 0 0 1 9 】

一実装では、第 1 のインジケータの値が閾値より小さいか、又は第 2 のインジケータの値がプリセット値に等しくないときに、取得モジュールは、幾何学的パーティショニング・マージ・モード候補の最大数の値を 0 にセットするように構成されている。

【 0 0 2 0 】

一実装では、閾値は 2 である。

【 0 0 2 1 】

一実装では、プリセット値は 1 である。

【 0 0 2 2 】

一実装では、第 2 のインジケータの値を取得することは、第 1 のインジケータの値を取得することの後に実行される。

【 0 0 2 3 】

一実装では、第 1 のインジケータの値が閾値以上であるときに、第 2 のインジケータの値は、ビットストリームのシーケンス・パラメータ・セット S P S から解析される。

【 0 0 2 4 】

一実装では、第 2 のインジケータの値は、ビットストリームのシーケンス・パラメータ・セット S P S から取得される。

【 0 0 2 5 】

一実装では、第 3 のインジケータの値は、ビットストリームのシーケンス・パラメータ・セット S P S から取得される。

【 0 0 2 6 】

一実装では、ビデオ復号のための幾何学的パーティショニング・マージ・モード候補の最大数を取得する方法が開示されており、方法は、ビデオ・シーケンスのためのビットストリームを取得することと、ビットストリームに従って第 1 のインジケータの値を取得することとであって、第 1 のインジケータは、モーション・ベクトル予測 M V P 候補をマージする最大数を表す、取得することと、取得

10

20

30

40

50

された第1のインジケータの値が閾値より大きい場合にのみ、ビットストリームに従って第2のインジケータの値を取得することであって、第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得することと、第1のインジケータの値が閾値より大きく、第2のインジケータの値がプリセット値に等しいときにのみ、ビットストリームから第3のインジケータの値を解析することであって、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析することと、を含む。

【0027】

本発明の第3の態様は、幾何学的パーティショニング・マージャ・モード候補の最大数を符号化する方法を提供し、

方法は、第1のインジケータの値を決定することであって、第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数を表す、決定することと、第2のインジケータの値を決定することであって、第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、決定することと、第1のインジケータの値が閾値より大きいとき、及び第2のインジケータの値がプリセット値に等しいときに、ビットストリームに第3のインジケータの値を符号化することであって、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、符号化することと、を含む。

【0028】

本発明の実施形態によれば、マージ・モード候補の数のインジケータのシグナリング方式が開示される。幾何学的パーティショニング・マージ・モード候補の最大数は、条件付きでシグナリングされる。したがって、ビットストリーム利用及び復号効率が改善されている。

【0029】

一実装では、方法は、第1のインジケータの値が閾値に等しく、かつ第2のインジケータの値がプリセット値に等しいときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を2にセットすることをさらに含む。

【0030】

一実装では、方法は、第1のインジケータの値が閾値より小さいか、又は第2のインジケータの値がプリセット値に等しくないときに、何学的パーティショニング・マージ・モード候補の最大数の値を0にセットすることをさらに含む。

【0031】

一実装では、閾値は2である。

【0032】

一実装では、プリセット値は1である。

【0033】

一実装では、第2のインジケータの値を決定することは、第1のインジケータの値を決定することの後に実行される。

【0034】

一実装では、第1のインジケータの値が閾値以上であるときに、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSに符号化される。

【0035】

一実装では、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSに符号化される。

【0036】

一実装では、第3のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSに符号化される。

【0037】

10

20

30

40

50

本発明の第４の態様は、ビデオ符号化装置を提供し、ビデオ符号化装置は、ビットストリームに従って第１のインジケータの値を決定するように構成されている決定モジュールであって、第１のインジケータは、モーション・ベクトル予測ＭＶＰ候補をマージする最大数を表し、決定モジュールは、第２のインジケータの値を決定するように構成されており、第２のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、決定モジュールと、第１のインジケータの値が閾値より大きいとき、及び第２のインジケータの値がプリセット値に等しいときに、ビットストリームから第３のインジケータの値を解析するように構成されている解析モジュールであって、第３のインジケータは、第１のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析モジュールと、を含む。

10

【００３８】

本発明の第３の態様による方法は、本発明の第４の態様による装置によって実行され得る。本発明の第３の態様による方法のさらなる特徴及び実装は、本発明の第４の態様による装置の特徴及び実装形式に対応する。

【００３９】

一実装では、決定モジュールは、第１のインジケータの値が閾値に等しく、かつ第２のインジケータの値がプリセット値に等しいときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を２にセットするように構成されている。

20

【００４０】

一実装では、決定モジュールは、第１のインジケータの値が閾値より小さいか、又は第２のインジケータの値がプリセット値に等しくないときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を０にセットするように構成されている。

【００４１】

一実装では、閾値は２である。

【００４２】

一実装では、プリセット値は１である。

【００４３】

一実装では、第２のインジケータの値を決定することは、第１のインジケータの値を決定することの後に実行される。

30

【００４４】

一実装では、第１のインジケータの値が閾値以上であるときに、第２のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットＳＰＳに符号化される。

【００４５】

一実装では、第２のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットＳＰＳに符号化される。

【００４６】

一実装では、第３のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットＳＰＳに符号化される。

【００４７】

本発明の第５の態様は、第１の態様及び第１の態様の実装のうちのいずれかによる方法を実施するための処理回路を含む復号器を提供する。

40

【００４８】

本発明の第６の態様は、第３の態様及び第３の態様の実装のいずれかによる方法を実施するための処理回路を含む符号化器を提供する。

【００４９】

本発明の第７の態様は、コンピュータ又はプロセッサで実行されるときに、第１の態様、第３の態様、及び第１の態様、第３の態様の実装のいずれかによる方法を実行するためのプログラム・コードを含むコンピュータ・プログラム製品を提供する。

【００５０】

50

本発明の第 8 の態様は、1 つ以上のプロセッサと、プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する非一時的なコンピュータ可読記憶媒体と、を含み、プログラミングは、プロセッサによって実行されるときに、第 1 の態様、第 3 の態様のいずれか、及び第 1 の態様、第 3 の態様の実装のいずれかによる方法を実施するように復号器を設定する、復号器を提供する。

【0051】

本発明の第 9 の態様は、プログラム・コードを搬送する非一時的なコンピュータ可読媒体であって、プログラム・コードは、コンピュータ・デバイスによって実行されるときに、コンピュータ・デバイスに第 1 の態様、第 3 の態様のいずれか、及び第 1 の態様、第 3 の態様の実装のいずれかによる方法を実行させる、非一時的なコンピュータ可読媒体を提供する。

10

【0052】

本発明の第 10 の態様は、第 3 の態様及び第 3 の態様の実装のいずれかによる方法を実施するための処理回路を含む符号化器を提供する。

【0053】

本発明の第 11 の態様は、1 つ以上のプロセッサと、プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する非一時的なコンピュータ可読記憶媒体と、を含み、プログラミングは、プロセッサによって実行されるときに、第 3 の態様のいずれか、及び第 3 の態様の実装のいずれかによる方法を実施するように符号化器を設定する、符号化器を提供する。

20

【0054】

本発明の第 12 の態様は、上記の実施形態のいずれかの方法によって符号化／復号されるビットストリームを含む非一時的な記憶媒体を提供する。

【0055】

本発明の第 13 の態様は、複数の構文要素を含むことによってビデオ信号に対して符号化されたビットストリームを提供し、複数の構文要素は、第 2 のインジケータ (`sps_geo_enabled_flag` など) を含み、第 3 のインジケータ `sps_max_num_merge_cand_minus_max_num_geo_cand` は、`sps_geo_enabled_flag` の値に少なくとも部分的に基づいて条件付きでシグナリングされる。

30

【0056】

本発明の第 14 の態様は、画像復号デバイスによって復号される符号化されたビットストリームを含む非一時的な記憶媒体であって、ビットストリームは、ビデオ信号又は画像信号のフレームを複数のブロックに分割することによって生成され、かつ複数の構文要素を含み、複数の構文要素は、先行する態様のいずれかによる第 3 のインジケータ (`sps_max_num_merge_cand_minus_max_num_geo_cand` など) を含む、非一時的な記憶媒体を提供する。

【0057】

本発明の第 15 の態様は、ビデオ復号のための方法を提供し、方法は、ビデオ・シーケンスのためのビットストリームを取得することと、ビットストリームに従って第 1 のインジケータの値を取得することとであって、第 1 のインジケータは、モーション・ベクトル予測 MVP 候補をマージする最大数を表す、取得することと、ビットストリームに従って第 2 のインジケータの値を取得することとであって、第 2 のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得することと、第 1 のインジケータの値が閾値より大きいとき、及び第 2 のインジケータの値がプリセット値に等しいときに、ビットストリームから第 3 のインジケータの値を解析することとであって、第 3 のインジケータは、第 1 のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析することと、

40

現在のコーディング・ブロックの隣接ブロックのモーション・ベクトルにしたがって、現

50

在のコーディング・ブロックのためのマージ候補リストを構成することと、
第3のインジケータの値にしたがって、マージ・インデックスを取得することと、
マージ・インデックス及びマージャ候補リストにしたがって、現在のコーディング・ブロックのモーション・ベクトルを取得することと、
現在のコーディング・ブロックのモーション・ベクトルにしたがって、現在のコーディング・ブロックを再構成することと、を含む。

【0058】

本発明の第16の態様は、ビデオ復号装置を提供し、ビデオ復号装置は、ビデオ・シーケンスのためのビットストリームを取得するように構成されている受信モジュールと、ビットストリームに従って第1のインジケータの値を取得するように構成されている取得モジュールであって、第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数を表し、取得モジュールは、ビットストリームに従って第2のインジケータの値を取得するように構成されており、第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得モジュールと、第1のインジケータの値が閾値より大きいとき、及び第2のインジケータの値がプリセット値に等しいときに、ビットストリームから第3のインジケータの値を解析するように構成されている解析モジュールであって、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析モジュールと、

現在のコーディング・ブロックの隣接ブロックのモーション・ベクトルにしたがって、現在のコーディング・ブロックのためのマージ候補リストを構成するように構成されているマージ候補リスト構成モジュールと、

第3のインジケータの値にしたがって、マージ・インデックスを取得するマージ・インデックスを取得するように構成されているその取得モジュールと、

マージ・インデックス及びマージャ候補リストにしたがって、現在のコーディング・ブロックのモーション・ベクトルを取得するように構成されているモーション・ベクトル・モジュールと、

現在のコーディング・ブロックのモーション・ベクトルにしたがって、現在のコーディング・ブロックを再構成するように構成されている画素再構成モジュールと、を含む。

【0059】

本発明の第15の態様及び本発明の第16の態様に関する詳細又は例は、本発明の第1の態様から第14の態様において開示された上記の例を参照することができる。

【0060】

前述の目的及び他の目的は、独立請求項の主題事項によって達成される。さらなる実装は、従属請求項、明細書及び図面から明らかである。

1つ以上の実施形態の詳細は、添付の図面及び以下の説明に記述されている。他の特徴、目的、及び利点は、明細書、図面、及び特許請求の範囲から明らかであろう。

【図面の簡単な説明】

【0061】

以下、本発明の実施形態が、添付の図及び図面を参照して、より詳細に記載されている。

【0062】

【図1A】本発明の実施形態を実装するように構成されているビデオ・コーディング・システムの一例を示すブロック図である。

【図1B】本発明の実施形態を実装するように構成されているビデオ・コーディング・システムの一例を示すブロック図である。

【図2】本発明の実施形態を実装するように構成されているビデオ符号化器の一例を示すブロック図である。

【図3】本発明の実施形態を実装するように構成されているビデオ復号器の例示的な構造を示すブロック図である。

【図4】符号化装置又は復号装置の一例を示すブロック図である。

【図 5】符号化装置又は復号装置の別の例を示すブロック図である。

【図 6】重み付き予測符号化器側の意思決定及びパラメータ推定のためのフローチャートである。

【図 7】三角予測モードの例を例示する。

【図 8】幾何学的予測モードの例を例示する。

【図 9】幾何学的予測モードの別の例を例示する。

【図 10】コンテンツ配信サービスを実現するコンテンツ供給システム 3 1 0 0 の例示的な構造を示すブロック図である。

【図 11】端末デバイスの一例の構造を示すブロック図である。

【図 12】本出願によるインター予測方法の例を例示するブロック図である。

【図 13】本出願によるインター予測のための装置の例を例示するブロック図である。

【図 14】本出願によるインター予測のための装置の別の例を例示するブロック図である。

【図 15】本発明による方法の実施形態を示すフローチャートである。

【図 16】本発明による装置を示すブロック図である。

【0063】

以下、同一の参照符号は、他に明示的に指定されていない場合は、同一の又は少なくとも機能的に同等の特徴を指す。

【発明を実施するための形態】

【0064】

以下の説明では、本開示の一部を形成し、本発明の実施形態の特定の態様又は本発明の実施形態が使用され得る特定の態様を例示するために、添付の図面を参照する。本発明の実施形態は、他の態様で使うことができ、図に示されていない構造的又は論理的变化を含むことが理解される。したがって、以下の詳細な説明は、限定的な意味で解釈されるべきではなく、本発明の範囲は、添付の特許請求の範囲によって定義される。

【0065】

例えば、記載の方法に関連する開示は、当該方法を行うように構成された対応するデバイス又はシステムについても真であり、その逆もまた真であり得ることが理解される。例えば、1つ又は複数の特定の方法ステップが記載される場合、対応するデバイスは、記載された1つ又は複数の方法ステップを行う（例えば、1つのユニットが1つ又は複数のステップを行うか、又は複数のユニットの各々が複数のステップのうちの1つ以上を行う）ために、1つ又は複数のユニット、例えば、機能ユニットを、たとえそのような1つ以上のユニットが明示的に記載されるか、図に例示されていなくても、含んでもよい。一方、例えば、1つ又は複数のユニット、例えば、機能ユニットに基づいて特定の装置が記載される場合、対応する方法は、1つ又は複数のユニットの機能性を行う（例えば、1つのステップが1つ又は複数のユニットの機能性を行うか、又は複数のユニットのうちの各々が複数のユニットのうちの1つ以上の機能性を行う複数のステップ）ために、1つのステップを、たとえそのような1つ又は複数のステップが明示的に記載されるか、図に示されていなくても、含んでもよい。さらに、本明細書で記載された種々の例示的な実施形態及び／又は態様の特徴は、特に断らない限り、互いに組み合わせてもよいことが理解される。

【0066】

ビデオ・コーディングは、典型的には、ビデオ又はビデオ・シーケンスを形成する一連のピクチャの処理を指す。用語「ピクチャ」の代わりに、用語「フレーム」又は「画像」は、ビデオ・コーディングの分野では同義語として使用されてもよい。ビデオ・コーディング（又は一般的な符号化）は、2つの部分のビデオ符号化及びビデオ復号を含む。ビデオ・コーディングは、ソース側で行われ、典型的には、元のビデオ・ピクチャを（より効率的な記憶及び／又は伝送のために）表するのに必要なデータ量を低減するために、元のビデオ・ピクチャの（例えば、圧縮による）処理を含む。ビデオ復号は、宛先側で行われ、典型的には、ビデオ・ピクチャを再構成するために符号化器と比較して、逆方向の処理を含む。ビデオ・ピクチャ（又は、一般には、ピクチャ）の「コーディング」を参照する実施形態は、ビデオ・ピクチャ又はそれぞれのビデオ・シーケンスの「符号化」又は「復

10

20

30

40

50

号」に関係するものと理解されたい。符号化部分と複合部分の組み合わせは、CODEC (Coding and Decoding) と呼ばれる。

【0067】

可逆ビデオ・コーディングの場合、元のビデオ・ピクチャを再構成することができる。すなわち、再構成されたビデオ・ピクチャは、元のビデオ・ピクチャと同じ品質を有する（記憶又は伝送の間に伝送損失又は他のデータ損失がないと仮定する）。非可逆ビデオ・コーディングの場合、ビデオ・ピクチャを表すデータの量を低減するために、例えば量子化によるさらなる圧縮が行われるが、復号器で完全には再構成することができない。すなわち、再構成されたビデオ・ピクチャの品質は、元のビデオ・ピクチャの品質よりも低い

10

【0068】

いくつかのビデオ・コーディング標準は、「非可逆ハイブリッド・ビデオ・コーデック」のグループに属する（すなわち、サンプル・ドメインにおける空間的及び時間的予測と、トランスフォーム・ドメインにおける量子化を適用するための2Dトランスフォーム・コーディングを組み合わせる）。ビデオ・シーケンスの各ピクチャは、典型的には、重複しないブロックのセットにパーティショニングされ、コーディングは、典型的には、ブロックレベルで行われる。言い換えれば、符号化器では、ビデオは、典型的には、ブロック（ビデオ・ブロック）レベルで、例えば、空間的（ピクチャ内）予測及び／又は時間的（ピクチャ間）予測を使用して予測ブロックを生成し、現在のブロック（現在処理されている／処理されるブロック）から予測ブロックを減算して残差ブロックを取得して、残差ブロックをトランスフォームし、トランスフォーム・ドメイン内の残差ブロックを量子化して、送信されるデータ量を低減（圧縮）することによって、処理、すなわち符号化される一方、復号器では、符号化器と比較した逆処理が符号化されたか、又は圧縮されたブロックに適用されて、提示のために現在のブロックを再構成する。さらに、符号化器は復号器処理ループを複製し、両者が同一の予測（例えば、イントラ予測及びインター予測）を生成し、及び／又は処理、すなわち、後続のブロックをコーディングするための再構成を生成するようにする。

20

【0069】

ビデオ・コーディング・システム10の以下の実施形態では、図1～図3に基づいて、ビデオ符号化器20及びビデオ復号器30が記載される。

30

【0070】

図1Aは、例示的なコーディング・システム10、例えば、本出願の技術を利用することができるビデオ・コーディング・システム10（又は省略してコーディング・システム10）を示す概略ブロック図である。ビデオ・コーディング・システム10のビデオ符号化器20（又は省略して符号化器20）及びビデオ復号器30（又は省略して復号器30）は、本出願で記載される種々の例にしたがって技術を行うように構成され得るデバイスの例を表す。

【0071】

図1Aに示すように、コーディング・システム10は、例えば、符号化されたピクチャ・データ13を復号するための宛先デバイス14に、符号化されたピクチャ・データ21を提供するように構成されているソース・デバイス12を含む。

40

【0072】

ソース・デバイス12は、符号化器20を含み、追加的に、ピクチャ・ソース16、プリプロセッサ（又は前処理ユニット）18、例えばピクチャ・プリプロセッサ18、及び通信インターフェース又は通信ユニット22を含んでもよい。

【0073】

ピクチャ・ソース16は、任意の種類のピクチャ捕捉デバイス、例えば、実世界ピクチャを捕捉するためのカメラ、及び／又は任意の種類のピクチャ生成デバイス、例えば、コンピュータ・アニメーション・ピクチャを生成するためのコンピュータ・グラフィック・プロセッサ、又は、実世界ピクチャ、コンピュータ生成ピクチャ（例えば、スクリーン・

50

コンテンツ、バーチャル・リアリティ（VR）ピクチャ）、及び／又はそれらの任意の組み合わせ（例えば、拡張現実（AR）ピクチャ）を取得する、及び／又は提供するための任意の種類の他のデバイスを含んでもよい。ピクチャ・ソースは、上述のピクチャのいずれかを記憶する任意の種類のメモリ又はストレージであってもよい。

【0074】

プリプロセッサ18と、前処理ユニット18によって行われる処理とを区別して、ピクチャ又はピクチャ・データ17もまた、生のピクチャ又は生のピクチャ・データ17と呼ばれてもよい。

【0075】

プリプロセッサ18は、（生の）ピクチャ・データ17を受信し、ピクチャ・データ17に前処理を行って、前処理されたピクチャ・データ19又は前処理されたピクチャ・データ19を取得するように構成されている。プリプロセッサ18によって行われる前処理は、例えば、トリミング、色フォーマット変換（例えば、RGBからYCbCrへ）、色補正、又はノイズ除去を含んでもよい。前処理ユニット18は、任意のコンポーネントであってもよいと理解され得る。

【0076】

ビデオ符号化器20は、前処理されたピクチャ・データ19を受信し、符号化されたピクチャ・データ21を提供するように構成されている（例えば、図2に基づいて、さらなる詳細を以下に記載する）。

【0077】

ソース・デバイス12の通信インターフェース22は、符号化されたピクチャ・データ21を受信し、記憶又は直接再構成のために、通信チャネル13を介して、符号化されたピクチャ・データ21（又はその任意のさらなる処理されたバージョン）を他のデバイス、例えば宛先デバイス14又は任意の他のデバイスに送信するように構成されてもよい。

【0078】

宛先デバイス14は、復号器30（例えば、ビデオ復号器30）を含み、追加的、すなわち任意選択的に、通信インターフェース又は通信ユニット28、ポストプロセッサ32（又は後処理ユニット32）、及び表示デバイス34を含んでもよい。

【0079】

宛先デバイス14の通信インターフェース28は、符号化されたピクチャ・データ21（又はそのさらなる処理されたバージョン）を、例えばソース・デバイス12から直接、又は任意の他のソース、例えば記憶デバイス、例えば符号化されたピクチャ・データ記憶デバイスから受信し、符号化されたピクチャ・データ21を復号器30に提供するように構成されている。

【0080】

通信インターフェース22及び通信インターフェース28は、ソース・デバイス12と宛先デバイス14との間の直接通信リンク、例えば直接有線若しくは無線接続、又は任意の種類のネットワーク、例えば有線若しくは無線ネットワーク若しくはそれらの任意の組み合わせ、若しくは任意の種類のプライベート及びパブリック・ネットワーク、又はそれらの任意の種類の組み合わせを介して、符号化されたピクチャ・データ21又は符号化されたデータ13を送信又は受信するように構成されてもよい。

【0081】

通信インターフェース22は、例えば、符号化されたピクチャ・データ21を適切なフォーマット、例えば、パケットにパッケージングし、及び／又は通信リンク又は通信ネットワークを介した伝送のための任意の種類の伝送符号化又は処理を使用して符号化されたピクチャ・データを処理するように構成されてもよい。

【0082】

通信インターフェース22のカウナパートを形成する通信インターフェース28は、例えば、送信されたデータを受信し、任意の種類の対応する伝送復号又は処理及び／又はパッケージング解除を使用して伝送データを処理して、符号化されたピクチャ・データ2

10

20

30

40

50

1を取得するように構成されてもよい。

【0083】

通信インターフェース22及び通信インターフェース28は両方とも、ソース・デバイス12から宛先デバイス14へ指し示す図1Aの通信チャンネル13の矢印によって示されるように、一方向通信インターフェース、又は双方向通信インターフェースとして構成されてもよく、例えば、通信リンク及び／又はデータ伝送例えば、符号化されたピクチャ・データ伝送に関係する任意の他の情報を確認応答及び交換するために、メッセージを送信及び受信する、例えばコネクションをセット・アップするように構成されてもよい。

【0084】

復号器30は、符号化されたピクチャ・データ21を受信し、復号されたピクチャ・データ31又は復号されたピクチャ31を提供するように構成されている（例えば、図3又は図5に基づいて、さらなる詳細を以下に記載する）。

【0085】

宛先デバイス14のポストプロセッサ32は、復号されたピクチャ・データ31、例えば復号されたピクチャ31を後処理して、後処理されたピクチャ・データ33、例えば後処理されたピクチャ33を取得するように構成されている。後処理ユニット32によって行われる後処理は、例えば、色フォーマット変換（例えば、YCbCrからRGBへ）、色補正、トリミング、又は再サンプリング、又は、例えば、表示デバイス34による表示のために、復号されたピクチャ・データ31を準備するための任意の他の処理を含むことができる。

【0086】

宛先デバイス14の表示デバイス34は、例えばユーザ又はビューアに、ピクチャを表示するために後処理されたピクチャ・データ33を受信するように構成されている。表示デバイス34は、再構成されたピクチャを表すための任意の種類のディスプレイ、例えば、統合されたディスプレイ又は外部ディスプレイ若しくはモニタであってもよく、又はそれらを含んでもよい。ディスプレイは、例えば、液晶ディスプレイ（LCD）、有機発光ダイオード（OLED）ディスプレイ、プラズマ・ディスプレイ、プロジェクタ、マイクロLEDディスプレイ、液晶オン・シリコン（LCOS）、デジタル光プロセッサ（DLP）、又は任意の種類の他のディスプレイを含むことができる。

【0087】

図1Aは、ソース・デバイス12及び宛先デバイス14を別個のデバイスとして描いているのであるが、デバイスの実施形態は、両方又は両方の機能、ソース・デバイス12又は対応する機能性、及び宛先デバイス14又は対応する機能性を含んでもよい。そのような実施形態では、ソース・デバイス12又は対応する機能性と宛先デバイス14又は対応する機能性は、同じハードウェア及び／又はソフトウェア、別個のハードウェア及び／又はソフトウェア又はそれらの任意の組み合わせによって実装されてもよい。

【0088】

説明に基づいて当業者には明らかなように、図1Aに示すように、ソース・デバイス12及び／又は宛先デバイス14内の異なるユニット又は機能性の存在及び機能性の（正確な）分裂は、実際のデバイス及び用途に依存して変動し得る。

【0089】

符号化器20（例えば、ビデオ符号化器20）、復号器30（例えば、ビデオ復号器30）、又は符号化器20と復号器30の両方は、図1Bに示すような処理回路、1つ以上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、ハードウェア、専用のビデオ・コーディング、又はそれらの任意の組み合わせによって実装されてもよい。符号化器20は、図2の符号化器20及び／又は本明細書で記載される任意の他の符号化器システム又はサブシステムに関して記載したように、種々のモジュールを具体化するために、処理回路46を介して実装されてもよい。復号器30は、図3の復号器30及び／又は本明細書で記載される任意の他の復号器システム又はサブシ

10

20

30

40

50

テムに関して記載されるように、種々のモジュールを具体化するために、処理回路 46 を介して実装されてもよい。処理回路は、後述するように、種々の演算を行うように構成されてもよい。図 5 に示すように、技術が部分的にソフトウェアで実装される場合、デバイスは、好適な非一時的なコンピュータ可読記憶媒体にソフトウェアのための命令を記憶し、本開示の技術を行うために 1 つ以上のプロセッサを使用してハードウェアで命令を実行することができる。ビデオ符号化器 20 とビデオ復号器 30 のいずれかは、例えば図 1 B に示すように、単一のデバイス内の組み合わされた符号化器／復号器（CODED）の一部として一体化されてもよい。

【0090】

ソース・デバイス 12 及び宛先デバイス 14 は、ノートブック又はラップトップ・コンピュータ、携帯電話、スマートフォン、タブレット又はタブレット・コンピュータ、カメラ、デスクトップ・コンピュータ、セット・トップ・ボックス、テレビ、表示デバイス、デジタル・メディア・プレーヤ、ビデオゲーム機、ビデオ・ストリーミング装置（コンテンツ・サービス・サーバ又はコンテンツ配信サーバなど）、放送受信機デバイス、放送送信機装置など、任意の種類の手held・デバイス又は固定デバイスを含む広範囲のデバイスのいずれも含んでもよく、オペレーティング・システムを全く使用しないか、又は任意の種類のものを使用してもよい。場合によっては、ソース・デバイス 12 及び宛先デバイス 14 は、無線通信用に装備されてもよい。したがって、ソース・デバイス 12 及び宛先デバイス 14 は、無線通信デバイスであってもよい。

【0091】

場合によっては、図 1 A に示されたビデオ・コーディング・システム 10 は、単なる一例に過ぎず、本出願の技術は、必ずしも符号化デバイスと復号デバイスとの間の任意のデータ通信を含まないビデオ・コーディング・セッティング（例えば、ビデオ符号化又はビデオ復号）に適用されてもよい。他の例では、データはローカルメモリから検索され、ネットワークを介してストリーミングされるなどである。ビデオ符号化デバイスは、データを符号化し、メモリに記憶してもよく、及び／又はビデオ復号デバイスは、メモリからデータを検索し、復号してもよい。いくつかの例では、符号化及び復号は、互いに通信せず、単にデータをメモリに符号化し、及び／又はメモリからデータを検索及び復号するデバイスによって行われる。

【0092】

説明の便宜上、本発明の実施形態は、例えば、高効率ビデオ・コーディング（HEVC）又は汎用ビデオ・コーディング（VVC）の参照ソフトウェア、ITU-T ビデオ・コーディング・エキスパート・グループ（VCEG）及びISO/IEC モーション・ピクチャ・コーディング・エキスパート・グループ（MPEG）のジョイント・コラボレーション・チーム・オン・ビデオ・コーディング（JCT-VC）によって開発された次世代ビデオ符号化規格を参照して、本明細書に記載される。当業者は、本発明の実施形態が HEVC 又は VVC に限定されないことを理解するであろう。

【0093】

符号化器及び符号化方法

【0094】

図 2 は、本出願の技術を実装するように構成されている例示的なビデオ符号化器 20 の概略ブロック図を示す。図 2 の例では、ビデオ符号化器 20 は、入力 201（又は入力インターフェース 201）、残差計算ユニット 204、変換処理ユニット 206、量子化ユニット 208、逆量子化ユニット 210、逆変換処理ユニット 212、再構成ユニット 214、ループ・フィルタ・ユニット 220、復号されたピクチャ・バッファ（DPB）230、モード選択ユニット 260、エントロピー符号化ユニット 270、出力 272（又は出力インターフェース 272）を含む。モード選択ユニット 260 は、インター予測ユニット 244 と、イントラ予測ユニット 254 と、パーティションニング・ユニット 262 とを含んでもよい。インター予測ユニット 244 は、モーション推定ユニットと、モーション補償ユニット（図示せず）とを含んでもよい。図 2 に示す

ビデオ符号化器 20 は、ハイブリッド・ビデオ符号化器又はハイブリッド・ビデオ・コーデックにしたがったビデオ符号化器とも呼ばれることがある。

【0095】

残差演算ユニット 204、トランスフォーム処理ユニット 206、量子化ユニット 208、モード選択ユニット 260 は符号化器 20 の順方向信号経路を形成するものとして呼ばれることがあるが、逆量子化ユニット 210、逆トランスフォーム処理ユニット 212、再構成ユニット 214、バッファ 216、ループ・フィルタ 220、復号されたピクチャ・バッファ (DPB) 230、インター予測ユニット 244、及びイントラ予測ユニット 254 は、ビデオ符号化器 20 の逆方向信号経路を形成するものとして呼ばれることがあり、ビデオ符号化器 20 の逆方向信号経路は復号器の信号経路に対応する (図 3 のビデオ復号器 30 を参照)。また、逆量子化ユニット 210、逆トランスフォーム処理ユニット 212、再構成ユニット 214、ループ・フィルタ 220、復号されたピクチャ・バッファ (DRB) 230、インター予測ユニット 244、及びイントラ予測ユニット 254 も、ビデオ符号化器 20 の「内蔵復号器」を形成すると呼ばれる。

【0096】

ピクチャ & ピクチャのパーティショニング (ピクチャ & ブロック)

【0097】

符号化器 20 は、例えば入力 201 を介してピクチャ 17 (又はピクチャ・データ 17)、例えばビデオ又はビデオ・シーケンスを形成するピクチャのシーケンスのピクチャを受信するように構成されてもよい。受信されたピクチャ又はピクチャ・データは、前処理されたピクチャ 19 (又は前処理されたピクチャ・データ 19) でもよい。簡単のために、以下の説明は、ピクチャ 17 を指す。また、ピクチャ 17 は、現在のピクチャ又は (特に、現在のピクチャを他のピクチャ、例えば、同じビデオ・シーケンス、すなわち、現在のピクチャも含むビデオ・シーケンスの前に符号化及び/又は復号されたピクチャから区別するためのビデオ・コーディングにおいて) コーディング対象のピクチャ呼ばれることがある。

【0098】

(デジタル) ピクチャは、強度値を有するサンプルの二次元配列又はマトリックスであるか、又はこれとみなされることができる。アレイ中のサンプルは、画素 (ピクチャ要素の省略した形式) 又はペルとも呼ばれることがある。アレイ又はピクチャの水平方向及び垂直方向 (又は軸) のサンプル数は、ピクチャのサイズ及び/又は解像度を定義する。色の表現では、典型的には、3つの色コンポーネントが使用される。すなわち、ピクチャは、3つのサンプル・アレイで表現されるか、又はそれらを含んでもよい。RGB フォーマット又は色空間において、ピクチャは、対応する赤、緑及び青のサンプル・アレイを含む。しかしながら、ビデオ・コーディングにおいては、各画素は、典型的には、輝度及び色差フォーマット又は色空間、例えば、Y によって示される (L が代わりに使用されることもある) 輝度コンポーネント、及びCb及びCrによって示される2つの色差コンポーネントを含むYCbCrで表わされる。輝度 (又は省略してluma) コンポーネントYは、(例えば、グレースケール・ピクチャなどにおける) 明るさ又はグレーレベル強度を表し、一方、2つの色差 (又は省略してchroma) コンポーネントCb及びCrは、色差又は色情報コンポーネントを表す。したがって、YCbCrフォーマットのピクチャは、輝度サンプル値 (Y) の輝度サンプル・アレイと、色差値 (Cb及びCr) の2つの色差サンプル・アレイと、を含む。RGB フォーマットのピクチャはYCbCrフォーマットに変換又はトランスフォームされてもよく、その逆が行われてもよく、プロセスは色トランスフォーム又は変換としても知られる。ピクチャがモノクロである場合、ピクチャは、輝度サンプル・アレイのみを含んでもよい。したがって、ピクチャは、例えば、モノクロフォーマットの輝度サンプルのアレイ、又は輝度サンプルのアレイと4:2:0、4:2:2、及び4:4:4色フォーマットの色差サンプルの2つの対応するアレイであり得る。

【0099】

ビデオ符号化器 20 の実施形態は、ピクチャ 17 を複数の（典型的には重複しない）ピクチャ・ブロック 203 にパーティショニングするように構成されているピクチャ・パーティショニング・ユニット（図 2 には示さず）を含んでもよい。これらのブロックは、ルート・ブロック、マクロ・ブロック（H. 264 / AVC）又はコーディング・ツリー・ブロック（CTB）若しくはコーディング・ツリー・ユニット（CTU）（H. 265 / HEVC 及び VVC）とも呼ばれることがある。ピクチャ・パーティショニング・ユニットは、ビデオ・シーケンスのすべてのピクチャに対して同じブロック・サイズと、ブロック・サイズを定義する対応するグリッドを使用するか、又はピクチャ間、若しくはピクチャのサブセット若しくはグループ間でブロック・サイズを変更し、各ピクチャを対応するブロックにパーティショニングするように構成されてもよい。

10

【0100】

さらなる実施形態では、ビデオ符号化器は、ピクチャ 17 のブロック 203、例えば、ピクチャ 17 を形成する 1 つ、複数の、又はすべてのブロックを直接受信するように構成されてもよい。ピクチャ・ブロック 203 は、現在のピクチャ・ブロック又は符号化されるピクチャ・ブロックとも呼ばれることがある。

【0101】

ピクチャ 17 と同様に、ピクチャ・ブロック 203 は、再度ピクチャ 17 より小さい寸法ではあるが、強度値（サンプル値）を有するサンプルの 2 次元アレイ又はマトリックスであるか、又はそのように見なされる。言い換えれば、ブロック 203 は、例えば、1 つのサンプル・アレイ（例えば、モノクロ・ピクチャ 17 の場合は輝度アレイ、色ピクチャの場合は輝度又は色差アレイ）又は 3 つのサンプル・アレイ（例えば、色ピクチャ 17 の場合は輝度及び 2 つの色差アレイ）又は適用される色フォーマットに依存した任意の他の数及び／又は種類のアレイを含んでもよい。ブロック 203 の方向及び垂直方向（又は軸）のサンプル数は、ブロック 203 のサイズを定義する。したがって、ブロックは、例えば、サンプルの $M \times N$ （ M 列 $\times$ N 行）アレイ、又はトランスフォーム係数の $M \times N$ アレイであってもよい。

20

【0102】

図 2 に示すビデオ符号化器 20 の実施形態は、ブロック毎にピクチャ 17 を符号化するように構成されてもよく、例えば、符号化及び予測はブロック 203 毎に行われる。

【0103】

図 2 に示すビデオ符号化器 20 の実施形態は、スライス（ビデオ・スライスとも呼ばれる）を使用することによってピクチャをパーティショニングおよび／または符号化するようにさらに構成されてもよく、ピクチャは、1 つ以上のスライス（典型的には、重複しない）を使用してパーティショニングまたは符号化されてもよく、各スライスは 1 つ以上のブロック（例えば、CTU）を含んでもよい。

30

【0104】

図 2 に示すビデオ符号化器 20 の実施形態は、タイル・グループ（ビデオ・タイル・グループとも呼ばれる）および／またはタイル（ビデオタイルとも呼ばれる）を用いて、ピクチャをパーティショニングおよび／または符号化するようにさらに構成されてもよく、ピクチャは、1 つ以上のタイル・グループ（典型的には、重複しない）を使用してパーティショニングまたは符号化されてもよく、各タイル・グループは、例えば、1 つ以上のブロック（例えば、CTU）または 1 つ以上のタイルを含んでもよく、各タイルは、例えば、矩形形状であってもよく、1 つ以上のブロック（例えば、CTU）、例えば、完全ブロックまたは部分ブロックを含んでもよい。

40

【0105】

残差計算

【0106】

残差計算ユニット 204 は、サンプル毎（画素毎）に、例えば、ピクチャ・ブロック 203 のサンプル値から予測ブロック 265 のサンプル値を減算することにより、ピクチャ・ブロック 203 及び予測ブロック 265（予測ブロック 265 についてのさらなる詳細

50

は後述する)に基づいて残差ブロック205(残差205とも呼ばれる)を計算して、サンプル・ドメインにおける残差ブロック205を取得するように構成されてもよい。

【0107】

トランスフォーム

【0108】

トランスフォーム処理ユニット206は、残差ブロック205のサンプル値にトランスフォーム、例えば離散コサイン・トランスフォーム(DCT)又は離散サイン・トランスフォーム(DST)を適用して、トランスフォーム・ドメインにおけるトランスフォーム係数207を取得するように構成されてもよい。トランスフォーム係数207は、トランスフォーム残差係数とも呼ばれ、トランスフォーム・ドメインにおける残差ブロック205を表してもよい。

10

【0109】

トランスフォーム処理ユニット206は、H.265/HEVCに対して指定されたトランスフォームなど、DCT/DSTの整数近似を適用するように構成されてもよい。直交DCTトランスフォームと比較して、そのような整数近似は、典型的には、特定のファクタによってスケールされる。順トランスフォームと逆トランスフォームによって処理される残差ブロックのノルムを保存するために、トランスフォーム・プロセスの一部として付加的なスケール・ファクタが適用される。スケール・ファクタは、典型的には、シフト演算のための2のべき乗であるスケール・ファクタ、トランスフォーム係数のビット深度、精度と実装コストの間のトレードオフなどのような特定の制約に基づいて選択される。特定のスケール・ファクタは、例えば、逆トランスフォーム処理ユニット212(及び、例えばビデオ復号器30における逆トランスフォーム処理ユニット312による対応する逆トランスフォーム)によって、逆トランスフォームに対して指定され、符号化器20において、例えば、トランスフォーム処理ユニット206によって、順トランスフォームに対する対応するスケール・ファクタが、それに従って指定されてもよい。

20

【0110】

ビデオ符号化器20(それぞれ、トランスフォーム処理ユニット206)の実施形態は、例えば、エン트로ピー符号化ユニット270を介して直接又は符号化若しくは圧縮された、例えば、トランスフォーム又は複数のトランスフォームのパラメータのタイプを出力するように構成されてもよく、例えば、ビデオ復号器30は、復号のためにトランスフォーム・パラメータを受信及び使用してもよいようにする。

30

【0111】

量子化

【0112】

量子化ユニット208は、例えばスカラー量子化又はベクトル量子化を適用することによって、トランスフォーム係数207を量子化して量子化係数209を取得するように構成されてもよい。量子化係数209は、量子化トランスフォーム係数209又は量子化残差係数209とも呼ばれることがある。

【0113】

量子化プロセスは、トランスフォーム係数207の一部又は全部に関連するビット深度を低減してもよい。例えば、nビット・トランスフォーム係数は、量子化の間、mビット・トランスフォーム係数に丸められてもよく、nは、mより大きい。量子化の程度は、量子化パラメータ(QP)を調整することによって修正されてもよい。例えば、スカラー量子化では、異なるスケールを適用して、より微細又はより粗大な量子化を達成してもよい。より小さい量子化ステップ・サイズはより微細な量子化に対応し、より大きい量子化ステップ・サイズはより粗大な量子化に対応する。適用可能な量子化ステップ・サイズは、量子化パラメータ(QP)によって示されてもよい。量子化パラメータは、例えば、適用可能な量子化ステップ・サイズの予め定義されたセットに対するインデックスであってもよい。例えば、小さな量子化パラメータは、微細な量子化(小さな量子化ステップ・サ

40

50

イズ) に対応してもよく、大きな量子化パラメータは、粗大な量子化 (大きな量子化ステップ・サイズ) に対応してもよく、又はその逆に対応してもよい。量子化は、量子化ステップ・サイズによる除算を含んでもよく、例えば、逆量子化ユニット 210 による、対応する及び／又は逆の量子化解除は、量子化ステップ・サイズによる乗算を含んでもよい。いくつかの規格、例えば H E V C による実施形態は、量子化ステップ・サイズを決定するために量子化パラメータを使用するように構成されてもよい。一般に、量子化ステップ・サイズは、除算を含む式の固定点近似を使用して量子化パラメータに基づいて計算されてもよい。量子化ステップ・サイズ及び量子化パラメータの方程式の固定点近似に使用されるスケーリングのために修正され得る残差ブロックのノルムを復元するために、量子化及び量子化解除のために追加のスケーリング・ファクタを導入してもよい。一例では、逆トランスフォーム及び量子化解除のスケーリングが組み合わされてもよい。代替的には、カスタマイズされた量子化テーブルが使用され、符号化器から復号器へ、例えばビットストリームでシグナリングされてもよい。量子化は、損失が量子化ステップ・サイズの増加に伴って増加する不可逆演算である。

【0114】

ビデオ符号化器 20 (それぞれ、量子化ユニット 208) の実施形態は、例えば、直接又はエントロピー符号化ユニット 270 を介して符号化された量子化パラメータを出力するように構成されてもよく、これにより、例えば、ビデオ復号器 30 は、復号のために量子化パラメータを受信及び適用してもよい。

【0115】

逆量子化

【0116】

逆量子化ユニット 210 は、例えば、量子化ユニット 208 と同じ量子化ステップ・サイズに基づいて、又は同じ量子化ステップ・サイズを使用して、量子化ユニット 208 によって適用される量子化方式の逆を適用することによって、量子化ユニット 208 の逆量子化を量子化係数に適用して、量子化解除係数 211 を取得するように構成される。また、量子化解除係数 211 は、量子化解除残差係数 211 と呼ばれ、典型的には、量子化による損失のためにトランスフォーム係数と同一ではないが、トランスフォーム係数 207 に対応する。

【0117】

逆トランスフォーム

【0118】

逆トランスフォーム処理ユニット 212 は、トランスフォーム処理ユニット 206 によって適用されたトランスフォームの逆トランスフォーム、例えば逆離散コサイン・トランスフォーム (D C T)、逆離散サイン・トランスフォーム、又は他の逆トランスフォームを適用して、サンプル・ドメイン内の再構成された残差ブロック 213 (又は対応する量子化解除係数 213) を取得するように構成されている。再構成された残差ブロック 213 は、トランスフォーム・ブロック 213 と呼ばれる。

【0119】

再構成

【0120】

再構成ユニット 214 (例えば、加算器又は合計器 214) は、トランスフォーム・ブロック 213 (すなわち、再構成された残差ブロック 213) を予測ブロック 365 に加算し、例えば、再構成された残差ブロック 213 のサンプル値及び予測ブロック 265 のサンプル値をサンプル毎に加算することによって、サンプル・ドメイン内の再構成されたブロック 215 を取得するように構成されている。

【0121】

フィルタリング

【0122】

ループ・フィルタ・ユニット 220 (又は略して「ループ・フィルタ」220) は、再

10

20

30

40

50

構成されたブロック 2 1 5 をフィルタリングしてフィルタリングされたブロック 2 2 1 を得るか、又は一般に、再構成されたサンプルをフィルタリングしてフィルタリングされたサンプルを得るように構成されている。ループ・フィルタ・ユニットは、例えば、画素遷移を平滑化するように、又はそれ以外の方法でビデオ品質を改善するように構成されている。ループ・フィルタ・ユニット 2 2 0 は、ブロッキング解除フィルタ、サンプル・アダプティブ・オフセット (SAO) フィルタ、若しくは 1 つ以上の他のフィルタ、例えばバイラテラル・フィルタ、アダプティブ・ループ・フィルタ (ALF)、鮮鋭化フィルタ、平滑化フィルタ、若しくは共同フィルタ、又はそれらの任意の組み合わせなどの 1 つ以上のループ・フィルタを含んでもよい。ループ・フィルタ・ユニット 2 2 0 は、図 2 ではループ内フィルタとして示されているが、他の構成では、ループ・フィルタ・ユニット 2 2 0 は、ポスト・ループ・フィルタとして実装されてもよい。フィルタリングされたブロック 2 2 1 はまた、フィルタリングされた再構成されたブロック 2 2 1 と呼ばれることがある。

【0 1 2 3】

ビデオ符号化器 2 0 (それぞれ、ループ・フィルタ・ユニット 2 2 0) の実施形態は、例えば、直接又はエントロピー符号化ユニット 2 7 0 を介して符号化されたループ・フィルタ・パラメータを出力するように構成されてもよく、これにより、例えば、復号器 3 0 は、復号のために同じループ・フィルタ・パラメータ又はそれぞれのループ・フィルタを受信及び適用してもよい。

【0 1 2 4】

復号されたピクチャ・バッファ

【0 1 2 5】

復号されたピクチャ・バッファ (DRB) 2 3 0 は、ビデオ符号化器 2 0 によってビデオ・データを符号化するために、参照ピクチャ又は一般に、参照ピクチャ・データを記憶するメモリであってもよい。DPB 2 3 0 は、同期 DRAM (SDRAM)、磁気抵抗 RAM (MRAM)、抵抗 RAM (RRAM)、又は他のタイプのメモリ・デバイスを含む、ダイナミック・ランダム・アクセス・メモリなどの種々のメモリ・デバイスのいずれかによって形成されてもよい。復号されたピクチャ・バッファ 2 3 0 は、1 つ以上のフィルタリングされたブロック 2 2 1 を記憶するように構成されてもよい。復号されたピクチャ・バッファ 2 3 0 は、同じ現在のピクチャ又は異なるピクチャ、例えば以前に再構成されたピクチャの、他の以前にフィルタリングされたブロック、例えば以前に再構成され、フィルタリングされたブロック 2 2 1 を記憶するようにさらに構成されてもよく、例えば、インター予測のために、完全に以前に再構成された、すなわち復号されたピクチャ (並びに対応する参照ブロック及びサンプル)) 及び／又は部分的に再構成された現在のピクチャ (並びに対応する参照ブロック及びサンプル) を提供してもよい。復号されたピクチャ・バッファ (DRB) 2 3 0 はまた、例えば、再構成されたブロック 2 1 5 がループ・フィルタ・ユニット 2 2 0 によってフィルタリングされたものではないか、又は再構成されたブロック又はサンプルの任意の他のさらなる処理バージョンではない場合、1 つ以上のフィルタリングされていない構成されたブロック 2 1 5、又は一般に、フィルタリングされていない再構成されたサンプルを記憶するように構成されてもよい。

【0 1 2 6】

モード選択 (パーティショニング & 予測)

【0 1 2 7】

モード選択ユニット 2 6 0 は、パーティショニング・ユニット 2 6 2 と、インター予測ユニット 2 4 4 と、イントラ予測ユニット 2 5 4 を含み、元のピクチャ・データ、例えば元のブロック 2 0 3 (現在のピクチャ 1 7 の現在のブロック 2 0 3) と、再構成されたピクチャ・データ、例えば、フィルタリングされた、又はフィルタリングされていない、同じ (現在の) ピクチャ、及び／又は、1 つ又は複数の以前に復号されたピクチャ、例えば、復号されたピクチャ・バッファ 2 3 0 又は他のバッファ (例えば、ラインバッファ、図示せず) からの再構成されたサンプル又はブロックと、を受信し得るようにより構成さ

れている。再構成されたピクチャ・データは、予測、例えば、インター予測又はイントラ予測のための参照ピクチャ・データとして使用されて、予測ブロック 265 又は予測子 265 を取得する。

【0128】

モード選択ユニット 260 は、現在のブロック予測モード（パーティショニングなしを含む）及び予測モード（例えば、イントラ又はインター予測モード）に対するパーティショニングを決定又は選択し、残差ブロック 205 の計算及び再構成されるブロック 215 の再構成のために使用される対応する予測ブロック 265 を生成するように構成されてもよい。

【0129】

モード選択ユニット 260 の実施形態は、（例えば、モード選択ユニット 260 によってサポートされる、又は利用可能なものから）パーティショニング及び予測モードを選択するように構成されてもよく、これは、最良の一致、言い換えれば、最小の残差（最小の残差は、伝送又は記憶のためのより良い圧縮を意味する）、又は最小のシグナリング・オーバーヘッド（最小のシグナリングは、伝送又は記憶のためのより良いの圧縮を意味する）、又は両方を考慮若しくはバランスをとったものを提供する。モード選択ユニット 260 は、レート歪み最適化（RDO）に基づいて、パーティショニング及び予測モードを決定する、つまり、最小レート歪みを提供する予測モードを選択するように構成されてもよい。この文脈における「最良」、「最低」、「最適」などの用語は、必ずしも全体的な「最良」、「最低」、「最適」などを指すのではなく、終了基準若しくは例えば、閾値を超えるか、又は下回る値のような選択基準、又は「準最適の選択」につながる可能性があるが複雑性及び処理時間を短縮する他の制約条件の達成を指してもよい。

【0130】

言い換えれば、パーティショニング・ユニット 262 は、ブロック 203 を、例えば、クワッド・ツリー・パーティショニング（QT）、バイナリ・パーティショニング（BT）、トリプル・ツリー・パーティショニング（TT）又はそれらの任意の組み合わせを反復的に使用して、より小さなブロック・パーティション又はサブブロック（再度ブロックを形成する）にパーティショニングし、例えば、ブロック・パーティション又はサブブロックごとに予測を実行するように構成されてもよく、モード選択は、パーティショニングされたブロック 203 のツリー構造の選択を含み、予測モードは、ブロック・パーティション又はサブブロックの各々に適用される。

【0131】

以下、例示的なビデオ符号化器 20 によって行われるパーティショニング（例えば、パーティショニング・ユニット 260 による）及び予測処理（インター予測ユニット 244 及びイントラ予測ユニット 254 による）をより詳細に記載する。

【0132】

パーティショニング

【0133】

パーティショニング・ユニット 262 は、現在のブロック 203 を、例えば正方形又は長方形のサイズのより小さいブロックのような、より小さなパーティションにパーティショニング（又は分裂）することができる。これらのより小さなブロック（サブブロックとも呼ばれる）は、さらに小さなパーティションにパーティショニングされてもよい。これは、ツリー・パーティショニング又は階層ツリー・パーティショニングとも呼ばれ、ルート・ブロック、例えば、ルート・ツリー・レベル 0（階層レベル 0、深度 0）では、再帰的にパーティショニングされてもよく、例えば、ツリー・レベル 1（階層レベル 1、深度 1）でのノードのような次の下位ツリー・レベルの 2 つ以上のブロックにパーティショニングされてもよく、これらのブロックは、例えば、最大ツリー深度又は最小ブロック・サイズに達するなど、例えば、終了基準が満たされるために、パーティショニングが終了するまで、例えば、ツリー・レベル 2（階層レベル 2、深度 2）のような次の下位レベルの 2 つ以上のブロックに再度パーティショニングされてもよい。それ以上パーティショニン

グされないブロックは、ツリーのリーフ・ブロック又はリーフ・ノードとも呼ばれる。2つのパーティションへのパーティショニングを使用するツリーは2分木（BT）と呼ばれ、3つのパーティションへのパーティショニングを使用するツリーは3分木（TT）と呼ばれ、4つのパーティションへのパーティショニングを使用するツリーは4分木（QT）と呼ばれる。

【0134】

前述のように、本明細書で使用される場合、用語「ブロック」は、ピクチャの一部分、特に正方形又は長方形の部分であってもよい。例えば、HEVC及びVVCを参照すると、ブロックは、コーディング・ツリー・ユニット（CTU）、コーディング・ユニット（CU）、予測ユニット（PU）、及び変換ユニットに対応してもよく、及び／又は対応するブロック、例えば、コーディング・ツリー・ブロック（CTB）、コーディング・ブロック（CB）、変換ブロック（TB）、又は予測ブロック（PB）に対応してもよい。

【0135】

例えば、コーディング・ツリー・ユニット（CTU）は、輝度サンプルのCTB、3つのサンプル・アレイを有するピクチャの色差サンプルの2つの対応するCTB、又はモノクロ・ピクチャ若しくはサンプルをコーディングするために使用される3つの別々の色プレーン及び構文構造を使用してコーディングされるピクチャのサンプルのCTBであってもよく、又は、これらを含んでもよい。これに対応して、コーディング・ツリー・ブロック（CTB）は、あるコンポーネントのCTBへの分割がパーティショニングであるように、ある値のNのサンプルのN×Nブロックであってもよい。コーディング・ユニット（CU）は、輝度サンプルのCTB、3つのサンプル・アレイを有するピクチャの色差サンプルの2つの対応するコーディング・ブロック、又はモノクロ・ピクチャ若しくはサンプルをコーディングするために使用される3つの別々の色プレーン及び構文構造を使用してコーディングされるピクチャのサンプルのコーディング・ブロックであってもよく、又は、これらを含んでもよい。これに対応して、コーディング・ブロック（CB）は、あるCTBのコーディング・ブロックへの分割がパーティショニングであるように、ある値のM及びNのサンプル値のサンプルのN×Nブロックであってもよい。

【0136】

実施形態において、例えば、HEVCにしたがって、コーディング・ツリー・ユニット（CTU）は、コーディング・ツリーとして示される4分木構造を使用することによって、CUに分裂されてもよい。ピクチャ間（時間的）又はピクチャ内（空間的）予測を使用してピクチャ・エリアをコーディングするかどうかの判定は、CUレベルで行われる。各CUはさらに、PU分裂タイプに従って、1つ、2つ、又は4つのPUに分裂させることができる。1つのPU内では、同じ予測プロセスが適用され、関連情報がPUベースで復号器に送信される。PU分裂タイプに基づく予測プロセスを適用することによって残差ブロックを得た後、CUを、CUのコーディング・ツリーに類似した別のクワッド・ツリー構造に従って変換ユニット（TU）にパーティショニングすることができる。

【0137】

実施形態では、例えば、現在開発中の最新のビデオ・コーディング規格、汎用ビデオ・コーディング（VVC）と呼ばれるものにしたがって、クワッド・ツリーとバイナリー・ツリーの組み合わせ（QTBT）パーティショニングが、例えば、コーディング・ブロックをパーティショニングするために使用される。QTBTブロック構造では、CUは正方形又は長方形のいずれかの形状を有することができる。例えば、コーディング・ツリー・ユニット（CTU）は、最初にクワッド・ツリー構造によってパーティショニングされる。クワッド・ツリー・リーフ・ノードは、バイナリ・ツリー又はターナリー（若しくはトリプル）ツリー構造によってさらにパーティショニングされる。パーティショニング・ツリー・リーフ・ノードは、コーディング・ユニット（CU）と呼ばれ、セグメントは、それ以上のパーティショニングなしで予測及び変換処理のために使用される。これは、CU、PU、TUがQTBTコーディング・ブロック構造において同じブロックサイズを有することを意味する。並列では、複数のパーティション、例えば、トリプル・ツリー・パー

ティションが、Q T B T ブロック構造と共に使用されてもよい。

【0138】

一例では、ビデオ符号化器20のモード選択ユニット260は、本明細書で記載されるパーティショニング技術の任意の組み合わせを行うように構成されてもよい。

【0139】

上述のように、ビデオ符号化器20は、一組の（例えば、予め決定された）予測モードから最良又は最適な予測モードを決定又は選択するように構成される。予測モードのセットは、例えば、イントラ予測モード及び／又はインター予測モードを含んでもよい。

【0140】

イントラ予測

10

【0141】

イントラ予測モードのセットは、35の異なるイントラ予測モード、例えば、D C（又は平均）モード及び平面モードのような非方向モード、又は、例えば、H E V Cで定義されているような方向モードを含むことができ、又は、67の異なるイントラ予測モード、例えば、D C（又は平均）モード及び平面モードのような非方向モード、又は、例えば、V V Cに対して定義されているような方向モードを含むことができる。

【0142】

イントラ予測ユニット254は、イントラ予測モードのセットのイントラ予測モードに従って、同じの現在のピクチャの隣接ブロックの再構成されたサンプルを使用して、イントラ予測ブロック265を生成するように構成されている。

20

【0143】

イントラ予測ユニット254（又は、一般に、モード選択ユニット260）は、符号化されたピクチャ・データ21に含めるための構文要素266の形式で、エントロピー符号化ユニット270にイントラ予測パラメータ（又は、一般に、ブロックに対して選択されたイントラ予測モードを示す情報）を出力するようにさらに構成されており、これにより、例えば、ビデオ復号器30は、復号のために予測パラメータを受信し、使用することができる。

【0144】

インター予測

【0145】

30

インター予測のセット（または可能なインター予測モード）は、利用可能な参照ピクチャ（すなわち、例えば、D B P 2 3 0に記憶された少なくとも部分的に復号されたピクチャ）および他のインター予測パラメータ、例えば、参照ピクチャの全体または参照ピクチャの一部のみ、例えば、現在のブロックのエリアの周囲のサーチ・ウィンドウ・エリアが、最良一致の参照ブロックをサーチするために使用されるかどうか、および／または、例えば、画素内挿、例えば、半分／セミペルおよび／またはクォータ・ペル内挿が適用されるかどうか依存する。

【0146】

上記の予測モードに加えて、スキップ・モード及び／又は直接モードが適用されてもよい。

40

【0147】

インター予測ユニット244は、モーション推定（M E）ユニットと、モーション補償（M C）ユニット（両方とも図2には示さず）を含んでもよい。モーション推定ユニットは、モーション推定のために、ピクチャ・ブロック203（現在のピクチャ17の現在のピクチャ・ブロック203）及び復号されたピクチャ231、又は少なくとも1つ若しくは複数の以前に再構成されたブロック、例えば、1つ又は複数の他の／異なる以前に復号されたピクチャ231の再構成されたブロックを受信又は取得するように構成されてもよい。例えば、ビデオ・シーケンスは、現在のピクチャと以前に復号されたピクチャ231とを含んでもよく、言い換えれば、現在のピクチャと以前に復号されたピクチャ231は、ビデオ・シーケンスを形成するピクチャの一部であってもよく、又は、ピクチャのシ

50

ーケンスを形成してもよい。

【0148】

符号化器20は、例えば、複数の他のピクチャの同じ又は異なるピクチャの複数の参照ブロックから参照ブロックを選択し、参照ブロックの位置（x、y座標）と現在のブロックの位置との間の参照ピクチャ（又は参照ピクチャ・インデックス）及び／又はオフセット（空間オフセット）を、モーション推定ユニットへのインター予測パラメータとして提供するように構成されてもよい。このオフセットはモーション・ベクトル（MV）とも呼ばれる。

【0149】

モーション補償ユニットは、インター予測パラメータを取得、例えば、受信し、インター予測パラメータに基づいて、又はこれを使用して、インター予測を行って、インター予測ブロック265を取得するように構成されている。モーション補償ユニットによって行われるモーション補償は、モーション推定によって決定されたモーション／ブロック・ベクトルに基づいて予測ブロックをフェッチ又は生成することを伴ってもよく、おそらくサブピクセル精度までの補間を行う。補間フィルタリングは、既知の画素サンプルから追加の画素サンプルを生成することができ、したがって、ピクチャ・ブロックをコーディングするために使用され得る候補予測ブロックの数を潜在的に増加させる。現在のピクチャ・ブロックのPUに対するモーション・ベクトルを受信すると、モーション補償ユニットは、参照ピクチャ・リストのうちの1つにおいてモーション・ベクトルがポイントする予測ブロックを位置付けてもよい。

【0150】

モーション補償ユニットはまた、ビデオ・スライスのピクチャ・ブロックを復号する際にビデオ復号器30によって使用するために、ブロック及びビデオ・スライスに関連する構文要素を生成してもよい。スライス及びそれぞれの構文要素に追加して、又はそれらの代替として、タイルグループ及び／又はタイル及びそれぞれの構文要素が生成又は使用されてもよい。

【0151】

エントロピー・コーディング

【0152】

エントロピー符号化ユニット270は、例えば、エントロピー符号化アルゴリズム又は方式（例えば、可変長符号化（VLC）方式、コンテキスト適応VLC方式（CAVLC）、算術コーディング方式、バイナリ化、コンテキスト適応バイナリ算術コーディング（CABAC）、構文ベースのコンテキスト適応バイナリ算術コーディング（SBAC）、確率間隔パーティショニング・エントロピー（PIPE）コーディング、若しくは別のエントロピー符号化方法論又は技法）又はバイパス（非圧縮）を、量子化係数209、インター予測パラメータ、イントラ予測パラメータ、ループ・フィルタ・パラメータ及び／又は他の構文要素に適用して、例えば、符号化ビットストリーム21の形式で出力272を介して出力され得る符号化されたピクチャ・データ21を取得して、これにより、例えば、ビデオ復号器30は、復号のためのパラメータを受信及び使用することができる。符号化されたビットストリーム21は、ビデオ復号器30に送信されてもよいし、後の伝送又はビデオ復号器30による検索のためにメモリに記憶されてもよい。

【0153】

ビデオ符号化器20の他の構造的変形が、ビデオ・ストリームを符号化するために使用することができる。例えば、非トランスフォーム・ベースの符号化器20は、特定のブロック又はフレームに対してトランスフォーム処理ユニット206なしで残差信号を直接量子化することができる。別の実装では、符号化器20は、単一ユニットに組み合わされた量子化ユニット208及び逆量子化ユニット210を有することができる。

【0154】

復号器及び復号方法

【0155】

10

20

30

40

50

図3は、本出願の技術を実装するように構成されたビデオ復号器30の例を示す。ビデオ復号器30は、例えば、符号化器20によって符号化されたピクチャ・データ21（例えば、符号化されたビットストリーム21）を受信して、復号されたピクチャ331を取得するように構成されている。符号化されたピクチャ・データ又はビットストリームは、符号化されたピクチャ・データを復号するための情報、例えば、符号化されたビデオ・スライスのピクチャ・ブロック（及び／又はタイルグループ又はタイル）及び関連する構文要素を表すデータを含む。

【0156】

図3の例では、復号器30は、エントロピー復号ユニット304と、逆量子化ユニット310と、逆トランスフォーム処理ユニット312と、再構成ユニット314（例えば、合計器314）と、ループ・フィルタ320と、復号ピクチャ・バッファ330と、モード適用ユニット360と、インター予測ユニット344と、イントラ予測ユニット354とを含む。インター予測ユニット344は、モーション補償ユニットであってもよいし、又はこれを含んでもよい。ビデオ復号器30は、いくつかの例において、図2からのビデオ符号化器100に関して記載された符号化パスと概ね逆の復号パスを行ってもよい。

【0157】

符号化器20に関して記載したように、逆量子化ユニット210、逆変換処理ユニット212、再構成ユニット214、ループ・フィルタ220、復号されたピクチャ・バッファ（DRB）230、インター予測ユニット344、及びイントラ予測ユニット354も、ビデオ符号化器20の「内蔵復号器」を形成すると呼ばれる。したがって、逆量子化ユニット310は、逆量子化ユニット110と機能的に同一であってもよく、逆トランスフォーム処理ユニット312は、逆トランスフォーム処理ユニット212と機能的に同一であってもよく、再構成ユニット314は、再構成ユニット214と機能的に同一であってもよく、ループ・フィルタ320は、ループ・フィルタ220と機能的に同一であってもよく、復号されたピクチャ・バッファ330は、復号されたピクチャ・バッファ230と機能的に同一であってもよい。したがって、ビデオ復号器30のそれぞれのユニット及び機能には、ビデオ20符号化器のそれぞれのユニット及び機能に対して提供された説明が適用される。

【0158】

エントロピー復号

【0159】

エントロピー復号ユニット304は、ビットストリーム21（又は一般に、符号化されたピクチャ・データ21）を解析し、例えば、符号化されたピクチャ・データ21へのエントロピー復号を行って、例えば、量子化係数309及び／又は復号されたコーディング・パラメータ（図3には示さず）、例えば、インター予測パラメータ（例えば、参照ピクチャ・インデックス及びモーション・ベクトル）、イントラ予測パラメータ（例えば、イントラ予測モード又はインデックス）、変換パラメータ、量子化パラメータ、ループ・フィルタ・パラメータ、及び／又は他の構文要素のいずれか又はすべてを取得するように構成されている。エントロピー復号ユニット304は、符号化器20のエントロピー符号化ユニット270に関して記載したように、符号化方式に対応する復号アルゴリズム又は方式を適用するように構成されてもよい。エントロピー復号ユニット304は、インター予測パラメータ、イントラ予測パラメータ及び／又は他の構文要素をモード適用ユニット360に提供し、他のパラメータを復号器30の他のユニットに提供するようにさらに構成されてもよい。ビデオ復号器30は、ビデオ・スライス・レベル及び／又はビデオ・ブロック・レベルで構文要素を受信してもよい。スライス及びそれぞれの構文要素に追加して、又はそれらの代替として、タイルグループ及び／又はタイル及びそれぞれの構文要素が受信又は使用されてもよい。

【0160】

逆量子化

【0161】

逆量子化ユニット 310 は、（例えば、エントロピー復号ユニット 304 による、例えば、解析及び／又は復号による）符号化されたピクチャ・データ 21 からの量子化パラメータ（QP）（又は、一般には、逆量子化に関係する情報）及び量子化係数を受信し、量子化パラメータに基づいて逆量子化を復号された量子化係数 309 に適用して、トランスフォーム係数 311 とも呼ばれることがある量子化解除係数 311 を取得するように構成されてもよい。逆量子化プロセスは、量子化の程度、及び同様に適用されるべき逆量子化の程度を決定するために、ビデオ・スライス（又はタイル若しくはタイル・グループ）におけるビデオ・ブロックごとにビデオ符号化器 20 によって決定された量子化パラメータを使用することを含んでもよい。

【0162】

逆トランスフォーム

【0163】

逆トランスフォーム処理ユニット 312 は、トランスフォーム係数 311 とも呼ばれる量子化解除係数 311 を受信し、サンプル・ドメインにおいて再構成された残差ブロック 213 を取得するために、量子化解除係数 311 にトランスフォームを適用するように構成されてもよい。再構成された残差ブロック 213 は、トランスフォーム・ブロック 213 とも呼ばれることがある。トランスフォームは、逆変換、例えば、逆 DCT、逆 DST、逆整数変換、又は概念的に類似した逆変換プロセスであってもよい。逆トランスフォーム処理ユニット 312 は、トランスフォーム・パラメータ又は対応する情報を符号化されたピクチャ・データ 21 から受信して（例えば、エントロピー復号ユニット 304 による、例えば、解析及び／又は復号によって）、量子化解除係数 311 に適用されるトランスフォームを決定するようにさらに構成されてもよい。

【0164】

再構成

【0165】

再構成ユニット 314（例えば、加算器又は合計器 314）は、再構成された残差ブロック 313 を予測ブロック 365 に加算し、例えば、再構成された残差ブロック 313 のサンプル値及び予測ブロック 365 のサンプル値を追加することによって、サンプル・ドメインにおける再構成されたブロック 315 を取得するように構成されてもよい。

【0166】

フィルタリング

【0167】

ループ・フィルタ・ユニット 320（コーディング・ループ内又はコーディング・ループ後のいずれか）は、再構成されたブロック 315 をフィルタリングして、フィルタリングされたブロック 321 を取得して、例えば、画素遷移を平滑化するか、又は、それ以外の方法でビデオ品質を改善するように構成されている。ループ・フィルタ・ユニット 320 は、ブロッキング解除フィルタ、サンプル・アダプティブ・オフセット（SAO）フィルタ、若しくは 1 つ以上の他のフィルタ、例えばバイラテラル・フィルタ、アダプティブ・ループ・フィルタ（ALF）、鮮鋭化フィルタ、平滑化フィルタ、若しくは共同フィルタ、又はそれらの任意の組み合わせなどの 1 つ以上のループ・フィルタを含んでもよい。ループ・フィルタ・ユニット 320 は、図 3 ではループ内フィルタとして示されているが、他の構成では、ループ・フィルタ・ユニット 320 は、ポスト・ループ・フィルタとして実装されてもよい。

【0168】

復号されたピクチャ・バッファ

【0169】

次いで、ピクチャの復号されたビデオ・ブロック 321 は、復号されたピクチャ・バッファ 330 に記憶され、復号されたピクチャ・バッファ 330 は、他のピクチャ及び／又はそれぞれ出力表示するためのその後のモーション補償のための参照ピクチャとして記憶される。

10

20

30

40

50

【0170】

復号器30は、ユーザへの提示又は視認のために、例えば出力312を介して復号されたピクチャ311を出力するように構成されている。

【0171】

予測

【0172】

インター予測ユニット344は、インター予測ユニット244（特に、モーション補償ユニット）と同一であってもよく、イントラ予測ユニット354は、機能において、イントラ予測ユニット254と同一であってもよく、（例えば、エントロピー復号ユニット304による、例えば、解析及び／又は復号による）復号されたピクチャ・データ21から受信されたパーティショニング及び／又は予測パラメータ又はそれぞれの情報に基づいて、分裂又はパーティショニング決定及び予測を行う。モード適用ユニット360は、再構成されたピクチャ、ブロック又はそれぞれのサンプル（フィルタリングされた又はフィルタリングされていない）に基づいてブロックごとに予測（イントラ予測又はインター予測）を行って、予測ブロック365を取得するように構成されてもよい。

【0173】

ビデオ・スライスがイントラ・コーディング（I）されたスライスとしてコーディングされるときに、モード適用ユニット360のイントラ予測ユニット354は、シグナリングされたイントラ予測モードと、現在のピクチャの以前に復号されたブロックからのデータとに基づいて、現在のビデオ・スライスのピクチャ・ブロックのための予測ブロック365を生成するように構成されている。ビデオ・ピクチャがインター・コーディング（すなわち、B、又はP）されたスライスとしてコーディングされるときに、モード適用ユニット360のインター予測ユニット344（例えば、モーション補償ユニット）は、エントロピー復号ユニット304から受信されたモーション・ベクトル及び他の構文要素に基づいて、現在のビデオ・スライスのビデオ・ブロックのための予測ブロック365を生成するように構成されている。インター予測のために、予測ブロックは、参照ピクチャ・リストのうちの1つ内の参照ピクチャのうちの1つから生成されてもよい。ビデオ復号器30は、DPB330に記憶された参照ピクチャに基づくデフォルトの構成技術を使用して、参照フレーム・リスト、リスト0及びリスト1を構成してもよい。同じ又は類似のことが、タイル・グループ（例えば、ビデオ・タイル・グループ）及び／又はタイル（例えば、ビデオタイル）をスライス（例えば、ビデオ・スライス）に加えて、又は代替的に使用する実施形態に適用されるか、又はその実施形態によって適用することができ、例えば、ビデオは、I、P又はBタイル・グループ及び／又はタイルを用いてコーディングされてもよい。

【0174】

モード適用ユニット360は、モーション・ベクトル又は関係情報及び他の構文要素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・ブロックのための予測情報を決定するように構成されており、予測情報を使用して、復号される現在のビデオ・ブロックの予測ブロックを生成する。例えば、モード適用ユニット360は、受信された構文要素のいくつかを使用して、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（例えば、イントラ又はインター予測）、インター予測スライス・タイプ（例えば、Bスライス、Pスライス、又はGPBスライス）、スライスのための参照ピクチャ・リストのうちの1つ以上の構成情報、スライスの各インター符号化されたビデオ・ブロックのためのモーション・ベクトル、スライスの各インターコーディングされたビデオ・ブロックのためのインター予測ステータス、及び現在のビデオ・スライスにおけるビデオ・ブロックを復号するための他の情報を決定する。同じ又は類似のことが、タイル・グループ（例えば、ビデオ・タイル・グループ）及び／又はタイル（例えば、ビデオタイル）をスライス（例えば、ビデオ・スライス）に加えて、又は代替的に使用する実施形態に適用されるか、又はその実施形態によって適用することができ、例えば、ビデオは、I、P又はBタイル・グループ及び／又はタイルを用いてコーディングされてもよ

い。

【0175】

図3に示すビデオ符号化器30の実施形態は、スライス（ビデオ・スライスとも呼ばれる）を使用することによってピクチャをパーティショニングおよび／または符号化するようにさらに構成されてもよく、ピクチャは、1つ以上のスライス（典型的には、重複しない）を使用してパーティショニングまたは符号化されてもよく、各スライスは1つ以上のブロック（例えば、CTU）を含んでもよい。

【0176】

図3に示すビデオ復号器30の実施形態は、タイル・グループ（ビデオ・タイル・グループとも呼ばれる）および／またはタイル（ビデオタイルとも呼ばれる）を用いて、ピクチャをパーティショニングおよび／または復号するように構成されてもよく、ピクチャは、1つ以上のタイル・グループ（典型的には、重複しない）を使用してパーティショニングまたは復号されてもよく、各タイル・グループは、例えば、1つ以上のブロック（例えば、CTU）または1つ以上のタイルを含んでもよく、各タイルは、例えば、矩形形状であってもよく、1つ以上のブロック（例えば、CTU）、例えば、完全ブロックまたは部分ブロックを含んでもよい。

【0177】

ビデオ復号器30の他の変形を使用して、符号化されたピクチャ・データ21を復号することができる。例えば、復号器30は、ループ・フィルタリング・ユニット320なしで出力ビデオ・ストリームを生成することができる。例えば、非トランスフォーム・ベースの復号器30は、特定のブロック又はフレームに対して、逆トランスフォーム処理ユニット312なしで残差信号を直接逆量子化することができる。別の実装では、ビデオ復号器30は、単一のユニットに組み合わせられた逆量子化ユニット310及び逆トランスフォーム処理ユニット312を有することができる。

【0178】

符号化器20及び復号器30では、現在のステップの処理結果をさらに処理し、次のステップに出力してもよいと理解されたい。例えば、補間フィルタリング、モーション・ベクトル導出、又はループ・フィルタリングの後に、クリップ又はシフトなどのさらなる演算を、補間フィルタリング、モーション・ベクトル導出、又はループ・フィルタリングの処理結果に対して行ってもよい。

【0179】

さらなる演算は、現在のブロックの導出されたモーション・ベクトル（アフィン・モードの制御点モーション・ベクトル、アフィン、平面、ATMVPモードのサブブロック・モーション・ベクトル、時間的モーション・ベクトルなどを含むが、これらに限定されない）に適用され得ると留意されたい。例えば、モーション・ベクトルの値は、その表現ビットにしたがって予め定義された範囲に制約される。モーション・ベクトルの表現ビットが $bitDepth$ である場合、範囲は、 $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)} - 1$ であり、ここで、「 $\wedge$ 」はべき乗を意味する。例えば、 $bitDepth$ が16にセットされる場合、範囲は $-32768 \sim 32767$ であり、 $bitDepth$ が18にセットされる場合、 $-131072 \sim 131071$ である。例えば、導出されたモーション・ベクトルの値（例えば、1つの 8×8 ブロック内の 4×4 サブブロックのMV）は、4つの 4×4 サブブロックのMVの整数部の間の最大差が、1ピクセル以下などNピクセル以下になるように制約される。ここでは、 $bitDepth$ にしたがってモーション・ベクトルを制約するための2つの方法を提供する。

【0180】

方法1：フロー演算によりオーバーフローMSB（最上位ビット）を除去する。

【数1】

10

20

30

40

50

$$ux = (mvx + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (1)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (2)$$

$$uy = (mvy + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (3)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (4)$$

10

ここで、 mvx は、画像ブロックまたはサブブロックのモーション・ベクトルの水平コンポーネントであり、 mvy は、画像ブロックまたはサブブロックのモーション・ベクトルの垂直コンポーネントであり、 ux および uy は、中間値を示す。例えば、 mvx の値が -32769 である場合、式 (1) 及び (2) を適用した後、結果の値は 32767 である。コンピュータ・システムでは、 10 進数は 2 の補数として記憶される。 -32769 の 2 の補数は $1, 0111, 1111, 1111$ (17 ビット) であり、 MSB が破棄されるため、結果として得られる 2 つの補数は $0111, 1111, 1111, 1111$ (10 進数は 32767) となり、これは、式 (1) と (2) を適用した出力と同じである。

【数 2】

20

$$ux = (mvp_x + mvd_x + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (5)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (7)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

30

演算は、式 (5) ~ (8) に示すように、 mvp と mvd の和の間に適用されてもよい。

【0181】

方法 2：値をクリッピングすることによりオーバーフロー MSB を除去する。

【数 3】

$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vx)$$

$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, vy)$$

40

ここで、 vx は、画像ブロックまたはサブブロックのモーション・ベクトルの水平コンポーネントであり、 vy は、画像ブロックまたはサブブロックのモーション・ベクトルの垂直コンポーネントであり、 x 、 y および z はそれぞれ MV クリッピング・プロセスの 3 つの入力値に対応し、関数 Clip3 の定義は以下の通りである。

【数 4】

50

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

【0182】

図4は、本開示の一実施形態によるビデオ・コーディング・デバイス400の概略図である。ビデオ・コーディング・デバイス400は、本明細書で記載される開示される実施形態を実装するのに好適である。一実施形態では、ビデオ・コーディング・デバイス400は、図1Aのビデオ復号器30のような復号器、又は図1Aのビデオ符号器20のような符号器であってもよい。

【0183】

ビデオ・コーディング・デバイス400は、データを受信するための入口ポート410（又は入力ポート410）及び受信機ユニット（Rx）420と、データを処理するためのプロセッサ、論理ユニット、又は中央処理ユニット（CPU）430と、データを送信するための送信機ユニット（Tx）440及び出口ポート450（又は出力ポート450）と、データを記憶するためのメモリ460を含む。ビデオ・コーディング・デバイス400はまた、光信号又は電気信号の出入りのために、入口ポート410、受信機ユニット420、送信機ユニット440、及び出口ポート450に結合された光から電気（OE）コンポーネント及び電気から光（EO）コンポーネントを含んでもよい。プロセッサ430は、ハードウェア及びソフトウェアによって実装される。プロセッサ430は、1つ以上のCPUチップ、コア（例えば、マルチコアプロセッサ）、FPGA、ASIC、及びDSPとして実装されてもよい。プロセッサ430は、入口ポート410、受信機ユニット420、送信機ユニット440、出口ポート450、及びメモリ460と通信している。プロセッサ430は、コーディング・モジュール470を含む。コーディング・モジュール470は、上述の開示された実施形態を実装する。例えば、コーディング・モジュール470は、種々のコーディング演算を実装、処理、準備、又は提供する。したがって、符号化モジュール470を含めることは、ビデオ・コーディング・デバイス400の機能性への実質的な改善を提供し、ビデオ・コーディング・デバイス400の異なる状態への変換をもたらす。代替的には、コーディング・モジュール470は、メモリ460に記憶され、プロセッサ430によって実行される命令として実装される。

【0184】

メモリ460は、1つ又は複数のディスク、テープ・ドライブ、及びソリッドステート・ドライブを含んでもよく、オーバ・フロー・データ記憶デバイスとして使用され、そのようなプログラムが実行のために選択されたときにプログラムを記憶し、プログラム実行中に読み出された命令及びデータを記憶してもよい。メモリ460は、例えば、揮発性及び／又は不揮発性であってもよく、リード・オンリー・メモリ（ROM）、ランダム・アクセス・メモリ（RAM）、ターナリー・コンテンツ・アドレス指定可能メモリ（TCAM）、及び／又はスタティック・ランダム・アクセス・メモリ（SRAM）であってもよい。

【0185】

図5は、例示的な実施形態による図1からのソース・デバイス12及び宛先デバイス14のいずれか又は両方として使用されてもよい装置500の簡略ブロック図である。

【0186】

装置500内のプロセッサ502は、中央処理ユニットとすることができる。代替的には、プロセッサ502は、現在存在する、又は今後開発される情報を操作又は処理することが可能な、任意の他のタイプのデバイス又は複数のデバイスとすることができる。開示された実装は、示すように単一のプロセッサ、例えば、プロセッサ502を用いて実施することができるが、複数のプロセッサを使用して、速度及び効率における利点を達成する

ことができる。

【0187】

装置500内のメモリ504は、実装におけるリード・オンリー・メモリ（ROM）デバイス又はランダム・アクセス・メモリ（RAM）デバイスとすることができる。メモリ504として、任意の他の好適なタイプの記憶デバイスを使用することができる。メモリ504は、バス512を使用して、プロセッサ502によってアクセスされるコード及びデータ506を含むことができる。メモリ504は、オペレーティング・システム508及びアプリケーション・プログラム510をさらに含むことができ、アプリケーション・プログラム510は、プロセッサ502が本明細書で記載される方法を行うことを許可する少なくとも1つのプログラムを含む。例えば、アプリケーション・プログラム510は、アプリケーション1～Nを含むことができ、これは、本明細書で記載される方法を行うビデオ・コーディング・アプリケーションをさらに含む。

10

【0188】

装置500はまた、ディスプレイ518などの1つ以上の出力デバイスをさらに含むことができる。ディスプレイ518は、一例では、ディスプレイを、タッチ入力を感知するように動作可能なタッチ・センシティブ要素と組み合わせるタッチ・センシティブ・ディスプレイであってもよい。ディスプレイ518は、バス512を介してプロセッサ502に接続することができる。

【0189】

ここでは、単一のバスとして示されているが、装置500のバス512は、複数のバスから構成することができる。さらに、二次ストレージ514は、装置500の他のコンポーネントに直接結合することができ、又はネットワークを介してアクセスすることができ、メモリカードなどの単一の一体化ユニット又は複数のメモリカードなどの複数のユニットを含むことができる。したがって、装置500は、広範な構成で実装することができる。

20

【0190】

それぞれ三角形マージ・モード及び幾何学的マージ・モードとしても知られている三角形パーティショニング・モード（TPM）及び幾何学的モーション・パーティショニング（GEO）は、予測パーティション間の非水平及び非垂直境界を可能にするパーティショニング技術であり、予測ユニットPU1と予測ユニットPU1は、異なる色コンポーネントに関係するそれらのサンプルのサブセットの重み付け平均化手順を使用して領域内で組み合わされる、TPMは、矩形ブロック対角線に沿った予測パーティション間の境界を可能にするが、GEOによる境界は任意の位置に位置してもよい。重み付き平均化手順が適用される領域では、正方形内の整数は、予測ユニットPU1の輝度コンポーネントに適用される重み W_{PU1} を示す。一例では、予測ユニットPU2の輝度コンポーネントに適用される重み W_{PU2} は、以下のように計算される。

30

【数5】

$$W_{PU2} = 8 - W_{PU1}.$$

対応する予測ユニットの色差コンポーネントに適用される重みは、対応する予測ユニットの輝度コンポーネントに適用される重みとは異なってもよい。

40

【0191】

TPMに対する構文の詳細が表1に提示され、4つの構文要素が、TPMについての情報をシグナリングするために使用される。

`MergeTriangleFlag`は、TPMが選択されているかどうかを識別するフラグである（「0」は、TPMが選択されていないことを意味し、それ以外の場合、TPMが選択されている）。

`merge_triangle_split_dir`はTPMに対する分裂方向フラグ（「0」は、分裂方向が左上隅から右下隅までであり、それ以外の場合、分裂方向は、分裂右上隅から左下隅までである）。

50

merge_triangle_idx0及び**merge_triangle_idx1**
は、TPMのために使用されるマージ候補0及び1のインデックスである。

【表 1】

表 1 : TPMのための構文を含むマージ・データ構文

merge_data(x0, y0, cbWidth, cbHeight) {	記述子
if (CuPredMode[x0][y0] == MODE_IBC) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag cbWidth * cbHeight != 32)	
regular_merge_flag[x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_mmvd_enabled_flag && cbWidth * cbHeight != 32)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
if(MergeTriangleFlag[x0][y0]) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)

merge_triangle_idx1[x0][y0]	ae(v)
}	
}	
}	
}	
}	
}	

10

20

30

40

50

一例では、TPMは、

R-L. Liao and C. S. Lim 「CE10. 3. 1. b: Triangular prediction unit mode」、contribution JVET-L0124 to the 12th JVET meeting, Macao, China, October 2018の提案において記載されている。

GEOは、

S. Esenlik, H. Gao, A. Filippov, V. Ruffitskiy, A. M. Kotra, B. Wang, E. Alshina, M. Bläser, and J. Sauer, 「Non-CE4: Geometrical partitioning for inter blocks」、contribution JVET-O0489 to the 15th JVET meeting, Gothenburg, Sweden, July 2019

10

のペーパーにおいて記載されている。

TPM及び／又はGEOをWPと調和させるための開示された方法は、WPが適用されるときにTPM及び／又はGEOを無効にすることである。

第1の実装が表2に示されており、コーディング・ユニットに対してweightedPredFlag変数の値が0に等しいかどうかチェックされる。

20

変数weightedPredFlagは、以下のように導出される。

－ slice_typeがPに等しい場合、weightedPredFlagは、pps_weighted_pred_flagに等しくセットされる。

－ それ以外の場合 (slice_typeがBに等しい場合)、weightedPredFlagは、pps_weighted_bipred_flagに等しくセットされる。

重み付け予測プロセスは、pps_weighted_pred_flag及びsps_weighted_pred_flag構文要素をそれぞれ使用して、ピクチャ・レベル及びスライス・レベルで切り替えられ得る。

上記に開示されているように、変数weightedPredFlagは、スライスのインター予測サンプルを取得するときに、スライス・レベル重み付け予測が使用されるべきかどうかを示す。

30

40

50

【表 2】

表 2 : TPMをWPと調和させるための開示されたマージ・データ構文

merge_data(x0, y0, cbWidth, cbHeight, chType) {	記述子
if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && (sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) (sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B))	
regular_merge_flag [x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag [x0][y0]	ae(v)
if mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
}	
} else {	
if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && weightedPredFlag == 0 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag [x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	

10

20

30

40

50

<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>if(!ciip_flag[x0][y0] && MaxNumTriangleMergeCand > 1) {</code>	
<code>merge_triangle_split_dir[x0][y0]</code>	<code>ae(v)</code>
<code>merge_triangle_idx0[x0][y0]</code>	<code>ae(v)</code>
<code>merge_triangle_idx1[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

10

`ciip_flag[ x0 ][ y0 ]` は、現在のコーディング・ユニットに対して、インターピクチャ・マージとイントラピクチャ予測の組み合わせが適用されるどうかを指定する。配列インデックス `x0`、`y0` は、ピクチャの左上の輝度サンプルに対して考慮されるコーディング・ブロックの左上の輝度サンプルの位置 (`x0`, `y0`) を指定する。

20

`ciip_flag[ x0 ][ y0 ]` が存在しないときに、それは以下のように推論される。

－ 以下の条件がすべて真である場合、`ciip_flag[ x0 ][ y0 ]` は、1 に等しいと推論される。

- － `sps_ciip_enabled_flag` が 1 に等しい。
- － `general_merge_flag[ x0 ][ y0 ]` が 1 に等しい。
- － `merge_subblock_flag[ x0 ][ y0 ]` が 0 に等しい。
- － `regular_merge_flag[ x0 ][ y0 ]` が 0 に等しい。
- － `cbWidth` が 128 より小さい。
- － `cbHeight` が 128 より小さい。
- － `cbWidth * cbHeight` が 64 以上である。

30

－ それ以外の場合、`ciip_flag[ x0 ][ y0 ]` は 0 に等しいと推論される。

`ciip_flag[ x0 ][ y0 ]` が 1 に等しいときに、変数 `IntraPredModeY[ x ][ y ]` (`x = x0 . . x0 + cbWidth - 1` であり、`y = y0 . . y0 + cbHeight - 1` である) は、`INTRA_PLANAR` に等しくセットされる。変数 `MergeTriangleFlag[ x0 ][ y0 ]` は、B スライスを復号するときに、現在のコーディング・ユニットの予測サンプルを生成するために三角形の形状ベースのモーション補償が使用されるかどうかを指定し、以下のように導出される。

－ 以下の条件がすべて真である場合、`MergeTriangleFlag[ x0 ][ y0 ]` は、1 にセットされる。

40

- － `sps_triangle_enabled_flag` が 1 に等しい。
- － `slice_type` が B に等しい。
- － `general_merge_flag[ x0 ][ y0 ]` が 1 に等しい。
- － `MaxNumTriangleMergeCand` が 2 以上である。
- － `cbWidth * cbHeight` が 64 以上である。
- － `regular_merge_flag[ x0 ][ y0 ]` が 0 に等しい。
- － `merge_subblock_flag[ x0 ][ y0 ]` が 0 に等しい。
- － `ciip_flag[ x0 ][ y0 ]` が 0 に等しい。
- － `weightedPredFlag` が 0 に等しい。

50

ー それ以外の場合、MergeTriangleFlag[x0][y0]は0に等しくセットされる。

第2の実装が、表3に提示されている。

weightedPredFlagが1に等しい場合、構文要素max_num_merge_cand_minus_max_num_triangle_candは存在せず、MaxNumTriangleMergeCandが2より小さくなるような値で推論される。

【表3】

表3：TPMをWPと調和させるための開示された一般スライス・ヘッダ構文

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL nal_unit_type == IDR_N_LP nal_unit_type == CRA_NUT NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)

10

20

30

40

50

if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if(slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)

10

20

30

40

50

if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && slice_type == B && !weightedPredFlag && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if(sps_ibc_enabled_flag)	
slice_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)

10

20

30

40

50

slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)

10

20

30

40

50

if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

20

特に、以下のセマンティクスが、第2の実装のために使われ得る。

max_num_merge_cand_minus_max_num_triangle_candは、**MaxNumMergeCand**から差し引かれたスライスでサポートされる三角マージ・モード候補の最大数を指定する。

30

max_num_merge_cand_minus_max_num_triangle_candが存在せず、**sps_triangle_enabled_flag**が1に等しく、**slice_type**がBに等しく、**weightedPredFlag**が0に等しく、かつ**MaxNumMergeCand**が2以上であるときに、**max_num_merge_cand_minus_max_num_triangle_cand**は、**pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1+1**に等しいと推論される。

max_num_merge_cand_minus_max_num_triangle_candが存在せず、**sps_triangle_enabled_flag**が1に等しく、**slice_type**がBに等しく、**weightedPredFlag**が1に等しく、かつ**MaxNumMergeCand**が2以上であるときに、**max_num_merge_cand_minus_max_num_triangle_cand**は、**pMaxNumMergeCand**又は**MaxNumMergeCand-1**に等しいと推論される。

40

三角形マージ・モード候補の最大数**MaxNumTriangleMergeCand**は、以下のように導出される。

MaxNumTriangleMergeCand=**MaxNumMergeCand**—**max_num_merge_cand_minus_max_num_triangle**

50

`e_cand`

`max_num_merge_cand_minus_max_num_triangle_cand`が存在するときに、`MaxNumTriangleMergeCand`の値は、2～`MaxNumMergeCand`の範囲（両端を含む）にあるものとする。

`max_num_merge_cand_minus_max_num_triangle_cand`が存在しない（、かつ`sps_triangle_enabled_flag`が0に等しいか、又は`MaxNumMergeCand`が2より小さい）ときに、`MaxNumTriangleMergeCand`は、0に等しくセットされる。

`MaxNumTriangleMergeCand`が0に等しいときに、現在のスライスに対して三角マージ・モードは許可されない。

10

開示のメカニズムは、TPM及びGEOだけでなく、三角形パーティションを有する組み合わせされたイントラインター予測のような他の非矩形予測及びパーティショニング・モードにも適用可能である。

TPM及びGEOはBスライスにおいてのみ適用されるため、前述の実施形態における変数`weightedPredFlag`は、変数`pps_weighted_bipred_flag`によって直接置き換えられ得る。

第3の実装が表6に示されており、コーディング・ユニットに対して`weightedPredFlag`変数の値が0に等しいかどうかチェックされる。

変数`weightedPredFlag`は、以下のように導出される。

ー 以下の条件がすべて満たされている場合、`weightedPredFlag`は、0にセットされる。

20

`luma_weight_l0_flag[i]`は、0～`NumRefIdxActive[0]`の*i*に対して0に等しい。

`luma_weight_l1_flag[i]`は、0～`NumRefIdxActive[1]`の*i*に対して0に等しい。

`chroma_weight_l0_flag[i]`は、0～`NumRefIdxActive[0]`の*i*に対して0に等しい。

`chroma_weight_l1_flag[i]`は、0～`NumRefIdxActive[1]`の*i*に対して0に等しい。

ー それ以外の場合、`weightedPredFlag`は、1にセットされる。

30

`weightedPredFlag`の導出プロセスは、輝度及び色差コンポーネント、並びに現在のスライスのすべての参照インデックスに対するすべての重み付けフラグが0である場合、現在のスライスにおいて重み付け予測は無効であり、それ以外の場合、現在のスライスに対して重み付け予測が使用されてもよい。

上記に開示されているように、変数`weightedPredFlag`は、スライスのインター予測サンプルを取得するときに、スライス・レベル重み付け予測が使用されるべきかどうかを示す。

第4の実装が表2に示されており、`weightedPredFlag`は、`slice_weighted_pred_flag`に置き換えられており、これは表4に示されているように、スライス・ヘッダにおいてシグナリングされる。

40

上記に開示されているように、構文`slice_weighted_pred_flag`は、スライスのインター予測サンプルを取得するときに、スライス・レベル重み付け予測が使用されるべきかどうかを示す。

【表 4】

表 4：スライス・レベ重み付け予測フラグをシグナルするための開示された一般スライス・ヘッダ

構文	記述子
slice_header() {	
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL nal_unit_type == IDR_N_LP nal_unit_type == CRA_NUT NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)

10

20

30

40

50

if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lf[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if(slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)

10

20

30

40

50

if(slice_type == B && !pps_mvdl1_zero_idc)	
mvdl1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
slice_weighted_pred_flag	u(1)
if(slice_weighted_pred_flag)	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if(sps_ibc_enabled_flag)	
slice_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	

10

20

30

40

50

slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	

10

20

30

40

50

slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

20

特に、以下のセマンティクスが、第4の実装のために使われ得る。

slice_weighted_pred_flagが0に等しいことは、現在のスライスに重み付け予測が適用されないことを指定する。**slice_weighted_pred_flag**が1に等しいことは、現在のスライスに対して重み付け予測が適用されることを指定する。

30

提示されていないときは、**slice_weighted_pred_flag**の値は0に推論される。

第5の実装は、適合性制約によりブロック・レベルにおいてTPMを無効にすることである。

TPMコーディングされるブロックの場合に、インター予測子P₀710及びP₁720（図7に示すように）に対する参照ピクチャの輝度及び色差コンポーネントに対する重み付けファクタは存在するべきではない。

40

より詳細については、**refIdxA**及び**predListFlagA**は、インター予測子P₀の参照インデックス及び参照ピクチャ・リストを指定し、**refIdxB**及び**predListFlagB**は、インター予測子P₁の参照インデックス及び参照ピクチャ・リストを指定する。

変数**lumaWeightedFlag**及び**chromaWeightedFlag**は、以下のように導出される。

【数6】

50

```

lumaWeightedFlagA = predListFlagA ? luma_weight_11_flag[ refIdxA ] : luma_weight_10_flag[ refIdxA ]
lumaWeightedFlagB = predListFlagB ? luma_weight_11_flag[ refIdxB ] : luma_weight_10_flag[ refIdxB ]
chromaWeightedFlagA = predListFlagA ? chroma_weight_11_flag[ refIdxA ] : chroma
weight_10_flag[ refIdxA ]
chromaWeightedFlagB = predListFlagB ? chroma_weight_11_flag[ refIdxB ] : chroma
weight_10_flag[ refIdxB ]
lumaWeightedFlag = lumaWeightedFlagA || lumaWeightedFlagB
chromaWeightedFlag = chromaWeightedFlagA || chromaWeightedFlagB

```

10

lumaWeightedFlag 及び **chromaWeightedFlag** が 0 に等しくあるべきことはビットストリーム適合性の要件である。

第 6 の実装は、明示的な重み付け予測が使用されるときに、TPM コーディングされるブロックに対して混合重み付けサンプル予測プロセスを無効にすることである。

図 7 及び図 8 は、それぞれ TPM 及び GEO の例を例示する。

TPM に対する実施形態は、GEO モードに対しても実装されることがあると留意される。

TPM コーディングされるブロックの場合に、インター予測子 P_{0710} 又は P_{1720} に対する参照ピクチャの輝度又は色差コンポーネントに対する重み付けファクタが存在する場合、インター予測子ブロックを生成するために、WP パラメータ（それぞれ、 P_0 及び P_1 に対する WP パラメータ $730 \{w_0, o_0\}$ 及び WP パラメータ $740 \{w_1, o_1\}$ ）に従った重み付けプロセスが使用され、それ以外の場合、混合重み付けパラメータに従った重み付けプロセスがブロック 750 に対するインター予測子を生成するために使用される。

20

図 9 に示すように、インター予測子 901 は、重複領域 921 を有する 2 つ予測ブロック P_{0911} 及び P_{1912} を必要とし、非ゼロ重みが、予測子 P_{0911} 及び P_{1912} を部分的に混合するために両方のブロック 911 及び 912 に適用される。

ブロック 901 に隣接するブロックは、図 9 において 931 、 932 、 933 、 934 、 935 及び 936 として示される。

30

図 8 は、TPM と GEO マージ・モード間のいくつかの違いを例示する。

GEO マージ・モードの場合に、予測子 851 と 852 との間の重複領域は、インター予測ブロック 850 の対角線に沿ってだけでなく、位置することができる。

予測子 P_{0851} 及び P_{1852} は、ブロック 810 及び 820 それぞれに重み及びオフセット $\{w_0, o_0\}$ 830 及び $\{w_1, o_1\}$ 840 を適用するか又は適用せずに、他のピクチャからブロック 810 及び 820 をコピーすることによって受信することができる。

一例では、**refIdxA** 及び **predListFlagA** は、インター予測子 P_0 の参照インデックス及び参照ピクチャ・リストを指定し、**refIdxB** 及び **predListFlagB** は、インター予測子 P_1 の参照インデックス及び参照ピクチャ・リストを指定する。

40

変数 **lumaWeightedFlag** 及び **chromaWeightedFlag** は、以下のように導出される。

【数 7】

```

lumaWeightedFlagA = predListFlagA ? luma_weight_11_flag[ refIdxA ] : luma_weight_10_flag[ refIdxA ]
lumaWeightedFlagB = predListFlagB ? luma_weight_11_flag[ refIdxB ] : luma_weight_10_flag[ refIdxB ]
chromaWeightedFlagA = predListFlagA ? chroma_weight_11_flag[ refIdxA ] : chroma
weight_10_flag[ refIdxA ]
chromaWeightedFlagB = predListFlagB ? chroma_weight_11_flag[ refIdxB ] : chroma
weight_10_flag[ refIdxB ]
lumaWeightedFlag = lumaWeightedFlagA || lumaWeightedFlagB
chromaWeightedFlag = chromaWeightedFlagA || chromaWeightedFlagB

```

10

次いで、`lumaWeightedFlag`が真である場合、明示的な重み付けプロセスが呼び出され、`lumaWeightedFlag`が偽である場合、混合重み付けプロセスが呼び出される。

同様に、色差コンポーネントは、`chromaWeightedFlag`によって判定される。

代替的な実装の場合、すべてのコンポーネントに対する重み付けフラグは、一緒に考慮される。

`lumaWeightedFlag`又は`chromaWeightedFlag`のうちのいずれかが真である場合、明示的な重み付けプロセスが呼び出され、`lumaWeightedFlag`及び`chromaWeightedFlag`の両方が偽である場合、混合重み付けプロセスが呼び出される。

20

双方向予測メカニズムを使用して予測された矩形ブロックに対する明示的な重み付けプロセスは、以下に記載のように実行される。

このプロセスへの入力は、

- － 現在のコーディング・ブロックの幅及び高さを指定する2つの変数`nCbW`及び`nCbH`、
- － 2つの $(nCbW) \times (nCbH)$ 配列`predSamplesA`及び`predSamplesB`、
- － 予測リスト・フラグ`preListFlagA`及び`preListFlagB`、
- － 参照インデックス`refIdxA`及び`refIdxB`、
- － 色コンポーネント・インデックスを指定する変数`cIdx`、
- － サンプルビット深さ`bitDepth`、である。

30

このプロセスの出力は、予測サンプル値の $(nCbW) \times (nCbH)$ 配列`pSamples`である。

変数`shift1`は、 $\text{Max}(2, 14 - \text{bitDepth})$ に等しくセットされる。

変数`log2Wd`、`o0`、`o1`、`w0`及び`w1`は、以下のように導出される。

- － 輝度サンプルに対して`cIdx`が0に等しい場合、以下が適用される。

【数8】

40

```

log2Wd = luma_log2_weight_denom + shift1
w0 = predListFlagA ? LumaWeightL1[ refIdxA ] : LumaWeightL0[ refIdxA ]
w1 = predListFlagB ? LumaWeightL1[ refIdxB ] : LumaWeightL0[ refIdxB ]
o0 = ( predListFlagA ? luma_offset_11[ refIdxA ] : luma_offset_10[ refIdxA ] ) << ( BitDepthY - 8 )
o1 = ( predListFlagB ? luma_offset_11[ refIdxB ] : luma_offset_10[ refIdxB ] ) << ( BitDepthY - 8 )

```

- － それ以外の場合（色差サンプルに対して`cIdx`が0に等しくない場合）、以下が適用される。

50

【数 9】

```

log2Wd = ChromaLog2WeightDenom + shift1
w0 = predListFlagA ? ChromaWeightL1[ refIdxA ][ cIdx - 1 ] : ChromaWeightL0[ refIdxA ][ cIdx - 1 ]
w1 = predListFlagA ? ChromaWeightL1[ refIdxB ][ cIdx - 1 ] : ChromaWeightL0[ refIdxB ][ cIdx - 1 ]
o0 = ( predListFlagA ? ChromaOffsetL1[ refIdxA ][ cIdx - 1 ] : ChromaOffsetL0[ refIdxA ][ cIdx - 1 ] )
<<< ( BitDepthC - 8 )
o1 = ( predListFlagB ? ChromaOffsetL1[ refIdxB ][ cIdx - 1 ] : ChromaOffsetL0[ refIdxB ][ cIdx - 1 ] )
<<< ( BitDepthC - 8 )

```

10

予測サンプル $pbSamples[x][y]$ ($x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$) は、以下のように導出される。

【数 10】

```

pbSamples[x][y] = Clip3( 0, ( 1 << bitDepth ) - 1, ( predSamplesA[x][y] * w0
predSamplesB[x][y] * w1 + ( ( o0 + o1 + 1 ) << log2Wd ) ) >> ( log2Wd + 1 ) )

```

スライス・レベル重み付け予測のパラメータは、参照ピクチャ・リストの各要素に割り当てられた変数のセットとして表わされ得る。要素のインデックスは、さらに「i」として示される。これらのパラメータは、以下を含み得る。

20

- － $LumaWeightL0[i]$
- － $luma_offset_l0[i]$ は、 $RefPicList[0][i]$ を使用して、リスト0予測のための輝度予測値に適用される追加のオフセットである。 $luma_offset_l0[i]$ の値は、 $-128 \sim 127$ の範囲にある。 $luma_weight_l0_flag[i]$ が0に等しいときに、 $luma_offset_l0[i]$ は、0に等しいと推論される。

変数 $LumaWeightL0[i]$ は、 $(1 << luma_log2_weight_denom) + delta_luma_weight_l0[i]$ と等しくなるように導出される。 $luma_weight_l0_flag[i]$ が1に等しいときに、 $delta_luma_weight_l0[i]$ は、 $-128 \sim 127$ の範囲（両端を含む）にあるとする。 $luma_weight_l0_flag[i]$ が0に等しいときに、 $LumaWeightL0[i]$ は、 $2^{luma_log2_weight_denom}$ に等しいと推論される。

30

双方向予測メカニズムを使用して予測された矩形ブロックに対する混合重み付けプロセスは、以下に記載のように実行される。

このプロセスへの入力は、

- － 現在のコーディング・ブロックの幅及び高さを指定する2つの変数 $nCbW$ 及び $nCbH$ 、
- － 2つの $(nCbW) \times (nCbH)$ 配列 $predSamplesLA$ 及び $predSamplesB$ 、
- － パーティション方向を指定する変数 $triangleDir$ 、
- － 色コンポーネント・インデックスを指定する変数 $cIdx$ 、である。

40

このプロセスの出力は、予測サンプル値の $(nCbW) \times (nCbH)$ 配列 $pbSamples$ である。

変数 $nCbR$ は、以下のように導出される。

【数 11】

$$nCbR = (nCbW > nCbH) ? (nCbW / nCbH) : (nCbH / nCbW)$$

50

変数 $bitDepth$ は、以下のように導出される。

－ $cIdx$ が 0 に等しい場合、 $bitDepth$ は、 $BitDepth_Y$ に等しくセットされる。

－ それ以外の場合、 $bitDepth$ は、 $BitDepth_C$ に等しくセットされる。

変数 $shift1$ 及び $offset1$ は、以下のように導出される。

－ 変数 $shift1$ は、 $Max(5, 17 - bitDepth)$ に等しくセットされる。

－ 変数 $offset1$ は、 $1 << (shift1 - 1)$ にセットされる。

$TriangleDir$ 、 wS 及び $cIdx$ の値に応じて、予測サンプル $pbSamples[x][y]$ ($x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$) は、以下のように導出される。

10

－ 変数 $wIdx$ は、以下のように導出される。

－ $cIdx$ が 0 に等しく、 $triangleDir$ が 0 に等しい場合、以下が適用される。

【数 1 2】

$$wIdx = (nCbW > nCbH) ? (Clip3(0, 8, (x / nCbR - y) + 4)) \\ : (Clip3(0, 8, (x - y / nCbR) + 4))$$

－ それ以外の場合、 $cIdx$ が 0 に等しく、 $triangleDir$ が 1 に等しい場合、以下が適用される。

20

【数 1 3】

$$wIdx = (nCbW > nCbH) ? (Clip3(0, 8, (nCbH - 1 - x / nCbR - y) + 4)) \\ (Clip3(0, 8, (nCbW - 1 - x - y / nCbR) + 4))$$

－ それ以外の場合、 $cIdx$ が 0 より大きく、 $triangleDir$ が 0 に等しい場合、以下が適用される。

【数 1 4】

30

$$wIdx = (nCbW > nCbH) ? (Clip3(0, 4, (x / nCbR - y) + 2)) \\ : (Clip3(0, 4, (x - y / nCbR) + 2))$$

－ それ以外の場合 ($cIdx$ が 0 より大きく、 $triangleDir$ が 1 に等しい場合)、以下が適用される。

【数 1 5】

40

$$wIdx = (nCbW > nCbH) ? (Clip3(0, 4, (nCbH - 1 - x / nCbR - y) + 2)) \\ (Clip3(0, 4, (nCbW - 1 - x - y / nCbR) + 2))$$

－ 予測サンプルの重みを指定する変数 $wValue$ は、以下のように $wIdx$ 及び $cIdx$ を使用して導出される。

【数 1 6】

$$wValue = (cIdx == 0) ? Clip3(0, 8, wIdx) : Clip3(0, 8, wIdx * 2)$$

50

ー 変数サンプル値は、以下のように導出される。

【数 17】

$$\text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (\text{predSamplesLA}[x][y] * \text{wValue} + \text{predSamplesLB}[x][y] * (8 - \text{wValue}) + \text{offset1}) \gg \text{shift1})$$

幾何学的モードの場合、双方向予測メカニズムを使用して予測された矩形ブロックに対する混合重み付けプロセスは、以下のプロセスが、以下に記載のように実行される。

このプロセスへの入力は、

- ー 現在のコーディング・ブロックの幅及び高さを指定する2つの変数 $nCbW$ 及び $nCbH$ 、
- ー 2つの $(nCbW) \times (nCbH)$ 配列 predSamplesLA 及び predSamplesB 、
- ー 幾何学的パーティションの角度インデックスを指定する変数 angleIdx 、
- ー 幾何学的パーティションの距離 idx を指定する変数 distanceIdx 、
- ー 色コンポーネント・インデックスを指定する変数 $cIdx$ 、である。

このプロセスの出力は、予測サンプル値の $(nCbW) \times (nCbH)$ 配列 pbSamples 及び変数 partIdx である。

変数 bitDepth は、以下のように導出される。

- ー $cIdx$ が 0 に等しい場合、 bitDepth は、 BitDepth_Y に等しくセットされる。
 - ー それ以外の場合、 bitDepth は、 BitDepth_C に等しくセットされる。
- 変数 shift1 及び offset1 は、以下のように導出される。
- ー 変数 shift1 は、 $\text{Max}(5, 17 - \text{bitDepth})$ に等しくセットされる。
 - ー 変数 offset1 は、 $1 \ll (\text{shift1} - 1)$ にセットされる。

重み配列 $\text{sampleWeight}_L[x][y]$ (輝度に対するもの) 及び $\text{sampleWeight}_C[x][y]$ (色差に対するもの) ($x = 0 \dots nCbW - 1$ であり、 $y = 0 \dots nCbH - 1$) が、以下のように導出される。

以下の変数の値がセットされる。

- ー hwRatio が $nCbH / nCbW$ にセットされる。
- ー displacementX が angleIdx にセットされる。
- ー displacementY が $(\text{displacementX} + 8) \% 32$ にセットされる。
- ー partIdx が $\text{angleIdx} \geq 13 \ \& \ \text{angleIdx} \leq 27 ? 1 : 0$ にセットされる。
- ー rho は、表 8-12 で指定された Dis として示されるルックアップテーブルを使用して以下の値にセットされる。

【数 18】

$$\text{rho} = (\text{Dis}[\text{displacementX}] \ll 8) + (\text{Dis}[\text{displacementY}] \ll 8)$$

以下の条件うちの1つが真である場合、変数 shiftHor が 0 に等しくセットされる。

$\text{angleIdx} \% 16$ が 8 に等しく、
 $\text{angleIdx} \% 16$ が 0 と等しくなく、 $\text{hwRatio} \geq 1$ である。

それ以外の場合、 shiftHor が 1 にセットされる。

shiftHor が 0 に等しい場合、 offsetX と offsetY は、以下のように導出される。

【数 19】

10

20

30

40

50

```

offsetX = ( 256 - nCbW ) >> 1,
offsetY = ( 256 - nCbH ) >> 1 + angleIdx < 16 ? (distanceIdx * nCbH) >> 3 : -((distanceIdx
* nCbH) >> 3)

```

それ以外の場合、`shiftHor`が1に等しい場合、`offsetX`及び`offsetY`は、以下のように導出される。

【数 2 0】

```

offsetX = ( 256 - nCbW ) >> 1 + angleIdx < 16 ? (distanceIdx * nCbW) >> 3 : -((distanceIdx
* nCbW) >> 3)
offsetY = ( 256 - nCbH ) >> 1

```

10

変数`weightIdx`及び`weightIdxAbs`は、以下のようにルックアップ表9を使用して計算される（ $x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$ ）。

【数 2 1】

```

weightIdx = (((x + offsetX) << 1) + 1) * Dis[displacementX]
+ (((y + offsetY) << 1) + 1) * Dis[displacementY] - rho.
weightIdxAbs = Clip3(0, 26, abs(weightIdx)).

```

20

`SampleWeightL[x][y]`（ $x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$ ）の値は、`GeoFilter`と示される表10に従ってセットされる。

【数 2 2】

```

sampleWeightL[x][y] = weightIdx <= 0 ? GeoFilter[weightIdxAbs] : 8 - GeoFilter[weightIdxAbs]

```

`sampleWeightC[x][y]`（ $x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$ ）の値は、以下のようにセットされる。

30

【数 2 3】

```

sampleWeightC[x][y] = sampleWeightL[(x << (SubWidthC - 1))][(y << (SubHeightC - 1))]

```

注 - `SampleWeightL[x][y]`の値は、`SampleWeightL[x - shiftX][y - shiftY]`からも導出され得る。`angleIdx`が4より大きく12より小さい場合、又は`angleIdx`が20より大きく24より小さい場合、`shiftX`は分裂角度のタンジェントであり、`shiftY`は1であり、それ以外の場合、`shiftX`は分裂角度の1であり、シフトYは分割角度のコタンジェントである。タンジェント（それぞれコタンジェント）値が無大である場合、`shiftX`は1（それぞれ0）であるか、又は`shiftY`は0（それぞれ1）である。予測サンプル値は以下のように導出され、Xは、L又はCとして示され、`cIdx`は0に等しいか、又は0に等しくない。

40

【数 2 4】

50

pbSamples[x][y] = partIdx ?

Clip3(0, (1 << bitDepth) - 1, (predSamplesLA[x][y] * (8 - sampleWeightx[x][y]) +
predSamplesLB[x][y] * sampleWeightx[x][y] + offset1) >> shift1)

: Clip3(0, (1 << bitDepth) - 1, (predSamplesLA[x][y] * sampleWeightx[x][y] +
predSamplesLB[x][y] * (8 - sampleWeightx[x][y]) + offset1) >> shift1)

10

【表 5】

表 5－幾何学的パーティション化距離の導出のためのルックアップ表 Dis

idx	0	1	2	4	6	7	8	9	10	12	14	15
Dis[idx]	8	8	8	8	4	2	0	-2	-4	-8	-8	-8
idx	16	17	18	20	22	23	24	25	26	28	30	31
Dis[idx]	-8	-8	-8	-8	-4	-2	0	2	4	8	8	8

20

【表 6】

表 6－幾何学的パーティション化フィルタ重みの導出のためのフィルタ重みルックアップ表

GeoFilter

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
GeoFilter[idx]	4	4	4	4	5	5	5	5	5	5	5	6	6	6
idx	14	15	16	17	18	19	20	21	22	23	24	25	26	
GeoFilter[idx]	6	6	6	6	7	7	7	7	7	7	7	7	8	

30

VVC使用ドラフト7（文書JVET-P2001-vE: B. Bross, J. Chen, S. Liu, Y. K. Wang, 「Versatile Video Coding (Draft 7)」, the 16th JVET meeting, Geneva, Switzerlandの出力文書JVET-P2001、本文書は、ファイルJVET-P2001-v14: http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/16_Geneva/wg11/JVET-P2001-v14.zipに含まれる）では、PHに関連する各SHにおける同じ構文要素に等しい又は類似の値を割り当てることによって生じるシグナリング・オーバーヘッドを低減するために、スライス・ヘッダ（SH）からの構文要素の一部をピクチャヘッダ（PH）に移動することによって、PHの概念が導入された。表7に提示されるように、TPMマージ・モードに対するマージ候補の最大数を制御するための構文要素は、PHにおいてシグナリングされているが、重み付き予測パラメータは、表8及び表10に示すように依然としてSHにおけるものである。表8及び表9において使用される構文要素のセマンティクスは、以下に記載される。

40

50

【表 7】

表 7ーピクチャ・ヘッダ RBSP 構文

picture_header_rbsp() {	記述子
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rbbsp_trailing_bits()	
}	

10

20

ピクチャ・ヘッダ RBSP セマンティクス

PHは、PHに関連するコードディングされたピクチャのすべてのスライスに共通する情報を含む。

non_reference_picture_flagが1に等しいことは、PHに関連するピクチャが参照ピクチャとして使用されることはないことを指定する。non_reference_picture_flagが0に等しいことは、PHに関連するピクチャが参照ピクチャとして使用されることがあるかどうかを指定する。

30

gdr_pic_flagが1に等しいことは、PHに関連するピクチャが漸次復号リフレッシュ（GDR）ピクチャであることを指定する。gdr_pic_flagが0に等しいことは、PHに関連するピクチャがGDRピクチャではないことを指定する。

no_output_of_previous_pics_flagは、ビットストリームにおける第1のピクチャではないコーディングされたレイヤ・ビデオ・シーケンス開始（CLVSS）ピクチャの復号後に、復号されたピクチャ・バッファ（DPB）における以前に復号された画像の出力に影響を与える。

recovery_poc_cntは、復号されたピクチャのリカバリ・ポイントを出力順序で指定する。

40

現在のピクチャがPHに関連付けられたGDRピクチャであり、コーディングされたレイヤ・ビデオ・シーケンス（CLVS）において復号順序における現在のGDRピクチャに続き、かつ現在のGDRピクチャのPicOrderCntValにrecovery_poc_cntの値を加えたものに等しいPicOrderCntValを有するピクチャPicAがある場合、ピクチャpicAは、リカバリ・ポイント・ピクチャと呼ばれる。それ以外の場合、現在のピクチャのPicOrderCntValにrecovery_poc_cntの値を加えたものより大きいPicOrderCntValを持つ出力順序における最初のピクチャは、リカバリ・ポイント・ピクチャと呼ばれる。

リカバリ・ポイント画像は、復号順序において現在のGDR画像に先行しないものとする。recovery_poc_cntの値は、0～MaxPicOrderCntLsb-1の範囲（両端を含む）にあるものとする。

50

注1 — `gdr_enabled_flag`が1に等しく、現在のピクチャの`PicOrderCntVal`が関連するGDRピクチャの`RpPicOrderCntVal`以上であるときに、出力順序における現在及び後続の復号されるピクチャは、復号順序において関連するGDRピクチャから先行する、以前のイントラ・ランダム・アクセス・ポイント（RAP）（存在するとき）から復号処理を開始することによって生成された対応するピクチャに完全に一致する。

`ph_pic_parameter_set_id`は、使用中のPPSに対する`pps_pic_parameter_set_id`の値を指定する。

`ph_pic_parameter_set_id`の値は、0～63の範囲（両端を含む）にあるものとする。

10

PHの`TemporalId`の値が、`ph_pic_parameter_set_id`に等しい`pps_pic_parameter_set_id`を有するピクチャ・パラメータ・セット（PPS）の`TemporalId`の値以上であるものとすることは、ビットストリーム適合性の要件である。

`sps_poc_msb_flag`が1に等しいことは、`ph_poc_msb_cycle_present_flag`構文要素がシーケンス・パラメータ・セット（SPS）を参照するPHにおいて存在することを指定する。`sps_poc_msb_flag`が0に等しいことは、`ph_poc_msb_cycle_present_flag`構文要素がSPSを参照するPHにおいて存在しないことを指定する。

`ph_poc_msb_present_flag`が1に等しいことは、構文要素`poc_msb_val`がPHにおいて存在することを指定する。`ph_poc_msb_present_flag`が0に等しいことは、構文要素`poc_msb_val`がPHにおいて存在しないことを指定する。

20

`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]`が0に等しく、現在のレイヤの参照レイヤにおいて現在のアクセス・ユニット（AU）においてピクチャがあるときに、`ph_poc_msb_present_flag`の値は0に等しいものとする。

`poc_msb_val`は、現在のピクチャのピクチャ・オーダ・カウント（POC）最上位ビット（MSB）値を指定する。

構文要素`poc_msb_val`の長さは、`poc_msb_len_minus1+1`ビットである。

30

`sps_triangle_enabled_flag`は、三角形形状ベースのモーション補償がインター予測のために使用され得るかどうかを指定する。0に等しい`sps_triangle_enabled_flag`は、三角形形状ベースのモーション補償がコーディングされたレイヤ・ビデオ・シーケンス（CLVS）において使用されないように、構文が制約されるものとするを指定し、`merge_triangle_split_dir`、`merge_triangle_idx0`、及び`merge_triangle_idx1`は、CLVSのコーディング・ユニット構文に存在しない。1に等しい`sps_triangle_enabled_flag`は、三角形形状ベースのモーション補償がCLVSにおいて使用され得ることを指定する。

40

`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`が0に等しいことは、`pic_max_num_merge_cand_minus_max_num_triangle_cand`がピクチャ・パラメータ・セット（PPS）

を参照するスライスのPHにおいて存在することを指定する。`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`が0より大きいことは、`pic_max_num_merge_cand_minus_max_num_triangle_cand`が存在しないことを指定する。

`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`の値は、0～`MaxNumMergeCand-1`の

50

範囲（両端を含む）にあるものとする。

`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`が0に等しいことは、`pic_max_num_merge_cand_minus_max_num_triangle_cand`が、PPSを参照するスライスのPHにおいて存在することを指定する。`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`が0より大きいことは、`pic_max_num_merge_cand_minus_max_num_triangle_cand`が、PPSを参照するPHにおいて存在しないことを指定する。

`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`の値は、0～`MaxNumMergeCand-1`の範囲（両端を含む）にあるものとする。

`pic_six_minus_max_num_merge_cand`は、6から差し引かれた、PHに関連するスライスでサポートされるモーション・ベクトル予測（MVP）候補をマージする最大数を指定する。

MVP候補をマージする最大数`MaxNumMergeCand`は、以下のように導出される。

【数25】

$$\text{MaxNumMergeCand} = 6 - \text{picsix_minus_max_num_merge_cand}$$

`MaxNumMergeCand`の値は、1～6の範囲にあるものとする。

存在しないときに、`pic_six_minus_max_num_merge_cand`の値は、`pps_six_minus_max_num_merge_cand_plus1-1`に等しいと推論される。

10

20

30

40

50

【表 8】

表 8 一般スライス・ヘッダ構文

slice_header() {	記述子
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag && (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
slice_poc_lsb_ltrp[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_ltrp[i][j]	ue(v)
}	
}	
}	
if(pic_rpl_present_flag (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	

10

20

30

40

50

if((slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(slice_type != I) {	
...	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
}	
...	
byte_alignment()	
}	

10

20

一般的なスライス・ヘッダ・セマンティクス

存在するときに、スライス・ヘッダ構文要素 `slice_pic_order_cnt_lsb` の値は、コーディングされたピクチャのすべてのスライス・ヘッダにおいて同じである。

`cu_qp_delta_abs` を含むコーディング・ユニットの輝度量子化パラメータとその予測との差を指定する変数 `CuQpDeltaVal` は、0 に等しくセットされる。
`cu_chroma_qp_offset_flag` を含むコーディング・ユニットの $Q_p'_{cb}$ 、 $Q_p'_{cr}$ 、 $Q_p'_{cbcr}$ 量子化パラメータのそれぞれの値を決定するときに使用される値を指定する変数 `CuQpOffsetCb`、`CuQpOffsetCr`、`CuQpOffsetCbCr` は、すべて 0 に等しくセットされる。

30

`slice_pic_order_cnt_lsb` は、現在のピクチャのピクチャ・オーダー・カウント・モジュロ `MaxPicOrderCntLsb` を指定する。

`slice_pic_order_cnt_lsb` 構文要素の長さは、`log2_max_pic_order_cnt_lsb_minus4 + 4` ビットである。

`slice_pic_order_cnt_lsb` の値は、 $0 \sim \text{MaxPicOrderCntLsb} - 1$ の範囲（両端を含む）にあるものとする。

現在のピクチャが GDR ピクチャであるときに、変数 `RpPicOrderCntVal` は、以下のように導出される。

40

【数 26】

$$\text{RpPicOrderCntVal} = \text{PicOrderCntVal} + \text{recovery_poc_cnt}.$$

`slice_subpic_id` は、スライスを含むサブピクチャのサブピクチャ識別子を指定する。`slice_subpic_id` が存在するときに、変数 `SubPicIdx` の値は、`SubpicIdList[SubPicIdx]` が `slice_subpic_id` に等しくなるように導出される。それ以外の場合（`slice_subpic_id` が存在しない場合）、変数 `SubPicIdx` は、0 に等しくなるように導出される

50

。 `slice_subpic_id` のビット長は、以下のように導出される。

－ `sps_subpic_id_signalling_present_flag` が 1 に等しい場合、 `slice_subpic_id` の長さは、 `sps_subpic_id_len_minus1+1` に等しい。

－ それ以外の場合、 `ph_subpic_id_signalling_present_flag` が 1 に等しい場合、 `slice_subpic_id` の長さは、 `ph_subpic_id_len_minus1+1` に等しい。

－ それ以外の場合、 `pps_subpic_id_signalling_present_flag` が 1 に等しい場合、 `slice_subpic_id` の長さは、 `pps_subpic_id_len_minus1+1` に等しい。

－ それ以外の場合、 `slice_subpic_id` の長さは、 $\text{Ceil}(\text{Log}_2(\text{sps_num_subpics_minus1}+1))$ に等しい。

`slice_address` は、スライスのスライス・アドレスを指定する。

存在しないときに、 `slice_address` の値は、 0 に等しいと推論される。

`rect_slice_flag` が 0 に等しい場合、以下が適用される。

－ スライス・アドレスは、ラスタ・スキャン・タイル・インデックスである。

－ `slice_address` の長さは、 $\text{Ceil}(\text{Log}_2(\text{NumTilesInPic}))$ ビットである。

－ `slice_address` の値は、 $0 \sim \text{NumTilesInPic}-1$ の範囲（両端を含む）にあるものとする。

それ以外の場合（ `rect_slice_flag` が 1 に等しい場合）、以下が適用される。

－ スライス・アドレスは、 `SubPicIdx` 番目のサブピクチャ内のスライスのスライス・インデックスである。

－ `slice_address` の長さは、 $\text{Ceil}(\text{Log}_2(\text{NumSlicesInSubpic}[\text{SubPicIdx}]))$ ビットである。

－ `slice_address` の値は、 $0 \sim \text{NumSlicesInSubpic}[\text{SubPicIdx}]-1$ の範囲（両端を含む）にあるものとする。

以下の制約が適用されることがビットストリーム適合性の要件である。

－ `rect_slice_flag` が 0 に等しい場合、又は `subpics_present_flag` が 0 に等しい場合、 `slice_address` の値は、同じコーディングされたピクチャの任意の他のコーディングされたスライス・ネットワーク抽象化レイヤ（NAL）ユニットの `slice_address` の値に等しくないものとする。

－ それ以外の場合、 `slice_subpic_id` 及び `slice_address` の値の対は、同じコーディングされたピクチャの任意の他のコーディングされたスライス NAL ユニットの `slice_subpic_id` 及び `slice_address` の値の対に等しくないものとする。

－ `rect_slice_flag` が 0 に等しいときに、ピクチャのスライスはそれらの `slice_address` 値の増加順にあるものとする。

－ ピクチャのスライスの形状は、各コーディング・ツリー・ユニット（CTU）が、復号されるときに、ピクチャ境界からなるか、以前に復号された CTU（複数可）の境界からなるその全体の左境界及び全体の上境界を有するものとする。

`num_tiles_in_slice_minus1+1` は、存在するときに、スライス内のタイルの数を指定する。 `num_tiles_in_slice_minus1` の値は、 $0 \sim \text{NumTilesInPic}-1$ の範囲（両端を含む）にあるものとする。

現在のスライス内の CTU の数を指定する変数 `NumCtuInCurrSlice` と、スライス内の i 番目のコーディング・ツリー・ブロック（CTB）のピクチャ・ラスタ・スキャン・アドレスを指定する、 $0 \sim \text{NumCtuInCurrSlice}-1$ の範囲（両端を含む）の i に対するリスト `CtbAddrInCurrSlice[i]` は、以下のように導出される。

10

20

30

40

50

【数 2 7】

```

if( rect_slice_flag ) {

    picLevelSliceIdx = SliceSubpicToPicIdx[ SubPicIdx ][ slice_address ]

    NumCtuInCurrSlice = NumCtuInSlice[ picLevelSliceIdx ]

    for( i = 0; i < NumCtuInCurrSlice; i++ )

        CtbAddrInCurrSlice[ i ] = CtbAddrInSlice[ picLevelSliceIdx ][ i ]

} else {

    NumCtuInCurrSlice = 0

    for( tileIdx = slice_address; tileIdx <= slice_address + num_tiles_in_slice_minus1[ i ];
tileIdx++ ) {

        tileX = tileIdx % NumTileColumns

        tileY = tileIdx / NumTileColumns

        for( ctbY = tileRowBd[ tileY ]; ctbY < tileRowBd[ tileY + 1 ]; ctbY++ ) {

            for( ctbX = tileColBd[ tileX ]; ctbX < tileColBd[ tileX + 1 ]; ctbX++ ) {

                CtbAddrInCurrSlice[ NumCtuInCurrSlice ] = ctbY * PicWidthInCtb + ctbX

                NumCtuInCurrSlice++

            }

        }

    }

}

```

変数 **SubPicLeftBoundaryPos**、**SubPicTopBoundaryPos**、**SubPicRightBoundaryPos**、及び **SubPicBotBoundaryPos** は、以下のように導出される。

【数 2 8】

10

20

30

40

50

```

if( subpic_treated_as_pic_flag[ SubPicIdx ] ) {
    SubPicLeftBoundaryPos = subpic_ctu_top_left_x[ SubPicIdx ] * CtbSizeY
    SubPicRightBoundaryPos = Min( pic_width_max_in_luma_samples - 1,
        ( subpic_ctu_top_left_x[ SubPicIdx ] + subpic_width_minus1[ SubPicIdx ] + 1 ) * CtbSizeY -
    1)
    SubPicTopBoundaryPos = subpic_ctu_top_left_y[ SubPicIdx ] * CtbSizeY
    SubPicBotBoundaryPos = Min( pic_height_max_in_luma_samples - 1,
        ( subpic_ctu_top_left_y[ SubPicIdx ] + subpic_height_minus1[ SubPicIdx ] + 1 ) * CtbSizeY
    - 1)
}

```

10

`slice_type` は、表 13 に従ってスライスのコーディング・タイプを指定する。
【表 9】

表 9 — `slice_type` への名前の関連付け

20

<code>slice_type</code>	<code>slice_type</code> の名前
0	B (B slice)
1	P (P slice)
2	I (I slice)

30

`slice_rpl_sps_flag[i]` が 1 に等しいことは、現在のリストの参照ピクチャ・リスト `i` が、SPS において `i` に等しい `listIdx` を有する `ref_pic_list_struct(listIdx, rplIdx)` 構文構造のうちの 1 つに基づいて導出されることを指定する。`slice_rpl_sps_flag[i]` が 0 に等しいことは、現在のスライスの参照ピクチャ・リスト `i` が、現在のピクチャのスライス・ヘッダに直接含まれる `i` に等しい `listIdx` を有する `ref_pic_list_struct(listIdx, rplIdx)` 構文構造に基づいて導出されることを指定する。

40

`slice_rpl_sps_flag[i]` が存在しないときに、以下が適用される。
— `pic_rpl_present_flag` が 1 に等しい場合、`slice_rpl_sps_flag[i]` の値は `pic_rpl_sps_flag[i]` に等しいと推論される。

— それ以外の場合、`num_ref_pic_lists_in_sps[i]` が 0 に等しいときに、`ref_pic_list_sps_flag[i]` の値は、0 に等しいと推論される。

— それ以外の場合、`num_ref_pic_lists_in_sps[i]` が 0 よ

50

り大きい場合かであって、`rpl1_idx_present_flag`が0に等しい場合に、`slice_rpl_sps_flag[1]`の値は、`slice_rpl_sps_flag[0]`に等しいと推論される。

`slice_rpl_idx[i]`は、現在のピクチャの参照ピクチャ・リスト*i*の導出のために使用される*i*に等しい`listIdx`を有する`ref_pic_list_struct(listIdx, rplIdx)`構文構造の、SPSに含まれる*i*に等しい`listIdx`を有する`ref_pic_list_struct(listIdx, rplIdx)`構文構造のリストにインデックスを指定する。

構文要素`slice_rpl_idx[i]`は、`Ceil(Log2(num_ref_pic_lists_in_sps[i]))`ビットによって表される。

10

存在しないときに、`slice_rpl_idx[i]`の値は、0に等しいと推論される。

`slice_rpl_idx[i]`の値は、0～`num_ref_pic_lists_in_sps[i] - 1`の範囲（両端を含む）にあるものとする。

`slice_rpl_sps_flag[i]`が1に等しく、`num_ref_pic_lists_in_sps[i]`が1に等しいときに、`slice_rpl_idx[i]`の値は、0に等しいと推論される。

`slice_rpl_sps_flag[i]`が1に等しく、かつ`rpl1_idx_present_flag`が0に等しいときに、`slice_rpl_idx[1]`の値は、`slice_rpl_idx[0]`に等しいと推論される。

変数`RplIdx[i]`は、以下のように導出される。

20

【数29】

```
if( pic_rpl_present_flag )
```

```
    RplIdx[ i ] = PicRplIdx[ i ]
```

```
else
```

```
    RplIdx[ i ] = slice_rpl_sps_flag[ i ] ? slice_rpl_idx[ i ] : num_ref_pic_lists_in_sps[ i ]
```

30

`slice_poc_lsb_lt[i][j]`は、*i*番目の参照ピクチャ・リストにおける*j*番目のLTRPエントリのピクチャ・オーダ・カウント・モジュロ`MaxPicOrderCntLsb`の値を指定する。

`slice_poc_lsb_lt[i][j]`構文要素の長さは、`log2_max_pic_order_cnt_lsb_minus4 + 4`ビットである。

変数`PocLsbLt[i][j]`は、以下のように導出される。

【数30】

```
if( pic_rpl_present_flag )
```

```
    PocLsbLt[ i ][ j ] = PicPocLsbLt[ i ][ j ]
```

40

```
else
```

```
    PocLsbLt[ i ][ j ] = ltrp_in_slice_header_flag[ i ][ RplIdx[ i ] ] ?
```

```
        slice_poc_lsb_lt[ i ][ j ] : rpls_poc_lsb_lt[ listIdx ][ RplIdx[ i ] ][ j ]
```

`slice_delta_poc_msb_present_flag[i][j]`が1に等しいことは、`slice_delta_poc_msb_cycle_lt[i][j]`

50

j] が存在することを指定する。 $slice_delta_poc_msb_present_flag[i][j]$ が 0 に等しいことは、 $slice_delta_poc_msb_cycle_lt[i][j]$ が存在しないことを指定する。

$prevTidOPic$ が、現在のピクチャと同じ nuh_layer_id を有し、 $TemporalId$ が 0 で、ランダム・アクセス・スキップされたリーディング (RASL) 又はランダム・アクセス復号可能なリーディング (RADL) ピクチャではない、復号順序での以前のピクチャとする。 $setOfPrevPocVals$ を以下からなるセットとする。

— $prevTidOPic$ の $PicOrderCntVal$

— $prevTidOPic$ の $RefPicList[0]$ 又は $RefPicList[1]$ におけるエントリによって参照され、現在のピクチャと同じ nuh_layer_id を有する各ピクチャの $PicOrderCntVal$

10

— 復号順序で $prevTidOPic$ に続き、現在のピクチャと同じ nuh_layer_id を有し、復号順序で現在のピクチャに先行する各ピクチャの $PicOrderCntVal$

$pic_rpl_present_flag$ が 0 に等しく、値モジュロ $MaxPicOrderCntLsb$ が $PocLsbLt[i][j]$ に等しい $setOfPrevPocVals$ において複数の値があるときに、 $slice_delta_poc_msb_present_flag[i][j]$ の値は、1 に等しいものとする。

$slice_delta_poc_msb_cycle_lt[i][j]$ は、以下のように変数 $FullPocLt[i][j]$ の値を指定する。

20

【数 3 1】

```
if( pic_rpl_present_flag )
```

```
    FullPocLt[ i ][ j ] = PicFullPocLt[ i ][ j ]
```

```
else {
```

```
    if( j == 0 )
```

30

```
        DeltaPocMsbCycleLt[ i ][ j ] = delta_poc_msb_cycle_lt[ i ][ j ]
```

```
    else
```

```
        DeltaPocMsbCycleLt[ i ][ j ] = delta_poc_msb_cycle_lt[ i ][ j ] +
```

```
        DeltaPocMsbCycleLt[ i ][ j - 1 ]
```

```
        FullPocLt[ i ][ j ] = PicOrderCntVal - DeltaPocMsbCycleLt[ i ][ j ] * MaxPicOrderCntLsb
```

```
        -( PicOrderCntVal & ( MaxPicOrderCntLsb - 1 ) ) + PocLsbLt[ i ][ j ]
```

40

```
    }
```

$slice_delta_poc_msb_cycle_lt[i][j]$ の値は、 $0 \sim 2^{(32 - \log_2 max_pic_order_cnt_lsb - 4)}$ の範囲（両端を含む）にあるものとする。

存在しないときに、 $slice_delta_poc_msb_cycle_lt[i][j]$ の値は、0 に等しいと推論される。

$num_ref_idx_active_override_flag$ が 1 に等しいこと

50

は、構文要素 `num_ref_idx_active_minus1[0]` が、Pスライス及びBスライスに対して存在し、`num_ref_idx_active_minus1[1]` が、Bスライスに対して存在することを指定する。`num_ref_idx_active_override_flag` が0に等しいことは、構文要素 `num_ref_idx_active_minus1[0]` 及び `num_ref_idx_active_minus1[1]` が存在しないことを指定する。

存在しないときに、`slice_address` の値は1に等しいと推論される。

`num_ref_idx_active_minus1[i]` は、式145によって指定される変数 `NumRefIdxActive[i]` の導出のために使用される。

`num_ref_idx_active_minus1[i]` の値は、0～14の範囲にある。

10

0又は1に等しい *i* に対して、現在のスライスがBスライスであり、`num_ref_idx_active_override_flag` が1に等しく、`num_ref_idx_active_minus1[i]` が存在しないときに、`num_ref_idx_active_minus1[i]` は、0に等しいと推論される。

現在のスライスがPスライスであり、`num_ref_idx_active_override_flag` が1に等しく、`num_ref_idx_active_minus1[0]` が存在しないときに、`num_ref_idx_active_minus1[0]` は、0に等しいと推論される。

変数 `NumRefIdxActive[i]` は、以下のように導出される。

20

【数32】

```
for( i = 0; i < 2; i++ ) {
    if( slice_type == B || ( slice_type == P && i == 0 ) ) {
        if( num_ref_idx_active_override_flag )
            NumRefIdxActive[ i ] = num_ref_idx_active_minus1[ i ] + 1
        else {
            if( num_ref_entries[ i ][ RplIdx[ i ] ] >= num_ref_idx_default_active_minus1[ i ] + 1 )
                NumRefIdxActive[ i ] = num_ref_idx_default_active_minus1[ i ] + 1
            else
                NumRefIdxActive[ i ] = num_ref_entries[ i ][ RplIdx[ i ] ]
        }
    } else /* slice_type == I || ( slice_type == P && i == 1 ) */
        NumRefIdxActive[ i ] = 0
}
```

30

`NumRefIdxActive[i] - 1` の値は、スライスを復号するために使用されてもよい参照ピクチャ・リスト *i* に対する最大参照インデックスを指定する。`NumRefIdxActive[i]` の値が0に等しいときに、参照ピクチャ・リスト *i* に対する参照インデックスは、スライスを復号するために使用されなくてもよい。

40

現在のスライスがPスライスであるときに、`NumRefIdxActive[0]` の値は、0より大きいものとする。現在のスライスがBスライスであるときに、`NumRefIdxActive[0]` 及び `NumRefIdxActive[1]` は両方とも、0より大きいものとする。

50

【表 10】

pred_weight_table() {	記述子
luma_log2_weight_denom	ue(v)
if(ChromaArrayType != 0)	
delta_chroma_log2_weight_denom	se(v)
for(i = 0; i < NumRefIdxActive[0]; i++)	
luma_weight_l0_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[0]; i++)	
chroma_weight_l0_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[0]; i++) {	
if(luma_weight_l0_flag[i]) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if(chroma_weight_l0_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
if(slice_type == B) {	
for(i = 0; i < NumRefIdxActive[1]; i++)	
luma_weight_l1_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[1]; i++)	
chroma_weight_l1_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[1]; i++) {	
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_weight_l1_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)

}	
}	
}	

重み付け予測パラメータ・セマンティクス

luma_log2_weight_denomは、すべての輝度重み付けファクタに対

10

20

30

40

50

する分母の基数 2 の対数である。

$\text{luma_log2_weight_denom}[i]$ の値は、0 ～ 7 の範囲（両端を含む）にあるものとする。

$\text{delta_chroma_log2_weight_denom}$ は、すべての色差重み付けファクタに対する分母の基数 2 の対数の差である。

$\text{delta_chroma_log2_weight_denom}$ が存在しないときに、それは 0 に等しいと推論される。

変数 $\text{ChromaLog2WeightDenom}$ は、 $\text{luma_log2_weight_denom} + \text{delta_chroma_log2_weight_denom}$ と等しくなるように導出され、値は、0 ～ 7 の範囲（両端を含む）にあるものとする。

$\text{luma_weight_l0_flag}[i]$ が 1 に等しいことは、 $\text{RefPicList}[0][i]$ を使用したリスト 0 予測の輝度コンポーネントに対する重み付けファクタが存在することを指定する。 $\text{luma_weight_l0_flag}[i]$ が 0 に等しいことは、これらの重み付けファクタが存在しないことを指定する。

$\text{chroma_weight_l0_flag}[i]$ が 1 に等しいことは、 $\text{RefPicList}[0][i]$ を使用したリスト 0 予測の色差コンポーネントに対する重み付けファクタが存在することを指定する。 $\text{chroma_weight_l0_flag}[i]$ が 0 に等しいことは、これらの重み付けファクタが存在しないことを指定する。

$\text{chroma_weight_l0_flag}[i]$ が存在しないときに、0 に等しいと推論される。

$\text{delta_luma_weight_l0}[i]$ は、 $\text{RefPicList}[0][i]$ を使用したリスト 0 予測のための輝度予測値に適用される重み付けファクタの差である。

変数 $\text{LumaWeightL0}[i]$ は、 $(1 \ll \text{luma_log2_weight_denom}) + \text{delta_luma_weight_l0}[i]$ に等しくなるように導出される。

$\text{luma_weight_l0_flag}[i]$ が 1 に等しいときに、 $\text{delta_luma_weight_l0}[i]$ は、 $-128 \sim 127$ の範囲（両端を含む）にあるものとする。

$\text{luma_weight_l0_flag}[i]$ が 0 に等しいときに、 $\text{LumaWeightL0}[i]$ は、 $2 \text{luma_log2_weight_denom}$ に等しいと推論される。

$\text{luma_offset_l0}[i]$ は、 $\text{RefPicList}[0][i]$ を使用したリスト 0 予測のための輝度予測値に適用される追加のオフセットである。

$\text{luma_offset_l0}[i]$ の値は、 $-128 \sim 127$ の範囲にある。

$\text{luma_weight_l0_flag}[i]$ が 0 に等しいときに、 $\text{luma_offset_l0}[i]$ は、0 に等しいと推論される。

$\text{delta_chroma_weight_l0}[i][j]$ は、Cb に対して 0 に等しい j 、Cr に対して 1 に等しい j とした $\text{RefPicList}[0][i]$ を使用したリスト 0 予測のための色差予測値に適用される重み付けファクタの差である。

変数 $\text{ChromaWeightL0}[i][j]$ は、 $(1 \ll \text{ChromaLog2WeightDenom}) + \text{delta_chroma_weight_l0}[i][j]$ に等しくなるように導出される。

$\text{chroma_weight_l0_flag}[i]$ が 1 に等しいときに、 $\text{delta_chroma_weight_l0}[i][j]$ は、 $-128 \sim 127$ の範囲（両端を含む）にあるものとする。

$\text{chroma_weight_l0_flag}[i]$ が 0 に等しいときに、 $\text{ChromaWeightL0}[i][j]$ は、 $2 \text{ChromaLog2WeightDenom}$ に等しいと推論される。

$\text{delta_chroma_offset_l0}[i][j]$ は、Cb に対して 0 に等し

10

20

30

40

50

い j 、 Cr に対して 1 に等しい j とした $RefPicList[0][i]$ を使用したリスト 0 予測のための色差予測値に適用される追加のオフセットの差である。

変数 $ChromaOffsetL0[i][j]$ は、以下のように導出される。

【数 33】

$$ChromaOffsetL0[i][j] = Clip3(-128, 127, \\ (128 + delta_chroma_offset_l0[i][j] - \\ ((128 * ChromaWeightL0[i][j]) >> ChromaLog2WeightDenom)))$$

10

$delta_chroma_offset_l0[i][j]$ の値は、 $-4 * 128 \sim 4 * 127$ の範囲にあるものとする。

$chroma_weight_l0_flag[i]$ が 0 に等しいときに、 $ChromaOffsetL0[i]$ は、0 に等しいと推論される。

$luma_weight_l1_flag[i]$ 、 $chroma_weight_l1_flag[i]$ 、 $delta_luma_weight_l1[i]$ 、 $luma_offset_l1[i]$ 、 $delta_chroma_weight_l1[i][j]$ 、及び $delta_chroma_offset_l1[i][j]$ は、 $luma_weight_l0_flag[i]$ 、 $chroma_weight_l0_flag[i]$ 、 $delta_luma_weight_l0[i]$ 、 $luma_offset_l0[i]$ 、 $delta_chroma_weight_l0[i][j]$ 、及び $delta_chroma_offset_l0[i][j]$ と同じであり、 $l0$ 、 $L0$ 、リスト 0、及び $List0$ は、それぞれ $l1$ 、 $L1$ 、リスト 1 及び $List1$ に置き換えられる。

20

変数 $sumWeightL0Flags$ は、 $i = 0 \dots NumRefIdxActive[0] - 1$ に対して、 $luma_weight_l0_flag[i] + 2 * chroma_weight_l0_flag[i]$ の合計に等しくなるように導出される。

$slice_type$ が B に等しいときに、変数 $sumWeightL1Flags$ は、 $i = 0 \dots NumRefIdxActive[1] - 1$ に対して、 $luma_weight_l1_flag[i] + 2 * chroma_weight_l1_flag[i]$ の合計に等しくなるように導出される。

30

$slice_type$ が P に等しいときに、 $sumWeightL0Flags$ は 24 以下であるものとし、 $slice_type$ が B に等しいときに、 $sumWeightL0Flags$ と $sumWeightL1Flags$ の合計は 24 以下であるものとする。ことがビットストリーム適合性の要件である。

参照ピクチャ・リスト構造セマンティクス

$ref_pic_list_struct(listIdx, rplsIdx)$ 構文構造は、SPS 又はスライス・ヘッダにおいて存在してもよい。

構文構造がスライス・ヘッダに含まれるか SPS に含まれるかに応じて、以下が適用される。

40

— スライス・ヘッダに存在する場合、 $ref_pic_list_struct(listIdx, rplsIdx)$ 構文構造は、現在のピクチャ（スライスを含むピクチャ）の参照ピクチャ・リスト $listIdx$ を指定する。

— それ以外の場合（SPS に存在する場合）、 $ref_pic_list_struct(listIdx, rplsIdx)$ 構文構造は参照画像リスト $listIdx$ に対する候補を指定し、この節の残りの部分で指定されるセマンティクスにおける用語「現在のピクチャ」は、1) SPS に含まれる $ref_pic_list_struct(listIdx, rplsIdx)$ 構文構造のリストのインデックスに等しい $ref_pic_list_idx[listIdx]$ を含む 1 つ以上のスライス、2) SPS を参照するコード化されたビデオ・シーケンス (CVS) にある各ピクチャを指す。

50

`num_ref_entries [listIdx] [rplsIdx]` は、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造におけるエントリの数を指定する。

`num_ref_entries [listIdx] [rplsIdx]` の値は、 $0 \sim \text{MaxDecPicBufMin} + 14$ の範囲（両端を含む）にあるとする。

`ltrp_in_slice_header_flag [listIdx] [rplsIdx]` が 0 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造における LTRP エントリの POC LSB が、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造において存在することを指定する。`ltrp_in_slice_header_flag [listIdx] [rplsIdx]` が 1 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造におけるロングターム参照ピクチャ (LTRP) エントリの POC LSB が、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造において存在しないことを指定する。

`inter_layer_ref_pic_flag [listIdx] [rplsIdx] [i]` が 1 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造における i 番目のエントリがレイヤ間参照ピクチャ (ILRP) エントリであることを指定する。`inter_layer_ref_pic_flag [listIdx] [rplsIdx] [i]` が 0 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造における i 番目のエントリが ILRP エントリではないことを指定する。

存在しないときに、`inter_layer_ref_pic_flag [listIdx] [rplsIdx] [i]` の値は、0 に等しいと推論される。

`st_ref_pic_flag [listIdx] [rplsIdx] [i]` が 1 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造における i 番目のエントリが STRP エントリであることを指定する。`st_ref_pic_flag [listIdx] [rplsIdx] [i]` が 0 に等しいことは、`ref_pic_list_struct (listIdx, rplsIdx)` 構文構造における i 番目のエントリが LTRP エントリであることを指定する。

`inter_layer_ref_pic_flag [listIdx] [rplsIdx] [i]` が 0 に等しく、`st_ref_pic_flag [listIdx] [rplsIdx] [i]` が存在しないときに、`st_ref_pic_flag [listIdx] [rplsIdx] [i]` の値が 1 に等しいと推論される。

変数 `NumLtrpEntries [listIdx] [rplsIdx]` は、以下のよう導出される。

【数 3 4】

```
for( i = 0, NumLtrpEntries[ listIdx ][ rplsIdx ] = 0; i < num_ref_entries[ listIdx ][ rplsIdx ]; i++ )
    if( !inter_layer_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] && !st_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] )
        NumLtrpEntries[ listIdx ][ rplsIdx ]++
```

`abs_delta_poc_st [listIdx] [rplsIdx] [i]` は、変数 `AbsDeltaPocSt [listIdx] [rplsIdx] [i]` の値を以下のように指定する。

【数 3 5】

10

20

30

40

50

if(sps_weighted_pred_flag || sps_weighted_bipred_flag)

AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i]

else

AbsDeltaPocSt[listIdx][rplsIdx][i] = abs_delta_poc_st[listIdx][rplsIdx][i] + 1

abs_delta_poc_st[listIdx][rplsIdx][i] の値は、0～ $2^{15}-1$ の範囲（両端を含む）にあるものとする。

10

strp_entry_sign_flag[listIdx][rplsIdx][i] が1に等しいことは、構文構造ref_pic_list_struct(listIdx, rplsIdx)におけるi番目のエントリが0以上の値を有することを指定する。strp_entry_sign_flag[listIdx][rplsIdx][i] が0に等しいことは、構文構造ref_pic_list_struct(listIdx, rplsIdx)におけるi番目のエントリが0より小さい値を有することを指定する。

存在しないときに、strp_entry_sign_flag[listIdx][rplsIdx][i] の値は、1に等しいと推論される。

20

リストDeltaPocValSt[listIdx][rplsIdx] は、以下のよう

【数36】

```
for( i = 0; i < num_ref_entries[ listIdx ][ rplsIdx ]; i++ )
    if( !inter_layer_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] && st_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] )
        DeltaPocValSt[ listIdx ][ rplsIdx ][ i ] = ( strp_entry_sign_flag[ listIdx ][ rplsIdx ][ i ] ) ?
        AbsDeltaPocSt[ listIdx ][ rplsIdx ][ i ] : 0 - AbsDeltaPocSt[ listIdx ][ rplsIdx ][ i ]
```

30

rpls_poc_lsb_lt[listIdx][rplsIdx][i] は、ref_pic_list_struct(listIdx, rplsIdx) 構文構造におけるi番目のエントリによって参照される、ピクチャのピクチャ・オーダ・カウンタ・モジュロMaxPicOrderCntLsbの値を指定する。

rpls_poc_lsb_lt[listIdx][rplsIdx][i] 構文要素の長さは、 $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ ビットである。

ilrp_idx[listIdx][rplsIdx][i] は、ref_pic_list_struct(listIdx, rplsIdx) 構文構造におけるi番目のエントリのILRPの、直接参照レイヤのリストへのインデックスを指定する。

40

ilrp_idx[listIdx][rplsIdx][i] の値は、0～NumDirectRefLayers[GeneralLayerIdx[nuh_layer_id]] - 1の範囲（両端を含む）にあるものとする。

このように、参照ブロックP0及びP1が取られる参照ピクチャにWPが適用されるかどうかを受けてGEO/TPMマージ・モードを制御することを可能にするために、異なるメカニズムを使用することができる。すなわち、

- 表14において列挙されたWPパラメータをSHからPHに移動し、
- GEO/TPMパラメータをPHからSHに戻し、
- すなわち、WPを伴う参照ピクチャが使用され得るときに（例えば、フラグのlum

50

aWeightedFlagの少なくとも1つが真である)、そのようなスライスに対して、0又は1に等しいMaxNumTriangleMergeCandをセットすることによって、MaxNumTriangleMergeCandのセマンティクスを変更する。

TPMマージ・モードの場合、例示的な参照ブロックP0及びP1は、それぞれ、図7において710及び720によって示される。

GEOマージ・モードの場合、例示的な参照ブロックP0及びP1は、それぞれ、図8において810及び820によって示される。

このように、参照ブロックP0及びP1が取られる参照ピクチャにWPが適用されるかどうかを受けてGEO/TPMマージ・モードを制御することを可能にするために、異なるメカニズムを使用することができる。すなわち、

- ー 表14において列挙されたWPパラメータをSHからPHに移動し、
- ー GEO/TPMパラメータをPHからSHに戻し、
- ー すなわち、WPを伴う参照ピクチャが使用され得るときに（例えば、フラグのlumaWeightedFlagの少なくとも1つが真である）、そのようなスライスに対して、0又は1に等しいMaxNumTriangleMergeCandをセットすることによって、MaxNumTriangleMergeCandのセマンティクスを変更する。

TPMマージ・モードの場合、例示的な参照ブロックP0及びP1は、それぞれ、図7において710及び720によって示される。

GEOマージ・モードの場合、例示的な参照ブロックP0及びP1は、それぞれ、図8において810及び820によって示される。

一実施形態では、WPパラメータ及び非矩形モード（例えば、GEO及びTPM）の有効化がピクチャ・ヘッダにおいてシグナリングされるときに、以下の構文が、下記の表に示されるように使用され得る。

10

20

30

40

50

【表 1 1】

表 ピクチャ・ヘッダ RBSP 構文

picture_header_rbsp() {	記述子
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
...	
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
pic_poc_lsb_lf[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lf[i][j]	ue(v)
}	
}	
}	

10

20

30

40

50

...	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
...	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 && WPDisabled)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rbsp_trailing_bits()	
}	

10

変数 `WPDisabled` は、`luma_weight_l0_flag[i]`、`chroma_weight_l0_flag[i]`、`luma_weight_l1_flag[j]`、及び `chroma_weight_l1_flag[j]` のすべての値がゼロ、`i = 0..NumRefIdxActive[0]` の値、及び `j = 0..NumRefIdxActive[1]` の値にセットされるときに、1 に等しくセットされ、

20

それ以外の場合、`WPDisabled` の値は、0 に等しくセットされる。

変数 `WPDisabled` が 0 に等しくセットされるときに、`pic_max_num_merge_cand_minus_max_num_triangle_cand` の値が `MaxNumMergeCand` に等しくセットされる。

一例では、WP パラメータ及び非矩形モード（例えば、GEO 及び TPM）の有効化のシグナリングは、スライス・ヘッダにおいて実行される。

例示的な構文が以下の表に与えられる。

30

40

50

【表 1 2】

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL nal_unit_type == IDR_N_LP nal_unit_type == CRA_NUT NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
slice_poc_lsb_lf[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	

10

20

30

40

50

delta_poc_msb_cycle_lf[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if(slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mv_d_l1_zero_idc)	

10

20

30

40

50

mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
slice_weighted_pred_flag	u(1)
if(slice_weighted_pred_flag)	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1 && WPDIsabled) {	
max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
}	
if(sps_ibc_enabled_flag)	
slice_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	

10

20

30

40

50

slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(!pps_dep_quant_enabled_flag)	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	

10

20

30

40

50

slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
slice_scaling_list_present_flag	u(1)
if(slice_scaling_list_present_flag)	
slice_scaling_list_aps_id	u(3)
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

20

変数 **WPD is a b l e d** は、**luma_weight_l0_flag[i]**、**chroma_weight_l0_flag[i]**、**luma_weight_l1_flag[j]**、及び **chroma_weight_l1_flag[j]** のすべての値がゼロ、**i = 0 . . NumRefIdxActive[0]** の値、及び **j = 0 . . NumRefIdxActive[1]** の値にセットされるときに、1に等しくセットされ、それ以外の場合、**WPD is a b l e d** の値は、0に等しくセットされる。

30

変数 **WPD is a b l e d** が0に等しくセットされるときに、**max_num_merge_cand_minus_max_num_triangle_cand** の値が **MaxNumMergeCand** に等しくセットされる。

上記に開示の実施形態では、重み付き予測パラメータは、ピクチャ・ヘッダ又はスライス・ヘッダのいずれかにおいてシグナリングされてもよい。

【0192】

40

一例では、**TPM** 又は **GEO** が有効であるかどうかの決定は、ブロックが非矩形重み付け予測に使用してもよい参照ピクチャ・リストを考慮して実行される。ブロックに対するマージ・リストが1つの参照ピクチャ・リスト **k** のみからの要素を含むときに、変数 **WPD is a b l e d[k]** の値は、このマージ・モードが有効かどうかを決定する。

【0193】

一例では、非矩形予測モードのためのマージ・リストは、重み付き予測が有効でない要素のみを含むように構成される。

【0194】

明細書以下の部分は、この例を例示する。

このプロセスへの入力は、

50

- ー 現在のピクチャの左上の輝度サンプルに対する現在の輝度コーディング・ブロックの左上のサンプルの輝度位置 (x_{Cb} , y_{Cb})、
- ー 輝度サンプルにおける現在のコーディング・ブロックの幅を指定する変数 $cbWidth$ 、
- ー 輝度サンプルにおける現在のコーディング・ブロックの高さを指定する変数 $cbHeight$ 。

このプロセスの出力は以下のものであり、 X は0又は1で、

- ー 隣接するコーディング・ユニットの可用性フラグ $availableFlagA_0$ 、 $availableFlagA_1$ 、 $availableFlagB_0$ 、 $availableFlagB_1$ 、及び $availableFlagB_2$
- ー 隣接するコーディング・ユニットの参照インデックス $refIdxLXA_0$ 、 $refIdxLXA_1$ 、 $refIdxLXB_0$ 、 $refIdxLXB_1$ 、及び $refIdxLXB_2$ 、
- ー 隣接するコーディング・ユニットの利用フラグ $predFlagLXA_0$ 、 $predFlagLXA_1$ 、 $predFlagLXB_0$ 、 $predFlagLXB_1$ 、及び $predFlagLXB_2$ 、
- ー 隣接するコーディング・ユニットの $1/16$ 分数サンプル精度 $mvLXA_0$ 、 $mvLXA_1$ 、 $mvLXB_0$ 、 $mvLXB_1$ 及び $mvLXB_2$ のモーション・ベクトル、
- ー ハーフ・サンプル補間フィルタ指数 $hpelIdxA_0$ 、 $hpelIdxA_1$ 、 $hpelIdxB_0$ 、 $hpelIdxB_1$ 、及び $hpelIdxB_2$ 、
- ー 双方向予測重み指数 $bcwIdxA_0$ 、 $bcwIdxA_1$ 、 $bcwIdxB_0$ 、 $bcwIdxB_1$ 、及び $bcwIdxB_2$ 。

$availableFlagB_1$ 、 $refIdxLXB_1$ 、 $predFlagLXB_1$ 、 $mvLXB_1$ 、 $hpelIdxB_1$ 、及び $bcwIdxB_1$ の導出については、以下が適用される。

- ー 隣接する輝度コーディング・ブロック内の輝度位置 (x_{NbB_1} , y_{NbB_1}) は、($x_{Cb} + cbWidth - 1$, $y_{Cb} - 1$) に等しくセットされる。
- ー 6.4.4 節において指定されたように隣接するブロック可用性のための導出プロセスは、(x_{Cb} , y_{Cb}) に等しくセットされた現在の輝度位置 (x_{Curr} , y_{Curr})、隣接する輝度位置 (x_{NbB_1} , y_{NbB_1})、真に等しくセットされた $CheckPredModeY$ 、及び 0 に等しくセットされた $cIdx$ を入力として呼び出され、出力は、ブロック可用性フラグ $availableB_1$ に割り当てられる。

- ー 変数 $availableFlagB_1$ 、 $refIdxLXB_1$ 、 $predFlagLXB_1$ 、 $mvLXB_1$ 、 $hpelIdxB_1$ 、及び $bcwIdxB_1$ は、以下のように導出される。

ー $availableB_1$ が偽に等しい場合、 $availableFlagB_1$ が 0 に等しくセットされ、 $mvLXB_1$ の両方のコンポーネントが 0 に等しくセットされ、 $refIdxLXB_1$ が -1 に等しくセットされ、 $predFlagLXB_1$ が 0 に等しくセットされ、 X が 0 又は 1 で、 $hpelIdxB_1$ が 0 にセットされ、 $bcwIdxB_1$ が 0 に等しくセットされる。

ー それ以外の場合、 $availableFlagB_1$ は 1 に等しくセットされ、以下の割り当てが行われる。

【数 37】

10

20

30

40

50

$$mvLXB_1 = MvLX[xNbB_1][yNbB_1] \quad (501)$$

$$refIdxLXB_1 = RefIdxLX[xNbB_1][yNbB_1] \quad (502)$$

$$predFlagLXB_1 = PredFlagLX[xNbB_1][yNbB_1] \quad (503)$$

$$hpelIfIdxB_1 = HpelIfIdx[xNbB_1][yNbB_1] \quad (504)$$

$$bcwIdxB_1 = BcwIdx[xNbB_1][yNbB_1] \quad (505)$$

10

$availableFlagA_1$ 、 $refIdxLXA_1$ 、 $predFlagLXA_1$ 、 $mvLXA_1$ 、 $hpelIfIdxA_1$ 、及び $bcwIdxA_1$ の導出については、以下が適用される。

－ 隣接する輝度コーディング・ブロック内の輝度位置($xNbA_1$, $yNbA_1$)は、($xCb-1$, $yCb+cbHeight-1$)に等しくセットされる。

－ 6.4.4節において指定されたように隣接するブロック可用性のための導出プロセスは、(xCb , yCb)に等しくセットされた現在の輝度位置($xCurr$, $yCurr$)、隣接する輝度位置($xNbA_1$, $yNbA_1$)、真に等しくセットされた $CheckPredModeY$ 、及び0に等しくセットされた $cIdx$ を入力として呼び出され、出力は、ブロック可用性フラグ $availableA_1$ に割り当てられる。

20

－ 変数 $availableFlagA_1$ 、 $refIdxLXA_1$ 、 $predFlagLXA_1$ 、 $mvLXA_1$ 、 $hpelIfIdxA_1$ 、及び $bcwIdxA_1$ は、以下のように導出される。

－ 以下の条件のうちの1つ以上が真である場合、 $availableFlagA_1$ が0に等しくセットされ、 $mvLXA_1$ の両方のコンポーネントが0に等しくセットされ、 $refIdxLXA_1$ が-1に等しくセットされ、 $predFlagLXA_1$ が0に等しくセット、 X が0又は1で、 $hpelIfIdxA_1$ が0にセットされ、 $bcwIdxA_1$ が0に等しくセットされる。

30

－ $availableA_1$ は、偽に等しい。

－ $availableB_1$ は、真に等しく、輝度位置($xNbA_1$, $yNbA_1$)及び($xNbB_1$, $yNbB_1$)は同じ運動ベクトルと同じ参照インデックスを有する。

－ $WPDisabldX[RefIdxLX[xNbA_1][yNbA_1]]$ が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置($xCurr$, $yCurr$)におけるブロックに対して三角形フラグが1に等しくセット）である。

－ $WPDisabldX[RefIdxLX[xNbB_1][yNbB_1]]$ が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置($xCurr$, $yCurr$)におけるブロックに対して三角形フラグが1に等しくセット）である。

－ それ以外の場合、 $availableFlagA_1$ は1に等しくセットされ、以下の割り当てが行われる。

40

【数38】

$$mvLXA_1 = MvLX[xNbA_1][yNbA_1] \quad (506)$$

$$refIdxLXA_1 = RefIdxLX[xNbA_1][yNbA_1] \quad (507)$$

$$predFlagLXA_1 = PredFlagLX[xNbA_1][yNbA_1] \quad (508)$$

$$hpelIfIdxA_1 = HpelIfIdx[xNbA_1][yNbA_1] \quad (509)$$

$$bcwIdxA_1 = BcwIdx[xNbA_1][yNbA_1] \quad (510)$$

10

`availableFlagB0`、`refIdxLXB0`、`predFlagLXB0`、`mvLXB0`、`hpelIfIdxB0`、及び`bcwIdxB0`の導出については、以下が適用される。

－ 隣接する輝度コーディング・ブロック内の輝度位置(`xNbB0`、`yNbB0`)は、(`xCb+cbWidth`、`yCb-1`)に等しくセットされる。

－ 6.4.4節において指定されたように隣接するブロック可用性のための導出プロセスは、(`xCb`、`yCb`)に等しくセットされた現在の輝度位置(`xCurr`、`yCurr`)、隣接する輝度位置(`xNbB0`、`yNbB0`)、真に等しくセットされた`CheckPredModeY`、及び0に等しくセットされた`cIdx`を入力として呼び出され、出力は、ブロック可用性フラグ`availableB0`に割り当てられる。

20

－ 変数`availableFlagB0`、`refIdxLXB0`、`predFlagLXB0`、`mvLXB0`、`hpelIfIdxB0`、及び`bcwIdxB0`は、以下のように導出される。

－ 以下の条件のうちの1つ以上が真である場合、`availableFlagB0`が0に等しくセットされ、`mvLXB0`の両方のコンポーネントが0に等しくセットされ、`refIdxLXB0`が-1に等しくセットされ、`predFlagLXB0`が0に等しくセット、`X`が0又は1で、`hpelIfIdxB0`が0にセットされ、`bcwIdxB0`が0に等しくセットされる。

30

－ `availableB0`は、偽に等しい。

－ `availableB1`は、真に等しく、輝度位置(`xNbB1`、`yNbB1`)及び(`xNbB0`、`yNbB0`)は同じ運動ベクトルと同じ参照インデックスを有する。

－ `WPDIsabledX[RefIdxLX[xNbB0][yNbB0]]`が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置(`xCurr`、`yCurr`)におけるブルックに対して三角形フラグが1に等しくセット）である。

－ `WPDIsabledX[RefIdxLX[xNbB1][yNbB1]]`が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置(`xCurr`、`yCurr`)におけるブルックに対して三角形フラグが1に等しくセット）である。

－ それ以外の場合、`availableFlagB0`は1に等しくセットされ、以下の割り当てが行われる。

40

【数39】

50

$$mvLXB_0 = MvLX[xNbB_0][yNbB_0] \quad (511)$$

$$refIdxLXB_0 = RefIdxLX[xNbB_0][yNbB_0] \quad (512)$$

$$predFlagLXB_0 = PredFlagLX[xNbB_0][yNbB_0] \quad (513)$$

$$hpelIdxB_0 = HpelIdx[xNbB_0][yNbB_0] \quad (514)$$

$$bcwIdxB_0 = BcwIdx[xNbB_0][yNbB_0] \quad (515)$$

10

`availableFlagA0`、`refIdxLXA0`、`predFlagLXA0`、`mvLXA0`、`hpelIdxA0`、及び`bcwIdxA0`の導出については、以下が適用される。

－ 隣接する輝度コーディング・ブロック内の輝度位置(`xNbA0`、`yNbA0`)は、(`xCb-1`、`yCb+cbWidth`)に等しくセットされる。

－ 6.4.4節において指定されたように隣接するブロック可用性のための導出プロセスは、(`xCb`、`yCb`)に等しくセットされた現在の輝度位置(`xCurr`、`yCurr`)、隣接する輝度位置(`xNbA0`、`yNbA0`)、真に等しくセットされた`CheckPredModeY`、及び0に等しくセットされた`cIdx`を入力として呼び出され、出力は、ブロック可用性フラグ`availableA0`に割り当てられる。

20

－ 変数`availableFlagA0`、`refIdxLXA0`、`predFlagLXA0`、`mvLXA0`、`hpelIdxA0`、及び`bcwIdxA0`は、以下のように導出される。

－ 以下の条件のうちの1つ以上が真である場合、`availableFlagA0`が0に等しくセットされ、`mvLXA0`の両方のコンポーネントが0に等しくセットされ、`refIdxLXA0`が-1に等しくセットされ、`predFlagLXA0`が0に等しくセット、`X`が0又は1で、`hpelIdxA0`が0にセットされ、`bcwIdxA0`が0に等しくセットされる。

30

－ `availableA0`は、偽に等しい。

－ `availableA1`は、真に等しく、輝度位置(`xNbA1`、`yNbA1`)及び(`xNbA0`、`yNbA0`)は同じ運動ベクトルと同じ参照インデックスを有する。

－ `WPDisabledX[RefIdxLX[xNbA0][yNbA0]]`が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置(`xCurr`、`yCurr`)におけるブルックに対して三角形フラグが1に等しくセット）である。

－ `WPDisabledX[RefIdxLX[xNbA1][yNbA1]]`が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置(`xCurr`、`yCurr`)におけるブルックに対して三角形フラグが1に等しくセット）である。

－ それ以外の場合、`availableFlagA0`は1に等しくセットされ、以下の割り当てが行われる。

40

【数40】

50

$$mvLXA_0 = MvLX[xNbA_0][yNbA_0] \quad (516)$$

$$refIdxLXA_0 = RefIdxLX[xNbA_0][yNbA_0] \quad (517)$$

$$predFlagLXA_0 = PredFlagLX[xNbA_0][yNbA_0] \quad (518)$$

$$hpelIfIdxA_0 = HpelIfIdx[xNbA_0][yNbA_0] \quad (519)$$

$$bcwIdxA_0 = BcwIdx[xNbA_0][yNbA_0] \quad (520)$$

10

$availableFlagB_2$ 、 $refIdxLXB_2$ 、 $predFlagLXB_2$ 、 $mvLXB_2$ 、 $hpelIfIdxB_2$ 、及び $bcwIdxB_2$ の導出については、以下が適用される。

ー 隣接する輝度コーディング・ブロック内の輝度位置($xNbB_2$, $yNbB_2$)は、($xCb-1$, $yCb-1$)に等しくセットされる。

ー 6.4.4節において指定されたように隣接するブロック可用性のための導出プロセスは、(xCb , yCb)に等しくセットされた現在の輝度位置($xCurr$, $yCurr$)、隣接する輝度位置($xNbB_2$, $yNbB_2$)、真に等しくセットされた $CheckPredModeY$ 、及び0に等しくセットされた $cIdx$ を入力として呼び出され、出力は、ブロック可用性フラグ $availableB_2$ に割り当てられる。

20

ー 変数 $availableFlagB_2$ 、 $refIdxLXB_2$ 、 $predFlagLXB_2$ 、 $mvLXB_2$ 、 $hpelIfIdxB_2$ 、及び $bcwIdxB_2$ は、以下のように導出される。

ー 以下の条件のうちの1つ以上が真である場合、 $availableFlagB_2$ が0に等しくセットされ、 $mvLXB_2$ の両方のコンポーネントが0に等しくセットされ、 $refIdxLXB_2$ が-1に等しくセットされ、 $predFlagLXB_2$ が0に等しくセット、 X が0又は1で、 $hpelIfIdxB_2$ が0にセットされ、 $bcwIdxB_2$ が0に等しくセットされる。

30

ー $availableB_2$ は、偽に等しい。

ー $availableA_1$ は、真に等しく、輝度位置($xNbA_1$, $yNbA_1$)及び($xNbB_2$, $yNbB_2$)は同じ運動ベクトルと同じ参照インデックスを有する。

ー $availableB_1$ は、真に等しく、輝度位置($xNbB_1$, $yNbB_1$)及び($xNbB_2$, $yNbB_2$)は同じ運動ベクトルと同じ参照インデックスを有する。

ー $availableFlagA_0 + availableFlagA_1 + availableFlagB_0 + availableFlagB_1$ が4に等しい。

ー $WPDisabledX[RefIdxLX[xNbB_1][yNbB_1]]$ が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置($xCurr$, $yCurr$)におけるブルックに対して三角形フラグが1に等しくセット）である。

40

ー $WPDisabledX[RefIdxLX[xNbB_2][yNbB_2]]$ が0にセットされ、マージ・モードが非矩形（例えば、現在の輝度位置($xCurr$, $yCurr$)におけるブルックに対して三角形フラグが1に等しくセット）である。

ー それ以外の場合、 $availableFlagB_2$ は1に等しくセットされ、以下の割り当てが行われる。

【数41】

50

$$mvLXB_2 = MvLX[xNbB_2][yNbB_2] \quad (521)$$

$$refIdxLXB_2 = RefIdxLX[xNbB_2][yNbB_2] \quad (522)$$

$$predFlagLXB_2 = PredFlagLX[xNbB_2][yNbB_2] \quad (523)$$

$$hpelIdxB_2 = HpelIdx[xNbB_2][yNbB_2] \quad (524)$$

$$bcwIdxB_2 = BcwIdx[xNbB_2][yNbB_2] \quad (525)$$

10

上記に開示の例では、以下の変数定義が使用される。

変数 `WPDisabled0[i]` は、`luma_weight_l0_flag[i]` 及び `chroma_weight_l0_flag[i]` のすべての値がゼロ、
`i = 0..NumRefIdxActive[0]` の値にセットされるときに、1 に等しく
 セットされる。

それ以外の場合、`WPDisabled0[i]` の値は、0 に等しくセットされる。

変数 `WPDisabled1[i]` は、`luma_weight_l1_flag[i]` 及び `chroma_weight_l1_flag[i]` のすべての値がゼロ、
`i = 0..NumRefIdxActive[1]` の値にセットされるときに、1 に等しく
 セットされる。

20

それ以外の場合、`WPDisabled1[1]` の値は、0 に等しくセットされる。

別の例では、変数 `SliceMaxNumTriangleMergeCand` は、以下の
 うちの1つに従ってスライス・ヘッダで定義される。

【数42】

- `SliceMaxNumTriangleMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 0 :
MaxNumTriangleMergeCand ;`
- `SliceMaxNumTriangleMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 1 :
MaxNumTriangleMergeCand ;`
- `SliceMaxNumTriangleMergeCand = slice_weighted_pred_flag ? 0 : MaxNumTriangleMergeCand ;`

30

あるいは、

【数43】

- `SliceMaxNumTriangleMergeCand = slice_weighted_pred_flag ? 1 : MaxNumTriangleMergeCand`

`SliceMaxNumTriangleMergeCand` の値は、ブロック・レベル
 でのマージ情報の解析においてさらに使用される。

例示的な構文が以下の表に与えられる。

40

50

【表 1 3】

merge_data(x0, y0, cbWidth, cbHeight, chType) {	記述子
if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) (sps_triangle_enabled_flag && SliceMaxNumTriangleMergeCand > 1 && slice_type == B)))	
regular_merge_flag [x0][y0]	ae(v)
if (regular_merge_flag[x0][y0] == 1){	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag [x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else {	
if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
}	
} else {	
if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && SliceMaxNumTriangleMergeCand > 1 && weightedPredFlag == 0 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag [x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	

10

20

30

40

50

merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && SliceMaxNumTriangleMergeCand > 1) {	
merge_triangle_split_dir[x0][y0]	ae(v)
merge_triangle_idx0[x0][y0]	ae(v)
merge_triangle_idx1[x0][y0]	ae(v)
}	
}	
}	
}	
}	

10

非矩形インター予測モードがGEOモードである場合に、以下の例がさらに記載される。参照ブロックP0及びP1が取られる参照ピクチャにWPが適用されるかどうかを受けてGEO／TPMマージ・モードを制御することを可能にするために、異なるメカニズムを使用することができる。すなわち、

- － 表14において列挙されたWPパラメータをSHからPHに移動し、
 - － GEOパラメータをPHからSHに戻し、
 - － すなわち、WPを伴う参照ピクチャが使用され得るときに（例えば、フラグのlumaWeightedFlagの少なくとも1つが真である）、そのようなスライスに対して、0又は1に等しいMaxNumGeoMergeCandをセットすることによって、MaxNumGeoMergeCandのセマンティクスを変更する。
- GEOマージ・モードの場合、例示的な参照ブロックP0及びP1は、それぞれ、図8において810及び820によって示される。

20

一例では、WPパラメータ及び非矩形モード（例えば、GEO及びTPM）の有効化がピクチャ・ヘッダにおいてシグナリングされるときに、以下の構文が、下記の表に示されるように使用され得る。

30

40

50

【表 1 4】

表ーピクチャ・ヘッダ RBSP 構文

picture_header_rbsp() {	記述子
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
...	
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idx[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	

10

20

30

40

50

}	
...	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
...	
...	
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 && WPDIsabled)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
rsbsp_trailing_bits()	
}	

10

変数 `WPDIsabled` は、`luma_weight_l0_flag[i]`、`chroma_weight_l0_flag[i]`、`luma_weight_l1_flag[j]`、及び `chroma_weight_l1_flag[j]` のすべての値がゼロ、`i = 0..NumRefIdxActive[0]` の値、及び

20

`j = 0..NumRefIdxActive[1]` の値にセットされるときに、1 に等しくセットされ、

それ以外の場合、`WPDIsabled` の値は、0 に等しくセットされる。

変数 `WPDIsabled` が 0 に等しくセットされるときに、`pic_max_num_merge_cand_minus_max_num_geo_cand` の値が `MaxNumMergeCand` に等しくセットされる。

別の例では、`pic_max_num_merge_cand_minus_max_num_geo_cand` が、`MaxNumMergeCand - 1` に等しくセットされる。一例では、WP パラメータ及び非矩形モード（例えば、GEO 及び TPM）の有効化のシグナリングは、スライス・ヘッダにおいて実行される。

30

例示的な構文が以下の表に与えられる。

40

50

【表 1 5】

slice_header() {	記述子
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag && (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
}	
if(pic_rpl_present_flag (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	

10

20

30

40

50

if((slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(pic_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if(slice_weighted_pred_flag)	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1 && WPDisabled) {	
max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

10

20

30

40

50

}	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !pic_sao_enabled_present_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !pic_deblocking_filter_override_present_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)

10

20

30

40

50

}	
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

変数 `WPDisabled` は、`luma_weight_l0_flag[ i ]`、`chroma_weight_l0_flag[ i ]`、`luma_weight_l1_flag[ j ]`、及び `chroma_weight_l1_flag[ j ]` のすべての値がゼロ、`i = 0 . . NumRefIdxActive[ 0 ]` の値、及び `j = 0 . . NumRefIdxActive[ 1 ]` の値にセットされるときに、1 に等しくセットされ、

20

それ以外の場合、`WPDisabled` の値は、0 に等しくセットされる。

変数 `WPDisabled` が 0 に等しくセットされるときに、`max_num_merge_cand_minus_max_num_geo_cand` の値が `MaxNumMergeCand` に等しくセットされる。

別の実施形態では、変数 `WPDisabled` が 0 に等しくセットされるときに、`max_num_merge_cand_minus_max_num_geo_cand` の値が `MaxNumMergeCand - 1` に等しくセットされる。

30

上記の例では、重み付き予測パラメータは、ピクチャ・ヘッダ又はスライス・ヘッダのいずれかにおいてシグナリングされてもよい。

別の実施形態では、変数 `SliceMaxNumGeoMergeCand` は、以下のうちのいずれかに従ってスライス・ヘッダで定義される。

【数 4 4】

- `SliceMaxNumGeoMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 0 : MaxNumGeoMergeCand ;`
- `SliceMaxNumGeoMergeCand = (lumaWeightedFlag || chromaWeightedFlag) ? 1 : MaxNumGeoMergeCand ;`
- `SliceMaxNumGeoMergeCand = slice_weighted_pred_flag ? 0 : MaxNumGeoMergeCand ;`

40

あるいは、

【数 4 5】

`SliceMaxNumGeoMergeCand = slice_weighted_pred_flag ? 1 : MaxNumGeoMergeCand`

50

異なる実施形態は、上記に列挙された異なる場合を使用する。

変数 `SliceMaxNumGeoMergeCand` の値は、ブロック・レベルでのマージ情報の解析においてさらに使用される。

例示的な構文が以下の表に与えられる。

【表 16】

7.3.9.7 マージ・データ構文

merge_data(x0, y0, cbWidth, cbHeight, chType) {	記述子
if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) (sps_geo_enabled_flag && SliceMaxNumGeoMergeCand > 1 && cbWidth >= 8 && cbHeight >= 8 && slice_type == B))	
regular_merge_flag [x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag [x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && sps_geo_enabled_flag && SliceMaxNumGeoMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < 128 && cbHeight < 128)	
ciip_flag [x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	

10

20

30

40

50

<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>if(!ciip_flag[x0][y0] && SliceMaxNumGeoMergeCand > 1) {</code>	
<code>merge_geo_partition_idx[x0][y0]</code>	<code>ae(v)</code>
<code>merge_geo_idx0[x0][y0]</code>	<code>ae(v)</code>
<code>if(SliceMaxNumGeoMergeCand > 2)</code>	
<code>merge_geo_idx1[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

10

関係するピクチャ・ヘッダ・セマンティクスは、以下のようである。

`pic_max_num_merge_cand_minus_max_num_geo_cand`は、`MaxNumMergeCand`から差し引かれたピクチャのヘッダに関連するスライスでサポートされる`geoマージ・モード`候補の最大数を指定する。

`pic_max_num_merge_cand_minus_max_num_geo_cand`が存在せず、`sps_geo_enabled_flag`が1に等しく、かつ`MaxNumMergeCand`が2以上であるときに、`max_num_merge_cand_minus_max_num_geo_cand`は、`pps_max_num_merge_cand_minus_max_num_geo_cand_plus1-1`に等しいと推論される。

20

`geoマージ・モード`候補の最大数`MaxNumGeoMergeCand`は、以下のよう
に導出される。

【数46】

$$\text{MaxNumGeoMergeCand} = \text{MaxNumMergeCand} - \text{pic_max_num_merge_cand_minus_max_num_geo_cand}$$

30

`pic_max_num_merge_cand_minus_max_num_geo_cand`が存在するときに、`MaxNumGeoMergeCand`の値は、2～`MaxNumMergeCand`の範囲（両端を含む）にあるものとする。

`pic_max_num_merge_cand_minus_max_num_geo_cand`が存在しない（、かつ`sps_geo_enabled_flag`が0に等しいか、又は`MaxNumMergeCand`が2より小さい）ときに、`MaxNumGeoMergeCand`は0に等しくセットされる。

`MaxNumGeoMergeCand`が0に等しいときに、`PH`に関連付けられたスライスに対して`geoマージ・モード`は許可されない。

以下の例では、いくつかのシグナリング関係の態様が考慮される。

40

すなわち、これらの態様は以下のようである。

ー マージ・モード（）のための候補の数に
関係する構文要素は、シーケンス・パラメータ・セット（`SPS`）においてシグナリングされ、これは、特定の
実装が`SPS`レベルで非矩形モード・マージ候補の数（`MaxNumGeoMergeCand`）を導出することを可能にする。

ー ピクチャがただ1つのスライスを含むときに、`PH`は、`SH`においてシグナリングされ得る。

ー 以下のように、以下を伴って`PH`／`SH`パラメータのオーバーライド・メカニズムを定義する。

関係するコーディング・ツールの構文要素が`PH`において存在するか、`SH`において

50

存在するか（両方ではない）を指定する P P S フラグ。

特に、参照ピクチャ・リストと重み付き予測表はこのメカニズムを使用することができる。

- － 予測重み表は、P H 又は S H (A L F、デブロッキング、R P L、及び S A O など) のいずれかにおいてシグナリングすることができる第 5 のタイプのデータ。
- － ピクチャに対して重み付き予測が有効であるときに、ピクチャのすべてのスライスが同じ参照ピクチャ・リストを有することが必要である。
- － P H に関連するピクチャにおいて特定のスライス・タイプのみが使用される場合、インター及びイントラ関係構文要素が条件付きでシグナリングされる。

特に、`pic__inter__slice__present__flag` 及び `pic__intra__slice__present__flag` の 2 つのフラグが導入される。

一例では、マージ・モード () のための候補の数に関する構文要素は、シーケンス・パラメータ・セット (S P S) においてシグナリングされ、これは、特定の実装が S P S レベルで非矩形モード・マージ候補の数 (`MaxNumGeoMergeCand`) を導出することを可能にする。

この態様は、以下の構文に基づいた符号化又は復号プロセスによって実装され得る。

10

20

30

40

50

【表 1 7】

7.3.2.3 シーケンス・パラメータ・セット RBSP 構文

seq_parameter_set_rbsp() {	記述子
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
sps_seq_parameter_set_id	u(4)
...	
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
sps_affine_type_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
if(chroma_format_idc == 3) {	
sps_palette_enabled_flag	u(1)
sps_act_enabled_flag	u(1)
}	
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_geo_enabled_flag	u(1)
sps_six_minus_max_num_merge_cand_plus1	ue(v)
if(sps_geo_enabled_flag)	
sps_max_num_merge_cand_minus_max_num_geo_cand_plus1	ue(v)
sps_lmcs_enabled_flag	u(1)
...	
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
...	

上記に記載の構文は以下のセマンティクスを有する。

`sps_six_minus_max_num_merge_cand_plus1`が0に等しいことは、`pic_six_minus_max_num_merge_cand`が、PPSを参照するスライスのPHにおいて存在することを指定する。`sps_six_minus_max_num_merge_cand_plus1`が0より大きいこと

は、`pic_six_minus_max_num_merge_cand`が、PPSを参照するPHにおいて存在しないことを指定する。

`sps_six_minus_max_num_merge_cand_plus1[i]`の値は、0～6の範囲（両端を含む）にあるものとする。

`sps_max_num_merge_cand_minus_max_num_geo_cand_plus1`が0に等しいことは、`pic_max_num_merge_cand_minus_max_num_geo_cand`が、PPSを参照するスライスのPHにおいて存在することを指定する。`sps_max_num_merge_cand_minus_max_num_geo_cand_plus1`が0より大きいことは、`pic_max_num_merge_cand_minus_max_num_geo_cand`が、PPSを参照するPHにおいて存在しないことを指定する。

10

`sps_max_num_merge_cand_minus_max_num_geo_cand_plus1`の値は、0～`MaxNumMergeCand-1`の範囲（両端を含む）にあるものとする。

PHの対応する要素のセマンティクスは、以下のものである。

`pic_six_minus_max_num_merge_cand`は、6から差し引かれた、PHに関連するスライスでサポートされるモーション・ベクトル予測（MVP）候補をマージする最大数を指定する。

MVP候補をマージする最大数`MaxNumMergeCand`は、以下のように導出される。

20

【数47】

$$\text{MaxNumMergeCand} = 6 - \text{pic_six_minus_max_num_merge_cand}$$

`MaxNumMergeCand`の値は、1～6の範囲にあるものとする。

存在しないときに、`pic_six_minus_max_num_merge_cand`の値は、`sps_six_minus_max_num_merge_cand_plus1-1`に等しいと推論される。

`pic_max_num_merge_cand_minus_max_num_geo_cand`は、`MaxNumMergeCand`から差し引かれたピクチャのヘッダに関連するスライスでサポートされるgeoマージ・モード候補の最大数を指定する。

30

`sps_max_num_merge_cand_minus_max_num_geo_cand`が存在せず、`sps_geo_enabled_flag`が1に等しく、かつ`MaxNumMergeCand`が2以上であるときに、`pic_max_num_merge_cand_minus_max_num_geo_cand`は、`sps_max_num_merge_cand_minus_max_num_geo_cand_plus1-1`に等しいと推論される。

geoマージ・モード候補の最大数`MaxNumGeoMergeCand`は、以下のように導出される。

【数48】

40

$$\text{MaxNumGeoMergeCand} = \text{MaxNumMergeCand} - \text{pic_max_num_merge_cand_minus_max_num_geo_cand}$$

`pic_max_num_merge_cand_minus_max_num_geo_cand`が存在するときに、`MaxNumGeoMergeCand`の値は、2～`MaxNumMergeCand`の範囲（両端を含む）にあるものとする。

`pic_max_num_merge_cand_minus_max_num_geo_cand`が存在しない（、かつ`spg_geo_enabled_flag`が0に等しいか、又は`MaxNumMergeCand`が2より小さい）ときに、`MaxNumGeoMergeCand`は0に等しくセットされる。

`MaxNumGeoMergeCand`が0に等しいときに、PHに関連付けられたスラ

50

イスに対して `geo` マージ・モードは許可されない。

あるいは、`max_num_merge_cand_minus_max_num_geo_cand` は、`MaxNumMergeCand` から差し引かれた `SPS` でサポートされる `GEO` マージ・モード候補の最大数を指定する。

に `sps_geo_enabled_flag` が 1 に等しく、`MaxNumMergeCand` が 3 以上であるときに、`GEO` マージ・モード候補の最大数 `MaxNumGeoMergeCand` は、以下のように導出される。

【数 49】

$$\text{MaxNumGeoMergeCand} = \text{MaxNumMergeCand} - \text{max_num_merge_cand_minus_max_num_geo_cand}$$

10

`sps_geo_enabled_flag` の値が 1 に等しい場合、`MaxNumGeoMergeCand` の値は、2 ～ `MaxNumMergeCand` までの範囲（両端を含む）にあるものとする。

それ以外の場合、`sps_geo_enabled_flag` が 1 に等しく、`MaxNumMergeCand` が 2 に等しいときに、`MaxNumGeoMergeCand` は 2 にセットされる。

それ以外の場合、`MaxNumGeoMergeCand` は 0 に等しくセットされる。

この例に対する代替的な構文及びセマンティクスは、以下のである。

【表 18】

...	
<code>sps_geo_enabled_flag</code>	<code>u(1)</code>
<code>sps_six_minus_max_num_merge_cand</code>	<code>ue(v)</code>
<code>if(sps_geo_enabled_flag)</code>	
<code>sps_max_num_merge_cand_minus_max_num_geo_cand</code>	<code>ue(v)</code>
...	

20

`sps_six_minus_max_num_merge_cand` は、6 から差し引かれた、`PH` に関連するスライスでサポートされるモーション・ベクトル予測（`MVP`）候補をマージする最大数を指定する。

30

`MVP` 候補をマージする最大数 `MaxNumMergeCand` は、以下のように導出される。

【数 50】

$$\text{MaxNumMergeCand} = 6 - \text{sps_six_minus_max_num_merge_cand}$$

`MaxNumMergeCand` の値は、1 ～ 6 の範囲にあるものとする。

`sps_max_num_merge_cand_minus_max_num_geo_cand` は、`MaxNumMergeCand` から差し引かれた、ピクチャのヘッダに関連するスライスでサポートされる `geo` マージ・モード候補の最大数を指定する。

40

`geo` マージ・モード候補の最大数 `MaxNumGeoMergeCand` は、以下のように導出される。

【数 51】

$$\text{MaxNumGeoMergeCand} = \text{MaxNumMergeCand} - \text{sps_max_num_merge_cand_minus_max_num_geo_cand}$$

`sps_max_num_merge_cand_minus_max_num_geo_cand` が存在するときに、`MaxNumGeoMergeCand` の値は、2 ～ `MaxNumMergeCand` の範囲（両端を含む）にあるものとする。

`sps_max_num_merge_cand_minus_max_num_geo`

50

— `cand`が存在しない（、かつ`sps_geo_enabled_flag`が0に等しいか、又は`MaxNumMergeCand`が2より小さい）ときに、`MaxNumGeoMergeCand`は0に等しくセットされる。

`MaxNumGeoMergeCand`が0に等しいときに、`geoマージ・モード`は許可されない。上記に記載の例及び両方の代替的な構文定義について、重み付き予測が有効かどうかについてチェックが実行される。このチェックは`MaxNumGeoMergeCand`変数の導出に影響を与え、`MaxNumGeoMergeCand`の値は以下の場合の1つにおいてゼロにセットされる。

— $i = 0 \dots \text{NumRefIdxActive}[0]$ の値及び $j = 0 \dots \text{NumRefIdxActive}[1]$ の値に対して、`luma_weight_l0_flag[i]`、`chroma_weight_l0_flag[i]`、`luma_weight_l1_flag[j]` 及び `chroma_weight_l1_flag[j]` のすべての値がゼロにセットされるか、又は存在しないとき、

— `SPS`又は`PPS`におけるフラグ（`pps_weighted_bipred_flag`）が双方向重み付き予測の存在を示すとき、

— 画像ヘッダ（`PH`）又はスライス・ヘッダ（`SH`）のいずれかにおいて、双方向重み付け予測の存在が示されるとき、

重み付き予測パラメータの存在を示す`SPS`レベルのフラグは、以下のようにシグナリングされ得る。

【表 19】

<code>sps_bcw_enabled_flag</code>	<code>u(1)</code>
<code>sps_ibc_enabled_flag</code>	<code>u(1)</code>
<code>sps_ciip_enabled_flag</code>	<code>u(1)</code>
<code>if(sps_mmvd_enabled_flag)</code>	
<code>sps_fpel_mmvd_enabled_flag</code>	<code>u(1)</code>
<code>sps_wp_enabled_flag</code>	
<code>if(!sps_wp_enabled_flag)</code>	
<code>sps_geo_enabled_flag</code>	<code>u(1)</code>
<code>sps_six_minus_max_num_merge_cand_plus1</code>	<code>ue(v)</code>
<code>if(sps_geo_enabled_flag)</code>	
<code>sps_max_num_merge_cand_minus_max_num_geo_cand_plus1</code>	<code>ue(v)</code>
<code>sps_lmcs_enabled_flag</code>	<code>u(1)</code>
<code>sps_lfst_enabled_flag</code>	<code>u(1)</code>
<code>sps_ladf_enabled_flag</code>	<code>u(1)</code>
...	

構文要素「`sps_wp_enabled_flag`」は、重み付き予測がより低いレベル（`PPS`、`PH`、又は`SH`）で有効にできるかどうかを決定する。例示的な実装が、以下に与えられる。

【表 2 0】

if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbr_qp_offset_present_flag)	
joint_cbr_qp_offset_list[i]	se(v)
}	
}	
if(sps_wp_enabled_flag) {	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
}	

10

上記の表では、pps_weighted_pred_flagとpps_weighted_pred_flagはmビットストリームにおけるフラグであり、単方向予測ブロック及び双方向予測ブロックに対して重み付け予測が有効かどうかを示す。

20

一例では、pic_weighted_pred_flag及びpic_weighted_bipred_flagなどのように、ピクチャ・ヘッダにおいて重み付け予測フラグが指定されている場合、sps_wp_enabled_flagへの以下の依存関係がビットストリーム構文において指定されてもよい。

【表 2 1】

...
if(sps_wp_enabled_flag) {
pic_weighted_pred_flag
pic_weighted_bipred_flag
}
...

30

一例では、weighted_pred_flag及びweighted_bipred_flagなどのように、スライス・ヘッダにおいて重み付け予測フラグが指定されている場合、sps_wp_enabled_flagへの以下の依存関係がビットストリーム構文において指定されてもよい。

【表 2 2】

...
if(sps_wp_enabled_flag) {
weighted_pred_flag
weighted_bipred_flag
}
...

40

50

一例では、参照ピクチャ・リストは、PPS又はPH若しくはSH（両方ではない）のいずれかにおいて示されてもよい。いくつかの例では、参照ピクチャ・リストのシグナリングは、重み付き予測の存在を示す構文要素（例えば、pps_weighted_pred_flag及びpps_weighted_bipred_flag）に依存する。したがって、参照ピクチャ・リストがPPS、PH又はSHにおいて示されるかどうかに応じて、重み付き予測パラメータが、参照画像リストの前にPPS、PH又はSHにおいて対応してシグナリングされる。

本実施形態に対して、以下の構文が指定され得る。

【表 2 3】

ピクチャ・パラメータ・セット構文

pic_parameter_set_rbsp() {	記述子
...	
rpl_present_in_ph_flag	u(1)
sao_present_in_ph_flag	u(1)
alf_present_in_ph_flag	u(1)
...	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
if(pps_weighted_pred_flag pps_weighted_bipred_flag rpl_present_in_ph_flag)	
weighted_pred_table_present_in_ph_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_present_in_ph_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
...	
}	

rpl_present_in_ph_flagが1に等しいことは、PPSを参照するスライス・ヘッダにおいて参照ピクチャ・リスト・シグナリングが存在しないが、PPSを参照するPHにおいて存在してもよいことを指定する。rpl_present_in_ph_flagが0に等しいことは、PPSを参照するPHにおいて参照ピクチャ・リスト・シグナリングが存在しないが、PPSを参照するスライス・ヘッダにおいて存在してもよいことを指定する。

sao_present_in_ph_flagが1に等しいことは、PPSを参照するスライス・ヘッダにおいてSAOの使用を可能にするための構文要素が存在しないが、P

PPSを参照するPHにおいて存在してもよいことを指定する。sao_present_in_ph_flagが0に等しいことは、PPSを参照するPHにおいてSAOの使用を可能にするための構文要素が存在しないが、PPSを参照するスライス・ヘッダにおいて存在してもよいことを指定する。

alf_present_in_ph_flagが1に等しいことは、PPSを参照するスライス・ヘッダにおいてALFの使用を可能にするための構文要素が存在しないが、PPSを参照するPHにおいて存在してもよいことを指定する。alf_present_in_ph_flagが0に等しいことは、PPSを参照するPHにおいてALFの使用を可能にするための構文要素が存在しないが、PPSを参照するスライス・ヘッダにおいて存在してもよいことを指定する。

10

...

weighted_pred_table_present_in_ph_flagが1に等しいことは、PPSを参照するスライス・ヘッダにおいて重み付け予測表が存在しないが、PPSを参照するPHにおいて存在してもよいことを指定する。weighted_pred_table_present_in_ph_flagが0に等しいことは、PPSを参照するPHにおいて重み付け予測表が存在しないが、PPSを参照するスライス・ヘッダにおいて存在してもよいことを指定する。存在しないときに、weighted_pred_table_present_in_ph_flagの値は、0に等しいと推論される。

20

...

deblocking_filter_override_enabled_flagが1に等しいことは、PPSを参照するPHまたはスライス・ヘッダにおいてデブロッキング・フィルタ・オーバーライドが存在してもよいことを指定する。deblocking_filter_override_enabled_flagが0に等しいことは、PPSを参照するPH又はスライス・ヘッダのいずれにおいてもデブロッキング・フィルタ・オーバーライドが存在しないことを指定する。存在しないときに、deblocking_filter_override_enabled_flagの値は、0に等しいと推論される。

deblocking_filter_override_present_in_ph_flagが1に等しいことは、PPSを参照するスライス・ヘッダにおいてデブロッキング・フィルタ・オーバーライドが存在しないが、PPSを参照するPHにおいて存在してもよいことを指定する。deblocking_filter_override_present_in_ph_flagが0に等しいことは、PPSを参照するPHにおいてデブロッキング・フィルタ・オーバーライドが存在しないが、PPSを参照するスライス・ヘッダにおいて存在してもよいことを指定する。

30

40

50

【表 2 4】

...	
picture_header_rbsp() {	記述子
...	
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	
if(rpl_present_in_ph_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
pic_poc_lsb_ltrp[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_ltrp[i][j]	ue(v)
}	
}	
}	
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
——}	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	

10

20

30

40

50

pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
...	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag)	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
...	
}	

10

20

30

40

50

【表 2 5】

slice_header() {	記述子
...	
if(!rpl_present_in_ph_flag &&((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	10
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	20
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	30
if(rpl_present_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) {	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	40

10

20

30

40

50

if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
...	
if(slice_type != I) {	
...	
if(((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)) && !weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
}	
if(sps_sao_enabled_flag && !sao_present_in_ph_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !alf_present_in_ph_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !deblocking_filter_override_present_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)

10

20

30

40

50

if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
...	
}	

10

ピクチャ・ヘッダに対する代替的な構文は以下のようである。

20

30

40

50

【表 2 6】

picture_header_rbsp() {	記述子
...	
if(rpl_present_in_ph_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
——}	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	
pic_alf_enabled_flag	u(1)

10

20

30

40

50

if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
...	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag)	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
...	
}	

10

20

30

他の例では、ピクチャ・ヘッダ要素とスライス・ヘッダ要素のシグナリングが単一のプロセスにおいて組み合わせられ得る。

この例では、ピクチャ・ヘッダ及びスライス・ヘッダが組み合わせられるかどうかを示すフラグ（「**picture_header_in_slice_header_flag**」）を導入する。この例に従ったビットストリームに対する構文は以下のようである。

【表 2 7】

ピクチャ・ヘッダ RBSP 構文

picture_header_rbsp() {	記述子
picture_header_structure()	
}	

40

50

【表 2 8】

ピクチャ・ヘッダ構造構文

picture_header_structure() {	記述子
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
...	
}	

10

【表 2 9】

一般スライス・ヘッダ構文

slice_header() {	記述子
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag && (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
...	
}	

20

30

40

picture_header_in_slice_header_flag 及び関係するビットストリーム制約に対するセマンティクスは、以下のようである。

picture_header_in_slice_header_flag が 1 に等しいことは、スライス・ヘッダにおいてピクチャ・ヘッダ構文構造が存在することを指定する。**picture_header_in_slice_header_flag** が 0 に等しいことは、スライス・ヘッダにおいてピクチャ・ヘッダ構文構造が存在しないことを指定する。

picture_header_in_slice_header_flag の値が **CLVS** のすべてのスライスにおいて同じであることが、ビットストリーム適合性の要件であ

50

る。

`picture_header_in_slice_header_flag`が1に等しいときに、`PH_NUT`に等しいNALユニット・タイプを有するNALユニットがCLVSにおいて存在しないことがビットストリーム適合性の要件である。

`picture_header_in_slice_header_flag`が0に等しいときに、`PH_NUT`に等しいNALユニット・タイプを持つNALユニットが、PUの最初のVCL NALユニットを先行する、PUにおいて存在することがビットストリーム適合性の要件である。

これらの例の態様の組み合わせは、以下のようである。

`picture_header_in_slice_header_flag`が0に等しいときに、関連するコーディング・ツールの構文要素がPH又はSHのいずれか（両方ではない）に存在するかどうか指定するフラグ、

10

それ以外るとき（`picture_header_in_slice_header_flag`が1に等しいとき）、これらのフラグは、スライス・レベルでツール・パラメータ・シグナリングを示す0に推論される。

代替的な実装は以下のようである。

`picture_header_in_slice_header_flag`が0に等しいときに、関連するコーディング・ツールの構文要素がPH又はSHのいずれか（両方ではない）に存在するかどうか指定するフラグ、

それ以外るとき（`picture_header_in_slice_header_flag`が1に等しいとき）、これらのフラグは、ピクチャ・ヘッダ・レベルでツール・パラメータ・シグナリングを示す0に推論される。

20

この組み合わせは、以下の構文を有する。

30

40

50

【表 3 0】

ピクチャ・パラメータ・セット構文

pic_parameter_set_rbsp() {	記述子
picture_header_in_slice_header_flag	u(1)
...	
if (picture_header_in_slice_header_flag)	
{	
rpl_present_in_ph_flag	u(1)
sao_present_in_ph_flag	u(1)
alf_present_in_ph_flag	u(1)
}	
...	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
if(pps_weighted_pred_flag pps_weighted_bipred_flag rpl_present_in_ph_flag)	
weighted_pred_table_present_in_ph_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_present_in_ph_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
...	
}	

この例では、重み付き予測が有効かどうかのチェックは、重み付き予測で参照される参照ピクチャ・リストにおけるエントリの数を示すことによって実行される。
この例における代替的な構文及びセマンティクスは、以下のようである。

10

20

30

40

50

【表 3 1】

pred_weight_table() {	記述子
luma_log2_weight_denom	ue(v)
if(ChromaArrayType != 0)	
delta_chroma_log2_weight_denom	se(v)
num_l0_weighted_ref_pics	ue(v)
for(i = 0; i < num_l0_weighted_ref_pics; i++)	
luma_weight_l0_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[0]; i++)	
chroma_weight_l0_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[0]; i++) {	
if(luma_weight_l0_flag[i]) {	
delta_luma_weight_l0[i]	se(v)
luma_offset_l0[i]	se(v)
}	
if(chroma_weight_l0_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l0[i][j]	se(v)
delta_chroma_offset_l0[i][j]	se(v)
}	
}	
num_l1_weighted_ref_pics	ue(v)
for(i = 0; i < num_l1_weighted_ref_pics; i++)	
luma_weight_l1_flag[i]	u(1)
if(ChromaArrayType != 0)	
for(i = 0; i < NumRefIdxActive[1]; i++)	
chroma_weight_l1_flag[i]	u(1)
for(i = 0; i < NumRefIdxActive[1]; i++) {	
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_weight_l1_flag[i])	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)

delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	

num_l0_weighted_ref_picsは、重み付けされる参照ピクチャ・リスト0における参照ピクチャの数を指定する。num_l0_weighted_re

`f_pics`の値は、 $0 \sim \text{MaxDecPicBuffMinus1} + 14$ の範囲（両端を含む）にわたるものとする。

`num_l0_weighted_ref_pics`の値が、存在するときに、ピクチャ・ヘッダに関連するピクチャ内の任意のスライスのL0に対するアクティブな参照ピクチャの数より小さくないものとするのがビットストリーム適合性の要件である。

`num_l1_weighted_ref_pics`は、重み付けされる参照ピクチャ・リスト1における参照ピクチャの数を指定する。`num_l1_weighted_ref_pics`の値は、 $0 \sim \text{MaxDecPicBuffMinus1} + 14$ の範囲（両端を含む）にわたるものとする。

`num_l1_weighted_ref_pics`の値が、存在するときに、ピクチャ・ヘッダに関連するピクチャ内の任意のスライスのL1に対するアクティブな参照ピクチャの数より小さくないものとするのがビットストリーム適合性の要件である。

...

`MaxNumGeoMergeCand`は、`num_l0_weighted_ref_pics`又は`num_l1_weighted_ref_pics`のいずれかが非ゼロであるときに、ゼロにセットされる。以下の構文は、この依存関係がどのように利用され得るかの例である。

【表 3 2】

<code>if(sps_prof_pic_present_flag)</code>	
<code>pic_disable_prof_flag</code>	<code>u(1)</code>
<code>if(sps_geo_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_geo_cand_plus1 && num_l0_weighted_ref_pics==0 && num_l1_weighted_ref_pics==0)</code>	
<code>pic_max_num_merge_cand_minus_max_num_geo_cand</code>	<code>ue(v)</code>
<code>if(sps_ibc_enabled_flag)</code>	
<code>pic_six_minus_max_num_ibc_merge_cand</code>	<code>ue(v)</code>
<code>if(sps_joint_cbr_enabled_flag)</code>	

本実施形態における`pic_max_num_merge_cand_minus_max_num_geo_cand`のセマンティクスは、前述の実施形態に対するものと同じである。

一例では、PHに関連するピクチャにおいて特定のスライス・タイプのみが使用される場合、インター及びイントラ関係構文要素が条件付きでシグナリングされる。

この例に対する構文は、以下に与えられる。

10

20

30

40

50

【表 3 3】

picture_header_rbsp() {	記述子
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_virtual_boundaries_present_flag) {	
ph_virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
}	

10

20

30

40

50

if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
pic_poc_lsb_lf[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lf[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(pic_intra_slice_present_flag) {	
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	

10

20

30

40

50

pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
→}	
if(cu_qp_delta_enabled_flag) {	
pic_cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(pic_inter_slice_present_flag) {	
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mv_d_l1_zero_idc)	
mv_d_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)

10

20

30

40

50

if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if(sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
...	

10

【表 3 4】

20

7.3.7.1 一般スライス・ヘッダ構文

slice_header() {	記述子
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
if(pic_inter_slice_present_flag)	
slice_type	ue(v)
...	

30

7. 4. 3. 6 ピクチャ・ヘッダRBSPセマンティクス

pic_inter_slice_present_flagが1に等しいことは、0 (B) 又は1 (P) に等しい**slice_type**を有する1つ以上のスライスが、PHに関連するピクチャにおいて存在してもよいことを指定する。**pic_inter_slice_present_flag**が0に等しいことは、0 (B) 又は1 (P) に等しい**slice_type**を有するスライスが、PHに関連するピクチャにおいて存在し得ないことを指定する。

40

pic_intra_slice_present_flagが1に等しいことは、2 (I) に等しい**slice_type**を有する1つ以上のスライスが、PHに関連するピクチャにおいて存在してもよいことを指定する。**pic_intra_slice_present_flag**が0に等しいことは、2 (I) に等しい**slice_type**を有するスライスが、PHに関連するピクチャにおいて存在し得ないことを指定する。

存在しないときに、**pic_intra_slice_only_flag**の値は、1に等しいと推論される。

注 - : **pic_inter_slice_present_flag**と**pic_intra_slice_present_flag**の両方の値は、インターコーディングされ

50

たスライス（複数可）を含む1つ以上のサブピクチャ（複数可）とマージされてもよい、イントラコーディングされたスライス（複数可）を含む1つ以上のサブピクチャを含むピクチャに関連するピクチャ・ヘッダにおいて1に等しくセットされる。

7. 4. 8. 1 一般的なスライス・ヘッダ・セマンティクス

`slice_type`は、表7-5に従ってスライスのコーディング・タイプを指定する。
【表35】

表7-5 `slice_type` への名前の関連付け

<code>slice_type</code>	<code>slice_type</code> の名前
0	B (B slice)
1	P (P slice)
2	I (I slice)

`nal_unit_type`が`IDR_W_RADL`～`CRA_NUT`の範囲（両端を含む）の`nal_unit_type`の値で、現在のピクチャがアクセス・ユニットにおける第1のピクチャであるときに、`slice_type`は2に等しいものとする。

存在しないときに、`slice_type`の値は、2に等しいと推論される。

`pic_intra_slice_present_flag`が0に等しいときに、`slice_type`の値は、0～1の範囲（両端を含む）にあるものとする。

この例は、ピクチャ・ヘッダの`pred_weight_table()`のシグナリングと組み合和され得る。

ピクチャ・ヘッダにおける`pred_weight_table()`のシグナリングは、先の例において開示されている。

代替的な実装は、以下のようである。

【表36】

<code>picture_header_rbsp() {</code>	記述子
...	
...	
<code>if((pps_weighted_pred_flag pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag)</code>	
<code>pred_weight_table()</code>	
...	

ピクチャ・ヘッダにおいて`pred_weight_table()`が存在することを示すときに、以下の構文が使用され得る。

10

20

30

40

50

【表 3 7】

picture_header_rbsp() {	記述子
...	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && pic_inter_slice_present_flag)	
pred_weight_table()	
...	

10

代替的な例は、以下の構文を使用してもよい。

【表 3 8】

picture_header_rbsp() {	記述子
...	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

20

30

代替的な例は、以下の構文を使用してもよい。

【表 3 9】

picture_header_rbsp() {	記述子
...	
pic_inter_bipred_slice_present_flag	u(1)
if(!pic_inter_bipred_slice_present_flag)	
pic_inter_slice_present_flag	u(1)
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

40

上記の構文では、**pic_inter_bipred_slice_present_f**

50

lag は、ピクチャ・ヘッダを参照するすべてのスライス・タイプ、I スライス、B スライス、P スライスの存在を示す。

pic_inter_bipred_slice_present_flag が 0 であるときに、ピクチャは、I タイプ又は B タイプのスライスのみを含む。

この場合に、非矩形モードは無効である。

一例では、上記の例の組み合わせが開示される。例示的な構文は、以下のように記載される。

【表 40】

picture_header_rbsp() {	記述子	
...		
pic_inter_slice_present_flag	u(1)	
if(pic_inter_slice_present_flag)		
pic_intra_slice_present_flag	u(1)	
pic_rpl_present_flag	u(1)	
if(pic_rpl_present_flag) {		
if(rpl_present_in_ph_flag) {		
for(i = 0; i < 2; i++) {		
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))		20
pic_rpl_sps_flag[i]	u(1)	
if(pic_rpl_sps_flag[i]) {		
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))		
pic_rpl_idx[i]	u(v)	
} else		
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])		
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {		30
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])		
pic_poc_lsb_lt[i][j]	u(v)	
pic_delta_poc_msb_present_flag[i][j]	u(1)	
if(pic_delta_poc_msb_present_flag[i][j])		
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)	
}		
}		
}		
...		40
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && weighted_pred_table_present_in_ph_flag && pic_inter_slice_present_flag)		
pred_weight_table()		
...		
if(sps_sao_enabled_flag && sao_present_in_ph_flag) {		
pic_sao_enabled_present_flag	u(1)	

10

20

30

40

50

— if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag && alf_present_in_ph_flag) {	
— pic_alf_enabled_present_flag	u(1)
— if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
}	
...	
if(deblocking_filter_override_enabled_flag && deblocking_filter_override_present_in_ph_flag) {	
— pic_deblocking_filter_override_present_flag	u(1)
— if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	

10

20

30

40

— }	
...	
}	

一例では、重み付き予測ファクタのないピクチャを参照する非矩形（例えば、GEO）モードを選択することが許可される。

この例では、セマンティクスは、以下のように定義される。

7. 4. 10. 7 マージ・データ・セマンティクス

50

...

変数 `MergeGeoFlag[x0][y0]` は、Bスライスを復号するときに、現在のコーディング・ユニットの予測サンプルを生成するために `geo` 形状ベースのモーション補償が使用されるかどうかを指定し、以下のように導出される。

ー 以下の条件がすべて真である場合、`MergeGeoFlag[x0][y0]` は、1に等しくセットされる。

- ー `sps_geo_enabled_flag` が1に等しい。
- ー `slice_type` がBに等しい。
- ー `general_merge_flag[x0][y0]` が1に等しい。
- ー `MaxNumGeoMergeCand` が2つ以上である。
- ー `cbWidth` が8以上である。
- ー `cbHeight` が8以上である。
- ー `cbWidth` が $8 * cbHeight$ より小さい。
- ー `cbHeight` が $8 * cbWidth$ より小さい。
- ー `regular_merge_flag[x0][y0]` が0に等しい。
- ー `merge_subblock_flag[x0][y0]` が0に等しい。
- ー `cip_flag[x0][y0]` が0に等しい。

ー それ以外の場合、`MergeGeoFlag[x0][y0]` は、0に等しくセットされる。

CUの輝度又は色差明示的重み付けフラグの1つが真である場合、`MergeGeoFlag[x0][y0]` が0に等しいことがビットストリーム適合性の要件である。

一例では、VVC仕様の一部は、以下のように記載される。

8.5.7 geoインターブロックのための復号プロセス

8.5.7.1 概要

このプロセスは、`MergeGeoFlag[xCb][yCb]` が1に等しいコーディング・ユニットを復号するときに呼び出される。

このプロセスへの入力は、

- ー 現在のピクチャの左上の輝度サンプルに対する現在の輝度コーディング・ブロックの左上のサンプルの輝度位置 (`xCb`, `yCb`)、
- ー 輝度サンプルにおける現在のコーディング・ブロックの幅を指定する変数 `cbWidth`、
- ー 輝度サンプルにおける現在のコーディング・ブロックの高さを指定する変数 `cbHeight`、
- ー $1/16$ の分数サンプル精度 `mvA` 及び `mvB` における輝度モーション・ベクトル、
- ー 色差モーション・ベクトル `mvCA` 及び `mvCB`、
- ー 参照インデックス `refIdxA` 及び `refIdxB`、
- ー 予測リスト・フラグ `predListFlagA` 及び `predListFlagB`、

...

`predSamplesLAL` 及び `predSamplesLBL` を、予測された輝度サンプル値の $(cbWidth) \times (cbHeight)$ 配列とし、`predSamplesLAcB`、`predSamplesLBcB`、`predSamplesLAcR` 及び `predSamplesLBcR` を、予測された色差サンプル値の $(cbWidth / SubWidthC) \times (cbHeight / SubHeightC)$ 配列とする。`predSamplesL`、`predSamplescB` 及び `predSamplescR` は、以下の順序付けられたステップによって導出される。

1. `N` がAとBの各々である場合、以下が適用される：

...

2. 表36に指定されるように、`merge_geo_partition_idx[xCb][yCb]` の値に従って、マージ `geo` モード変数 `angleIdx` 及び `distanceIdx` のパーティション角度及び距離がセットされる。

3. 変数 `explicitWeightedFlag` は、以下のように導出される。

【数 5 2】

```
lumaWeightedFlagA = predListFlagA ? luma_weight_11_flag[ refIdxA ] : luma_weight_10_flag[ refIdxA ]
lumaWeightedFlagB = predListFlagB ? luma_weight_11_flag[ refIdxB ] : luma_weight_10_flag[ refIdxB ]
chromaWeightedFlagA = predListFlagA ? chroma_weight_11_flag[ refIdxA ] : chroma_weight_10_flag[ refIdxA ]
chromaWeightedFlagB = predListFlagB ? chroma_weight_11_flag[ refIdxB ] : chroma_weight_10_flag[ refIdxB ]
weightedFlag = lumaWeightedFlagA || lumaWeightedFlagB || chromaWeightedFlagA || chromaWeightedFlagB
```

10

4. 現在の輝度コーディング・ブロック内の予測サンプル `predSamplesL[xL][yL]` ($xL=0..cbWidth-1$ 及び $yL=0..cbHeight-1$) は、`weightedFlag` が 0 に等しい場合、8.5.7.2 節において指定されている `geo` マージ・モードのための重み付けサンプル予測プロセス、及び `weightFlag` が 1 に等しい場合、8.5.6.6.3 節における明示的な重み付けサンプル予測プロセスを、`cbWidth` に等しくセットされたコーディング・ブロック幅 `nCbW`、`cbHeight` に等しくセットされたコーディング・ブロック高さ `nCbH`、サンプル配列 `predSamplesLAL` 及び `predSamplesLBL`、変数 `angleIdx` 及び `distanceIdx`、並びに 0 に等しい `cIdx` を入力として呼び出すことによって導出される。

20

5. 現在の色差コンポーネント `Cb` コーディング・ブロック内の予測サンプル `predSamplescb[xc][yc]` ($xc=0..cbWidth/SubWidthC-1$ 及び $yc=0..cbHeight/SubHeightC-1$) は、`weightedFlag` が 0 に等しい場合、8.5.7.2 節において指定されている `geo` マージ・モードのための重み付けサンプル予測プロセス、及び `weightFlag` が 1 に等しい場合、8.5.6.6.3 節における明示的な重み付けサンプル予測プロセスを、`cbWidth/SubWidthC` に等しくセットされたコーディング・ブロック幅 `nCbW`、`cbHeight/SubHeightC` に等しくセットされたコーディング・ブロック高さ `nCbH`、サンプル配列 `predSamplesLAcB` 及び `predSamplesLBcB`、変数 `angleIdx` 及び `distanceIdx`、並びに 1 に等しい `cIdx` を入力として呼び出すことによって導出される。

30

6. 現在の色差コンポーネント `Cr` コーディング・ブロック内の予測サンプル `predSamplescr[xc][yc]` ($xc=0..cbWidth/SubWidthC-1$ 及び $yc=0..cbHeight/SubHeightC-1$) は、`weightedFlag` が 0 に等しい場合、8.5.7.2 節において指定されている `geo` マージ・モードのための重み付けサンプル予測プロセス、及び `weightFlag` が 1 に等しい場合、8.5.6.6.3 節における明示的な重み付けサンプル予測プロセスを、`cbWidth/SubWidthC` に等しくセットされたコーディング・ブロック幅 `nCbW`、`cbHeight/SubHeightC` に等しくセットされたコーディング・ブロック高さ `nCbH`、サンプル配列 `predSamplesLAcr` 及び `predSamplesLBcr`、変数 `angleIdx` 及び `distanceIdx`、並びに 2 に等しい `cIdx` を入力として呼び出すことによって導出される。

40

7. 8.5.7.3 節において指定されているマージ `geo` モードのためのモーション・ベクトル記憶処理は、輝度コーディング・ブロック位置 (xCb , yCb)、輝度コーディング・ブロック幅 `cbWidth`、輝度コーディング・ブロック高さ `cbHeight`、パーティション方向 `angleIdx` 及び `distanceIdx`、輝度モーション・ベクトル `mvA` 及び `mvB`、参照インデックス `refIdxA` 及び `refIdxB`、並

50

びに予測リスト・フラグ predListFlagA 及び predListFlagB を入力として呼び出される。

【表 4 1】

表 3 6 — Table 36—merge_geo_partition_idx 値に基づいた angleIdx 及び distanceIdx の仕様

merge_geo_p artition_idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
angleIdx	0	0	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
distanceIdx	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2
merge_geo_p artition_idx	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
angleIdx	4	6	6	8	8	8	8	9	9	9	9	10	10	10	10	11	11
distanceIdx	3	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
merge_geo_p artition_idx	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
angleIdx	11	11	12	12	13	13	13	14	14	14	15	15	15	16	16	16	18
distanceIdx	2	3	1	3	1	2	3	1	2	3	1	2	3	1	2	3	1
merge_geo_p artition_idx	51	52	53	54	55	56	57	58	59	60	61	62	63				
angleIdx	18	20	20	20	21	21	21	22	22	22	23	23	23				
distanceIdx	3	1	2	3	1	2	3	1	2	3	1	2	3				

8. 5. 6. 6. 3 明示的な重み付けサンプル予測プロセス

このプロセスへの入力は、

- 現在のコーディング・ブロックの幅及び高さを指定する2つの変数 $nCbW$ 及び $nCbH$ 、
- 2つの $(nCbW) \times (nCbH)$ 配列 presamplesL0 及び predSamplesL1 、
- 予測リスト利用フラグ predFlagL0 及び predFlagL1 、
- 参照インデックス refIdxL0 及び refIdxL1 、
- 色コンポーネント・インデックスを指定する変数 $cIdx$ 、
- サンプルビット深さ $bitDepth$ 、である。

このプロセスの出力は、予測サンプル値の $(nCbW) \times (nCbH)$ 配列 pbSamples である。変数 shift1 は、 $\text{Max}(2, 14 - bitDepth)$ に等しくセットされる。変数 log2Wd 、 $o0$ 、 $o1$ 、 $w0$ 及び $w1$ は、以下のように導出される。

— 輝度サンプルに対して $cIdx$ が 0 に等しい場合、以下が適用される。

【数 5 3】

$$\text{log2Wd} = \text{luma_log2_weight_denom} + \text{shift1} \quad (1010)$$

$$w0 = \text{LumaWeightL0}[\text{refIdxL0}] \quad (1011)$$

$$w1 = \text{LumaWeightL1}[\text{refIdxL1}] \quad (1012)$$

$$o0 = \text{luma_offset_l0}[\text{refIdxL0}] \ll (\text{bitDepth} - 8) \quad (1013)$$

$$o1 = \text{luma_offset_l1}[\text{refIdxL1}] \ll (\text{bitDepth} - 8) \quad (1014)$$

— それ以外の場合（色差サンプルに対して $cIdx$ が 0 に等しくない場合）、以下が適用される。

【数 5 4】

10

20

30

40

50

$$\log 2 W d = \text{ChromaLog2WeightDenom} + \text{shift1} \quad (1015)$$

$$w0 = \text{ChromaWeightL0}[\text{refIdxL0}][\text{cIdx} - 1] \quad (1016)$$

$$w1 = \text{ChromaWeightL1}[\text{refIdxL1}][\text{cIdx} - 1] \quad (1017)$$

$$o0 = \text{ChromaOffsetL0}[\text{refIdxL0}][\text{cIdx} - 1] \ll (\text{bitDepth} - 8) \quad (1018)$$

$$o1 = \text{ChromaOffsetL1}[\text{refIdxL1}][\text{cIdx} - 1] \ll (\text{bitDepth} - 8) \quad (1019)$$

10

予測サンプル $\text{pbSamples}[x][y]$ ($x = 0 \dots nCbW - 1$ 及び $y = 0 \dots nCbH - 1$) は、以下のように導出される。

－ predFlagL0 が 1 に等しく、 predFlagL1 が 0 に等しい場合、予測サンプル値は、以下のように導出される。

【数 5 5】

$$\begin{aligned} & \text{if}(\log 2 W d \geq 1) \\ & \quad \text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \\ & \quad \quad ((\text{predSamplesL0}[x][y] * w0 + 2^{\log 2 W d - 1}) \gg \log 2 W d) + o0) \\ & \quad \quad \quad (1020) \\ & \text{else} \\ & \quad \text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \text{predSamplesL0}[x][y] * w0 + o0) \end{aligned}$$

20

－ それ以外の場合、 predFlagL0 が 0 に等しく、 predFlagL1 が 1 に等しい場合、予測サンプル値は、以下のように導出される。

【数 5 6】

$$\begin{aligned} & \text{if}(\log 2 W d \geq 1) \\ & \quad \text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \\ & \quad \quad ((\text{predSamplesL1}[x][y] * w1 + 2^{\log 2 W d - 1}) \gg \log 2 W d) + o1) \\ & \quad \quad \quad (1021) \\ & \text{else} \\ & \quad \text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \text{predSamplesL1}[x][y] * w1 + o1) \end{aligned}$$

30

－ それ以外の場合 (predFlagL0 が 1 に等しく、 predFlagL1 が 1 に等しい場合)、予測サンプル値は、以下のように導出される。

【数 5 7】

$$\begin{aligned} & \text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, \\ & \quad (\text{predSamplesL0}[x][y] * w0 + \text{predSamplesL1}[x][y] * w1 + \\ & \quad \quad ((o0 + o1 + 1) \ll \log 2 W d)) \gg (\log 2 W d + 1)) \\ & \quad \quad \quad (1022) \end{aligned}$$

40

この例では、非矩形マージ・モード（例えば、GEOモード）の存在を示す変数のチェックを含むマージ・データ・パラメータの構文が開示される。構文例が以下に与えられる。

【表 4 2】

merge_data(x0, y0, cbWidth, cbHeight, chType) {	記述子
if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if(cbWidth < 128 && cbHeight < 128 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64) (sps_geo_enabled_flag && MaxNumGeoMergeCand > 1 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < 8 * cbHeight && cbHeight < 8 * cbWidth && slice_type == B)))	
regular_merge_flag [x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag [x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && sps_geo_enabled_flag && MaxNumGeoMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < 8 * cbHeight && cbHeight < 8 * cbWidth && cbWidth < 128 && cbHeight < 128)	
ciip_flag [x0][y0]	ae(v)

10

20

30

40

50

if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && MaxNumGeoMergeCand > 1) {	
merge_geo_partition_idx[x0][y0]	ae(v)
merge_geo_idx0[x0][y0]	ae(v)
if(MaxNumGeoMergeCand > 2)	
merge_geo_idx1[x0][y0]	ae(v)
}	
}	
}	
}	
}	

10

変数 **MaxNumGeoMergeCand** は、先の例のいずれかに従って導出される。**MaxNumGeoMergeCand** 変数から導出される代替的な変数 **SliceMaxNumGeoMergeCand** が使用されてもよい。**MaxNumGeoMergeCand** の値は、より高いシグナリング・レベル（例えば、PH、PPS又はSPS）で取得される。

20

一例では、**SliceMaxNumGeoMergeCand** は、**MaxNumGeoMergeCand** の値と、スライスに対して実行される追加のチェックに基づいて導出される。

例えば、

【数 5 8】

$\text{SliceMaxNumGeoMergeCand} = (\text{num_l0_weighted_ref_pics} > 0 \parallel \text{num_l1_weighted_ref_pics} > 0) ? 0 :$

$\text{MaxNumGeoMergeCand}.$

別の例では、以下の式を仕様して **MaxNumGeoMergeCand** 値を決定する。

30

【数 5 9】

$\text{SliceMaxNumGeoMergeCand} = (!\text{pic_inter_slice_present_flag}) ? 0 :$

$\text{MaxNumGeoMergeCand}.$

一例では、

以下の構文テーブルが定義される。

40

50

【表 4 3】

picture_header_rbsp() {	記述子
...	
pic_inter_bipred_slice_present_flag	
if(!pic_inter_bipred_slice_present_flag)	
pic_inter_slice_present_flag	
if(pic_inter_slice_present_flag)	
pic_intra_slice_present_flag	u(1)
...	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && pic_inter_slice_present_flag && weighted_pred_table_present_in_ph_flag)	
pred_weight_table()	
...	

10

変数MaxNumGeoMergeCandは、以下のように導出される。

【数 6 0】

SliceMaxNumGeoMergeCand = (!pic_inter_bipred_slice_present_flag) ? 0: MaxNumGeoMergeCand.

20

矩形モード及び非矩形モードのためのマージ候補の数を示す方法が開示される。矩形モード及び非矩形モードのためのマージ候補の数は相互に依存しており、矩形モードのためのマージ候補の数が閾値より少ないことが示されたときには、非矩形モードのマージ候補の数を示す必要はなくてもよい。

特に、TPM又はGeoマージ・モードの場合、それらの非矩形マージ・モードのいずれかを使用して予測されるブロックは、それらに対して指定される異なるMVを有する2つのインター予測子を必要とするため、マージ・モードに対して少なくとも2つの候補があるべきである。実施形態では、マージ・モード候補の数がシーケンス・パラメータ・セット（SPS）において示されるときに、以下の構文が使用され得る。

30

40

50

【表 4 4】

7.3.2.3 シーケンス・パラメータ・セット RBSP 構文

seq_parameter_set_rbsp() {	記述子
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
sps_seq_parameter_set_id	u(4)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
ref_pic_resampling_enabled_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_log2_ctu_size_minus5	u(2)
subpics_present_flag	u(1)
if(subpics_present_flag) {	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
subpic_ctu_top_left_x[i]	u(v)
subpic_ctu_top_left_y[i]	u(v)
subpic_width_minus1[i]	u(v)
subpic_height_minus1[i]	u(v)
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_present_flag	u(1)
if(sps_subpic_id_present_flag) {	
sps_subpic_id_signalling_present_flag	u(1)
if(sps_subpic_id_signalling_present_flag) {	
sps_subpic_id_len_minus1	ue(v)

10

20

30

40

50

for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
bit_depth_minus8	ue(v)
min_qp_prime_ts_minus4	ue(v)
sps_weighted_pred_flag	u(1)
sps_weighted_bipred_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	u(4)
sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
dpb_parameters(0, sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)

10

20

30

40

50

sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
sps_joint_cbr_enabled_flag	u(1)
if(ChromaArrayType != 0) {	
same_qp_table_for_chroma	u(1)
numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbr_enabled_flag ? 3 : 2)	
for(i = 0; i < numQpTables; i++) {	
qp_table_start_minus26[i]	se(v)
num_points_in_qp_table_minus1[i]	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
delta_qp_in_val_minus1[i][j]	ue(v)
delta_qp_diff_val[i][j]	ue(v)
}	
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag)	
sps_bdpcm_enabled_flag	u(1)
if(sps_bdpcm_enabled_flag && chroma_format_idc == 3)	
sps_bdpcm_chroma_enabled_flag	u(1)

10

20

30

40

50

sps_ref_wraparound_enabled_flag	u(1)
if(sps_ref_wraparound_enabled_flag)	
sps_ref_wraparound_offset_minus1	ue(v)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
if(sps_bdof_enabled_flag)	
sps_bdof_pic_present_flag	u(1)
sps_smvd_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
if(sps_dmvr_enabled_flag)	
sps_dmvr_pic_present_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(chroma_format_idc == 1) {	
sps_chroma_horizontal_collocated_flag	u(1)
sps_chroma_vertical_collocated_flag	u(1)
}	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
sps_six_minus_max_num_merge_cand	ue(v)
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
sps_five_minus_max_num_subblock_merge_cand	ue(v)
sps_affine_type_flag	u(1)

10

20

30

40

50

sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
if(chroma_format_idc == 3) {	
sps_act_enabled_flag	u(1)
sps_palette_enabled_flag	u(1)
}	
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
if(sps_ibc_enabled_flag)	
sps_six_minus_max_num_ibc_merge_cand	ue(v)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
sps_geo_enabled_flag	u(1)
if(sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_scaling_list_enabled_flag	u(1)
sps_virtual_boundaries_present_flag	u(1)
if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	

10

20

30

40

50

sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)
if(sps_general_hrd_params_present_flag) {	
general_hrd_parameters()	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_cpb_params_present_flag	u(1)
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 :	
sps_max_sublayers_minus1	
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)	
}	
}	
field_seq_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters() /* Specified in ITU-T H.SEI ISO/IEC 23002-7 */	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbbsp_trailing_bits()	
}	

10

20

30

本発明の一実施形態によれば、SPSにおけるマージ・モード候補の数を示すために、以下のステップが実行され、

ー 通常モードに対するマージ・モード候補の数 (MaxNumMergeCand) を示すことと、

ー 非矩形モードが非矩形マージ有効フラグ (sps_geo_enabled_flag) によって有効であるかどうかを示すことと、

ー 非矩形マージ有効フラグ値が非ゼロである場合に、通常のマージ・モードに対するマージ・モード候補の数が第1の閾値を超えるとときに、非矩形モード・モードの数 (sps_max_num_merge_cand_minus_max_num_geo_cand) を示すことと、を含む。

40

通常モードに対するマージ・モード候補の数が第2の閾値、例えば1を超えるとときに、非矩形マージ有効フラグを示すことが実行される。

実施形態1では、このステップのシーケンスは、VVC仕様のSPS構文の以下の部分として示される。

50

【表 4 5】

...	
if (MaxNumMergeCand > 1)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

10

この実施形態では、2つの順次チェックが実行され、第2のチェックは、第1のチェックの結果に従ってシグナリングされるか、又はシグナリングされないフラグの値に依存する。実施形態2は、実施形態1に対して記載されたプロセスと比較して異なって第2のチェックを実行した。特に、実施形態1は、「以上」の代わりに「より大きい」条件を使用する。このステップのシーケンスは、VVC仕様のSPS構文の以下の部分として示されている。

【表 4 6】

...	
if (MaxNumMergeCand > 1)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

20

実施形態3は、第1のチェックが偽の値をもたらすときに、第2のチェックは実行されず、非矩形マージ有効化フラグ値 (sps_geo_enabled_flag) は、sps_six_minus_max_num_merge_cand構文要素からのMaxNumMergeCand値の導出のプロセスが終了した後に決定されるという点で実施形態1と異なるが、sps_geo_enabled_flagの値は、MaxNumMergeCandのいくつかの値に対して参照されず、解析プロセスにおける処理から省略することができるため、技術的効果がある。実施形態3に従って実行されるステップのこのシーケンスは、VVC仕様のSPS構文の以下の部分として示される。

30

【表 4 7】

if (MaxNumMergeCand > 1) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

40

実施形態4は、実施形態2及び実施形態3の態様の組み合わせである。実施形態4に従って実行されるステップのシーケンスは、VVC仕様のSPS構文の以下の部分として示される。

50

【表 4 8】

...	
if (MaxNumMergeCand > 1) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

10

実施形態 5～8 は、第 1 のチェック及び第 2 のチェックの異なる定式化を開示する。
これらの実施形態は、以下のように記載されてもよい。

実施形態 5

【表 4 9】

...	
if (MaxNumMergeCand >= 2)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

20

実施形態 6

【表 5 0】

...	
if (MaxNumMergeCand >= 2)	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
...	

30

実施形態 7

【表 5 1】

...	
if (MaxNumMergeCand >= 2) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

40

実施形態 8

50

【表 5 2】

...	
if (MaxNumMergeCand >= 2) {	
sps_geo_enabled_flag	u(1)
if (sps_geo_enabled_flag && MaxNumMergeCand > 2)	
sps_max_num_merge_cand_minus_max_num_geo_cand	ue(v)
}	
...	

10

【0195】

図15に示すような一実装では、ビデオ復号のための幾何分的パーティショニング・マージャ・モード候補の最大数を取得する方法が開示されており、方法は、以下を含む。ビデオ・シーケンスのためのビットストリームを取得する（S1501）。

【0196】

ビットストリームは、無線ネットワーク又は有線ネットワークに従って取得されてもよい。ビットストリームは、同軸ケーブル、光ファイバケーブル、ツイスト・ペア、デジタル加入者線、又は赤外線、ラジオ、マイクロ波、WIFI、Bluetooth、LTE若しくは5Gの無線技術を使用して、ウェブサイト、サーバ、又は他のリモート・ソースから送信されてもよい。

20

【0197】

一実施形態では、ビットストリームは、1つ以上のコーディングされたビデオ・シーケンス（CVS）を形成するアクセス・ユニット（AU）のシーケンスの表現を形成するネットワーク抽象化レイヤ（NAL）ユニット・ストリーム又はバイト・ストリームの形式のビットのシーケンスである。

【0198】

いくつかの実施形態では、復号プロセスの場合、復号器側がビットストリームを読み、ビットストリームから復号されたピクチャを導出し、符号化プロセスの場合、符号化器側がビットストリームを生成する。

30

【0199】

通常、ビットストリームは、構文構造によって形成される構文要素を含むだろう。

構文要素：ビットストリームにおいて表されたデータの要素

構文構造：指定された順序でビットストリームにおいて一緒に存在する0個以上の構文要素

【0200】

特定の例では、ビットストリームフォーマットは、ネットワーク抽象化レイヤ（NAL）ユニット・ストリームとバイト・ストリームとの間の関係を指定し、これらはいずれもビットストリームと呼ばれる。

【0201】

ビットストリームは、NALユニット・ストリーム・フォーマット又はバイト・ストリーム・フォーマットの2つのフォーマットうちの1つであり得る。NALユニット・ストリーム・フォーマットは概念的には、より「基本的な」タイプである。NALユニット・ストリーム・フォーマットは、NALユニットと呼ばれる構文構造のシーケンスを含む。このシーケンスは、復号順序で順序付けされる。NALユニット・ストリームにおけるNALユニットの復号順序（及び）に課される制約がある。

40

【0202】

バイト・ストリーム・フォーマットは、NALユニットを復号順に順序付け、各NALユニットを開始コードプレフィックス及びゼロ以上のゼロ値バイトを伴ってプレフィックスしてバイトのストリームを形成することによって、NALユニット・ストリーム・フォ

50

ーマットから構成され得る。NALユニット・ストリーム・フォーマットは、このバイトのストリームにおけるユニークな開始コード・プレフィックス・パターンの位置をサーチすることによって、バイト・ストリーム・フォーマットから抽出され得る。

【0203】

この項は、ビットストリームを介して与えられる、ソースと復号されたピクチャとの間の関係を指定する。

【0204】

ビットストリームによって表されるビデオ・ソースは、復号順序のピクチャのシーケンスである。

【0205】

ソース及び復号されたピクチャは各々、1つ以上のサンプル配列から構成される。

- － 輝度（Y）のみ（モノクロ）。
- － 輝度及び2つの色差（YCbCr又はYCGCo）。
- － 緑、青、赤（GBR、RGBとしても知られている）。
- － 他の指定されていないモノクロ又は三刺激色のサンプリングを表す配列（例えば、YZX、XYZとしても知られている）。

【0206】

これらの配列に関連する変数及び用語は、輝度（又はL若しくはY）及び色差と呼ばれ、2つの色差配列は、実際に使用されている色表現方法にかかわらず、Cb及びCrと呼ばれる。実際に使用されている色表現方法は、ITU-T H.261 | ISO/IEC 23002-7において指定されているように、VUIパラメータにおいて指定されている構文において示され得る。

【0207】

ビットストリームに従って第1のインジケータの値を取得する（S1502）。

【0208】

第1のインジケータは、モーション・ベクトル予測MVP候補をマージする最大数MVP候補を表わす。

【0209】

一例では、第1のインジケータは、変数MaxNumMergeCandに従って表される。

【0210】

一例では、MVP候補をマージする最大数MaxNumMergeCandは、以下のように導出される。

【数61】

$$\text{MaxNumMergeCand} = 6 - \text{sps_six_minus_max_num_merge_cand.}$$

ここで、sps_six_minus_max_num_merge_candは、6から差し引かれた、SPSでサポートされるモーション・ベクトル予測（MVP）候補をマージする最大数を指定する。sps_six_minus_max_num_merge_candの値は、0～5の範囲（両端を含む）にあるものとする。

一例では、sps_six_minus_max_num_merge_candは、ビットストリームにおけるシーケンス・パラメータ・セットRBSP構文構造から解析される。

【0211】

ビットストリームに従って第2のインジケータの値を取得する（S1503）。

第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表わす。

【0212】

10

20

30

40

50

一例では、第2のインジケータは、`sps_gpm_enabled_flag` (`sps_gpm_enabled_flag`) に従って表される。`sps_gpm_enabled_flag` が1に等しいことは、CLVSに対して幾何学的パーティション・ベースのモーション補償が有効であることを指定し、`merge_gpm_partition_idx`、`merge_gpm_idx0`、及び`merge_gpm_idx1`は、CLVSのコーディング・ユニット構文において存在することができる。`sps_gpm_enabled_flag` が0に等しいことは、CLVSに対して幾何学的パーティション・ベースのモーション補償が無効であることを指定し、`merge_gpm_partition_idx`、`merge_gpm_idx0`、及び`merge_gpm_idx1`は、CLVSのコーディング・ユニット構文において存在しないことを指定する。存在しないときに、`sps_gpm_enabled_flag` の値は、0に等しいと推論される。

10

【0213】

一実装では、第2のインジケータの値を取得することは、第1のインジケータの値を取得することの後に実行される。

【0214】

一実装では、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから取得される。

【0215】

一実装では、第1のインジケータの値が閾値以上であるときに、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから解析される。しきい値は整数値であり、一例では、閾値は2である。

20

【0216】

例えば、第2のインジケータ`sps_gpm_enabled_flag`の値は、以下に従って取得される。

シーケンス・パラメータ・セットRBSP構文

【表53】

<code>if(MaxNumMergeCand >= 2){</code>	
<code>sps_gpm_enabled_flag</code>	<code>u(1)</code>

30

【0217】

ビットストリームから第3のインジケータの値を解析する (S1504)。

【0218】

一実装では、第1のインジケータの値が閾値より大きいとき、及び第2のインジケータの値がプリセット値に等しいときに、ビットストリームから第3のインジケータの値を解析し、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す。閾値は整数値であり、プリセット値は整数値である。一例では、閾値は2である。

【0219】

一例では、プリセット値は1である。

40

【0220】

一例では、第3のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから取得される。

【0221】

一例では、第3のインジケータは、`sps_max_num_merge_cand_minus_max_num_geo_cand` (`sps_max_num_merge_cand_minus_max_num_gpm_cand`) に従って表される。

例えば、第3インジケータ`sps_max_num_merge_cand_minus_max_num_gpm_cand`の値は、以下に従って取得される。

50

シーケンス・パラメータ・セット R B S P 構文

【表 5 4】

if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
sps_max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	

10

【0 2 2 2】

一実装では、方法は、第 1 のインジケータの値が閾値に等しく、かつ第 2 のインジケータの値がプリセット値に等しいときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を 2 にセットすることをさらに含む。

【0 2 2 3】

一実装では、方法は、第 1 のインジケータの値が閾値より小さいか、又は第 2 のインジケータの値がプリセット値に等しくないときに、何学的パーティショニング・マージ・モード候補の最大数の値を 0 にセットすることをさらに含む。

【0 2 2 4】

一例では、sps_max_num_merge_cand_minus_max_num_gpm_cand は、MaxNumMergeCand から差し引かれた、SPS でサポートされる幾何学的パーティショニング・マージ・モード候補の最大数を指定する。sps_max_num_merge_cand_minus_max_num_gpm_cand の値は、0 ~ MaxNumMergeCand - 2 の範囲（両端を含む）にあるものとする。

20

幾何学的パーティショニング・マージ・モード候補の最大数 MaxNumGpmMergeCand (MaxNumGeoMergeCand) は、以下のように導出される。

【数 6 2】

```
if( sps_gpm_enabled_flag && MaxNumMergeCand >= 3 )
```

30

```
    MaxNumGpmMergeCand = MaxNumMergeCand -
```

```
        sps_max_num_merge_cand_minus_max_num_gpm_cand
```

```
else if( sps_gpm_enabled_flag && MaxNumMergeCand == 2 )
```

```
    MaxNumGpmMergeCand = 2
```

```
else
```

```
    MaxNumGpmMergeCand = 0.
```

40

【0 2 2 5】

図 1 6 に示すような実装では、ビデオ復号装置 1 6 0 0 が開示されており、ビデオ復号装置は、ビデオ・シーケンスのためのビットストリームを取得するように構成されている受信モジュール 1 6 0 1 と、ビットストリームに従って第 1 のインジケータの値を取得するように構成されている取得モジュール 1 6 0 2 であって、第 1 のインジケータは、モーション・ベクトル予測 MVP 候補をマージする最大数を表し、取得モジュール 1 6 0 2 は、ビットストリームに従って第 2 のインジケータの値を取得するように構成されており、

50

第2のインジケータは、ビデオ・シーケンスに対して幾何学的パーティション・ベースのモーション補償が有効であるかどうかを表す、取得モジュール1602と、第1のインジケータの値が閾値より大きいとき、及び第2のインジケータの値がプリセット値に等しいときに、ビットストリームから第3のインジケータの値を解析するように構成されている解析モジュール1603であって、第3のインジケータは、第1のインジケータの値から差し引かれた幾何学的パーティショニング・マージ・モード候補の最大数を表す、解析モジュールと、を含む。

【0226】

一実装では、決定モジュール1602は、第1のインジケータの値が閾値に等しく、かつ第2のインジケータの値がプリセット値に等しいときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を2にセットするように構成されている。

10

【0227】

一実装では、決定モジュール1602は、第1のインジケータの値が閾値より小さいか、又は第2のインジケータの値がプリセット値に等しくないときに、幾何学的パーティショニング・マージ・モード候補の最大数の値を0にセットするように構成されている。

【0228】

一実装では、閾値は2である。

【0229】

一実装では、プリセット値は1である。

【0230】

一実装では、第2のインジケータの値を取得することは、第1のインジケータの値を取得することの後に実行される。

20

【0231】

一実装では、第1のインジケータの値が閾値以上であるときに、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから解析される。

【0232】

一実装では、第2のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから取得される。

【0233】

一実装では、第3のインジケータの値は、ビットストリームのシーケンス・パラメータ・セットSPSから取得される。

30

【0234】

受信モジュール1601、取得モジュール1602、及び解析モジュール1603に関するさらなる詳細は、上記の方法の例及び実装を参照することができる。

【0235】

例1．マージ・モード候補数のシグナリングを含むビデオ・コーディングの方法であって、

一 通常モードに対するマージ・モード候補の数 (MaxNumMergeCand) を示すことと、

一 非矩形モードが非矩形マージ有効フラグ (sps_geo_enabled_flag) によって有効であるかどうかを示すことと、

40

一 非矩形マージ有効フラグ値が非ゼロであり、通常のマージ・モードに対するマージ・モード候補の数が第1の閾値を超えるとときに、非矩形モード・モードの数 (sps_max_num_merge_cand_minus_max_num_geo_cand) を示すことと、を含み、

一 通常モードに対するマージ・モード候補の数が第2の閾値 (1) を超えるとときに、非矩形マージ有効フラグを示すことが実行される。

例2．非矩形マージ有効フラグ値は、sps_six_minus_max_num_merge_cand構文要素からMaxNumMergeCand値を導出するプロセスが終了した後に決定される、例1の方法。

50

例 3．閾値チェックは、通常のマージ・モードに対するマージ・モード候補の数が 2 より大きいかどうかの比較である、先の例の方法。

例 4．第 1 の閾値チェックは、通常のマージ・モードに対するマージ・モード候補の数が 3 より大きいかどうかの比較である、例 1 又は例 2 の方法。

【0236】

一例では、インター予測方法が開示されており、方法は、ブロックのグループに対して非矩形のインター予測モードが許可されているかどうかを決定することと、ブロックのグループに対する 1 つ以上のインター予測モード・パラメータ及び重み付き予測パラメータを取得することと、1 つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値を取得することであって、インター予測モード・パラメータのうちの 1 つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む、取得することと、を含む。

10

【0237】

一例では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付き予測が有効である場合に無効である。

【0238】

実現可能な実装では、非矩形インター予測モードは、重み付き予測が無効である場合に有効である。

【0239】

一例において、非矩形インター予測モードが許可されていると決定することは、三角マージ候補の最大数 (MaxNumTriangleMergeCand) が、1 より大きいことを示すことを示す。

20

【0240】

一例では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

【0241】

一例では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、スライスのスライス・ヘッダにおけるものである。

30

【0242】

一例では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0243】

一例では、非矩形インター予測モードは、幾何学的 (GEO) パーティショニング・モードである。

【0244】

一例では、重み付き予測パラメータは、スライス・レベル輝度補正のために使用される。

【0245】

一例では、重み付き予測パラメータは、ブロック・レベル輝度補正のために使用される。

40

【0246】

一例において、重み付き予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

【0247】

一例では、インター予測のための装置が開示されており、装置は、プロセッサ実行可能な命令が記憶された非一時的なメモリと、メモリに結合されたプロセッサと、を含み、プロセッサは、方法の例のいずれか 1 つを容易にするために、プロセッサ実行可能な命令を実行するように構成されている。

【0248】

50

一例では、インター予測のためのビットストリームが開示されており、ビットストリームは、ブロックのグループに対して非矩形インター予測モードが許可されているかどうかを決定するための指示情報情報と、ブロックのグループに対する1つ以上のインター予測モード及び重み付け予測パラメータと、を含み、1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値が取得され、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む。

【0249】

一例では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付き予測が有効である場合に無効である。

10

【0250】

一例では、非矩形インター予測モードは、重み付き予測が無効である場合に有効である。

【0251】

一例では、指示情報は、三角マージ候補の最大数 (MaxNumTriangleMergeCand) は、1より大きいことを含む。

【0252】

一例では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

【0253】

一例では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び指示情報は、ピクチャのスライス・ヘッダにおけるものである。

20

【0254】

一例では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0255】

一例では、非矩形インター予測モードは、幾何学的 (GEO) パーティショニング・モードである。

【0256】

一例では、重み付き予測パラメータは、スライス・レベル輝度補正のために使用される。

【0257】

一例では、重み付き予測パラメータは、ブロック・レベル輝度補正のために使用される。

30

【0258】

一例において、重み付き予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

【0259】

一例では、ブロックのグループに対して非矩形のインター予測モードが許可されているかどうかを決定するように構成されている決定モジュールと、ブロックのグループに対する1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータを取得するように構成されている取得モジュールと、1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値を取得するように構成されている予測モジュールであって、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む、予測モジュールと、を含むインター予測装置が開示されている。

40

【0260】

一例では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付き予測が有効である場合に無効である。

【0261】

一例では、非矩形インター予測モードは、重み付き予測が無効である場合に有効である。

50

【0262】

一例において、決定モジュールは、具体的には、三角マージ候補の最大数（MaxNumTriangleMergeCand）は、1より大きいことを示すように構成されている。

【0263】

一例では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

【0264】

一例では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、スライスのスライス・ヘッダにおけるものである。

【0265】

一例では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0266】

一例では、非矩形インター予測モードは、幾何学的（GEO）パーティショニング・モードである。

【0267】

一例では、重み付き予測パラメータは、スライス・レベル輝度補正のために使用される。

【0268】

一例では、重み付き予測パラメータは、ブロック・レベル輝度補正のために使用される。

【0269】

一例において、重み付き予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

【0270】

実施形態は、例えば、Bスライスとも呼ばれる双方向（B）予測スライスにおいて、双方向のインター予測を可能又は有効にするスライスに対してのみ、スライス・ヘッダにおいて信号関係情報を使用する効率的な符号化及び／又は復号を提供する。

【0271】

以下、上記の実施形態に示す符号化方式及び復号方式の適用とそれらを用いたシステムの説明である。

【0272】

図10は、コンテンツ配信サービスを実現するためのコンテンツ供給システム3100を示すブロック図である。このコンテンツ供給システム3100は、捕捉デバイス3102、端末デバイス3106を含み、任意選択で、ディスプレイ3126を含む。捕捉デバイス3102は、通信リンク3104を介して端末デバイス3106と通信する。通信リンクは、上述の通信チャネル13を含んでもよい。通信リンク3104は、WIFI、イーサネット、ケーブル、無線（3G／4G／5G）、USB、又はそれらの任意の種類の組み合わせなどを含むが、これらに限定されない。

【0273】

捕捉デバイス3102は、データを生成し、上記の実施形態に示す符号化方法によってデータを符号化してもよい。代替的には、捕捉デバイス3102は、ストリーミングサーバ（図示せず）にデータを配信することができ、サーバは、データを符号化し、符号化されたデータを端末デバイス3106に送信する。捕捉デバイス3102は、カメラ、スマートフォン又はパッド、コンピュータ若しくはラップトップ、ビデオ会議システム、PDA、車載デバイス、又はそれらのいずれかの組み合わせなどを含むが、これらに限定されない。例えば、捕捉デバイス3102は、上述のようにソース・デバイス12を含んでもよい。データがビデオを含む場合に、捕捉デバイス3102に含まれるビデオ符号化器20は、実際にビデオ符号化処理を実行することができる。データがオーディオ（すなわち

10

20

30

40

50

、音声)を含む場合、捕捉デバイス3102に含まれるオーディオ符号化器は、実際にオーディオ符号化処理を実行することができる。いくつかの実際的なシナリオでは、捕捉デバイス3102は、符号化されたビデオ及びオーディオ・データを、それらを一緒に多重化することによって配布する。他の実際的なシナリオ、例えば、ビデオ会議システムにおいては、符号化されたオーディオ・データ及び符号化されたビデオ・データは多重化されない。捕捉デバイス3102は、符号化されたオーディオ・データ及び符号化されたビデオ・データを、端末デバイス3106に別々に配信する。

【0274】

コンテンツ供給システム3100では、端末デバイス310は、符号化されたデータを受信及び再生する。端末デバイス3106は、スマートフォン又はPad3108、コンピュータ若しくはラップトップ3110、ネットワーク・ビデオ・レコーダ(NVR)／デジタル・ビデオ・レコーダ(DVR)3112、TV3114、セット・トップ・ボックス3116、ビデオ会議システム3118、ビデオ監視システム3120、パーソナル・デジタル・アシスタント(PDA)3122、車載デバイス3124、又はこれらのいずれかの組み合わせなどの、データ受信及び復元能力を有し、前述の符号化されたデータを復号することが可能なデバイスとすることができる。例えば、捕捉デバイス3106は、上述のようにソース・デバイス14を含んでもよい。符号化されたデータがビデオを含む場合に、端末デバイスに含まれるビデオ復号器30は、ビデオ復号を実行するように優先される。符号化されたデータがオーディオを含む場合に、端末デバイスに含まれるオーディオ復号器は、オーディオ復号処理を実行するように優先される。

【0275】

ディスプレイを有する端末デバイス、例えば、スマートフォン又はパッド3108、コンピュータ又はラップトップ3110、ネットワーク・ビデオ・レコーダ(NVR)／デジタル・ビデオ・レコーダ(DVR)3112、TV3114、パーソナル・デジタル・アシスタント(PDA)3122、又は車両搭載デバイス3124では、端末デバイスは、復号されたデータをそのディスプレイに供給することができる。ディスプレイを搭載しないデバイス、例えば、STB3116、ビデオ会議システム3118、又はビデオ監視システム3120では、外部ディスプレイ3126がそこに接触されて、復号されたデータを受信し、示す。

【0276】

このシステムの各デバイスが符号化又は復号を実行する場合に、前述の実施形態に示すように、ピクチャ符号化デバイス又はピクチャ復号デバイスを使用することができる。

【0277】

図11は、端末デバイス3106の一例の構成を示す図である。端末デバイス3106が、捕捉デバイス3102からストリームを受信した後、プロトコル進行ユニット3202は、ストリームの伝送プロトコルを分析する。このプロトコルは、リアルタイム・ストリーミング・プロトコル(RTSP)、ハイパー・テキスト・トランスファー・プロトコル(HTTP)、HTTPライブ・ストリーミング・プロトコル(HLS)、MPEG-DASH、リアルタイム・トランスポート・プロトコル(RTP)、リアルタイム・メッセージング・プロトコル(RTMP)、又はそれらの任意の種類の組み合わせなどを含むが、これらに限定されない。

【0278】

プロトコル進行ユニット3202がストリームを処理した後に、ストリームファイルは、生成される。ファイルは、逆多重化ユニット3204に出力される。逆多重化ユニット3204は、多重化されたデータを符号化されたオーディオ・データと符号化されたビデオ・データとに分離することができる。上述したように、いくつかの実際的なシナリオ、例えば、ビデオ会議システムにおいては、符号化されたオーディオ・データ及び符号化されたビデオ・データは多重化されない。この状況では、符号化データは、逆多重化部3204を介することなく、ビデオ復号器3206及び音声復号器3208に送信される。

【0279】

逆多重化処理を介して、ビデオ・エレメントリー・ストリーム（ES）、オーディオES、及び任意選択で字幕が生成される。前述の実施形態で説明されたビデオ復号器30を含むビデオ復号器3206は、上述の実施形態に示す復号方法によってビデオESを復号して、ビデオ・フレームを生成し、このデータを同期ユニット3212に送信する。オーディオ復号器3208は、オーディオESを復号して、オーディオ・フレームを生成し、このデータを同期ユニット3212に送信する。代替的には、ビデオ・フレームは、同期ユニット3212に供給する前に、バッファ（図11には示さず）に記憶させてもよい。代替的には、オーディオ・フレームは、同期ユニット3212に供給する前に、バッファ（図11には示さず）に記憶させてもよい。

【0280】

同期ユニット3212は、ビデオ・フレームとオーディオ・フレームを同期させ、ビデオ／オーディオ・ディスプレイ3214にビデオ／オーディオを供給する。例えば、同期ユニット3212は、ビデオ情報及びオーディオ情報の提示を同期させる。情報は、符号化されたオーディオ及びビジュアル・データの提示に関するタイム・スタンプ及びデータストリーム自体の送達に関するタイム・スタンプを使用して、構文でコーディングすることができる。字幕がストリームに含まれる場合、字幕復号器3210は、字幕を復号し、それをビデオ・フレーム及びオーディオ・フレームと同期させ、ビデオ／オーディオ／字幕をビデオ／オーディオ／字幕ディスプレイ3216に供給する。

【0281】

本発明は、前述のシステムに限定されるものではなく、前述の実施形態におけるピクチャ符号化デバイス又はピクチャ復号デバイスのいずれかを、他のシステム、例えば、自動車システムに組み込むことができる。

【0282】

数学演算子

この出願で使用される数学演算子は、Cプログラミング言語で使用されるものに類似する。しかし、整数除算演算と算術シフト演算の結果は、より正確に定義され、べき乗や実数値除算などの追加演算が定義される。番号付けとカウントの規則は一般に0から開始する。例えば、「第1」は、0番に等価であり、「第2」は、1番に等価であるなどである。

【0283】

算術演算子

次の算術演算子は、次のように定義される。

+ 加算

− 減算（2引数演算子として）又は否定（単項接頭演算子として）

* 行列乗算を含む乗算

x^y べき乗。 x の指数 y のべき乗を指定する。他の文脈では、そのような表記は、べき乗として解釈することを意図しない上付き文字として使用される。

/ 結果がゼロに向かって切り捨てられる整数除算。たとえば、 $7/4$ と $-7/-4$ は1に切り捨てられ、 $-7/4$ と $7/-4$ は−1に切り捨てられる。

÷ 切り捨てや丸めが意図されていない数式の除算を示すために使用される。

【0284】

【数63】

10

20

30

40

50

$$\frac{x}{y}$$

10

切り捨てや丸めが意図されていない数式の除算を示すために使用される。

【0285】

【数64】

20

$$\sum_{i=x}^y f(i)$$

30

x から y までのすべての整数値をとる i の $f(i)$ の和である。

$x \% y$ 剰余。 x の残りを y で割った余りであり、 $x \geq 0$ 及び $y > 0$ の整数 x 及び y に対してのみ定義される。

【0286】

論理演算子

次の論理演算子は、次のように定義される。

$x \&\& y$ x と y のブール論理「and」

$x \mid\mid y$ x と y のブール論理「or」

! ブール論理「否定」

$x ? y : z$ x が TRUE 又は 0 でない場合、 y の値、そうでない場合は z の値で評価される。

40

【0287】

関係演算子

次の関係演算子は、次のように定義される。

> より大きい

>= 以上

< より小さい

<= 以下

== 等しい

!= 等しくない

50

値「na」（適用不可）が割り当てられた構文要素又は変数に関係演算子が適用される場合に、値「na」はその構文要素又は変数の重複しない値として扱われる。値「na」は他の値と等しくないとみなされる。

【0288】

ビット単位演算子

次のビット単位演算子は、次のように定義される。

& ビット単位の「and」である。整数引数について操作する場合に、整数値の2の補数表現について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合に、より短い引数は、0に等しいより大きなビットを加えることによって拡張される。

| ビット単位の「or」である。整数引数について操作する場合に、整数値の2の補数表現について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合に、より短い引数は、0に等しいより大きなビットを加えることによって拡張される。

^ ビット単位の「exclusive or」である。整数引数について操作する場合に、整数値の2の補数表現について操作する。別の引数よりも少ないビットを含むバイナリ引数について操作する場合に、より短い引数は、0に等しいより大きなビットを加えることによって拡張される。

$x \gg y$ x を y バイナリ・ディジットの2の補数整数表現の算術右シフトである。この関数は、 y の負でない整数値に対してのみ定義される。右シフトの結果として最上位ビット(MSB)にシフトされたビットは、シフト演算の前に x のMSBに等しい値を持つ。
 $x \ll y$ x を y バイナリ・ディジットの2の補数整数表現の算術左シフトである。この関数は、 y の負でない整数値に対してのみ定義される。左シフトの結果として最下位ビット(LSB)にシフトされたビットは、0に等しい値を持つ。

【0289】

割り当て演算子

次の代入演算子は、次のように定義される。

= 代入演算子

++ インクリメント、すなわち $x++$ は $x = x + 1$ と等価であり、配列インデックスで 사용되는場合に、インクリメント演算の前に変数の値で評価される。

-- デクリメント、すなわち $x--$ は $x = x - 1$ と等価であり、配列インデックスで 사용되는場合に、デクリメント演算の前に変数の値で評価される。

+= 指定された量だけインクリメントする、すなわち $x += 3$ は $x = x + 3$ と等価であり、 $x += (-3)$ は $x = x + (-3)$ と等価である。

-= 指定された量だけデクリメントする、すなわち $x -= 3$ は $x = x - 3$ と等価であり、 $x -= (-3)$ は $x = x - (-3)$ と等価である。

【0290】

範囲表記

値の範囲を指定するには、次の表記が使用される。

$x = y..z$ x は、 y から z までの整数値を取り、 x 、 y 及び z は、整数であり、 z は、 y より大きい。

【0291】

数学関数

次の数学関数が定義される。

【0292】

【数65】

10

20

30

40

50

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

A s i n (x) 正弦の逆三角関数であって、 $-1.0 \sim 1.0$ の範囲の引数 x について操作し、

10

$-\pi \div 2 \sim \pi \div 2$ (ラジアン単位) の範囲の出力値である。

A t a n (x) 正接の逆三角関数であって、引数 x について操作し、 $-\pi \div 2 \sim \pi \div 2$ (ラジアン単位) の範囲の出力値である。

【0293】

【数66】

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{それ以外} \end{cases}$$

20

C e i l (x) x 以上の最小整数である。

【0294】

【数67】

$$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x)$$

30

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$$

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{それ以外} \end{cases}$$

C o s (x) ラジアン単位の引数 x について操作する余弦の三角関数である。

40

F l o o r (x) x 以下の最大整数である。

【0295】

【数68】

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d/2 \\ c - d & ; \quad a - b > d/2 \\ c & ; \quad \text{それ以外} \end{cases}$$

L n (x) x の自然対数 (底 e 対数、 e は自然基本定数 $2.718281828\dots$)

50

.) 。

L o g 2 (x) x の底 2 の対数

L o g 1 0 (x) x の底 1 0 の対数

【 0 2 9 6 】

【数 6 9】

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

10

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases}$$

20

S i n (x) ラジアン単位の引数 x について操作する正弦の三角関数である。

【 0 2 9 7 】

【数 7 0】

$$\text{Sqrt}(x) = \sqrt{x}$$

30

$$\text{Swap}(x, y) = (y, x)$$

T a n (x) ラジアン単位の引数 x について操作する正接の三角関数である。

【 0 2 9 8 】

演算優先度順位

式の優先順位が丸かっこを使用して明示的に示されない場合、次の規則が適用される。

- － より高い優先度の演算は、より低い優先度の演算の前に評価される。
- － 同じ優先度の演算は、左から右へ順次評価される。

以下の表は、最高から最低までの演算の優先度を指定しており、表中のより高い位置は、より高い優先度を示す。

40

Cプログラミング言語でも使用されるこれらの演算子については、本明細書で使用される優先順位は、Cプログラミング言語で使用される優先順位と同じである。

【 0 2 9 9 】

50

【表 5 5】

演算 (オペランド x, y , 及び z)	
" $x++$ ", " $x--$ "	
" $!x$ ", " $-x$ " (単項前置演算子として)	
x^y	
" $x * y$ ", " x / y ", " $x \div y$ ", " $\frac{x}{y}$ ", " $x \% y$ "	10
" $x + y$ ", " $x - y$ " (2つの引数の演算子として), " $\sum_{i=x}^y f(i)$ "	
" $x << y$ ", " $x >> y$ "	
" $x < y$ ", " $x \leq y$ ", " $x > y$ ", " $x \geq y$ "	
" $x == y$ ", " $x != y$ "	20
" $x \& y$ "	
" $x y$ "	
" $x \&\& y$ "	
" $x y$ "	
" $x ? y : z$ "	
" $x..y$ "	30
" $x = y$ ", " $x += y$ ", " $x -= y$ "	

表：（表の上部）最高から（表の下部）最低までの演算優先度

【0 3 0 0】

論理演算のテキスト記載

本テキストにおいて、次の形式で数学的に記載される論理演算の記述

【0 3 0 1】

【数 7 1】

10

20

30

40

50

if(condition 0)

statement 0

else if(condition 1)

statement 1

...

else /* informative remark on remaining condition */

statement n

10

は、以下の方式で記載されてもよい。

【 0 3 0 2 】

【 数 7 2 】

... as follows / ... the following applies:

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- ...
- Otherwise (informative remark on remaining condition), statement n

20

本テキストにおいて、各「If . . . Otherwise, if . . . Otherwise, . . .」文は、直後に「If . . . 」が続く、「. . . as follows」又は「. . . the following applies」で導入される。「If . . . Otherwise, if . . . Otherwise, . . .」の最後の条件は、常に「Otherwise, . . .」である。「If . . . Otherwise, if . . . Otherwise, . . .」文は、「. . . as follows」又は「. . . the following applies」を終わりの「Otherwise, . . .」と一致させることによって識別することができる。

30

本テキストにおいて、次の形式で数学的に記載される論理演算の記述

【 0 3 0 3 】

【 数 7 3 】

40

50

```

if( condition 0a  &&  condition 0b )
    statement 0
else if( condition 1a  ||  condition 1b )
    statement 1
...
else
    statement n

```

10

は、以下の方式で記載されてもよい。

【 0 3 0 4 】

【数 7 4 】

... as follows / ... the following applies:

- If all of the following conditions are true, statement 0:
 - condition 0a
 - condition 0b
- Otherwise, if one or more of the following conditions are true, statement 1:
 - condition 1a
 - condition 1b
- ...
- Otherwise, statement n

20

30

本テキストにおいて、次の形式で数学的に記載される論理演算の記述

【 0 3 0 5 】

【数 7 5 】

```

if( condition 0 )
    statement 0
if( condition 1 )
    statement 1

```

40

は、以下の方式で記載されてもよい。

【 0 3 0 6 】

【数 7 6 】

50

When condition 0, statement 0

When condition 1, statement 1.

例えば、符号化器 20 及び復号器 30 の実施形態、並びに、例えば、符号化器 20 及び復号器 30 を参照して、本明細書で説明される機能は、ハードウェア、ソフトウェア、ファームウェア、又はそれらの任意の組み合わせで実装されてもよい。ソフトウェアで実装される場合、機能は、コンピュータ可読媒体に記憶されるか、又は 1 つ以上の命令又はコードとして通信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る。コンピュータ可読媒体は、データ記憶媒体のような有形媒体に対応するコンピュータ可読記憶媒体、又は、例えば通信プロトコルにしたがって、ある場所から他の場所へのコンピュータ・プログラムの転送を容易にする任意の媒体を含む通信媒体を含んでもよい。このようにして、コンピュータ可読媒体は、一般に、(1) 非一時的である有形のコンピュータ可読記憶媒体、又は (2) 信号又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、本開示で説明される技術の実装のための命令、コード及び／又はデータ構造を検索するために、1 つ以上のコンピュータ又は 1 つ以上のプロセッサによってアクセス可能な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、コンピュータ可読記憶媒体を含んでもよい。

【0307】

一例として、限定するものではないが、このようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROM 又は他の光ディスク記憶デバイス、磁気ディスクストレージ、又は他の磁気記憶デバイス、フラッシュメモリ、又は命令又はデータ構造の形式で所望のプログラム・コードを記憶するために使用することができ、コンピュータによってアクセスすることができる他の任意の媒体を含むことができる。また、任意の接続は、適切にコンピュータ可読媒体とも称される。例えば、同軸ケーブル、光ファイバケーブル、ツイスト・ペア、デジタル加入者線 (DSL)、又は赤外線、無線、及びマイクロ波などの無線技術を用いて、ウェブサイト、サーバ、又は他のリモート・ソースから命令が送信される場合、同軸ケーブル、光ファイバケーブル、ツイスト・ペア、DSL、又は赤外線、無線、及びマイクロ波のような無線技術は、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及びデータ記憶媒体は、接続、搬送波、信号、又は他の一時的な媒体を含まず、代わりに非一時的で有形の記憶媒体に向けられると理解されたい。ディスクは、本明細書で使用する場合、コンパクトディスク (CD)、レーザディスク、光ディスク、デジタル多用途ディスク (DVD)、フロッピー・ディスク及びブルーレイ・ディスクを含み、ディスクは、通常、磁気的にデータを再生し、ディスクは光学的にレーザでデータを再生する。上記の組み合わせはまた、コンピュータ可読媒体の範囲に含まれるべきである。

【0308】

命令は、1 つ以上のデジタル信号プロセッサ (DSP)、汎用マイクロプロセッサ、特定用途向け集積回路 (ASIC)、フィールド・プログラマブル論理アレイ (FPGA)、又は他の同等の集積又は個別論理回路などの 1 つ以上のプロセッサによって実行されてもよい。したがって、本明細書で使用する用語「プロセッサ」は、前述の構造のいずれか、又は本明細書で説明される技術の実装に好適な他の構造を指してもよい。追加的に、いくつかの態様において、本明細書で説明される機能は、符号化及び復号のために構成される専用ハードウェア及び／又はソフトウェア・モジュール内に提供されてもよく、又は組み合わされたコーデックに組み込まれてもよい。また、この技術は、1 つ以上の回路又は論理素子で完全に実装することができる。

【0309】

本開示の技術は、無線ハンドセット、集積回路（ＩＣ）、又はＩＣのセット（例えば、チップセット）を含む、広範な種類のデバイス又は装置で実装されてもよい。本開示では、開示された技術を実行するように構成されているデバイスの機能的態様を強調するために、種々のコンポーネント、モジュール、又はユニットが説明されるが、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しない。むしろ、上述のように、種々のユニットは、コーデック・ハードウェア・ユニット内で組み合わせられてもよく、又は、適切なソフトウェア及び／又はファームウェアと共に、上述の１つ以上のプロセッサを含む、相互動作可能なハードウェアユニットの集合によって提供されてもよい。

【０３１０】

以下、本発明のさらなる実施形態が提供される。以下のセクションにおいて使用される番号付けは、先のセクションにおいて使用された番号付けに必ずしも準拠しないことに留意されたい。

【０３１１】

実施形態１：ピクチャのブロックのインター予測の方法であって、重み付け予測パラメータ及び非矩形インター予測の有効化のシグナリングが、予測ブロックのグループに対して実行され、方法は、ブロックに対するインター予測モード・パラメータを取得することであって、取得することは、予測ブロックを含むブロックのグループに対して、非矩形インター予測モードが有効であるかどうかのチェックを含む、取得することと、ブロックに関連する重み付け予測パラメータ及びブロックに対するインター予測モード・パラメータを取得することであって、参照ピクチャがブロックに対して示されており、重み付け予測パラメータがブロックのグループに対して指定されている、取得することと、を含む。

実施形態２：非矩形インター予測を可能にすることは、１より大きい三角形マージ候補の最大数（`MaxNumTriangleMergeCand`）を示すことによって実行される、実施形態１に記載の方法。

実施形態３：重み付け予測パラメータが少なくとも１つの参照インデックスに対して有効な重み付け予測を指定するときに、非矩形インター予測は無効であると推論される、実施形態１又は２に記載の方法。

実施形態４：ブロックのグループがピクチャであり、重み付け予測パラメータ及び非矩形インター予測モード・パラメータを有効にすることがピクチャ・ヘッダにおいて示される、任意の実施形態１～３に記載の方法。

実施形態５：ブロックのグループがスライスであり、重み付け予測パラメータ及びインター予測の非矩形モード・パラメータを有効にすることがスライス・ヘッダにおいて示される、任意の実施形態１～４に記載の方法。

実施形態６：インター予測モード・パラメータは、参照ピクチャを決定するために使用される参照インデックスと、参照ピクチャ内の参照ブロックの位置を決定するために使用されるモーション・ベクトル情報とを含む、任意の実施形態１～５の方法。

実施形態７：非矩形マージ・モードが三角形パーティショニング・モードである、任意の実施形態１～６の方法。

実施形態８：非矩形マージ・モードがＧＥＯモードである、任意の実施形態１～７の方法。

実施形態９：重み付き予測がスライス・レベル輝度補償メカニズム（グローバル重み付き予測など）である、任意の実施形態１～８の方法。

実施形態１０：重み付き予測がブロック・レベル輝度補償メカニズム（ローカル輝度予測など）（`LIC`）である、任意の実施形態１～９の方法。

実施形態１１：重み付き予測パラメータが、重み付き予測が予測ブロックの輝度及び色差コンポーネントに適用されるかどうかを示すフラグのセットと、予測ブロックの値の線形トランスフォームを指定する線形モデル・パラメータ α 及び β と、を含む、実施形態１～１０の方法。

【０３１２】

本出願の一態様では、図１２に示すように、インター予測方法１２００が開示されており、方法は、ブロックのグループに対して非矩形のインター予測モードが許可されている

かどうかを決定すること（S1201）と、ブロックのグループに対する1つ以上のインター予測モード・パラメータ及び重み付き予測パラメータを取得すること（S1202）と、1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値を取得することであって、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む、取得すること（S1203）と、を含む。

【0313】

実現可能な実装では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付き予測が有効である場合に無効である。

10

【0314】

実現可能な実装では、非矩形のインター予測モードは、重み付き予測が無効である場合に有効である。

【0315】

実現可能な実装では、非矩形インター予測モードが許可されていると決定することは、三角マージ候補の最大数（MaxNumTriangleMergeCand）が、1より大きいことを示すことを示す。

【0316】

実現可能な実装では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

20

【0317】

実現可能な実装では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、スライスのスライス・ヘッダにおけるものである。

【0318】

実現可能な実装では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0319】

実現可能な実装では、非矩形インター予測モードは、幾何学的（GEO）パーティショニング・モードである。

30

【0320】

実現可能な実装では、重み付き予測パラメータは、スライス・レベル輝度補正のために使用される。

【0321】

実現可能な実装では、重み付き予測パラメータは、ブロック・レベル輝度補正のために使用される。

【0322】

実現可能な実装では、重み付き予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

40

【0323】

本出願の第2の態様では、図13に示すようなインター予測のための装置1300は、プロセッサ実行可能な命令を記憶した非一時的なメモリ1301と、メモリ1301に結合されたプロセッサ1302とを含み、プロセッサ1302は、本出願の第1の態様における実現可能な実装のいずれか1つを容易にするために、プロセッサ実行可能な命令を実行するように構成されている。

【0324】

本出願の第3の態様では、インター予測のためのビットストリームが開示されており、ビットストリームは、ブロックのグループに対して非矩形インター予測モードが許可され

50

ているかどうかを決定するための指示情報と、ブロックのグループに対する1つ以上のインター予測モード及び重み付け予測パラメータと、を含み、1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値が取得され、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む。

【0325】

実現可能な実装では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付け予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付け予測が有効である場合に無効である。

【0326】

実現可能な実装では、非矩形のインター予測モードは、重み付け予測が無効である場合に有効である。

【0327】

実現可能な実装では、指示情報は、三角マージ候補の最大数 (MaxNumTriangleMergeCand) は、1より大きいことを含む。

【0328】

実現可能な実装では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

【0329】

実現可能な実装では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び指示情報は、ピクチャのスライス・ヘッダにおけるものである。

【0330】

実現可能な実装では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0331】

実現可能な実装では、非矩形インター予測モードは、幾何学的 (GEO) パーティショニング・モードである。

【0332】

実現可能な実装では、重み付け予測パラメータは、スライス・レベル輝度補正のために使用される。

【0333】

実現可能な実装では、重み付け予測パラメータは、ブロック・レベル輝度補正のために使用される。

【0334】

実現可能な実装では、重み付け予測パラメータは、重み付け予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

【0335】

本出願の第4の態様では、図14に示すように、インター予測装置1400が開示されており、装置は、ブロックのグループに対して非矩形のインター予測モードが許可されているかどうかを決定するように構成されている決定モジュール1401と、ブロックのグループに対する1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータを取得するように構成されている取得モジュール1402と、1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値を取得するように構成されている予測モジュールであって、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む、予測モジュール1403と、を含む。

【0336】

実現可能な実装では、参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付け予測が有効であるかどうかを含み、非矩形のインター予測モードは、重み付け予測が

10

20

30

40

50

有効である場合に無効である。

【0337】

実現可能な実装では、非矩形のインター予測モードは、重み付き予測が無効である場合に有効である。

【0338】

実現可能な実装では、決定モジュール1401は、具体的には、三角マージ候補の最大数（MaxNumTriangleMergeCand）は、1より大きいことを示すように構成されている。

【0339】

実現可能な実装では、ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、ピクチャのピクチャ・ヘッダにおけるものである。

【0340】

実現可能な実装では、ブロックのグループは、スライスからなり、重み付け予測パラメータ及び非矩形の予測モードが許可されていると決定するための指示情報は、スライスのスライス・ヘッダにおけるものである。

【0341】

実現可能な実装では、非矩形インター予測モードは、三角形パーティショニング・モードである。

【0342】

実現可能な実装では、非矩形インター予測モードは、幾何学的（GEO）パーティショニング・モードである。

【0343】

実現可能な実装では、重み付き予測パラメータは、スライス・レベル輝度補正のために使用される。

【0344】

実現可能な実装では、重み付き予測パラメータは、ブロック・レベル輝度補正のために使用される。

【0345】

実現可能な実装では、重み付き予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む。

【0346】

先行技術の方法は、以下の態様のリストに要約され得る。

【0347】

態様1．インター予測方法であって、
ブロックのグループに対して非矩形インター予測モードが許可されるかどうかを決定することと、
ブロックのグループに対する1つ以上のインター予測モード・パラメータ及び重み付き予測パラメータを取得することと、
1つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに基づいて、現在のブロックの予測値を取得することであって、インター予測モード・パラメータのうちの1つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現在のブロックを含む、取得することと、を含む、方法。

【0348】

態様2．参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が有効であるかどうかを含み、非矩形インター予測モードは、重み付き予測が有効である場合に無効である、態様1に記載の方法。

【0349】

態様3．非矩形インター予測モードは、重み付き予測が無効である場合に有効である、

10

20

30

40

50

態様 1 又は 2 に記載の方法。

【0350】

態様 4 . 非矩形インター予測モードが許可されていると決定することは、三角マージ候補の最大数 (MaxNumTriangleMergeCand) は、1 より大きいことを示すことを含む、態様 1 ~ 3 のいずれか 1 つに記載の方法。

【0351】

態様 5 . ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び非矩形予測モードが許可されていると決定するための指示情報は、ピクチャのピクチャ・ヘッダにおけるものである、態様 1 ~ 4 のいずれか 1 つに記載の方法。

【0352】

態様 6 . ブロックのグループは、スライスからなり、重み付け予測パラメータ及び非矩形予測モードが許可されていると決定するための指示情報は、スライスのスライス・ヘッダにおけるものである、態様 1 ~ 4 のいずれか 1 つに記載の方法。

【0353】

態様 7 . 非矩形インター予測モードは、三角形パーティショニング・モードである、態様 1 ~ 6 のいずれか 1 つに記載の方法。

【0354】

態様 8 . 非矩形インター予測モードは、幾何学的 (GEO) パーティショニング・モードである、態様 1 ~ 6 のいずれか 1 つに記載の方法。

【0355】

態様 8 a . マージ・モード候補の数に関する構文要素 (非矩形インター予測を決定するための情報を示す) は、シーケンス・パラメータ・セット (SPS) においてシグナリングされる、態様 1 ~ 8 のいずれか 1 つに記載の方法。

【0356】

態様 8 b . ピクチャが 1 つのスライスのみを含むときに、ピクチャ・ヘッダは、スライス・ヘッダにおいてシグナリングされる、態様 1 ~ 8 a のいずれか 1 つに記載の方法。

【0357】

態様 8 c . ピクチャが 1 つのスライスのみを含むときに、ピクチャ・ヘッダは、スライス・ヘッダにおいてシグナリングされる、態様 1 ~ 8 b のいずれか 1 つに記載の方法。

【0358】

態様 8 d . ピクチャ・パラメータ・セットは、フラグを含み、フラグの値は、重み付けパラメータがピクチャ・ヘッダ又はスライス・ヘッダにおいて存在するかどうかを定義する、態様 1 ~ 8 c のいずれか 1 つに記載の方法。

【0359】

態様 8 e . ピクチャ・ヘッダにおけるフラグは、非イントラ・タイプのスライスが存在するかどうか、及びこのスライスに対してインター予測モード・パラメータがシグナリングされているかどうかを示す、対応 1 ~ 8 d のいずれか 1 つに記載の方法。

【0360】

態様 9 . 重み付け予測パラメータは、スライス・レベル輝度補償のために使用される、態様 1 ~ 8 のいずれか 1 つに記載の方法。

【0361】

態様 10 . 重み付け予測パラメータは、ブロック・レベル輝度補償のために使用される、態様 1 ~ 8 のいずれか 1 つに記載の方法。

【0362】

態様 11 . 重み付け予測パラメータは、重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどうかを示すフラグと、予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと、を含む、態様 1 ~ 10 のいずれか一項に記載の方法。

【0363】

10

20

30

40

50

態様 12. インター予測のための装置であって、
プロセッサ実行可能な命令が記憶された非一的なメモリと、
メモリに結合されたプロセッサと、を含み、プロセッサは、態様 1 ~ 11 のいずれか 1 つ
を容易にするためのプロセッサ実行可能な命令を実行する、装置。

【0364】

態様 13. インター予測のためのビットストリームであって、
ブロックのグループに対して非矩形インター予測モードが許可されているかどうかを決定
するための指示情報と、
ブロックのグループに対する 1 つ以上のインター予測モード及び重み付け予測パラメータ
と、を含み、1 つ以上のインター予測モード・パラメータ及び重み付け予測パラメータに
基づいて、現在のブロックの予測値が取得され、インター予測モード・パラメータのうち
の 1 つは、現在のブロックに対する参照ピクチャ情報を示し、ブロックのグループは、現
在のブロックを含む、ビットストリーム。

10

【0365】

態様 14. 参照ピクチャ情報は、参照ピクチャ・インデックスに対して重み付き予測が
有効であるかどうかを含み、非矩形インター予測モードは、重み付き予測が有効である場
合に無効である、態様 13 に記載のビットストリーム。

【0366】

態様 15. 非矩形インター予測モードは、重み付き予測が無効である場合に有効である
、態様 13 又は 14 に記載のビットストリーム。

20

【0367】

態様 16. 指示情報は、三角形マージ候補の最大数 (MaxNumTriangleMergeCand) が 1 より大きい、態様 13 ~ 15 のいずれか 1 つのビットストリーム。

【0368】

態様 17. ブロックのグループは、ピクチャからなり、重み付け予測パラメータ及び指
示情報は、ピクチャのピクチャ・ヘッダにおけるものである、態様 13 ~ 16 のいずれか
1 つに記載のビットストリーム。

【0369】

態様 18. ブロックのグループは、スライスからなり、重み付け予測パラメータ及び指
示情報は、ピクチャのスライス・ヘッダにおけるものである、態様 13 ~ 17 のいずれか
1 つに記載のビットストリーム。

30

【0370】

態様 19. 非矩形インター予測モードは、三角形パーティショニング・モードである、
態様 13 ~ 18 のいずれか 1 つに記載のビットストリーム。

【0371】

態様 20. 非矩形インター予測モードは、幾何学的 (GEO) パーティショニング・モ
ードである、態様 13 ~ 19 のいずれか 1 つに記載のビットストリーム。

【0372】

態様 21. 重み付け予測パラメータは、スライス・レベル輝度補償のために使用される
、態様 13 ~ 20 のいずれか 1 つに記載のビットストリーム。

40

【0373】

態様 22. 重み付け予測パラメータは、ブロック・レベル輝度補償のために使用される
、態様 13 ~ 20 のいずれか 1 つに記載のビットストリーム。

【0374】

態様 23. 重み付け予測パラメータは、
重み付き予測が、予測ブロックの輝度及び／又は色差コンポーネントに適用されるかどう
かを示すフラグと、
予測ブロックの値の線形トランスフォーメーションを指定する線形モデル・パラメータと
、を含む、態様 13 ~ 22 のいずれか 1 つに記載のビットストリーム。

50

【図面】

【図 1 A】

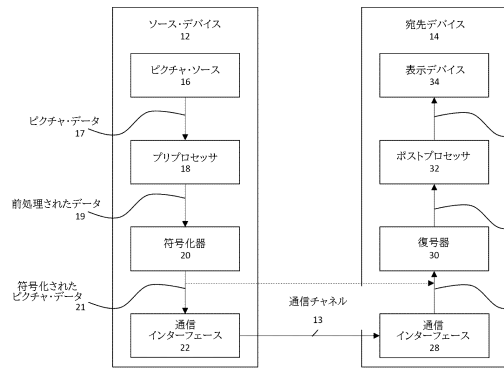


FIG. 1A

【図 1 B】

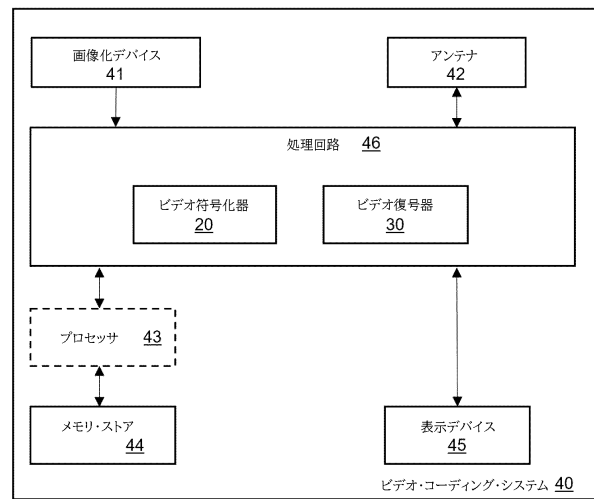


FIG. 1B

【図 2】

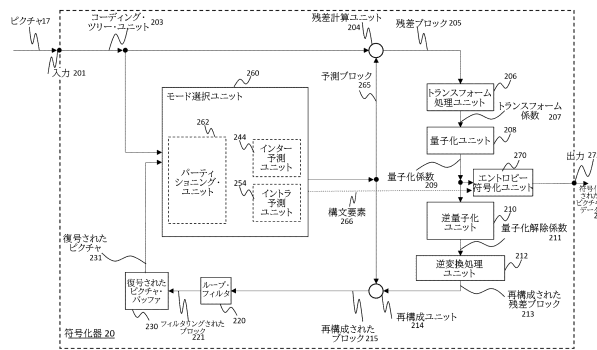


FIG. 2

【図 3】

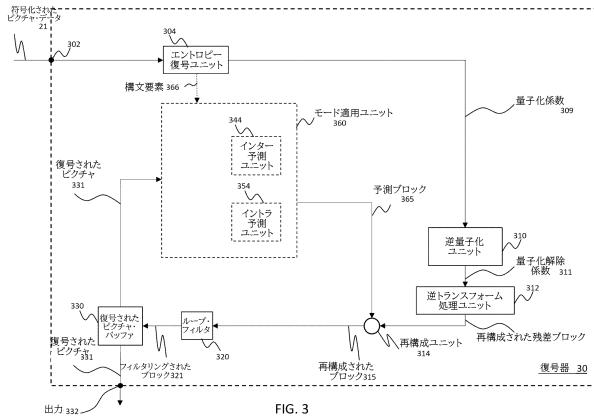


FIG. 3

10

20

30

40

50

【図 4】

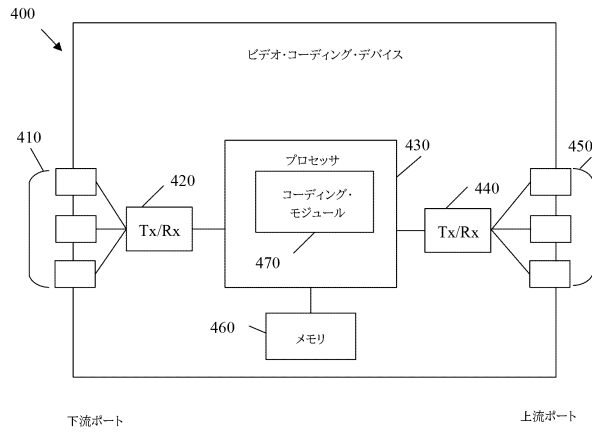
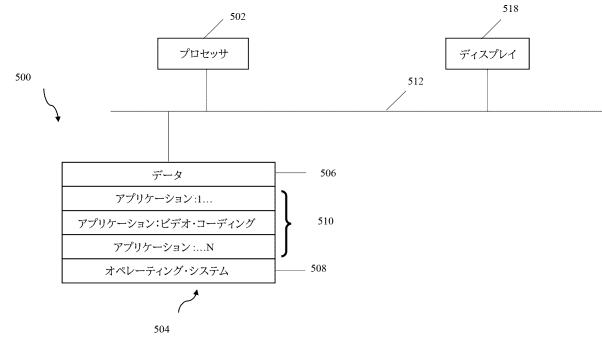


FIG. 4

【図 5】



10

【図 6】

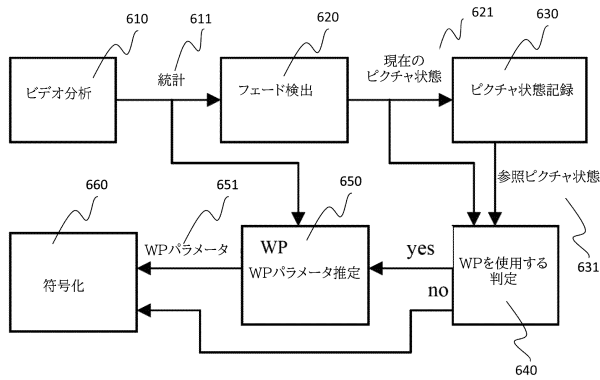


FIG. 6

【図 7】

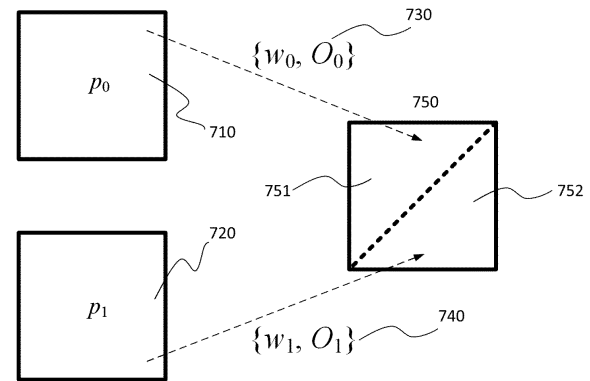


FIG. 7

20

30

40

50

【図 8】

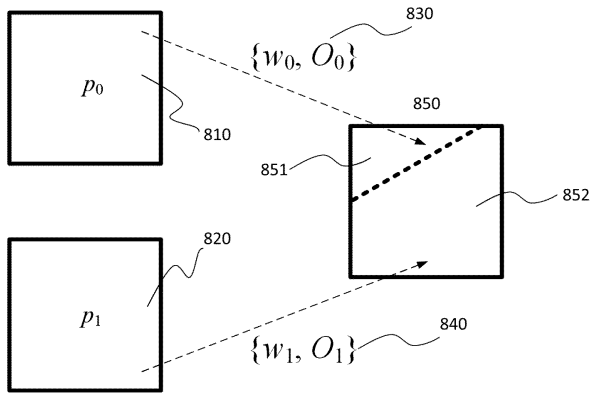


FIG. 8

【図 9】

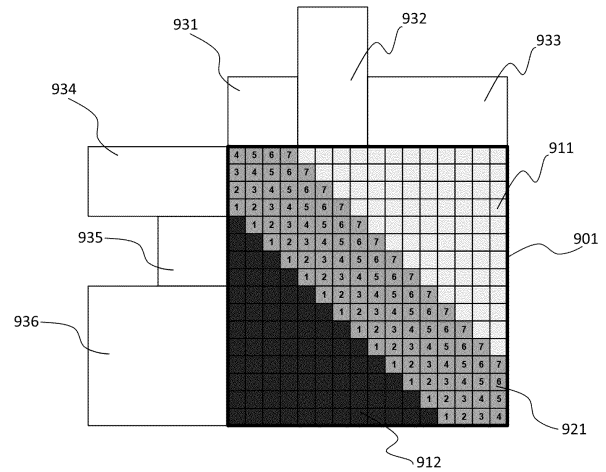


FIG. 9

【図 10】

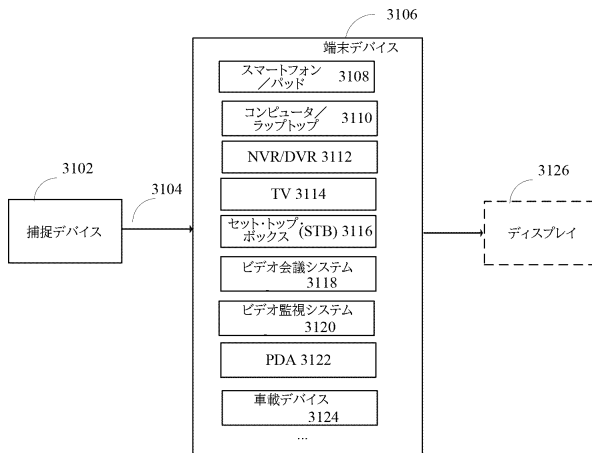


FIG. 10

【図 11】

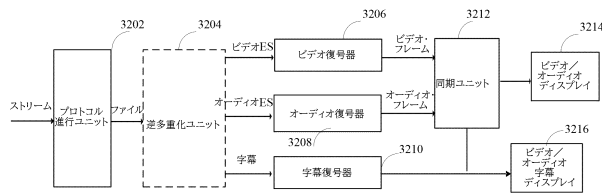


FIG. 11

10

20

30

40

50

【図 1 2】

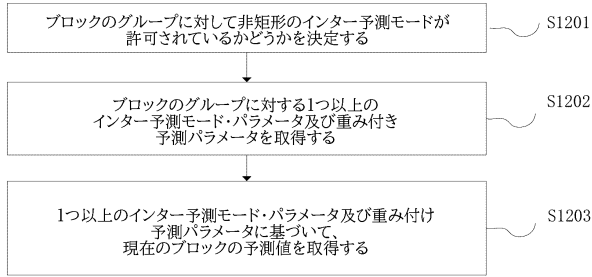


FIG. 12

【図 1 3】

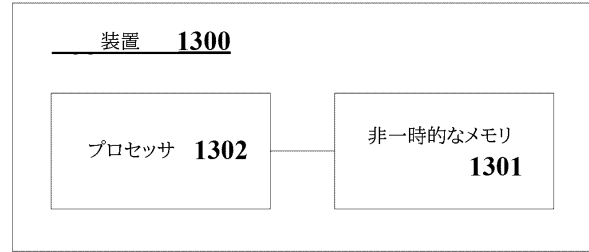


FIG. 13

【図 1 4】



FIG. 14

【図 1 5】

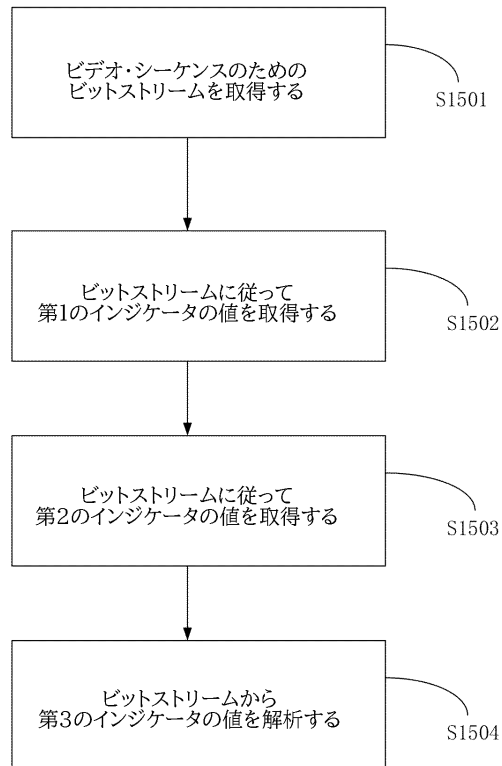


FIG. 15

10

20

30

40

50

【図 16】

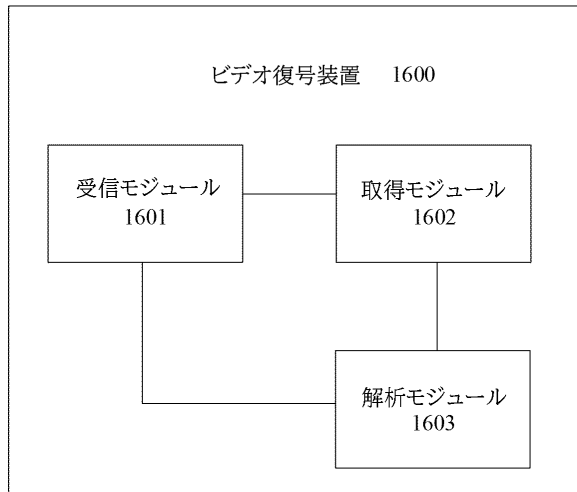


FIG. 16

10

20

30

40

50

フロントページの続き

- (74)代理人 100070150
弁理士 伊東 忠彦
- (74)代理人 100135079
弁理士 宮崎 修
- (72)発明者 フィリップフ, アレクセイ コンスタンティノヴィチ
中国 5 1 8 1 2 9 グアンドン シェンチェン ロンガン・ディストリクト バンティエン ホアウ
エイ・アドミニストレーション・ビルディング
- (72)発明者 ルフィツキー, ヴァシリイ アレクセヴィチ
中国 5 1 8 1 2 9 グアンドン シェンチェン ロンガン・ディストリクト バンティエン ホアウ
エイ・アドミニストレーション・ビルディング
- (72)発明者 アルシナ, エレナ アレキサンドロヴナ
ドイツ 8 0 9 9 2 ミュンヘン リースシュトラッセ 25 ホアウェイ・テクノロジーズ・デュッ
セルドルフ・ゲーエムベーハー
- 審査官 間宮 嘉誉
- (56)参考文献 国際公開第2021/136747 (WO, A1)
国際公開第2021/133552 (WO, A1)
国際公開第2020/219956 (WO, A1)
国際公開第2021/141477 (WO, A1)
国際公開第2021/045658 (WO, A2)
米国特許出願公開第2021/0021863 (US, A1)
特表2022-531443 (JP, A)
特表2023-508020 (JP, A)
特表2023-510858 (JP, A)
特表2023-510846 (JP, A)
ZHOU, Minhua, CE4-related: CE4-1 Spec Text with Suggested Fixes, JVET-Q0160 (version 1), ITU, 2019年12月28日, pp.39-43, 46-49, 97-108, 114-122, [online], [retrieved on 2023-09-29], Retrieved from the Internet: <URL: https://jvet-experts.org/doc_end_user/documents/17_Brussels/wg11/JVET-Q0160-v1.zip>, CE4_1_WD(on_top_of_CE4_CommonBaseWD)_w_fixes.docx
LI, Ling et al., SPS Level Control over the Number of Merge Candidates, JVET-Q0798 (version 1), ITU, 2020年01月15日, pp.1-4, [online], [retrieved on 2023-09-29], Retrieved from the Internet: <URL: https://jvet-experts.org/doc_end_user/documents/17_Brussels/wg11/JVET-Q0798-v1.zip>, JVET-Q0798-v1.docx
- (58)調査した分野 (Int.Cl., DB名)
H04N 7/12
H04N 19/00-19/98