

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7384910号

(P7384910)

(45)発行日 令和5年11月21日(2023.11.21)

(24)登録日 令和5年11月13日(2023.11.13)

(51)国際特許分類

F I

H 0 4 N 19/433 (2014.01)

H 0 4 N 19/433

H 0 4 N 19/593 (2014.01)

H 0 4 N 19/593

H 0 4 N 19/70 (2014.01)

H 0 4 N 19/70

請求項の数 17 (全93頁)

(21)出願番号 特願2021-543340(P2021-543340)
 (86)(22)出願日 令和2年2月2日(2020.2.2)
 (65)公表番号 特表2022-517691(P2022-517691 A)
 (43)公表日 令和4年3月9日(2022.3.9)
 (86)国際出願番号 PCT/CN2020/074155
 (87)国際公開番号 WO2020/156540
 (87)国際公開日 令和2年8月6日(2020.8.6)
 審査請求日 令和3年7月26日(2021.7.26)
 (31)優先権主張番号 PCT/CN2019/074598
 (32)優先日 平成31年2月2日(2019.2.2)
 (33)優先権主張国・地域又は機関 中国(CN)
 (31)優先権主張番号 PCT/CN2019/095297
 (32)優先日 令和1年7月9日(2019.7.9)
 最終頁に続く

(73)特許権者 520476341
 北京字節跳動網絡技術有限公司
 Beijing Bytedance Network Technology Co., Ltd.
 中華人民共和国100041北京市石景山区実興大街30号院3号楼2層B-0035房間
 Room B-0035, 2/F, No. 30 Building, No. 30, Shixing Road, Shijingshan District Beijing 100041 China
 (73)特許権者 520477474
 バイトダンス インコーポレイテッド
 最終頁に続く

(54)【発明の名称】 映像符号化におけるイントラブロックコピーのためのバッファ管理

(57)【特許請求の範囲】

【請求項1】

輝度ブロックである映像の現在の映像ブロックと前記映像のビットストリームとの変換について、前記現在の映像ブロックに第1のコーディングモードが適用されていることを判定することと、

前記現在の映像ブロックに対する第1のブロックベクトルを導出することと、

前記第1のブロックベクトルと輝度バッファとに基づいて、前記現在の映像ブロックの予測サンプルを生成することと、

前記予測サンプルに基づいて前記変換を行うことと、を含み、

前記輝度バッファにはフィルタリング処理が施されていない前の映像ブロックの再構成サンプルが記憶され、前記第1のコーディングモードでは前記予測サンプルが前記映像ブロックを含む同じピクチャから導出され、

前記輝度バッファの寸法および形状が、前記現在の映像ブロックと、前記現在の映像ブロックと同じ映像領域に位置する他の輝度ブロックとに対して同じであり、
 前記輝度バッファの前記寸法は、前記ビットストリームに含まれるフィールドに基づいて示される、

視覚メディア処理方法。

【請求項2】

前記輝度バッファは、矩形領域である、

請求項1に記載の方法。

【請求項 3】

前記フィールドは、シーケンスパラメータセットに含まれる、
請求項 1 に記載の方法。

【請求項 4】

前記フィールドは、映像パラメータセット、ピクチャパラメータセット、スライスパラメータセットまたはタイルパラメータセットに含まれる、
請求項 1 に記載の方法。

【請求項 5】

前記現在の映像ブロックに対応するクロマブロックに対して、前記輝度バッファの前記寸法と、前記クロマブロックおよび前記現在の映像ブロックのカラーフォーマットとに基づいて、クロマバッファの寸法を導出する、
請求項 1 ～ 4 のいずれか 1 項に記載の方法。

10

【請求項 6】

前記カラーフォーマットが 4 : 2 : 0 である場合、前記クロマバッファの幅は前記輝度バッファの幅の半分に等しく、前記クロマバッファの高さは前記輝度バッファの高さの半分に等しい、
請求項 5 に記載の方法。

【請求項 7】

前記カラーフォーマットが 4 : 4 : 4 である場合、前記クロマバッファの幅は前記輝度バッファの幅に等しく、前記クロマバッファの高さは前記輝度バッファの高さに等しい、
請求項 5 に記載の方法。

20

【請求項 8】

前記カラーフォーマットが 4 : 2 : 2 である場合、前記クロマバッファの幅は前記輝度バッファの幅の半分に等しく、前記クロマバッファの高さは前記輝度バッファの高さに等しい、
請求項 5 に記載の方法。

【請求項 9】

前記クロマバッファは、C b チャンネルおよび C r チャンネルを含む、
請求項 5 ～ 8 のいずれか 1 項に記載の方法。

【請求項 10】

前記フィルタリング処理は、非ブロック化フィルタリング、サンプル適応オフセットフィルタリング、または適応ループフィルタリングを含む、
請求項 1 ～ 9 のいずれか 1 項に記載の方法。

30

【請求項 11】

前記前の映像ブロックの一部は、前記現在の映像ブロックを含む現在のコーディングツリーブロックとは異なるコーディングツリーブロックに位置する、
請求項 1 ～ 10 のいずれか 1 項に記載の方法。

【請求項 12】

前記映像領域は、コーディングツリーブロックまたはスライスである、
請求項 1 ～ 11 のいずれか 1 項に記載の方法。

40

【請求項 13】

前記変換は、前記現在の映像ブロックを前記ビットストリームに符号化することを含む、
請求項 1 ～ 12 のいずれか 1 項に記載の方法。

【請求項 14】

前記変換は、前記現在の映像ブロックを前記ビットストリームから復号化することを含む、
請求項 1 ～ 13 のいずれか 1 項に記載の方法。

【請求項 15】

処理装置と、前記処理装置に命令が記憶された非一時的メモリと、を備える映像データ処理装置であって、前記命令が前記処理装置によって実行されると、前記処理装置に、

50

輝度ブロックである映像の現在の映像ブロックと前記映像のビットストリームとの変換について、前記現在の映像ブロックに第1のコーディングモードが適用されていることを判定することと、

前記現在の映像ブロックに対する第1のブロックベクトルを導出することと、

前記第1ブロックベクトルと輝度バッファとに基づいて、前記現在の映像ブロックの予測サンプルを生成することと、

前記予測サンプルに基づいて前記変換を行うことと、を行わせ、

前記輝度バッファにはフィルタリング処理が施されていない前の映像ブロックの再構成サンプルが記憶され、前記第1のコーディングモードでは前記予測サンプルが前記映像ブロックを含む同じピクチャから導出され、

前記輝度バッファの寸法および形状が、前記現在の映像ブロックと、前記現在の映像ブロックと同じ映像領域に位置する他の輝度ブロックに対して同じであり、

前記輝度バッファの前記寸法は、前記ビットストリームに含まれるフィールドに基づいて示される、

装置。

【請求項16】

処理装置に、

輝度ブロックである映像の現在の映像ブロックと前記映像のビットストリームとの変換について、前記現在の映像ブロックに第1のコーディングモードが適用されていることを判定することと、

前記現在の映像ブロックに対する第1のブロックベクトルを導出することと、

前記第1ブロックベクトルと輝度バッファとに基づいて、前記現在の映像ブロックの予測サンプルを生成することと、

前記予測サンプルに基づいて前記変換を行うことと、を行わせる命令を記憶し、

前記輝度バッファにはフィルタリング処理が施されていない前の映像ブロックの再構成サンプルが記憶され、前記第1のコーディングモードでは前記予測サンプルが前記映像ブロックを含む同じピクチャから導出され、

前記輝度バッファの寸法および形状が、前記現在の映像ブロックと、前記現在の映像ブロックと同じ映像領域に位置する他の輝度ブロックに対して同じであり、

前記輝度バッファの前記寸法は、前記ビットストリームに含まれるフィールドに基づいて示される、

非一時的なコンピュータ可読記憶媒体。

【請求項17】

映像のビットストリームを記憶するための方法であって、

前記方法は、

第1のコーディングモードが輝度ブロックである前記映像の現在の映像ブロックに適用されることを判定することと、

前記現在の映像ブロックに対する第1のブロックベクトルを導出することと、

前記第1ブロックベクトルと輝度バッファとに基づいて、前記現在の映像ブロックの予測サンプルを生成することと、

前記予測サンプルに基づいて前記ビットストリームを生成することと、

前記ビットストリームを非一時的なコンピュータ可読記録媒体に記憶することと、を含み、

前記輝度バッファにはフィルタリング処理が施されていない前の映像ブロックの再構成サンプルが記憶され、前記第1のコーディングモードでは前記予測サンプルが前記映像ブロックを含む同じピクチャから導出され、

前記輝度バッファの寸法および形状が、前記現在の映像ブロックと、前記現在の映像ブロックと同じ映像領域に位置する他の輝度ブロックに対して同じであり、

前記輝度バッファの前記寸法は、前記ビットストリームに含まれるフィールドに基づいて示される、

10

20

30

40

50

方法。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本願は、2019年2月2日出願の国際特許出願PCT/CN2019/074598号の優先権及び利益を適時に主張する、2020年2月2日出願の国際特許出願PCT/CN2020/074155号に基づく。法に基づくすべての目的のために、上記出願の開示全体は、本明細書の開示の一部として参照により援用される。

【0002】

この特許明細書は、映像符号化および復号化技術、デバイスおよびシステムに関する。

【背景技術】

【0003】

映像圧縮の進歩にもかかわらず、デジタル映像は、依然として、インターネット及び他のデジタル通信ネットワークにおいて最大の帯域幅の使用量を占めている。映像の受信及び表示が可能な接続されたユーザ機器の数が増加するにつれ、デジタル映像の使用に対する帯域幅需要は増大し続けることが予測される。

【発明の概要】

【0004】

本明細書は、映像または画像を復号化または符号化するためのイントラブロックコピーモードにおけるバッファ管理およびブロックベクトル符号化のための様々な実施形態および技術を説明する。

【0005】

一つの例示的な態様において、映像および／またはイメージ（画像）処理の方法が開示される。この方法は、イントラブロックコピーモードにおいて、予測のための参照サンプルを記憶するバッファサイズを判定することと、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うこととを含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0006】

別の例示的な態様において、映像データ処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定することと、このバッファが、ループフィルタリングステップの前に再構成サンプルを記憶するために使用され、バッファに記憶された再構成サンプルを使用してこの変換を行うことと、を含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0007】

さらに、別の例示的な態様において、映像データ処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定することと、このバッファが、ループフィルタリングステップの後に再構成サンプルを記憶するために使用され、バッファに記憶された再構成サンプルを使用してこの変換を行うことと、を含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0008】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視

10

20

30

40

50

覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定することと、このバッファが、ループフィルタリングステップの前、およびループフィルタリングステップの後に、再構成サンプルを記憶するために使用され、バッファに記憶された再構成サンプルを使用してこの変換を行うことと、を含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0009】

別の例示的な態様において、映像処理の別の方法が開示される。この方法は、バッファを使用してイントラブロックコピーモードにおいて、予測のための参照サンプルを記憶することと、バッファの第1のビット深度は、ビットストリーム表現の視覚メディアデータを表すのに使用される第2のビット深度とは異なり、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うこととを含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

10

【0010】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、イントラブロックコピーモードにおいて、バッファを初期化して、予測のための参照サンプルを記憶することと、バッファが第1の値で初期化され、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うこととを含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

20

【0011】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、イントラブロックコピーモードにおいて、バッファを初期化して、予測のための参照サンプルを記憶することと、1つ以上の映像ブロックの可用性に基づいて、バッファが視覚メディアデータの1つ以上の映像ブロックの画素値で初期化され、バッファに記憶された参照サンプルを使用して、視覚メディアデータの1つ以上の映像ブロックに属さない現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うこととを含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

30

【0012】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定することと、バッファに記憶された参照サンプルを使用してこの変換を行うことと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、空間的に、位置 (x_0, y_0) に位置し、動き情報に含まれるブロックベクトル (BV_x, BV_y) を有する画素に対し、参照位置 $(P \bmod M, Q \bmod N)$ 、(但し $\bmod$ はモデュロ演算、 M および N は、バッファの x および y 寸法を表す整数である) に基づいてバッファ内の対応する参照を計算することと、を含み、参照位置 (P, Q) は、ブロックベクトル (BV_x, BV_y) および位置 (x_0, y_0) を使用して判定する。

40

【0013】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視

50

覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定することと、バッファに記憶された参照サンプルを使用してこの変換を行うことと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、空間的に、位置 (x_0, y_0) に位置し、動き情報に含まれるブロックベクトル (BV_x, BV_y) を有する画素に対し、参照位置 (P, Q) に基づいてバッファ内の対応する参照を計算することと、を含み、参照位置 (P, Q) は、ブロックベクトル (BV_x, BV_y) および位置 (x_0, y_0) を使用して判定する。

【0014】

10

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定することと、バッファ内の画素位置が、 x および y の数字を使用してアドレス指定され、 x および y の数字に基づいてこの変換を行うことと、この変換は、バッファに記憶された参照サンプルを使用し、を含み、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0015】

20

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定することと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、空間的に、現在の映像ブロックの位置 (x_0, y_0) に位置し、ブロックベクトル (BV_x, BV_y) を有する画素に対し、参照位置 (P, Q) に基づいてバッファ内の対応する参照を計算することと、参照位置 (P, Q) は、ブロックベクトル (BV_x, BV_y) および位置 (x_0, y_0) を使用して判定し、バッファの外に参照位置 (P, Q) があることを判定すると、バッファ内のサンプルを使用して参照位置を再計算することと、を含む。

30

【0016】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定することと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、空間的に、現在の映像ブロックを含む符号化ツリーユニットの左上位置に対する現在の映像ブロックの位置 (x_0, y_0) に位置し、ブロックベクトル (BV_x, BV_y) を有する画素に対し、参照位置 (P, Q) に基づいてバッファ内の対応する参照を計算することと、参照位置 (P, Q) は、ブロックベクトル (BV_x, BV_y) および位置 (x_0, y_0) を使用して判定し、バッファの外部に参照位置 (P, Q) があることを判定すると、参照位置の少なくとも一部を抑制して、予め規定した範囲内に位置させることと、を含む。

40

【0017】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定することと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、空間的に、現在の映像ブロックを含む符号化ツリー

50

ユニットの左上位置に対する現在の映像ブロックの位置 (x_0 , y_0) に位置し、ブロックベクトル (BV_x , BV_y) を有する画素に対し、参照位置 (P , Q) に基づいてバッファ内の対応する参照を計算することと、参照位置 (P , Q) は、ブロックベクトル (BV_x , BV_y) および位置 (x_0 , y_0) を使用して判定し、バッファの外部にブロックベクトル (BV_x , BV_y) があることを判定すると、バッファ内のサンプル値のブロックベクトルによるブロックベクトル (BV_x , BV_y) をパディングすることと、を含む。
【0018】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、映像とその映像のビットストリーム表現との間での変換中に、映像境界におけるイントラブロックコピーモードで予測のために参照サンプルを記憶するバッファをリセットすることと、バッファに記憶された参照サンプルを使用してこの変換を行うことと、を含み、この映像の映像ブロックの変換は、参照ピクチャを参照することなく、映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

10

【0019】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行うことと、イントラブロックコピーモードにおいて予測のための参照サンプルを記憶するために使用されるバッファを更新することと、を含み、このバッファは、後続の映像ブロックと後続の映像ブロックのビットストリーム表現との間での変換のために使用され、後続の映像ブロックと後続の映像ブロックのビットストリーム表現との間での変換は、参照ピクチャを参照することなく、後続の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで行われる。

20

【0020】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶する為に使用するバッファを判定することと、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われ、バッファに記憶された再構成されたサンプルが変換中にサンプル値を予測するために使用されると判定すると、バッファに記憶された再構成されたサンプルに前処理演算を行うことと、を含む。

30

【0021】

さらに別の例示的な態様において、映像処理の別の方法が開示される。この方法は、映像領域の現在の仮想パイプラインデータユニット ($VPDU$) の現在の映像ブロックと、現在の映像ブロックのビットストリーム表現との間での変換のため、選択的に、映像領域の偶数行目の K_1 個の前処理済み $VPDU$ および／または奇数行目の K_2 個の前処理済み $VPDU$ を使用するかどうかを判定することと、変換を行うことと、を含み、この変換は、現在の $VPDU$ の残りを使用することを除外し、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

40

【0022】

さらに別の例示的な態様において、上述された方法を実装するように構成された処理装置を備える映像エンコーダまたはデコーダ装置が開示される。

【0023】

別の例示的な態様において、コンピュータ可読プログラム媒体が開示される。この媒体は、開示された方法のうちの1つを実装するための処理装置実行可能命令を実施するコードを記憶する。

【0024】

これらの、および他の態様は、本明細書でより詳細に説明される。

50

【図面の簡単な説明】

【0025】

【図1】現在のピクチャ参照またはイントラブロックコピー映像または画像符号化技術を例示する。

【図2】ダイナミック参照領域を例示する。

【図3】(x, y) から始まるブロックの符号化を例示する。

【図4】前回符号化された64×64ブロックを選択するための、考えられる代替方法を例示する。

【図5】64×64ブロックの符号化／復号化の順序を変更するための、考えられる代替方法を例示する。

【図6】映像または画像の処理方法の一例を示すフローチャートである。

【図7】映像または画像の符号化または復号化を行うハードウェアプラットフォームを示すブロック図である。

【図8】64×64ブロックの復号化の順序が上から下へ、左から右である場合、前回の符号化された64×64ブロックを選択するための、別の考えられる代替方法を示す。

【図9】前回の符号化された64×64ブロックを選択するための、別の考えられる代替方法を示す。

【図10】リシェイプを伴う復号化処理の例示的なフローチャートを示す。

【図11】64×64ブロックの復号化の順序が左から右、上から下である場合、前回の符号化された64×64ブロックを選択するための、別の考えられる代替方法を示す。

【図12】IBC参照バッファステータスを示す図であり、1つのブロックは64×64CTUを示す。

【図13】IBCのための参照領域の1つの配置を示す。

【図14】IBCのための参照領域の別の配置を示す。

【図15】現在の仮想パイプラインデータユニット(VPDU)がピクチャ境界の右側にある場合のIBCのための参照領域の別の配置を示す。

【図16】1つのCTU行内のVPDUを順次復号化する場合の仮想バッファの状態の例を示す。

【図17】開示された技術を実装することができる例示的な映像処理システムを示すブロック図である。

【図18】映像データ処理方法の一例を示すフローチャートである。

【図19】映像データ処理方法の一例を示すフローチャートである。

【図20】映像データ処理方法の一例を示すフローチャートである。

【図21】映像データ処理方法の一例を示すフローチャートである。

【図22】映像データ処理方法の一例を示すフローチャートである。

【図23】映像データ処理方法の一例を示すフローチャートである。

【図24】映像データ処理方法の一例を示すフローチャートである。

【図25】映像データ処理方法の一例を示すフローチャートである。

【図26】映像データ処理方法の一例を示すフローチャートである。

【図27】映像データ処理方法の一例を示すフローチャートである。

【図28】映像データ処理方法の一例を示すフローチャートである。

【図29】映像データ処理方法の一例を示すフローチャートである。

【図30】映像データ処理方法の一例を示すフローチャートである。

【図31】映像データ処理方法の一例を示すフローチャートである。

【図32】映像データ処理方法の一例を示すフローチャートである。

【図33】映像データ処理方法の一例を示すフローチャートである。

【図34】映像データ処理方法の一例を示すフローチャートである。

【発明を実施するための形態】

【0026】

本明細書では、理解を容易にするために章の見出しを使用しており、1つの章に開示さ

10

20

30

40

50

れた実施形態をその章にのみ限定するものではない。本明細書は、映像または画像を復号化または符号化するためのイントラブロックコピーモードにおけるバッファ管理およびブロックベクトル符号化のための様々な実施形態および技術を説明する。

【0027】

1. 概要

【0028】

本明細書は、映像符号化技術に関する。具体的には、映像符号化におけるイントラブロックコピーに関する。これは、例えば、汎用映像符号化など、開発中の規格に適用されてもよい。本発明は、将来の映像符号化規格または映像コーデックにも適用可能である。

【0029】

2. 簡単な説明

【0030】

映像符号化規格は、主に周知のITU-TおよびISO/IEC規格の開発によって発展してきた。ITU-TはH.261とH.263を作り、ISO/IECはMPEG-1とMPEG-4 Visualを作り、両団体はH.262/MPEG-2 VideoとH.264/MPEG-4 AVC (Advanced Video Coding)とH.265/HEVC規格を共同で作った。H.262以来、映像符号化規格は、時間予測と変換符号化が利用されるハイブリッド映像符号化構造に基づく。HEVCを超えた将来の映像符号化技術を探索するため、2015年には、VCEGとMPEGが共同でJVET (Joint Video Exploration Team) を設立した。それ以来、多くの新しい方法がJVETによって採用され、JEM (Joint Exploration Mode) と呼ばれる参照ソフトウェアに組み込まれてきた。2018年4月には、VCEG (Q6/16) とISO/IEC JTC1 SC29/WG11 (MPEG) との間にJoint Video Expert Team (JVET) が発足し、HEVCと比較して50%のビットレート削減を目標にVVC規格の策定に取り組んでいる。

【0031】

2.1 HEVC/H.265におけるインター予測

【0032】

各インター予測されたPUは、1つまたは2つの参照ピクチャリストのための動きパラメータを有する。動きパラメータは、動きベクトルおよび参照ピクチャインデックスを含む。2つの参照ピクチャリストのうちの1つの参照ピクチャリストの使用は、inter_pred_idcを使用して信号通知されてもよい。動きベクトルは、予測子に対するデルタ (delta) として明確にコーディングされてもよい。

【0033】

1つのCUがスキップモードにて符号化される場合、1つのPUがこのCUに関連付けられ、有意な残差係数がなく、符号化された動きベクトル差分も参照ピクチャインデックスもない。マージモードを指定し、これにより、現在のPUのための動きパラメータを、空間的および時間的候補を含む近傍のPUから取得する。マージモードは、スキップモードのためだけでなく、任意のインター予測されたPUに適用することができる。マージモードの代替として、動きパラメータの明示的な送信があり、動きベクトル（より正確には、動きベクトル予測子と比較した動きベクトル差分 (MVD) ）、各参照ピクチャリストの対応する参照ピクチャインデックス、および参照ピクチャリストの使用状況が、各PUに明確に信号通知される。このようなモードを、本開示では高度動きベクトル予測 (AMVP) と呼ぶ。

【0034】

2つの参照ピクチャリストのうちの1つを使用することを信号通知が示す場合、サンプルのうちの1つのブロックからPUを生成する。これを「単一予測」と呼ぶ。PスライスおよびBスライスの両方に対して単一予測が利用可能である。

【0035】

10

20

30

40

50

両方の参照ピクチャリストを使用することを信号通知が示す場合、サンプルのうちの2つのブロックからPUを生成する。これを「双方向予測」と呼ぶ。Bスライスのみにも双方向予測が利用可能である。

【0036】

以下、HEVCに規定されるインター予測モードについて詳細に説明する。まず、マージモードについて説明する。

【0037】

2.2 現在のピクチャの参照

【0038】

現在のピクチャの参照(CPR)は、イントラブロックコピー(IBC)と呼ばれたこともあるが、HEVCスクリーンコンテンツ符号化拡張機能(HEVC-SCC)と現在のVVCテストモデルに採用されている。IBCは、動き補償の概念をフレーム間符号化からイントラフレーム符号化に拡張する。図1に示すように、現在のブロックは、CPRが適用される場合、同じピクチャ内の1つの参照ブロックによって予測される。現在のブロックを符号化または復号化する前に、参照ブロックにおけるサンプルは既に再構成されていなければならない。CPRは、カメラでキャプチャされたほとんどのシーケンスに対してそれほど効率的ではないが、スクリーンコンテンツに対しては、有意な符号化利得を示す。その理由は、スクリーンコンテンツピクチャにおいて、アイコン、文字等の繰り返しパターンが多いためである。CPRは、これらの繰り返しパターン間の冗長性を有効に除去することができる。HEVC-SCCにおいて、インター符号化ユニット(CU)は、現在のピクチャをその参照ピクチャとして選択する場合、CPRを適用することができる。この場合、MVをブロックベクトル(BV)と改称し、BVは常に整数画素精度を有する。メインプロファイルHEVCに適合するように、現在のピクチャは、復号化ピクチャバッファ(DPB)における「長期」参照ピクチャとしてマークされる。なお、同様に、マルチプルビュー／3D映像符号化規格において、ビュー間の参照ピクチャも「長期」参照ピクチャとしてマークされる。

【0039】

BVがその参照ブロックを見つけた後、この参照ブロックをコピーすることで予測を生成することができる。残差は、元の信号から参照画素を減算することによって得ることができる。そして、他の符号化モードと同様に、変換および量子化を適用することができる。

【0040】

図1は、現在のピクチャの参照の例示である。

【0041】

しかしながら、参照ブロックがピクチャの外にある場合、または現在のブロックと重複する場合、または再構成された領域の外にある場合、或いは何らかの制約によって制限された有効領域の外にある場合、画素値の一部または全部は規定されない。基本的に、このような問題に対処するために2つの解決策がある。1つは、このような状況、例えばビットストリーム適合性を許可しないことである。もう1つは、これらの未定義の画素値にパディングを適用することである。以下のサブセッションでは、解決策を詳細に説明する。

【0042】

2.3 HEVCスクリーンコンテンツ符号化拡張機能におけるCPR

【0043】

HEVCのスクリーンコンテンツ符号化拡張機能において、1つのブロックが現在のピクチャを参照として使用する場合、以下の仕様のテキストに示すように、参照ブロック全体が利用可能な再構成された領域内にあることを保証すべきである。

変数offsetXおよびoffsetYは、以下のように導出される。

$$\text{offsetX} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[0] \& 0x7 ? 2 : 0) \quad (8-104)$$

$$\text{offsetY} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[1] \& 0x7 ? 2 : 0) \quad (8-105)$$

参照ピクチャが現在のピクチャである場合、輝度動きベクトル $mvLX$ は、以下の制約に従うべきであることが、ビットストリーム適合性の要件である。

－ 6.4.1項で規定されたような z 走査順序ブロックの可用性に対する導出処理が、 (xCb, yCb) と等しく設定された $(xCurr, yCurr)$ と、 $(xPb + (mvLX[0] >> 2) - offsetX, yPb + (mvLX[1] >> 2) - offsetY)$ に等しく設定された近傍の輝度位置 $(xNbY, yNbY)$ と、が入力として呼び出されると、出力は `TRUE` となる。

－ 6.4.1項に規定されたような z 走査順序ブロックの可用性に対する導出処理が、 (xCb, yCb) と等しく設定された $(xCurr, yCurr)$ と、 $(xPb + (mvLX[0] >> 2) + nPbW - 1 + offsetX, yPb + (mvLX[1] >> 2) + nPbH - 1 + offsetY)$ と等しく設定された近傍の輝度位置 $(xNbY, yNbY)$ と、が入力として呼び出されると、出力は `TRUE` となる。

－ 次の条件の一方または両方が `true` であること

－ $(mvLX[0] >> 2)$ の値 $+ nPbW + xB1 + offsetX$ が 0 以下である。

－ $(mvLX[1] >> 2)$ の値 $+ nPbH + yB1 + offsetY$ が 0 以下である。

－ 次の条件が `true` となること。

$(xPb + (mvLX[0] >> 2) + nPbSw - 1 + offsetX) / CtbSizeY - xCb / CtbSizeY \leq yCb / CtbSizeY - (yPb + (mvLX[1] >> 2) + nPbSh - 1 + offsetY) / CtbSizeY$ (8-106) .

【0044】

このように、参照ブロックが現在のブロックと重複するケース、または参照ブロックがピクチャの外にあるケースは発生しない。参照ブロックまたは予測ブロックを埋める必要がない。

【0045】

2.4 CPR / IBC の例

【0046】

VVC 試験モデルにおいて、参照ブロック全体は現在の符号化ツリーユニット (CTU) を有するべきであり、現在のブロックと重複しない。よって、参照または予測ブロックをパディングする必要がない。

【0047】

デュアルツリーが有効化される場合、分割構造は、輝度 CTU とクロマ CTU とで異なってもよい。従って、4:2:0 カラーフォーマットの場合、1つのクロマブロック (例えば、CU) は、複数の輝度 CU に分割されたコロケーションされた 1つの輝度領域に対応してもよい。

【0048】

クロマブロックは、以下の条件が `true` である場合に、CPRモードでのみ符号化することができる。

1) コロケーションされた輝度ブロックの輝度 CU の各々は、CPRモードで符号化されるべきである。

2) 輝度 4×4 ブロック' BV の各々は、まず、クロマブロックの BV に変換され、クロマブロックの BV は、有効な BV である。

【0049】

2つの条件のいずれかが `false` である場合、クロマブロックは、CPRモードで符号化されない。

【0050】

なお、有効な BV の定義には、以下の制約がある。

1) BV によって特定される参照ブロックのすべてのサンプルは、制限された検索範囲内

10

20

30

40

50

にあるべきである（例えば、現在のVVC設計において同じCTU内にある）。

2) BVによって特定された参照ブロックのすべてのサンプルが再構成されている。

【0051】

2.5 CPR/IBCの例

【0052】

いくつかの実施例において、CPR/IBCのための参照領域は、現在のCTUに制限され、最大 128×128 である。CPR/IBCブロックがより多くの参照候補を有することができる一方、CPR/IBCの参照バッファを1つのCTUから維持または低減することができるように、参照領域を動的に変更してメモリを再利用し、CPR/IBCの参照サンプルを記憶する。

【0053】

図2は1つの方法を示しており、1つのブロックは 64×64 であり、1つのCTUは4つの 64×64 ブロックを含む。 64×64 ブロックを符号化する場合、前の3つの 64×64 ブロックを参照として使用できる。そうすることによって、デコーダは、CPR/IBCをサポートするために、4つの 64×64 ブロックを記憶するだけでよい。

【0054】

ピクチャの左上隅に対する現在の輝度CUの位置を (x, y) とし、ブロックベクトルを (BVx, BVy) とする。現在の設計において、BVが有効であるかどうかは、輝度位置 $((x + BVx) \gg 6 < 6 + (1 < 7))$ 、 $((y + BVy) \gg 6 < 6)$ が再構成されておらず、 $((x + BVx) \gg 6 < 6 + (1 < 7))$ 、 $((y + BVy) \gg 6 < 6)$ が $(x \gg 6 < 6, y \gg 6 < 6)$ と等しくないことによって知ることができる。

【0055】

2.6 インループリシェイプ(ILR)

【0056】

インループリシェイプ(ILR)の基本的な考えは、元の(第1のドメインにおける)信号(予測/再構成信号)を第2のドメイン(リシェイプされたドメイン)に変換することである。

【0057】

インループリシェイパは、1対のルックアップテーブル(LUT)として実装されるが、信号通知されたLUTから他方を計算することができるので、2つのLUTのうちの一方のみを信号通知する必要がある。各LUTは、1次元10ビット1024エントリマッピングテーブル(1D-LUT)である。1つのLUTは、フォワードLUT、FwdLUTであり、これは、入力輝度コード値 Y_i を変更された値 Y_r ： $Y_r = \text{FwdLUT}[Y_i]$ にマッピングする。他方のLUTは、逆LUT、InvLUTであり、変更されたコード値 Y_r を $\hat{Y}_i$ ： $\hat{Y}_i = \text{InvLUT}[Y_r]$ にマッピングする。 $(\hat{Y}_i$ は Y_i の再構成値を表す)。

【0058】

2.6.1 PWLモデル

【0059】

概念的には、区分線形(PWL)は、以下のように実装される。

【0060】

x_1 、 x_2 を2つの入力支点とし、 y_1 、 y_2 を1つのピースに対応する出力支点とする。 x_1 と x_2 との間の任意の入力値 x に対する出力値 y は、以下の式によって補間することができる

【0061】

$$y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$$

【0062】

固定点実装では、この式は、以下のように書き換えることができる。

$$y = ((m * x + 2^{FP_PREC} - 1) \gg FP_PREC) + c$$

10

20

30

40

50

【0063】

mはスカラーであり、cはオフセットであり、FP__PRECは精度を規定するための定数である。

【0064】

いくつかの実施例において、PWLモデルは、1024エントリのFwdLUTマッピングテーブルおよびInvLUTマッピングテーブルを予め計算するために使用されるが、PWLモデルは、LUTを予め計算することなく、実装において同一のマッピング値をオンザフライで計算することも可能にする。

【0065】

2.6.2.1 輝度のリシェイプ

【0066】

インループ輝度リシェイプの方法は、より複雑性低いパイプラインを提供し、且つインタースライス再構成におけるブロック単位のイントラ予測のための復号化待ち時間を排除する。イントラ予測は、インタースライスおよびイントラスライスの両方のために、リシェイプされたドメインにおいて行われる。

【0067】

イントラ予測は、スライスタイプにかかわらず、常にリシェイプされたドメインで行われる。このような構成によれば、前回のTU再構成を行った直後にイントラ予測を開始することができる。このような構成は、スライスに依存する代わりに、イントラモードのための統一された処理を提供することもできる。図10は、CE12-2の方式に基づく復号化処理を示すブロック図である。

【0068】

32ピース区分線形(PWL)モデルの代わりに、16ピースのPWLモデルを使用し、輝度およびクロマ残差スケーリングを行った。

【0069】

インループ輝度リシェイプを用いたインタースライス再構成(薄い緑色の影が付けられたブロックは、リシェイプされたドメインにおける信号を示す。輝度残差、輝度内予測、および輝度内再構成)

【0070】

2.6.2.2 輝度依存クロマ残差スケーリング

【0071】

輝度依存クロマ残差スケーリングは、固定小数点整数演算で実装される乗算処理である。クロマ残差スケーリングは、輝度信号のクロマ信号との相互作用を補償する。TUレベルでクロマ残差スケーリングを適用する。具体的には、以下が適用される。

- ー イントラの場合、再構成された輝度を平均する。
- ー インターの場合、予測輝度を平均する。

【0072】

この平均値は、PWLモデルにおけるインデックスを特定するために使用される。このインデックスは、スケーリング係数cScaleInvを特定する。クロマ残差にその数を乗算する。

【0073】

なお、クロマスケール係数は、再構成された輝度値ではなく、フォワードマップされた予測輝度値から算出される。

【0074】

2.6.2.3 ILR側情報の信号通知

【0075】

パラメータは(現在)タイルグループヘッダ(ALFに類似)で送信される。これらは、40~100ビットを要すると報告されている。

【0076】

いくつかの実施例において、追加された構文は、イタリック体で強調されている。

10

20

30

40

50

【0077】

【表1】

7. 3. 2. 1におけるシーケンスパラメータセットR B S P構文

seq_parameter_set_rbsp() {	ディスクリプタ
sps_seq_parameter_set_id	• ue(v)
...	•
sps_triangle_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)
}	•
}	•
<i>sps_resaper_enabled_flag</i>	• <i>u(1)</i>
rbps_trailing_bits()	•
}	•

10

20

【0078】

【表2】

7. 3. 3. 1におけるタイルグループヘッダ構文

tile_group_header() {	ディスクリプタ
...	
if(num_tiles_in_tile_group_minus1 > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_tiles_in_tile_group_minus1; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if (sps_resaper_enabled_flag) {	•
<i>tile_group_resaper_model_present_flag</i>	• <i>u(1)</i>
if (<i>tile_group_resaper_model_present_flag</i>)	•
<i>tile_group_resaper_model</i> ()	•
<i>tile_group_resaper_enable_flag</i>	• <i>u(1)</i>
if (<i>tile_group_resaper_enable_flag</i> && (!(<i>qtbtt_dual_tree_intra_flag</i> && <i>tile_group_type</i> == <i>I</i>)))	•
<i>tile_group_resaper_chroma_residual_scale_flag</i>	• <i>u(1)</i>
}	•
byte_alignment()	•
}	•

30

40

新しい構文表タイルグループリシェイパを追加する。

【0079】

50

【表 3】

<i>tile_group_reshaper_model () {</i>	ディスクリプ タ
<i>reshaper_model_min_bin_idx</i>	<i>ue(v)</i>
<i>reshaper_model_delta_max_bin_idx</i>	<i>ue(v)</i>
<i>reshaper_model_bin_delta_abs_cw_prec_minus1</i>	<i>ue(v)</i>
<i>for (i = reshaper_model_min_bin_idx; i <=</i> <i>reshaper_model_max_bin_idx; i++) {</i>	
<i>reshape_model_bin_delta_abs_CW [i]</i>	<i>u(v)</i>
<i>if (reshaper_model_bin_delta_abs_CW[i]</i> <i>)>0)</i>	
<i>reshaper_model_bin_delta_sign_CW_flag[i]</i>	<i>u(1)</i>
<i>}</i>	
<i>}</i>	

10

一般的に、シーケンスパラメータセットR B S Pの意味論では、次の意味論を追加する。
*sps__reshaper__enabled__flag*が1と等しい場合は、符号化映像
 シーケンス (C V S) においてリシェイパが使用されていることを規定している。
*sps__reshaper__enabled__flag*が0と等しい場合は、C V Sにおいてリ
 シェイパが使用されていないことを規定している。

20

タイルグループヘッダ構文において、以下の意味論を追加する。

*tile__group__reshaper__model__present__flag*が1
 と等しい場合は、*tile__group__reshaper__model ()*がタイルグ
 ループ内に存在することを規定している。

*tile__group__reshaper__model__present__flag*が0
 と等しい場合は、*tile__group__reshaper__model ()*がタイルグ
 ループヘッダに存在しないことを規定している。

30

*tile__group__reshaper__model__present__flag*が存
 在しない場合は、0に等しいと推論される。

*tile__group__reshaper__enabled__flag*が1と等しい場合
 は、現在のタイルグループに対してリシェイパが有効になっていることを規定している。

*tile__group__reshaper__enabled__flag*が0と等しい場合
 は、現在のタイルグループに対してリシェイパが有効になっていないことを規定している。

*tile__group__reshaper__enable__flag*が存在しない場合、
 0であると推論される。

*tile__group__reshaper__chroma__residual__scal
 e__flag*が1と等しい場合は、現在のタイルグループに対してクロマ残差スケーリン
 グを有効であることを規定している。

40

*tile__group__reshaper__chroma__residual__scal
 e__flag*が0と等しい場合は、現在のタイルグループに対してクロマ残差スケーリン
 グが有効でないことを規定している。

*tile__group__reshaper__chroma__residual__scal
 e__flag*が存在しない場合は、0であると推測される。

tile__group__reshaper__model () 構文を追加する

*reshape__model__min__bin__idx*は、最小のビン (またはピース)
 インデックスを、リシェイパ構築処理に使用するように規定している。*reshape__
 model__min__bin__idx*の値が0 ~ *MaxBinIdx*の範囲内にあるもの

50

とする。

MaxBinIdxの値は15と等しいとする。

reshape_model_delta_max_bin_idxは、最大許容ビン（またはピース）インデックスMaxBinIdxから最大ビンインデックスを引いたものが、リシェイパ構築処理で使用されることを規定している。reshape_model_delta_max_bin_idxの値は、MaxBinIdx-reshape_model_delta_max_bin_idxに等しく設定される。

reshaper_model_bin_delta_abs_cw_prec_min_us1+1は、構文reshape_model_bin_delta_abs_CW[i]の表現に使用するビット数を規定している。

reshape_model_bin_delta_abs_CW[i]は、i番目のビンの絶対デルタコード名値を規定する。

reshaper_model_bin_delta_sign_CW_flag[i]は、reshape_model_bin_delta_abs_CW[i]のサインを次のように記述する。

— reshape_model_bin_delta_sign_CW_flag[i]が0と等しい場合、対応する変数RspDeltaCW[i]は正の値である。

— そうでない場合（reshape_model_bin_delta_sign_CW_flag[i]が0と等しくない）、対応する変数RspDeltaCW[i]は負の値である。

reshape_model_bin_delta_sign_CW_flag[i]が存在しない場合、0に等しいと推測される。

変数RspDeltaCW[i] = (1 2 * reshape_model_bin_delta_sign_CW[i]) * reshape_model_bin_delta_abs_CW[i] ;

変数RspCW[i]は、以下のステップとして導出される。

変数OrgCWは、(1 << BitDepthY) / (MaxBinIdx + 1) に等しく設定される。

— reshaper_model_min_bin_idx ≤ i ≤ reshaper_model_max_bin_idxの場合、RspCW[i] = OrgCW + RspDeltaCW[i] である。

— そうでない場合、RspCW[i] = 0である。

BitDepthYの値が10に等しい場合、RspCW[i]の値は、32 ~ 2 * OrgCW_1の範囲内にくる。

iが0 ~ MaxBinIdx + 1の範囲にある変数InputPivot[i]は、以下のように導出される。

InputPivot[i] = i * OrgCW

iが0 ~ MaxBinIdx + 1の範囲にあり、変数ScaleCoef[i]およびInvScaleCoef[i]がそれぞれ0 ~ MaxBinIdxの範囲にあり、iが0 ~ MaxBinIdxの範囲にある変数ReshapePivot[i]は、以下のように導出される。

shiftY = 14

ReshapePivot[0] = 0 ;

for (i = 0 ; i ≤ MaxBinIdx ; i++) {

ReshapePivot[i + 1] = ReshapePivot[i] + RspCW[i]

ScaleCoef[i] = (RspCW[i] * (1 << shiftY) + (1 < (Log2(OrgCW) - 1))) >> (Log2(OrgCW))

if (RspCW[i] == 0)

InvScaleCoef[i] = 0

10

20

30

40

50

```

    else
        InvScaleCoeff[i] = OrgCW * (1 << shiftY) / RspCW[i]
    }

```

i が 0 ~ MaxBinIdx の範囲内にある変数 ChromaScaleCoeff[i] は、以下のように導出される。

```

ChromaResidualScaleLut[64] = {16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192,
8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096,
4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731,
2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048,
1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489,
1489, 1489, 1365, 1365, 1365, 1365, 1260,
1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092,
1092, 1092, 1092, 1024, 1024, 1024, 1024};

```

```

    shiftC = 11
    - if (RspCW[i] == 0)
        ChromaScaleCoeff[i] = (1 << shiftC)
    - Otherwise (RspCW[i] != 0), ChromaScaleCoeff[i] =
        ChromaResidualScaleLut[RspCW[i] >> 1]

```

【0080】

2.6.2.4 ILR の使用方法

【0081】

エンコーダ側では、まず、各ピクチャ（またはタイルグループ）をリシェイプされたドメインに変換する。そして、すべての符号化処理は、リシェイプされたドメインで行われる。イントラ予測の場合、近傍のブロックはリシェイプされたドメインにあり、インター予測の場合、まず、参照ブロック（復号化ピクチャバッファからの元のドメインから生成される）をリシェイプされたドメインに変換する。そして、残差を生成し、ビットストリームに符号化する。

【0082】

ピクチャ全体（またはタイルグループ）の符号化／復号化が終わった後、リシェイプされたドメインのサンプルを元のドメインに変換し、次いで、非ブロック化フィルタおよび他のフィルタを適用する。

【0083】

以下の場合、予測信号へのフォワードリシェイプは無効化される。

【0084】

現在のブロックはイントラ符号化されている。

【0085】

現在のブロックは CPR（現在のピクチャを参照し、別名イントラブロックコピー、IBC）として符号化される。

【0086】

現在のブロックは、結合されたインターイントラモード（CIIP）として符号化され、イントラ予測ブロックのためにフォワードリシェイプは無効化される。

【0087】

3. 様々な実施形態によって解決される問題の例

【0088】

現在の CPR／IBC の設計において、いくつかの問題が存在する。

1) 参照領域が動的に変化するため、エンコーダ／デコーダの処理が複雑になる。

2) 無効なブロックベクトルは容易に生成され、チェックしにくいため、エンコーダお

10

20

30

40

50

よびデコーダの両方を複雑にする。

3) 参照領域が不規則であると、ブロックベクトルの符号化が非効率になる。

4) 128×128 より小さいCTUサイズに対してどのように対処するかは不明である。

5) BVが有効であるか無効であるかの判定処理において、クロマブロックの場合、この決定は、輝度サンプルの可用性に基づいて行われ、このことは、デュアルツリーパーティション構造に起因して誤った判定をもたらす可能性がある。

【0089】

4. 例示的な実施形態

【0090】

いくつかの実施形態において、CPR/IBCブロックが参照を得るために、通常のバッファを使用できる。

【0091】

関数 $isRec(x, y)$ は、画素 (x, y) が再構成され、IBCモードによって参照されたかどうかを示すように規定される。 (x, y) が異なるスライス/タイル/ブリックのピクチャから外れている場合、 $isRec(x, y)$ は $false$ を返し、 (x, y) が再構成されていない場合、 $isRec(x, y)$ は $false$ を返す。別の例において、サンプル (x, y) が再構成されたが、何らかの他の条件が満たされた場合、そのサンプルは、参照領域外/異なるVPDU内など、利用不可能であるとマークされてもよく、 $isRec(x, y)$ は $false$ を返す。

【0092】

関数 $isRec(c, x, y)$ は、成分 c のサンプル (x, y) が利用可能であるかどうかを示すように規定される。例えば、サンプル (x, y) がまだ再構成されていない場合、それは利用不可能であるとマークされる。別の例において、サンプル (x, y) が再構成されたが、何らかの他の条件が満たされた場合、該サンプル (x, y) はまた、使用不可能であるとしてマークされてもよく、例えば、ピクチャから外れている/異なるスライス/タイル/ブリック内にある/異なるVPDU内にある、許可された参照領域のなどのようにサンプル (x, y) が利用不可能である場合、 $isRec(c, x, y)$ は $false$ を返し、そうでない場合、 $true$ を返す。

【0093】

以下の説明において、参照サンプルは再構成サンプルであってもよい。なお、「画素バッファ」は、「1つの色成分のバッファ」または「複数の色成分のバッファ」に応答してもよい。

【0094】

CPR/IBC用参照バッファ

1. CPR/IBCのために輝度参照サンプルを記憶するために、 $M \times N$ 個の画素バッファを使用することが提案される。

a. 一例において、バッファサイズは 64×64 である。

b. 一例において、バッファサイズは 128×128 である。

c. 一例において、バッファサイズは 64×128 である。

d. 一例において、バッファサイズは 128×64 である。

e. 一例において、 N はCTUの高さに等しい。

f. 一例において、 $N = nH$ であり、 H はCTUの高さであり、 n は正の整数である。

g. 一例において、 M はCTUの幅に等しい。

h. 一例において、 $M = mW$ であり、 W はCTUの幅であり、 m は正の整数である。

i. 一例において、バッファサイズは、CTUサイズに等しくなく、例えば、 96×128 または 128×96 である。

j. 一例において、バッファサイズはCTUサイズに等しい。

k. 一例において、 $M = mW$ 、 $N = H$ であり、 W および H は、CTUの幅および高さであり、 m は、正の整数である。

10

20

30

40

50

l. 一例において、 $M=W$ 、 $N=nH$ であり、 W および H は、 CTU の幅および高さであり、 n は、正の整数である。

m. 一例において、 $M=mW$ 、 $N=nH$ であり、 W および H は、 CTU の幅および高さであり、 m および n は、正の整数である。

n. 上記例において、 m および n は CTU のサイズに依存してもよい。

i. 一例において、 CTU のサイズが 128×128 である場合、 $m=1$ であり、 $n=1$ である。

ii. 一例において、 CTU のサイズが 64×64 である場合、 $m=4$ であり、 $n=1$ である。

iii. 一例において、 CTU のサイズが 32×32 である場合、 $m=16$ であり、 $n=1$ である。

iv. 一例において、 CTU のサイズが 16×16 である場合、 $m=64$ であり、 $n=1$ である。

o. 代替的に、バッファサイズは CTU サイズに対応する。

p. 代替的に、このバッファサイズは、仮想パイプラインデータユニット ($VPDU$) のサイズに対応する。

q. M および/または N は、エンコーダからデコーダに、例えば $VPS/SPS/PPS$ /ピクチャヘッダ/スライスヘッダ/タイルグループヘッダ等で信号通知されてもよい。

2. M および/または N は、規格で規定された異なるプロファイル/レベル/層で異なってもよい。別の $M_c \times N_c$ 画素バッファを使用して、 CPR/IBC のためのクロマ参照サンプルを記憶することが提案される。

a. 一例において、 $4:2:0$ 映像の場合、 $M_c=M/2$ および $N_c=N/2$ である。

b. 一例において、 $4:4:4$ 映像の場合、 $M_c=M$ および $N_c=N$ である。

c. 一例において、 $4:2:2$ 映像の場合、 $M_c=M$ および $N_c=N/2$ である。

d. 代替的に、 M_c および N_c は、 M および N から独立してもよい。

e. 一例において、クロマバッファは、 C_b および C_r に対応する2つのチャンネルを含む。

f. 一例において、 $M_c=M$ であり、 $N_c=N$ である。

3. $M \times N$ サンプルバッファを使用して PR/IBC のために RGB 参照サンプルを記憶ことが提案される。

a. 一例において、バッファサイズは 64×64 である。

b. 一例において、バッファサイズは 128×128 である。

c. 一例において、バッファサイズは 64×128 である。

d. 一例において、バッファサイズは 128×64 である。

e. 代替的に、バッファサイズは CTU サイズに対応する。

f. 代替的に、このバッファサイズは、仮想パイプラインデータユニット ($VPDU$) のサイズに対応する。

4. バッファは、ループフィルタリングの前に再構成された画素を記憶することができることが提案される。ループフィルタリングは、非ブロック化フィルタ、適応ループフィルタ (ALF)、サンプル適応オフセット (SAO)、クロス成分 ALF 、または任意の他のフィルタを参照してもよい。

a. 一例において、バッファは、現在の CTU にサンプルを記憶することができる。

b. 一例において、バッファは、現在の CTU の外部にサンプルを記憶することができる。

c. 一例において、バッファは、現在のピクチャの任意の部分からのサンプルを記憶することができる。

d. 一例において、バッファは、他のピクチャからのサンプルを記憶することができる。

5. バッファは、ループフィルタリングの後に再構成された画素を記憶することができ

10

20

30

40

50

ることが提案される。ループフィルタリングは、非ブロック化フィルタ、適応ループフィルタ（ALF）、サンプル適応オフセット（SAO）、クロス成分ALF、または任意の他のフィルタを参照してもよい。

- a. 一例において、バッファは、現在のCTUにサンプルを記憶することができる。
- b. 一例において、バッファは、現在のCTUの外部にサンプルを記憶することができる。
- c. 一例において、バッファは、現在のピクチャの任意の部分からのサンプルを記憶することができる。
- d. 一例において、バッファは、他のピクチャからのサンプルを記憶することができる。

10

6. バッファは、ループフィルタリング前およびループフィルタリング後の両方で再構成されたサンプルを記憶することができることが提案される。ループフィルタリングは、非ブロック化フィルタ、適応ループフィルタ（ALF）、サンプル適応オフセット（SAO）、クロス成分ALF、または任意の他のフィルタを参照してもよい。

- a. 一例において、バッファは、現在のピクチャからのサンプルと他のピクチャからのサンプルとを、それらのサンプルの可用性に基づいて記憶することができる。
- b. 一例において、他のピクチャからの参照サンプルは、ループフィルタリング後の再構成サンプルからのものである。
- c. 一例において、他のピクチャからの参照サンプルは、ループフィルタリングの前に再構成されたサンプルからのものである。

20

7. バッファは、符号化映像データのビット深度とは異なり得る所与のビット深度を有するサンプルを記憶することが提案される。

- a. 一例において、再構成バッファ／符号化映像データのビット深度は、バッファに記憶されたIBC参照サンプルのビット深度よりも大きい。
- b. 一例において、内部ビット深度が1つの映像シーケンスのための入力ビット深度と異なる場合、例えば（10ビット対8ビット）であっても、IBC参照サンプルは、入力ビット深度に位置合わせされて記憶される。
- c. 一例において、ビット深度は再構成バッファのビット深度と同じである。
- d. 一例において、ビット深度は、入力画像／映像のビット深度と同じである。
- e. 一例において、ビット深度は、予め規定された数と同じである。
- f. 一例において、ビット深度は、標準のプロファイルに依存する。
- g. 一例において、出力ビット深度／入力ビット深度／内部ビット深度と比較したビット深度またはビット深度の差は、SPS／PPS／シーケンスヘッダ／ピクチャヘッダ／スライスヘッダ／タイルグループヘッダ／タイルヘッダまたは他の種類の映像データユニットにおいて信号通知されてもよい。
- h. 提案された方法は、他の黒丸印で言及された提案されたバッファ定義に適用されてもよく、代替的に、IBCの既存の設計にも適用されてもよい。
- i. バッファの各色成分のビット深度は異なってもよい。

30

【0095】

バッファ開始

40

8. 所与の値でバッファを初期化することが提案される。

- a. 一例において、バッファは所与の値で初期化される。
 - i. 一例において、所与の値は、入力ビット深度および／または内部ビット深度に依存してもよい。
 - ii. 一例において、バッファは、中間グレー値で初期化され、例えば、8ビット信号の場合、128が設定され、10ビット信号の場合、512が設定される。
 - iii. 一例において、ILRが使用される場合、バッファはforwardLUT(m)で初期化される。例 $m = 1 \ll (Bitdepth - 1)$ 。
- b. 代替的に、バッファは、SPS／VP S／APS／PPS／シーケンスヘッダ／タイルグループヘッダ／ピクチャヘッダ／tile／CTU／符号化ユニット／VPDU

50

／領域で信号通知される値で初期化される。

c. 一例において、この所与の値は、前に復号化されたピクチャまたはスライスまたはCTUの行またはCTUまたはCUのサンプルから導出されてもよい。

d. この所与の値は、異なる色成分ごとに異なってもよい。

9. 代替的に、前回符号化されたブロックからの復号化された画素を使用してバッファを初期化することが提案される。

a. 一例において、復号化された画素は、インループフィルタリング前の画素である。

b. 一例において、バッファサイズがCTUである場合、バッファは、利用可能である場合、前回の復号化されたCTUの復号化された画素で初期化される。

c. 一例において、バッファサイズが 64×64 である場合、そのバッファサイズは、利用可能である場合、前回の復号化された 64×64 ブロックの復号化された画素で初期化される。

d. 代替的に、さらに、前回符号化されたブロックが利用可能でない場合、黒丸印8の方法が適用されてもよい。

【0096】

バッファの参照

10. 1つのブロックがバッファ内の画素を参照として使用するために、このブロックは、バッファ内の位置 (x, y) 、 $x = 0, 1, 2, \dots, M-1$; $y = 0, 1, 2, \dots, N-1$ 、を使って参照先を示すことができる。

11. 代替的に、この参照位置は、 $l = y * M + x$ 、 $l = 0, 1, \dots, M * N - 1$ として表すことができる。

12. なお、現在のCTUに関するブロックの左上の位置を (x_0, y_0) として、ブロックベクトル $(BVx, BVy) = (x - x_0, y - y_0)$ をデコーダに送信し、バッファにおける参照先を示してもよい。

13. 代替的に、ブロックベクトル (BVx, BVy) は、 $(x - x_0 + Tx, y - y_0 + Ty)$ として規定されてもよく、 Tx および Ty は、予め規定されたオフセットである。

14. 任意の画素 (x_0, y_0) および (BVx, BVy) の場合、バッファにおけるその参照は、 $(x_0 + BVx, y_0 + BVy)$ に見出すことができる。

a. 一例において、 $(x_0 + BVx, y_0 + BVy)$ がバッファの外にある場合、それは境界にクリップされる。

b. 代替的に、 $(x_0 + BVx, y_0 + BVy)$ がバッファの外にある場合、その基準値は、所与の値、例えば、中間グレーとして予め規定される。

c. 代替的に、参照位置は、常にバッファ内にあるように、 $((x_0 + BVx) \bmod M, (y_0 + BVy) \bmod N)$ として規定される。

15. 任意の画素 (x_0, y_0) および (BVx, BVy) に対して、 $(x_0 + BVx, y_0 + BVy)$ がバッファの外にある場合、その基準値はバッファにおける値から導出されてもよい。

a. 一例において、この値は、バッファにおけるサンプル $((x_0 + BVx) \bmod M, (y_0 + BVy) \bmod N)$ から導出される。

b. 一例において、この値は、バッファにおけるサンプル $((x_0 + BVx) \bmod M, \text{clip}(y_0 + BVy, 0, N-1))$ から導出される。

c. 一例において、この値は、バッファにおけるサンプル $(\text{clip}(x_0 + BVx, 0, M-1), (y_0 + BVy) \bmod N)$ から導出される。

d. 一例において、この値は、バッファにおけるサンプル $(\text{clip}(x_0 + BVx, 0, M-1), \text{clip}(y_0 + BVy, 0, N-1))$ から導出される。

16. バッファ範囲外の特定の座標を許可しない場合がある。

a. 一例において、CTUの左上隅およびブロックベクトル (BVx, BVy) に対する任意の画素 (x_0, y_0) について、 $y_0 + BVy$ が $[0, \dots, N-1]$ の範囲内にあるべきであることが、ビットストリーム制約である。

10

20

30

40

50

b. 一例において、CTUおよびブロックベクトル (BV_x , BV_y) の左上隅の任意の画素 (x_0 , y_0) に対して、 $x_0 + BV_x$ が $[0, \dots, M-1]$ の範囲内にあるべきであることが、ビットストリーム制約である。

c. 一例において、CTUおよびブロックベクトル (BV_x , BV_y) の左上隅の任意の画素 (x_0 , y_0) に対して、 $y_0 + BV_y$ が $[0, \dots, N-1]$ の範囲内にあるべきであり、且つ $x_0 + BV_x$ が $[0, \dots, M-1]$ の範囲内にあるべきであることが、ビットストリーム制約である。

17. 1つのブロックの信号通知されたまたは導出されたブロックベクトルがバッファの外のどこかを指している場合、バッファに基づいてパディングを行ってもよい。

a. 一例において、バッファの外の任意のサンプルの値は、予め規定された値で規定される。

i. 一例において、この値は、 $1 < (\text{ビット深度} - 1)$ であってもよく、例えば、8ビット信号の場合、128であり、10ビット信号の場合、512である。

ii. 一例において、ILRが使用される場合、この値は $\text{forwardLUT}(m)$ であってもよい。例 $m = 1 \ll (\text{ビット深度} - 1)$ 。

iii. 代替的に、予め規定された値の表示は、SPS/PPS/シーケンスヘッダ/ピクチャヘッダ/スライスヘッダ/タイルグループ/タイル/CTU/CUレベルで信号通知または表示されてもよい。

b. 一例において、バッファの外の任意のサンプルは、バッファにおける最も近いサンプルの値として規定される。

18. バッファ参照からの取り扱い方法は、水平方向と垂直方向とで異なってもよく、または現在のブロックの位置によって（例えば、ピクチャ境界により近いかな否か）異なってもよい。

a. 一例において、 $y_0 + BV_y$ が $[0, N-1]$ の外にある場合、 $(x_0 + BV_x, y_0 + BV_y)$ のサンプル値が予め規定された値として割り当てられる。

b. 一例において、 $x_0 + BV_x$ が $[0, M-1]$ の外にある場合、 $(x_0 + BV_x, y_0 + BV_y)$ のサンプル値が予め規定された値として割り当てられる。

c. 代替的に、 $(x_0 + BV_x, y_0 + BV_y)$ のサンプル値を $((x_0 + BV_x) \bmod M, y_0 + BV_y)$ のサンプル値として割り当て、これにより、 $((x_0 + BV_x) \bmod M, y_0 + BV_y)$ がまだバッファの外にある場合には、その値をさらに導出するために他の方法と呼び出してもよい。

d. 代替的に、 $(x_0 + BV_x, y_0 + BV_y)$ のサンプル値を $(x_0 + BV_x, (y_0 + BV_y) \bmod N)$ のサンプル値として割り当て、これにより、 $(x_0 + BV_x, (y_0 + BV_y) \bmod N)$ がまだバッファの外にある場合には、その値をさらに導出するために他の方法と呼び出してもよい。

【0097】

ブロックベクトル表現

19. ブロックベクトル (BV_x , BV_y) の各成分またはその成分の1つを一定の範囲に正規化してもよい。

a. 一例において、 BV_x は、 $(BV_x \bmod M)$ に置き換えられてもよい。

b. 代替的に、 BV_x は、 $((BV_x + X) \bmod M) - X$ に置き換えられてもよく、 X は、予め規定された値である。

i. 一例において、 X は64。

ii. 一例において、 X は $M/2$ 。

iii. 一例において、 X は、現在のCTUに対するブロックの水平座標である。

c. 一例において、 BV_y は、 $(BV_y \bmod N)$ に置き換えられてもよい。

d. 代替的に、 BV_y を $((BV_y + Y) \bmod N) - Y$ で置き換えてもよく、 Y は予め規定された値である。

i. 一例において、 Y は64。

ii. 一例において、 Y は $N/2$ 。

- i i i . 一例において、Yは、現在のCTUに対するブロックの垂直座標である。
- 20 . BVxおよびBVyは、異なる正規化範囲を有してもよい。
- 21 . ブロックベクトル差(BVDx, BVDy)を一定の範囲に正規化することができる。
- a . 一例において、BVDxは、 $(BVDx \bmod M)$ に置き換えられてもよく、関数modは、リマインダを返す。
- b . 代替的に、BVDxは、 $((BVDx + X) \bmod M) - X$ に置き換えられてもよく、Xは、予め規定された値である。
- i . 一例において、Xは64。
- i i . 一例において、Xは $M/2$ 。
- c . 一例において、BVyは、 $(BVDy \bmod N)$ に置き換えられてもよい。
- d . 代替的に、BVyを $((BVDy + Y) \bmod N) - Y$ で置き換えてもよく、Yは予め規定された値である。
- i . 一例において、Yは64。
- i i . 一例において、Yは $N/2$ 。
- 22 . BVDxおよびBVDyは、異なる正規化範囲を有してもよい。

【0098】

ブロックベクトルの妥当性検査

IBCバッファの幅および高さをWbufおよびHbufとする。ピクチャの左上隅に対して(X, Y)から始まるW×Hブロック(輝度ブロック、クロマブロック、CU、TU、4×4、2×2、または他のサブブロックであってもよい)の場合、ブロックベクトル(BVx, BVy)が有効であるかどうかを示すために、以下を適用してもよい。Wpic、Hpicを1つのピクチャの幅および高さとし、Wctu、Hctuを1つのCTUの幅および高さとする。関数floor(x)は、xより大きくない最大の整数を返す。関数isRec(x, y)は、サンプル(x, y)を再構成した場合、それを返す。

23 . いずれかの参照位置がピクチャ境界の外にあっても、ブロックベクトル(BVx, BVy)を有効であるとして設定してもよい。

a . 一例において、ブロックベクトルは、 $X + BVx < 0$ であっても有効であるとして設定してもよい。

b . 一例において、ブロックベクトルは、 $X + W + BVx > Wpic$ であっても、有効であるとして設定してもよい。

c . 一例において、 $Y + BVy < 0$ であっても、ブロックベクトルを有効であるとして設定してもよい。

d . 一例において、 $Y + H + BVy > Hpic$ であっても、ブロックベクトルを有効であるとして設定してもよい。

24 . 現在のCTU行の外側に参照位置が存在しても、ブロックベクトル(BVx, BVy)を有効であるとして設定してもよい。

a . 一例において、ブロックベクトルは、 $Y + BVy < \text{floor}(Y/Hctu) * Hctu$ であっても有効であるとして設定してもよい。

b . 一例において、ブロックベクトルは、 $Y + H + BVy \geq \text{floor}(Y/Hctu) * Hctu + Hctu$ であっても有効であるとして設定してもよい。

25 . ブロックベクトル(BVx, BVy)は、いずれかの参照位置が現在のCTUおよび左(n-1)CTUの外にあったとしても、有効であるとして設定してもよく、nは、IBCのための参照領域として使用できるCTU(現在のCTUを含む、または除く)の数である。

a . 一例において、ブロックベクトルは、 $X + BVx < \text{floor}(X/Wctu) * Wctu - (n-1) * Wctu$ であっても有効であるとして設定してもよい。

b . 一例において、ブロックベクトルは、 $X + W + BVx > \text{floor}(X/Wctu) * Wctu + Wctu$ であっても、有効であるとして設定してもよい。

26 . 特定のサンプルが再構成されていなくても、ブロックベクトル(BVx, BVy

10

20

30

40

50

)を有効として設定してもよい。

a. 一例において、 $\text{isRec}(X + BV_x, Y + BV_y)$ が false であっても、ブロックベクトルを有効であるとして設定してもよい。

b. 一例において、 $\text{isRec}(X + BV_x + W - 1, Y + BV_y)$ が false であっても、ブロックベクトルを有効であるとして設定してもよい。

c. 一例において、 $\text{isRec}(X + BV_x, Y + BV_y + H - 1)$ が false であっても、ブロックベクトルを有効であるとして設定してもよい。

d. 一例において、 $\text{isRec}(X + BV_x + W - 1, Y + BV_y + H - 1)$ が false であっても、ブロックベクトルを有効であるとして設定してもよい。

27. ブロックベクトル (BV_x, BV_y) は、1つのブロックがCTU行における第1のCTUのブロックでない場合、常に有効であるとして設定してもよい。

a. 代替的に、ブロックベクトルは、常に有効であるとして設定してもよい。

28. 以下の3つの条件をすべて満たす場合、ブロックベクトル (BV_x, BV_y) を常に有効であるとして設定してもよい。

・ $X + BV_x \geq 0$

・ $Y + BV_y \geq \text{floor}(Y / H_{ctu})$

・ $\text{isRec}(X + BV_x + W - 1, Y + BV_y + H - 1) == \text{true}$

a. 代替的に、1つのCTU行における第1のCTUのブロックについて、3つの条件をすべて満たす場合、ブロックベクトルを常に有効であるとして設定してもよい。

29. ブロックベクトル (BV_x, BV_y) が有効である場合、ブロックのサンプルコピーはブロックベクトルに基づいて行われてもよい。

a. 一例において、サンプル (X, Y) の予測は、 $((X + BV_x) \% W_{buf}, (Y + BV_y) \% H_{buf})$ から得ることができる。

【0099】

バッファ更新

30. 新しいピクチャまたはタイルを符号化するとき、バッファはリセットされてもよい。

a. 「リセット」という用語は、バッファが初期化されることを指すことができる。

b. 用語「リセット」は、バッファ内のすべてのサンプル／画素が所与の値（例えば、0または-1）に設定されることを指すことができる。

31. VPDUの符号化を終えたら、バッファはVPDUの再構成された値で更新されてもよい。

32. CTUの符号化を終えたら、バッファはCTUの再構成された値で更新されてもよい。

a. 一例において、バッファが満杯でない場合、バッファはCTUによってCTUを順次更新することができる。

b. 一例において、バッファが満杯である場合、最も古いCTUに対応するバッファ領域を更新する。

c. 一例において、 $M = mW$ 、 $N = H$ （ W 、 H はCTUのサイズであり、 M 、 N はバッファサイズである）であり、前回更新された領域が $(kW, 0)$ から始まる場合、更新する次の開始位置は、 $((k + 1)W \bmod M, 0)$ となる。

33. バッファは、各CTU行の最初にリセットされてもよい。

a. 代替的に、バッファは、各CTUの復号化の開始時にリセットされてもよい。

b. 代替的に、バッファは、1つのタイルを復号化する開始時にリセットされてもよい。

c. 代替的に、バッファは、1つのタイルグループ／ピクチャの復号化の開始時にリセットされてもよい。

34. (x, y) から始まるブロックの符号化を終えたら、 (x, y) から始まるバッファの対応する領域は、ブロックからの再構成によって更新される。

a. 一例において、 (x, y) は、CTUの左上隅に対する位置である。

35. ピクチャに対するブロックの符号化を終えたら、バッファの対応する領域は、ブロックからの再構成によって更新される。

a. 一例において、バッファにおける位置 $(x \bmod M, y \bmod N)$ の値は、ピクチャの左上隅に対する位置 (x, y) の再構成された画素値で更新されてもよい。

b. 一例において、バッファにおける位置 $(x \bmod M, y \bmod N)$ の値は、現在のタイルの左上隅に対する位置 (x, y) の再構成された画素値で更新されてもよい。

c. 一例において、バッファにおける位置 $(x \bmod M, y \bmod N)$ の値は、現在のCTU行の左上隅に対する位置 (x, y) の再構成された画素値で更新されてもよい。

d. 一例において、バッファ内の値は、ビット深度整列後の再構成された画素値で更新されてもよい。

36. (x, y) から始まるブロックの符号化を終えたら、 (x_b, y_b) から始まるバッファの対応する領域は、ブロックからの再構成によって更新され、 (x_b, y_b) および (x, y) は2つの異なる座標である。

a. 一例において、 (x, y) は、CTUの左上隅に関する位置であり、 (x_b, y_b) は、 $(x + \text{update_x}, y + \text{update_y})$ であり、 update_x および update_y は、バッファにおける更新可能

37. 上記の例では、1つのブロックの再構成された値は、フィルタ（例えば、非ブロック化フィルタ）を適用する前の再構成された値を示してもよい。

a. 代替的に、1つのブロックの再構成された値は、フィルタ（例えば、非ブロック化フィルタ）が適用された後の再構成された値を示してもよい。

38. バッファが再構成されたサンプルから更新されるとき、再構成されたサンプルは、記憶される前にまず修正されてもよく、例えば、サンプルのビット深度を変更してもよい。

a. 一例において、バッファは、バッファのビット深度へのビット深度整列後、再構成されたサンプル値で更新される。

b. 一例において、バッファ値は、値 $\{p + [1 \ll (b - 1)]\} \gg b$ に基づいて更新され、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値である。

c. 一例において、バッファ値は、 $\text{clip}(\{p + [1 \ll (b - 1)]\} \gg b, 0, (1 \ll \text{bitdepth}) - 1)$ の値に基づいて更新され、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値であり、ビット深度は、バッファビット深度である。

d. 一例において、バッファ値は、値 $\{p + [1 \ll (b - 1) - 1]\} \gg b$ に基づいて更新され、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値である。

e. 一例において、バッファ値は、 $\text{clip}(\{p + [1 \ll (b - 1) - 1]\} \gg b, 0, (1 \ll \text{bitdepth}) - 1)$ の値に基づいて更新され、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値であり、 bitdepth は、バッファビット深度である。

f. 一例において、バッファ値は、値 $p \gg b$ に基づいて更新される。

g. 一例において、バッファ値は、 $\text{clip}(p \gg b, 0, (1 \ll \text{bitdepth}) - 1)$ の値に基づいて更新され、 bitdepth は、バッファビット深度である。

h. 上記例において、 b は、ビット深度から入力サンプルビット深度を引いたものとして再構成されてもよい。

39. バッファサンプルを使用して予測を形成する場合、前処理を適用することができる。

a. 一例において、予測値は、 $p \ll b$ であり、 p は、バッファにおけるサンプル値であり、 b は、予め規定された値である。

10

20

30

40

50

b. 一例において、予測値は、 $\text{clip}(p \ll b, 0, 1 \ll \text{bitdepth})$ であり、 bitdepth は、再構成サンプルのビット深度である。

c. 一例において、予測値は、 $(p \ll b) + (1 \ll (\text{bitdepth} - 1))$ であり、 p は、バッファにおけるサンプル値であり、 b は、予め規定された値であり、 bitdepth は、再構成サンプルのビット深度である。

d. 上記例において、 b は、ビット深度から入力サンプルビット深度を引いたものとして再構成されてもよい。

40. バッファは所与の順番で更新されてもよい。

a. 一例において、バッファは順次更新されてもよい。

b. 一例において、バッファは、再構成されたブロックの順に基づいて更新されてもよい。

10

41. バッファが満杯である場合、バッファ内のサンプルを最新の再構成サンプルに置き換えることができる。

a. 一例において、サンプルは、先入れ先出し方式で更新されてもよい。

b. 一例において、最も古いサンプルが置き換えられる。

c. 一例において、サンプルに優先順位を割り当て、この優先順位に基づいてサンプルを置き換えることができる。

d. 一例において、サンプルは、他のサンプルが最初に置き換えられるように、「長期」としてマークされ得る。

e. 一例において、高い優先順位を示すために、1つのブロックとともに1つのフラグを送信することができる。

20

f. 一例において、優先順位を示すために、1つのブロックとともに1つの数字を送信することができる。

g. 一例において、ある特徴を有する再構成されたブロックからのサンプルは、他のサンプルが最初に置き換えられるように、より高い優先順位を割り当てられる。

i. 一例において、IBCモードで符号化されたサンプルのパーセンテージが閾値よりも大きい場合、ブロックのすべてのサンプルに高い優先順位を割り当てることができる。

ii. 一例において、Paletteモードで符号化されたサンプルのパーセンテージが閾値よりも大きい場合、ブロックのすべてのサンプルに高い優先順位を割り当てることができる。

30

iii. 一例において、IBCまたはPaletteモードで符号化されたサンプルのパーセンテージが閾値よりも大きい場合、ブロックのすべてのサンプルに高い優先順位を割り当てることができる。

iv. 一例において、変換スキップモードで符号化されたサンプルのパーセンテージが閾値よりも大きい場合、ブロックのすべてのサンプルに高い優先順位を割り当てることができる。

v. この閾値は、ブロックサイズ、色成分、CTUサイズに基づいて異なってもよい。

vi. 閾値は、SPS/PPS/シーケンスヘッダ/スライスヘッダ/タイルグループ/タイルレベル/領域で信号通知されてもよい。

40

h. 一例において、このバッファが一杯であることは、このバッファにおける利用可能なサンプルの数が所与の閾値に等しいかまたはそれよりも多いことを意味し得る。

i. 一例において、バッファにおける利用可能なサンプルの数が $64 \times 64 \times 3$ 輝度サンプル以上である場合、バッファは満杯であると判定されてもよい。

【0100】

代替バッファ組み合わせ

42. 前回符号化された3つの 64×64 ブロックを常に参照領域として使用する代わりに、現在のブロック（またはVPDU）の位置に基づいてそれを適応的に変更することが提案される。

50

a. 一例において、 64×64 ブロックを符号化／復号化するとき、前回の3つの 64×64 ブロックを参照として使用できる。図2に比べ、前回の 64×64 個のブロックをより多く組み合わせてもよい。図2は、前の 64×64 ブロックの異なる組み合わせの例を示す。

43. z走査順序を用いる代わりに、垂直走査順序を使用してもよい。

a. 一例において、1つのブロックがz走査順序にインデックス0...3を有する4つのVPDUに分割される場合、符号化／復号化の順序は、0、2、1、3である。

b. 一例において、 64×64 個のブロックを符号化／復号化するとき、前回の3つの 64×64 個のブロックを参照として使用できる。図2と比較して、より多くの種類の 64×64 ブロックの符号化／復号化の順序を適用することができる。図4は、 64×64 ブロックの異なる符号化／復号化の順序の例を示す。

c. 代替的に、上記方法は、スクリーンコンテンツの符号化にのみ適用されてもよい。

d. 代替的に、上記方法は、1つのタイル／タイルグループ／ピクチャに対してCPRが有効化される場合にのみ適用されてもよい。

e. 代替的に、上記方法は、1つのCTUまたは1つのCTU行に対してCPRが有効化される場合にのみ適用されてもよい。

【0101】

仮想IBCバッファ

以下、輝度サンプルにおいて、VPDUの幅および高さを、それぞれWVPDU（例えば、64）およびHVPDU（例えば、64）と表す。代替的に、WVPDUおよび／またはHVPDUは、他の映像ユニット（例えば、CTU）の幅および／または高さを表してもよい。

44. IBC参照領域の状態を追跡し続けるために、仮想バッファを維持してもよい。

a. 一例において、仮想バッファサイズは、 $mWVPDU \times nHVPDU$ である。

i. 一例において、mは3に等しく、nは2に等しい。

ii. 一例において、mおよび／またはnは、ピクチャ解像度、CTUサイズに依存してもよい。

iii. 一例において、mおよび／またはnは、信号通知されてもよく、または予め規定されてもよい。

b. 一例において、上記の黒丸印および副黒丸印で記載された方法は、仮想バッファに適用されてもよい。

c. 一例において、ピクチャ／スライス／タイル／ブリックの左上隅に対するサンプル(x, y)は、 $(x \% (mWVPDU), y \% (nHVPDU))$ にマッピングされてもよい。

45. アレイを使用して、仮想バッファに関連付けられた各サンプルの可用性を追跡してもよい。

a. 一例において、フラグは、仮想バッファにおけるサンプルに関連付けて、バッファにおけるサンプルがIBC参照として使用できるか否かを特定してもよい。

b. 一例において、輝度サンプルおよびクロマサンプルを含む各 4×4 ブロックは、1つのフラグを共有して、そのブロックに関連付けられたいずれかのサンプルがIBC参照として使用できるかどうかを示してもよい。

46. VPDUまたは映像ユニットの復号化を終えた後、仮想バッファに関連付けられた特定のサンプルは、IBC参照に利用不可能なものとしてマークされてもよい。

a. 一例において、どのサンプルが利用不可能であるかは、直前に復号化されたVPDUの位置に依存してもよい。

b. 1つのサンプルが利用不可能とマークされた場合、このサンプルからの予測は許可されない。

i. 代替的に、他の方法（例えば、デフォルト値を使用する）をさらに適用して、予測子を導出し、利用不可能なサンプルを置き換えてもよい。

47. 最近復号化されたVPDUの位置を記録し、仮想バッファに関連付けられたどの

サンプルが利用不可能であるとマークされているかを識別しやすくすることができる。

a. 一例において、VPDUを復号化する開始時に、最近復号化されたVPDUの位置に基づいて、仮想バッファに関連付けられたいくつかのサンプルに利用不可能なマークを付けることができる。

i. 一例において、直近に復号化されたVPDUのピクチャ／スライス／タイル／ブリック／他の映像処理ユニットの左上隅に対して($x_{PrevVPDU}$, $y_{PrevVPDU}$)を左上位置として表し、 $y_{PrevVPDU} \% (n_{HVPDU})$ が0に等しいと、特定の位置(x , y)は、利用不可能とマークを付けることができる。

1. 一例において、 x は、 $[x_{PrevVPDU} - 2W_{VPDU} + 2mW_{VPDU}] \% mW_{VPDU}$, $((x_{PrevVPDU} - 2W_{VPDU} + 2mW_{VPDU}) \% mW_{VPDU}) - 1 + W_{VPDU}$ のような範囲内にあってもよい。

10

2. 一例において、 y は、 $[y_{PrevVPDU} \% (n_{HVPDU})]$, $(y_{PrevVPDU} \% (n_{HVPDU})) - 1 + H_{VPDU}$ のような範囲内にあってもよい。

3. 一例において、 x は、 $[x_{PrevVPDU} - 2W_{VPDU} + 2mW_{VPDU}] \% mW_{VPDU}$, $((x_{PrevVPDU} - 2W_{VPDU} + 2mW_{VPDU}) \% mW_{VPDU}) - 1 + W_{VPDU}$ のような範囲内にあってもよく、 y は、 $[y_{PrevVPDU} \% (n_{HVPDU})]$, $(y_{PrevVPDU} \% (n_{HVPDU})) - 1 + H_{VPDU}$ のような範囲内にあってもよい。

ii. 一例において、直近に復号化されたVPDUのピクチャ／スライス／タイル／ブリック／他の映像処理ユニットの左上隅に対して、($x_{PrevVPDU}$, $y_{PrevVPDU}$)を左上位置として表し、 $y_{PrevVPDU} \% (n_{HVPDU})$ が0に等しくない場合、特定の位置(x , y)は、利用不可能とマークを付けることができる。

20

1. 一例において、 x は、 $[x_{PrevVPDU} - W_{VPDU} + 2mW_{VPDU}] \% mW_{VPDU}$, $((x_{PrevVPDU} - W_{VPDU} + 2mW_{VPDU}) \% mW_{VPDU}) - 1 + W_{VPDU}$ のような範囲内にあってもよい。

2. 一例において、 y は、 $[y_{PrevVPDU} \% (n_{HVPDU})]$, $(y_{PrevVPDU} \% (n_{HVPDU})) - 1 + H_{VPDU}$ のような範囲内にあってもよい。

3. 一例において、 x は、 $[x_{PrevVPDU} - W_{VPDU} + 2mW_{VPDU}] \% mW_{VPDU}$, $((x_{PrevVPDU} - W_{VPDU} + 2mW_{VPDU}) \% mW_{VPDU}) - 1 + W_{VPDU}$ のような範囲内にあってもよく、 y は、 $[y_{PrevVPDU} \% (n_{HVPDU})]$, $(y_{PrevVPDU} \% (n_{HVPDU})) - 1 + H_{VPDU}$ のような範囲内にあってもよい。

30

48. 1つのCUが複数のVPDUを含む場合、VPDUに基づくIBC参照可用性マーキング処理を適用する代わりに、IBC参照可用性マーキング処理は以下のCUに従ってもよい。

a. 一例において、複数のVPDUを含むCUを復号化する開始時に、CU内のVPDUを復号化する前に、各VPDUごとにIBC参照可用性マーキング処理を適用してもよい。

b. このような場合、 128×64 個のIBCブロックと 64×128 個のIBCブロックとは許可されないことがある。

40

i. 一例において、 128×64 および 64×128 個のCUのための $pred_mode_ibc_flag$ は送信されなくてもよく、0に等しいと推測されてもよい。

49. 参照ブロックまたはサブブロックの場合、右上隅の参照可用性ステータスをチェックして、この参照ブロックに関連付けられたブロックベクトルが有効であるかどうかを判断する必要がない場合がある。

a. 一例において、ブロックベクトルが有効であるか否かを判断して、ブロック／サブブロックの左上、左下、右下の角のみをチェックする。

50. IBCバッファサイズは、VPDUのサイズ(幅／高さは $vSize$ で表される)および／またはCTB／CTUのサイズ(幅／高さは $ctbSize$ で表される)に依

50

存してもよい。

a. 一例において、バッファの高さは $ctbSize$ に等しくてもよい。

b. 一例において、バッファの幅は、 $\min(ctbSize, 64)$ に依存してもよい。

i. 一例において、バッファの幅は、 $(128 * 128 / vSize, \min(ctbSize, 64))$ であってもよい。

51. IBCバッファは、画素範囲外の値を含んでもよく、このことは、この位置がIBC参照に利用可能でない（例えば、他のサンプルの予測に利用されない）可能性があることを示す。

a. サンプル値は、サンプルが利用不可能であることを示す値に設定してもよい。

b. 一例において、この値は、 -1 であってもよい。

c. 一例において、この値は、 $[0, 1 << (\text{internal_bit_depth}) - 1]$ の外側の任意の値であってもよく、 $\text{internal_bit_depth}$ は正の整数値である。

例えば、 $\text{internal_bit_depth}$ は、1つの色成分のサンプルを符号化／復号化するために使用される内部ビット深度である。

d. 一例において、この値は、 $[0, 1 << (\text{input_bit_depth}) - 1]$ の範囲外の任意の値であってもよく、 input_bit_depth は正の整数値である。

例えば、 input_bit_depth は、1つの色成分のサンプルを符号化／復号化するために使用される入力ビット深度である。

52. IBCバッファにおけるサンプルの可用性マーキングは、現在のブロックの位置、現在のブロックのサイズ、 CTU/CTB のサイズ、および $VPDU$ のサイズに依存してもよい。一例において、 (xCb, yCb) は、ピクチャの左上に対するブロックの位置を表し、 $ctbSize$ は、 CTU/CTB のサイズ（即ち、幅および／または高さ）であり、 $vSize = \min(ctbSize, 64)$ であり、 $wIbcBuf$ および $hIbcBuf$ は、IBCバッファの幅および高さである。

a. 一例において、 $(xCb \% vSize)$ が 0 に等しく、 $(yCb \% vSize)$ が 0 に等しい場合、IBCバッファにおける特定の位置のセットは、利用不可能とマークを付けることができる。

b. 一例において、現在のブロックのサイズが $VPDU$ のサイズ、即ち、 $\min(ctbSize, 64)$ より小さい場合、利用不可能とマークを付けられた領域は、 $VPDU$ のサイズに従ってもよい。

c. 一例において、現在のブロックのサイズが $VPDU$ のサイズより大きい場合、即ち、 $\min(ctbSize, 64)$ である場合、利用不可能とマークを付けられた領域は、 CU のサイズに従ってもよい。

53. ピクチャの左上位置に対する映像ユニット（例えば、 $VPDU(xV, yV)$ ）の復号化の開始時に、IBCバッファにおける対応する位置は、画素範囲外の値に設定してもよい。

a. 一例において、バッファにおける位置 $(x \% wIbcBuf, y \% hIbcBuf)$ を有し、 $x = xV, \dots, xV + ctbSize - 1$ 、 $y = yV, \dots, yV + ctbSize - 1$ を有するバッファサンプルは、値 -1 に設定される。 $wIbcBuf$ および $hIbcBuf$ は、IBCバッファの幅および高さであり、 $ctbSize$ は、 CTU/CTB の幅である。

i. 一例において、 $hIbcBuf$ は $ctbSize$ に等しくてもよい。

54. ビットストリーム適合性の制約は、IBCバッファにおけるサンプルの値に従ってもよい。

a. 一例において、IBCバッファにおける1つのブロックベクトルに関連付けられた1つの参照ブロックが画素範囲外の値を含む場合、このビットストリームは不正である可能性がある。

10

20

30

40

50

55. IBCバッファにおける可用性表示に基づいて、ビットストリーム適合性の制約を設定してもよい。

a. 一例において、IBCバッファにマッピングされたいずれかの参照サンプルが、1つのブロックを符号化／復号化するために利用不可能とマークを付けられた場合には、このビットストリームは不正である可能性がある。

b. 一例において、single treeが使用される場合、1つのブロックを符号化／復号化するためにIBCバッファにマッピングされたいずれかの輝度参照サンプルが利用不可能とマークを付けられた場合には、このビットストリームは不正である可能性がある。

c. 1つの適合ビットストリームは、IBC符号化ブロックの場合、関連付けられたブロックベクトルがIBCバッファにマッピングされた1つの参照ブロックを指してもよく、1つのブロックを符号化／復号化するためにIBCバッファに位置する各輝度参照サンプルには利用可能なものとしてマークを付けることを満たすことができる（例えば、サンプルの値は、 $[K0, K1]$ の範囲内にあり、例えば、 $K0$ が0に設定され、 $K1$ が $(1 < \text{BitDepth} - 1)$ に設定され、 BitDepth は、内部ビット深度または入力ビット深度である）。

10

56. ビットストリーム適合性の制約は、分割木のタイプおよび現在のCUの符号化のtreeTypeに依存してもよい。

a. 一例において、dual treeが高レベル（例えば、スライス／ピクチャ／ブリック／タイル）で許可され、現在の映像ブロック（例えば、CU／PU／CB／PB）がシングルツリーで符号化される場合、ビットストリーム制約は、IBCバッファにマッピングされたすべての成分の位置が利用不可能としてマークされているかどうかをチェックする必要がある場合もある。

20

b. 一例において、dual treeがハイレベル（例えば、スライス／ピクチャ／ブリック／タイル）で許可され、現在の輝度映像ブロック（例えば、CU／PU／CB／PB）がデュアルツリーで符号化されている場合には、ビットストリームの制約により、クロマ成分のIBCバッファ内にマッピングされた位置が利用不可能としてマークされているかどうかを無視してもよい。

i. 代替的に、このような場合、ビットストリーム制約は、依然として、IBCバッファにマッピングされたすべての成分の位置が利用不可能としてマークされているかどうかをチェックすることもある。

30

c. 一例において、シングルツリーが使用される場合、ビットストリーム制約は、IBCバッファにマッピングされたクロマ成分の位置が利用不可能であるか否かを無視することができる。

【0102】

現在のVTM設計の改善

57. IBCの予測は、再構成よりも低い精度を有していてもよい。

a. 一例において、予測値は、値 $\text{clip} \{ \{ p + [1 < (b - 1)] \} \gg b, 0, (1 < \text{bitdepth}) - 1 \} < b$ に基づいており、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値であり、 bitdepth は、予測サンプルビットのビット深度である。

40

b. 一例において、予測値は、値 $\text{clip} \{ \{ p + [1 < (b - 1) - 1] \} \gg b, 0, (1 < \text{bitdepth}) - 1 \} < b$ に基づいており、 p は、再構成されたサンプル値であり、 b は、予め規定されたビットシフト値である。

c. 一例において、予測値は、値 $((p \gg b) + (1 < (\text{bitdepth} - 1))) < b$ に基づいており、ビット深度は、予測サンプルのビット深度である。

d. 一例において、予測値は、値 $(\text{clip} ((p \gg b), 0, (1 < (\text{bitdepth} - b))) + (1 < (\text{bitdepth} - 1))) < b$ に基づいており、ビット深度は、予測サンプルのビット深度である。

e. 一例において、予測値は、ILRが適用されるかどうかに基づいて、異なる方法

50

でクリップされる。

f. 上記例において、bは、ビット深度から入力サンプルビット深度を引いたものとして再構成されてもよい。

g. 一例において、出力ビット深度／入力ビット深度／内部ビット深度と比較したビット深度またはビット深度の差は、SPS／PPS／シーケンスヘッダ／ピクチャヘッダ／スライスヘッダ／タイルグループヘッダ／タイルヘッダまたは他の種類の映像データユニットにおいて信号通知されてもよい。

58. IBCの予測の一部はより精度が低くてもよく、他の部分は再構成と同じ精度を有する。

a. 一例において、許容される参照領域は、異なる精度（例えば、ビット深度）を有するサンプルを含んでもよい。

b. 一例において、現在の 64×64 ブロックが復号化されていない他の 64×64 ブロックからの参照は、精度が低く、現在の 64×64 ブロックからの参照は、再構成と同じ精度を有する。

c. 一例において、復号化されている現在のCTU以外のCTUからの参照は精度が低く、現在のCTUからの参照は再構成と同じ精度を有する。

d. 一例において、特定の色成分の集まりからの参照は精度が低く、他の色成分からの参照は再構成と同じ精度を有する。

59. CTUのサイズが $M \times M$ であり、参照領域のサイズが $nM \times nM$ である場合、参照領域は、CTUの行における最も近い利用可能な $n \times n$ CTUである。

a. 一例において、参照領域のサイズが 128×128 であり、CTUのサイズが 64×64 である場合、1つのCTU行における最も近い利用可能な4つのCTUをIBC参照に使用できる。

b. 一例において、参照領域のサイズが 128×128 であり、CTUのサイズが 32×32 である場合、1つのCTU行における最も近い利用可能な16個のCTUをIBC参照に使用できる。

60. CTUのサイズが M であり、参照領域のサイズが nM である場合、参照領域は、CTUの行／タイルにおける最も近い利用可能な $n - 1$ 個のCTUである。

a. 一例において、参照領域のサイズが 128×128 または 256×64 であり、CTUのサイズが 64×64 である場合、1つのCTU行における最も近い利用可能な3つのCTUをIBC参照に使用できる。

b. 一例において、参照領域のサイズが 128×128 または 512×32 であり、CTUのサイズが 32×32 である場合、1つのCTU行における最も近い利用可能な15個のCTUをIBC参照に使用できる。

61. CTUのサイズが M である場合、VPDUのサイズは kM であり、参照領域のサイズは nM であり、参照領域はCTUの行／タイルにおける最も近い利用可能な $n - k$ 個のCTUである。

a. 一例において、CTUのサイズは 64×64 であり、VPDUのサイズも 64×64 であり、参照のサイズは 128×128 であり、1つのCTUの行における最も近い3つのCTUをIBC参照のために使用できる。

b. 一例において、CTUのサイズは 32×32 であり、VPDUのサイズは 64×64 であり、参照のサイズは 128×128 であり、1つのCTU行における最も近い $(16 - 4) = 12$ 個のCTUをIBC参照のために使用できる。

62. IBCを使用して、左上隅が (x, y) である $w \times h$ ブロックの場合、参照ブロックをメモリ再利用のための特定の領域から守る制約があり、 w および h は、現在のブロックの幅および高さである。

a. 一例において、CTUのサイズが 128×128 であり、 $(x, y) = (m \times 64, n \times 64)$ である場合、参照ブロックは、 $((m - 2) \times 64, n \times 64)$ から始まる 64×64 の領域と重複することはできない。

b. 一例において、CTUのサイズが 128×128 である場合、参照ブロックは、

10

20

30

40

50

$w \times h$ ブロックの左上隅が $(x - 128, y)$ である場合に重複することはできない。

c. 一例において、CTUのサイズが 128×128 である場合、 $(x + BVx, y + BVy)$ は、左上隅は $(x - 128, y)$ である $w \times h$ ブロック内に存在することはない、 BVx および BVy は、現在のブロックのブロックベクトルを表す。

d. 一例において、CTUのサイズが $M \times M$ であり、IBCバッファサイズが $k \times M \times M$ である場合、参照ブロックは、 $w \times h$ ブロックと重複することができず、左上隅は $(x - k \times M, y)$ であり、 BVx および BVy は、現在のブロックのブロックベクトルを表す。

e. 一例において、CTUのサイズが $M \times M$ であり、IBCバッファサイズが $k \times M \times M$ である場合、 $(x + BVx, y + BVy)$ は、 $w \times h$ イントラブロック内に存在することはない、左上隅は $(x - k \times M, y)$ であり、 BVx および BVy は、現在のブロックのブロックベクトルを表す。

10

63. CTUのサイズが $M \times M$ でなく、参照領域のサイズが $nM \times nM$ である場合、参照領域は、CTUの行における最も近い利用可能な $n \times n - 1$ 個のCTUである。

a. 一例において、参照領域のサイズが 128×128 であり、CTUのサイズが 64×64 である場合、1つのCTU行における最も近い利用可能な3つのCTUをIBC参照に使用できる。

b. 一例において、参照領域のサイズが 128×128 であり、CTUのサイズが 32×32 である場合、1つのCTU行における最も近い利用可能な15個のCTUをIBC参照に使用できる。

20

64. $(2m \times 64, 2n \times 64)$ から始まる 64×64 のブロックのCU、即ち 128×128 CTUにおける左上の 64×64 ブロックの場合、そのIBC予測は、 $((2m - 2) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $((2m - 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、および $((2m - 1) \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロック、および現在の 64×64 ブロックにおける再構成サンプルから行うことができる。

65. $((2m + 1) \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロックのCU、すなわち、 128×128 CTUにおける右下の 64×64 ブロックの場合、そのIBC予測は、現在の 128×128 CTUからであってもよい。

66. $((2m + 1) \times 64, 2n \times 64)$ から始まる 64×64 のブロックのCU、即ち 128×128 CTUにおける右上の 64×64 ブロックの場合、そのIBC予測は、 $((2m - 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $((2m - 1) \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロック、および $(2m \times 64, 2n \times 64)$ から始まる 64×64 ブロック、および現在の 64×64 ブロックにおける再構成サンプルから行うことができる。

30

a. 代替的に、 $(2m \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロックを再構成した場合、IBC予測は、 $((2m - 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $(2m \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $(2m \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロック、および現在の 64×64 ブロックにおける再構成サンプルから行うことができる。

40

67. $(2m \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロックのCU、即ち 128×128 CTUにおける左下の 64×64 ブロックの場合、そのIBC予測は、 $((2m - 1) \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロック、 $(2m \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $((2m + 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、および現在の 64×64 ブロックにおける再構成サンプルから行うことができる。

a. 代替的に、 $((2m + 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロックを再構成されていない場合、IBC予測は、 $((2m - 1) \times 64, 2n \times 64)$ から始まる 64×64 ブロック、 $((2m - 1) \times 64, (2n + 1) \times 64)$ から始まる 64×64 ブロック、 $(2m \times 64, 2n \times 64)$ から始まる 64×64 ブロック、および現

50

在の 64×64 ブロックにおける再構成サンプルから行うことができる。

68. 現在のCUがどの 64×64 ブロックに属するかに基づいて、参照領域を調整することが提案される。

a. 一例において、 (x, y) から始まるCUの場合、 $(y \gg 6) \& 1 == 0$ のとき、 $((x \gg 6 < 6) - 128, y \gg 6 < 6)$ および $((x \gg 6 < 6) - 64, y \gg 6 < 6)$ から始まる2つまたは2つ前までの 64×64 ブロックは、IBCモードによって参照することができる。

b. 一例において、 (x, y) から始まるCUの場合、 $(y \gg 6) \& 1 == 1$ であるとき、 $((x \gg 6 < 6) - 64, y \gg 6 < 6)$ から始まる1つの前の 64×64 ブロックは、IBCモードによって参照することができる。

69. (x, y) から始まり、ブロックベクトル (BVx, BVy) を有するブロックの場合、 $isRec((x + BVx) \gg 6 < 6) + 128 - ((y + BVy) \gg 6) \& 1 * 64 + (x \% 64)$, $((y + BVy) \gg 6 < 6) + (y \% 64)$ が `true` ならば、ブロックベクトルは無効である。

a. 一例において、ブロックは輝度ブロックである。

b. 一例において、このブロックは、4:4:4フォーマットではクロマブロックである。

c. 一例において、このブロックは、輝度成分およびクロマ成分の両方を含む。

70. (x, y) から始まり、ブロックベクトル (BVx, BVy) を有する4:2:0形式のクロマブロックの場合、 $isRec((x + BVx) \gg 5 < 5) + 64 - ((y + BVy) \gg 5) \& 1 * 32 + (x \% 32)$, $((y + BVy) \gg 5 < 5) + (y \% 32)$ が `true` ならば、ブロックベクトルは無効である。

71. 成分cのブロックに対してBVが無効であるかどうかの判定は、輝度サンプルのみをチェックする代わりに、成分Xのサンプルの可用性に依存してもよい。

a. (x, y) から始まり、ブロックベクトル (BVx, BVy) を有する成分cのブロックの場合、 $isRec(c, (x + BVx) \gg 6 < 6) + 128 - ((y + BVy) \gg 6) \& 1 * 64 + (x \% 64)$, $((y + BVy) \gg 6 < 6) + (y \% 64)$ が `true` であれば、ブロックベクトルは無効として扱ってもよい。

i. 一例において、このブロックは、輝度ブロックである（例えば、cは、輝度成分であるか、またはRGB符号化のためのG成分である）。

ii. 一例において、このブロックは、4:4:4フォーマットのクロマブロックである（例えば、cは、cbまたはcr成分、またはRGB符号化のためのB/R成分である）。

iii. 一例において、例えば、このブロックは、輝度およびクロマ成分の両方を含むなど、輝度およびクロマ成分の両方のためのサンプルの可用性がチェックしてもよい。

b. 成分cの (x, y) から始まり、ブロックベクトル (BVx, BVy) を有する4:2:0形式のクロマブロックの場合、 $isRec(c, (x + BVx) \gg 5 < 5) + 64 - ((y + BVy) \gg 5) \& 1 * 32 + (x \% 32)$, $((y + BVy) \gg 5 < 5) + (y \% 32)$ が `true` である場合、ブロックベクトルは、無効として扱ってもよい。

c. 成分cの (x, y) から始まり、ブロックベクトル (BVx, BVy) を有するクロマブロックまたはサブブロックの場合、クロマ成分のための $isRec(c, x + BVx + Chroma_CTU_size, y)$ が `true` である場合、ブロックベクトルは無効として扱ってもよく、 $Chroma_CTU_size$ は、クロマ成分のCTUサイズである。

i. 一例において、4:2:0フォーマットの場合、 $Chroma_CTU_size$ は64であってもよい。

ii. 一例において、クロマサブブロックは、4:2:0フォーマットの 2×2 ブロックであってもよい。

iii. 一例において、クロマサブブロックは、4:4:4フォーマットの 4×4

10

20

30

40

50

ブロックであってもよい。

i v. 一例において、クロマサブブロックは、輝度成分における最小CUサイズに対応しうる。

1. 代替的に、クロマサブブロックは、クロマ成分の最小CUサイズに対応してもよい。

72. 上述したすべての黒丸印について、参照バッファは複数の $M \times M$ ブロック ($M = 64$) を含むと仮定する。しかしながら、これは、参照バッファが複数の $N \times M$ 個のブロックを含む (例えば、 $N = 128$ 、 $M = 64$) 等の他の場合にも拡張できる。

73. 上述したすべての黒丸印に対して、参照バッファが現在のブロックと同じブリック／タイル／タイルグループ／スライス内にあるべきであるという、さらなる制限を適用してもよい。

10

a. 一例において、参照バッファの一部が現在のブリック／タイル／タイルグループ／スライスの外にある場合、IBCの使用は無効にされ得る。IBC関連構文要素の信号通知はスキップしてもよい。

b. 代替的に、参照バッファの一部が現在のブリック／タイル／タイルグループ／スライスの外にある場合、IBCは1つのブロックに対して依然として有効化される可能性があるが、1つのブロックに関連付けられたブロックベクトルは残りの参照バッファのみを指してもよい。

74. IBCの参照領域として、現在のVPDUを除き、利用可能であれば、CTU／CTB行の第1のVPDU行において、K1個の最近符号化VPDUを有し、可能であれば、CTU／CTB行の第2のVPDU行において、K2個の最近符号化VPDUを有することが提案される。

20

a. 一例において、K1は2に等しく、K2は1に等しい。

b. 一例において、上記方法は、CTU／CTBサイズが 128×128 であり、VPDUサイズが 64×64 である場合に適用されてもよい。

c. 一例において、上記方法は、CTU／CTBサイズが 64×64 であり、VPDUサイズが 64×64 および／または 32×32 である場合に適用されてもよい。

d. 一例において、上記方法は、CTU／CTBサイズが 32×32 であり、VPDUサイズが 32×32 以下である場合に適用されてもよい。

75. 上述した方法は、異なる段階で適用されてもよい。

30

a. 一例において、ブロックベクトル (BV) のモジュール演算 (例えば、mod b) を、BVの可用性チェック処理において呼び出して、BVが有効であるか否かを判定してもよい。

b. 一例において、ブロックベクトル (BV) のモジュール演算 (例えば、mod b) を呼び出して、IBC仮想バッファまたは再構成ピクチャバッファにおける参照サンプルの位置 (例えば、現在のサンプルの位置およびBVのモジュール結果に基づいて) を特定してもよい (例えば、インループフィルタリング処理の前)。

【0103】

5. 実施形態

【0104】

40

5.1 実施形態 # 1

【0105】

IBCのためのバッファの1つの実装形態を以下に説明する。

【0106】

バッファサイズは 128×128 である。CTUのサイズも 128×128 である。1つのCTU行における第1のCTUの符号化のために、バッファは128 (8ビットの映像信号の場合) で初期化される。CTU行におけるk番目のCTUを符号化するために、バッファは、(k-1) 番目のCTUのループフィルタリングの前に、再構成で初期化される。

【0107】

50

図3は、 (x, y) から始まるブロックの符号化を例示する。

【0108】

現在のCTUに関する (x, y) から始まるブロックを符号化する場合、ブロックベクトル $(BVx, BVy) = (x - x_0, y - y_0)$ をデコーダに送信し、参照ブロックがIBCバッファにおける (x_0, y_0) からであることを示す。それぞれ、ブロックの幅を w とし、ブロックの高さを h とする。ブロックの符号化を終えるとき、IBCバッファにおける (x, y) から始まる $w \times h$ 領域は、ループフィルタリング前のブロックの再構成によって更新される。

【0109】

5.2 実施形態#2

10

【0110】

図4は、前回符号化された 64×64 ブロックを選択するための、考えられる代替方法を例示する。

【0111】

5.3 実施形態#3

【0112】

図5は、 64×64 ブロックの符号化／復号化の順序を変更することができる代替方法を例示する。

【0113】

5.4 実施形態#4

20

【0114】

図8は、 64×64 ブロックの復号化の順序が上から下へ、左から右である場合、前回の符号化された 64×64 ブロックを選択する別の考えられる代替方法を示す。

【0115】

5.5 実施形態#5

【0116】

図9は、前回の符号化された 64×64 ブロックを選択する別の考えられる代替方法を示す。

【0117】

5.6 実施形態#6

30

【0118】

図11は、 64×64 ブロックの復号化の順序が左から右、上から下である場合、前回の符号化された 64×64 ブロックを選択するための別の考えられる代替方法を示す。

【0119】

5.7 実施形態#7

【0120】

CTUのサイズが $W \times W$ であり、デコーダにおいて、サイズが $mW \times W$ であり、ビット深度が B であるIBCバッファを実装すると、以下のようになる。

【0121】

CTU行の復号化の開始時に、バッファを値 $(1 \ll (B - 1))$ で初期化し、更新の開始点 (x_b, y_b) を $(0, 0)$ に設定する。

40

【0122】

CTUの左上隅に関する、サイズが $w \times h$ である (x, y) から始まるCUを復号化する場合、ビット深度を B ビットに整列させた後、 $(x_b + x, y_b + y)$ および $w \times h$ サイズから始まる領域を、CUの再構成された画素値で更新する。

【0123】

CTUを復号化した後、更新開始点 (x_b, y_b) を $((x_b + W) \bmod mW, 0)$ と設定する。

【0124】

ブロックベクトル (BVx, BVy) を有するIBC CUを復号化するとき、CTU

50

の左上隅に関する任意の画素 (x, y) について、予測信号のビット深度にビット深度整理した後、バッファの位置 $((x + BVx) \bmod mW, (y + BVy) \bmod eW)$ からその予測を抽出する。

【0125】

一例において、 B は 7 または 8 に設定され、一方、ブロックの出力／入力ビット深度は 10 に等しくてもよい。

【0126】

5.8 実施形態 # 8

【0127】

ピクチャの左上隅に関する (x, y) およびブロックベクトル (BVx, BVy) から始まる輝度CUまたは結合輝度／クロマCUの場合、ブロックベクトルは、 $isRec((x + BVx) >> 6 << 6) + 128 - ((y + BVy) >> 6) \& 1 * 64 + (x \% 64), ((y + BVy) >> 6 << 6) + (y \% 64)$ が `true` の場合は無効である。

【0128】

ピクチャの左上隅に関する (x, y) およびブロックベクトル (BVx, BVy) から始まるクロマCUの場合、ブロックベクトルは、 $isRec(((x + BVx) >> 5 << 5) + 64 - ((y + BVy) >> 5) \& 1 * 32 + (x \% 32), ((y + BVy) >> 5 << 5) + (y \% 32))$ が `true` の場合は無効である。

【0129】

5.9 実施形態 # 9

【0130】

ピクチャの左上隅に関する 4:2:0 形式の (x, y) から始まるクロマブロックまたはサブブロック、およびブロックベクトル (BVx, BVy) の場合、 $isRec(c, (x + BVx + 64, y + BVy))$ が `true` で、 c がクロマ成分であるとき、ブロックベクトルは無効である。

【0131】

ピクチャの左上隅に関する、4:4:4 形式の (x, y) から始まるクロマブロックまたはサブブロック、およびブロックベクトル (BVx, BVy) の場合、 $isRec(c, (x + BVx + 64, y + BVy))$ が `true` で、 c がクロマ成分であるとき、ブロックベクトルは無効である。

【0132】

5.10 実施形態 # 10

【0133】

ピクチャの左上隅に関する (x, y) およびブロックベクトル (BVx, BVy) から始まる輝度CUまたは結合輝度／クロマCUの場合、ブロックベクトルは、 $isRec((x + BVx) >> 6 << 6) + 128 - ((y + BVy) >> 6) \& 1 * 64 + (x \% 64), ((y + BVy) >> 6 << 6) + (y \% 64)$ が `true` の場合は無効である。

【0134】

ピクチャの左上隅に関する、4:2:0 形式の (x, y) から始まるクロマブロックまたはサブブロック、およびブロックベクトル (BVx, BVy) の場合、 $isRec(c, ((x + BVx) >> 5 << 5) + 64 - ((y + BVy) >> 5) \& 1 * 32 + (x \% 32), ((y + BVy) >> 5 << 5) + (y \% 32))$ が `true` で、 c がクロマ成分であるとき、ブロックベクトルは無効である。

【0135】

5.11 実施形態 # 11

【0136】

本実施形態においては、現在のVPDUを除き、第1のVPDUの行にある2つの最も符号化されたVPDUと、第1のCTU／CTBの行の第2のVPDUの行にある1つの

10

20

30

40

50

最も符号化されたVPDUとを保持する実装が強調される。

【0137】

符号化順が上から下へ、左から右への場合、参照領域は図13のように示される。

【0138】

VPDUの符号化順が左から右および上から下であり、かつ、現在のVPDUがピクチャ境界の右側でない場合、参照領域は、図14のように示される。

【0139】

VPDUの符号化順が左から右および上から下であり、かつ現在のVPDUがピクチャ境界の右側にある場合、参照領域を図15として示すことができる。

【0140】

輝度ブロック(x, y)のサイズがw×hだとすると、ブロックベクトル(BVx, BVy)は有効であるか、どうかは以下の条件をチェックすることによって知ることができる。

【0141】

$$isRec((x + BVx + 128) >> 6 << 6) - (refy \& 0x40) + (x \% 64), ((y + BVy) >> 6 << 6) + (refy >> 6 == y >> 6) ? (y \% 64) : 0$$
。ここでは、 $refy = (y \& 0x40) ? (y + BVy) : (y + BVy + w - 1)$ 。

【0142】

上記関数がtrueを返す場合、ブロックベクトル(BVx, BVy)は無効であり、そうでない場合、ブロックベクトルは有効であり得る。

【0143】

5.12 実施形態#12

【0144】

CTUのサイズが192×128である場合、サイズが192×128の仮想バッファが維持されてIBCの参照サンプルを追跡する。

【0145】

ピクチャの左上隅に対するサンプル(x, y)は、バッファの左上隅に対する位置(x%192, y%128)に関連付けられる。以下のステップは、IBC参照のために仮想バッファに関連付けられたサンプルの可用性をどのようにマークするかを示す。

【0146】

ピクチャの左上隅に対する位置(xPrevVPDU, yPrevVPDU)を記録して、直近に復号化されたVPDUの左上のサンプルを表す。

1) 1つのVPDU行の復号化の開始時に、バッファのすべての位置が利用不可能とマークされる。(xPrevVPDU, yPrevVPDU)は、(0, 0)として設定される。

2) VPDUの第1のCUの復号化開始時に、 $x = (xPrevVPDU - 2WVPDU + 2mWVPDU) \% (mWVPDU)$, ..., $((xPrevVPDU - 2WVPDU + 2mWVPDU) \% (mWVPDU)) - 1 + WVPDU$; and $y = yPrevVPDU \% (nHVPDU)$, ..., $(yPrevVPDU \% (nHVPDU)) - 1 + HVPDU$ の場合の位置(x, y)を利用不可能としてマークしてもよい。そして、(xPrevVPDU, yPrevVPDU)を(xCU, yCU)として、即ち、CUのピクチャに対する左上位置に設定する。

3) 1つのCUを復号化した後、 $x = xCU \% (mWVPDU)$, ..., $(xCU + CU_width - 1) \% (mWVPDU)$ および $y = yCU \% (nHVPDU)$, ..., $(yCU + CU_height - 1) \% (nHVPDU)$ の場合の位置(x, y)は、利用可能としてマークされる。

4) ブロックベクトル(xBV, yBV)を有するIBCCUの場合、 $x = (xCU + xBV) \% (mWVPDU)$, ..., $(xCU + xBV + CU_width - 1) \% (mWVPDU)$ および $y = (yCU + yBV) \% (nHVPDU)$, ..., $(yCU +$

10

20

30

40

50

$yBV + CU_height - 1) \% (nHVPDU)$ の場合の位置 (x, y) は、利用不可能としてマークされ、ブロックベクトルは、無効であると考えられる。

【0147】

図16は、ピクチャにおけるバッファ状態をVPDUの復号化状態とともに示す。

【0148】

5.13 実施形態#13

【0149】

CTUのサイズが 128×128 であるか、またはCTUのサイズがVPDUのサイズよりも大きい（例えば、現在の設計においては、 64×64 ）場合、またはCTUのサイズがVPDUのサイズよりも大きい（例えば、現在の設計においては、 64×64 ）場合には、サイズ 192×128 の仮想バッファを維持してIBCのための参照サンプルを追跡する。以下、 $a < 0$ の場合、 $(a \% b)$ を $\text{floor}(a/b) * b$ として定義し、 $\text{floor}(c)$ は、 c 以下の最大の整数を返す。

【0150】

ピクチャの左上隅に対するサンプル (x, y) は、バッファの左上隅に対する位置 $(x \% 192, y \% 128)$ に関連付けられる。以下のステップは、IBC参照のために仮想バッファに関連付けられたサンプルの可用性をどのようにマークするかを示す。

【0151】

ピクチャの左上隅に対する位置 $(xPrevVPDU, yPrevVPDU)$ を記録して、直近に復号化されたVPDUの左上のサンプルを表す。

1) 1つのVPDU行の復号化の開始時に、バッファのすべての位置が利用不可能とマークされる。 $(xPrevVPDU, yPrevVPDU)$ は、 $(0, 0)$ として設定される。

2) VPDUの第1のCUの復号化の始めに、

a. $yPrevVPDU \% 64$ が0に等しい場合、 $x = (xPrevVPDU - 128) \% 192, \dots, ((xPrevVPDU - 128) \% 192) + 63$; および $y = yPrevVPDU \% 128, \dots, (yPrevVPDU \% 128) + 63$ の位置 (x, y) は、利用不可能としてマークされる。そして、 $(xPrevVPDU, yPrevVPDU)$ を (xCU, yCU) として、即ち、CUのピクチャに対する左上位置に設定する。

b. そうでない場合、 $x = (xPrevVPDU - 64) \% 192, \dots, ((xPrevVPDU - 64) \% 192) + 63$; および $y = yPrevVPDU \% 128, \dots, (yPrevVPDU \% 128) + 63$ の位置 (x, y) は、利用不可能としてマークされる。そして、 $(xPrevVPDU, yPrevVPDU)$ を (xCU, yCU) として、即ち、CUのピクチャに対する左上位置に設定する。

3) 1つのCUを復号化した後、 $x = xCU \% 192, \dots, (xCU + CU_width - 1) \% 192$ および $y = yCU \% 128, \dots, (yCU + CU_height - 1) \% 128$ の場合の位置 (x, y) は、利用可能としてマークされる。

4) ブロックベクトル (xBV, yBV) を有するIBCCUの場合、 $x = (xCU + xBV) \% 192, \dots, (xCU + xBV + CU_width - 1) \% 192$ および $y = (yCU + yBV) \% 128, \dots, (yCU + yBV + CU_height - 1) \% 128$ の場合の位置 (x, y) は、利用不可能としてマークされ、ブロックベクトルは、無効であると考えられる。

【0152】

CTUのサイズが $S \times S$ である場合、 S が 128 に等しくないので、 $Wbuf$ を $128 * 128 / S$ に等しくする。サイズ $Wbuf \times S$ の仮想バッファが維持して、IBCの参照サンプルを追跡する。この場合、VPDUのサイズはCTUのサイズに等しい。

【0153】

ピクチャの左上隅に対する位置 $(xPrevVPDU, yPrevVPDU)$ を記録して、直近に復号化されたVPDUの左上のサンプルを表す。

1) 1つのVPDU行の復号化の開始時に、バッファのすべての位置が利用不可能とマークされる。 $(x_{PrevVPDU}, y_{PrevVPDU})$ は、 $(0, 0)$ として設定される。

2) VPDUの第1のCUの復号化の開始時に、 $x = (x_{PrevVPDU} - W_{buf} * S) \% S, \dots, ((x_{PrevVPDU} - W_{buf} * S) \% S) + S - 1$; and $y = y_{PrevVPDU} \% S, \dots, (y_{PrevVPDU} \% S) + S - 1$ の場合の位置 (x, y) は、利用不可能であるとマークされる。そして、 $(x_{PrevVPDU}, y_{PrevVPDU})$ を (x_{CU}, y_{CU}) として、即ち、CUのピクチャに対する左上位置に設定する。

3) 1つのCUを復号化した後、 $x = x_{CU} \% (W_{buf}), \dots, (x_{CU} + CU_width - 1) \% (W_{buf})$ および $y = y_{CU} \% S, \dots, (y_{CU} + CU_height - 1) \% S$ である場合の位置 (x, y) は、利用可能としてマークされる。

4) ブロックベクトル (x_{BV}, y_{BV}) を有するIBCCUの場合、 $x = (x_{CU} + x_{BV}) \% (W_{buf}), \dots, (x_{CU} + x_{BV} + CU_width - 1) \% (W_{buf})$ および $y = (y_{CU} + y_{BV}) \% S, \dots, (y_{CU} + y_{BV} + CU_height - 1) \% S$ の場合の位置 (x, y) は、利用不可能としてマークされ、ブロックベクトルは、無効であると考えられる。

【0154】

5.14 実施形態#14

【0155】

CTUのサイズが 128×128 であるか、またはCTUのサイズがVPDUのサイズよりも大きい（例えば、現在の設計においては、 64×64 ）場合、またはCTUのサイズがVPDUのサイズよりも大きい（例えば、現在の設計においては、 64×64 ）場合には、サイズ 256×128 の仮想バッファを維持してIBCのための参照サンプルを追跡する。以下、 $a < 0$ の場合、 $(a \% b)$ を $\text{floor}(a / b) * b$ として定義し、 $\text{floor}(c)$ は、 c 以下の最大の整数を返す。

【0156】

ピクチャの左上隅に対するサンプル (x, y) は、バッファの左上隅に対する位置 $(x \% 256, y \% 128)$ に関連付けられる。以下のステップは、IBC参照のために仮想バッファに関連付けられたサンプルの可用性をどのようにマークするかを示す。

【0157】

ピクチャの左上隅に対する位置 $(x_{PrevVPDU}, y_{PrevVPDU})$ を記録して、直近に復号化されたVPDUの左上のサンプルを表す。

1) 1つのVPDU行の復号化の開始時に、バッファのすべての位置が利用不可能とマークされる。 $(x_{PrevVPDU}, y_{PrevVPDU})$ は、 $(0, 0)$ として設定される。

2) VPDUの第1のCUの復号化の始めに、

a. $y_{PrevVPDU} \% 64$ が0に等しい場合、 $x = (x_{PrevVPDU} - 128) \% 256, \dots, ((x_{PrevVPDU} - 128) \% 256) + 63$; および $y = y_{PrevVPDU} \% 128, \dots, (y_{PrevVPDU} \% 128) + 63$ の位置 (x, y) は、利用不可能としてマークされる。そして、 $(x_{PrevVPDU}, y_{PrevVPDU})$ を (x_{CU}, y_{CU}) として、即ち、CUのピクチャに対する左上位置に設定する。

b. そうでない場合、 $x = (x_{PrevVPDU} - 64) \% 256, \dots, ((x_{PrevVPDU} - 64) \% 256) + 63$; および $y = y_{PrevVPDU} \% 128, \dots, (y_{PrevVPDU} \% 128) + 63$ の位置 (x, y) は、利用不可能としてマークされる。そして、 $(x_{PrevVPDU}, y_{PrevVPDU})$ を (x_{CU}, y_{CU}) として、即ち、CUのピクチャに対する左上位置に設定する。

3) 1つのCUを復号化した後、 $x = x_{CU} \% 256, \dots, (x_{CU} + CU_width - 1) \% 256$ および $y = y_{CU} \% 128, \dots, (y_{CU} + CU_height - 1) \% 128$

10

20

30

40

50

$ht - 1) \% 128$ の場合の位置 (x, y) は、利用可能としてマークされる。

4) ブロックベクトル (xBV, yBV) を有する $IBCCU$ の場合、 $x = (xCU + xBV) \% 256, \dots, (xCU + xBV + CU_width - 1) \% 256$ および $y = (yCU + yBV) \% 128, \dots, (yCU + yBV + CU_height - 1) \% 128$ の場合の位置 (x, y) は、利用不可能としてマークされ、ブロックベクトルは、無効であると考えられる。

【0158】

CTU のサイズが 128×128 でない、または 64×64 未満である、または 64×64 未満である場合、前の実施例、すなわち、実施形態 # 14 と同じ処理が適用される。

【0159】

5. 15 実施形態 # 15

【0160】

IBC 参照可用性マーキング処理を以下のように説明する。この文書では、その変更を太字、下線、イタリック体で示している。

【0161】

【表 4】

7. 3. 7. 1 一般スライスデータ構文

slice_data() {	ディスクリ プタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u><i>xPrevVPDU = 0</i></u>	
<u><i>yPrevVPDU = 0</i></u>	
<u><i>if(CtbSizeY == 128)</i></u>	
<u><i>reset_ibc_isDecoded(0, 0, 256, CtbSizeY, BufWidth,</i></u>	
<u><i>BufHeight)</i></u>	
<u><i>else</i></u>	
<u><i>reset_ibc_isDecoded(0, 0, 128*128/CtbSizeY, CtbSizeY,</i></u>	
<u><i>BufWidth, BufHeight)</i></u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
.....	

【0162】

【表 5】

<u><i>reset_ibc_isDecoded(x0, y0, w, h, BufWidth, BufHeight) {</i></u>	ディスクリ プタ
<u><i>if(x0 >= 0)</i></u>	
<u><i>for (x = x0 % BufWidth; x < x0 + w; x+=4)</i></u>	
<u><i>for (y = y0 % BufHeight; y < y0 + h; y+=4)</i></u>	
<u><i>isDecoded[x >> 2][y >> 2] = 0</i></u>	
<u><i>}</i></u>	

$BufWidth$ は $(CtbSizeY == 128) ? 256 : (128 * 128 / C$

tbSizeY) に等しく、**BufHeight**は**CtbSizeY**に等しい。

【0163】

7.3.7.5 符号化ユニット構文

【0164】

【表6】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリ プタ	
<u>if(treeType != DUAL_TREE_CHROMA && (CtbSizeY == 128) && (x0 % 64)</u> <u>== 0 && (y0 % 64) == 0) {</u>		
<u>for(x = x0; x < x0 + cbWidth; x += 64)</u>		10
<u>for(y = y0; y < y0 + cbHeight; y += 64)</u>		
<u>if((yPrevVPDU % 64) == 0)</u>		
<u>reset_ibc_isDecoded(xPrevVPDU - 128, yPrevVPDU, 64,</u> <u>64, BufWidth, BufHeight)</u>		
<u>else</u>		
<u>reset_ibc_isDecoded(xPrevVPDU - 64, yPrevVPDU, 64,</u> <u>64, BufWidth, BufHeight)</u>		
<u>xPrevVPDU = x0</u>		
<u>yPrevVPDU = y0</u>		
<u>}</u>		
<u>if(treeType != DUAL_TREE_CHROMA && (CtbSizeY < 128) &&</u> <u>(x0 % CtbSizeY) == 0 && (y0 % CtbSizeY) == 0) {</u>		20
<u>reset_ibc_isDecoded(xPrevVPDU - (128*128/CtbSizeY -</u> <u>CtbSizeY), yPrevVPDU, 64, 64, BufWidth, BufHeight)</u>		
<u>xPrevVPDU = x0</u>		
<u>yPrevVPDU = y0</u>		
<u>}</u>		
if(slice_type != I sps_ibc_enabled_flag) {		
if(treeType != DUAL_TREE_CHROMA && !(cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag))		
cu_skip_flag[x0][y0]	ae(v)	30
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4))		
pred_mode_flag	ae(v)	
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && sps_ibc_enabled_flag && (cbWidth != 128 && cbHeight != 128))		
pred_mode_ibc_flag	ae(v)	40
}		
...		

【0165】

8.6.2 IBCブロックの動きベクトル成分の導出処理

【0166】

8.6.2.1 一般

...

8.6.3.2 項のブロックベクトルmvLでブロックベクトル妥当性検査処理が呼び出

10

20

30

40

50

される時に、`isBVvalid`が`true`であることが、ビットストリーム共同性能の要件である。

...

【0167】

8.6.3 `ibc`ブロックの復号化処理

【0168】

8.6.3.1 一般

【0169】

この処理は、`ibc`予測モードで符号化された符号化ユニットを復号化するとき呼び出される。

【0170】

この処理への入力は以下の通りである。

- 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符号化ブロックの左上のサンプルを規定する輝度位置 (`xCb`, `yCb`)、
- ー 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 `cbWidth`、
- ー 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 `cbHeight`、
- ー 水平および垂直方向の輝度符号化サブブロックの数を規定する変数 `numSbX` および `numSbY`、
- ー 動きベクトル `mv[xSbIdx][ySbIdx]` with `xSbIdx = 0...numSbX-1`、および `ySbIdx = 0...numSbY-1`、
- ー 現在のブロックの色成分インデックスを規定する変数 `cIdx`。
- ー `a (nIbcBufW) × (ctbSize) array ibcBuf`

...

【0171】

`xSbIdx = 0...numSbX-1`、および `ySbIdx = 0...numSbY-1` である場合のサブブロックインデックス (`xSbIdx`, `ySbIdx`) の各コーディングサブブロックに対して、以下が適用される。

- 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符号化サブブロックの左上のサンプルを規定する輝度位置 (`xSb`, `ySb`) は、以下のように導出される。

$(xSb, ySb) = (xCb + xSbIdx * sbWidth, yCb + ySbIdx * sbHeight)$ (8-913)

`cIdx` が 0 に等しい場合、`nIbcBufW` を `ibcBufferWidth` に設定し、そうでない場合、`nIbcBufW` を (`ibcBufferWidth / SubWidthC`) に設定する。以下のことが適用される。

$predSamples[xSb+x][ySb+y] = ibcBuf[(xSb+x + (mv[xSb][ySb][0] >> 4)) \% nIbcRefW][ySb+y + (mv[xSb][ySb][1] >> 4)]$ with `x = 0...sbWidth-1` and `y = 0...sbHeight-1`.

...

【0172】

8.6.3.2 ブロックベクトル妥当性検査処理

この処理への入力は以下の通りである。

- 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符号化ブロックの左上のサンプルを規定する輝度位置 (`xCb`, `yCb`)、
- ー 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 `cbWidth`、
- ー 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 `cbHeight`、
- ー 水平および垂直方向の輝度符号化サブブロックの数を規定する変数 `numSbX` および `numSbY`、
- ー `xSbIdx = 0...numSbX-1`、および `ySbIdx = 0...numSbY-1` の場合のブロックベクトル `mv[xSbIdx][ySbIdx]`、

10

20

30

40

50

- ー 現在のブロックの色成分インデックスを規定する変数 $cIdx$ 、
- ー $a(nIbcBufW) \times (ctbSize) array\ ibcBuf$

この処理の出力は、ブロックベクトルが有効であるかどうかを示すためのフラグ $isBVvalid$ である。

以下が適用される

1. $isBVvalid$ は $true$ に等しくなるように設定される。
2. $((yCb \& (ctbSize-1) + mv[0][0][1] + cbHeight) > ctbSize)$ の場合、 $isBVvalid$ は $false$ に等しく設定される。
3. そうでない場合、 $xSbIdx = 0 \dots numSbX-1$ 、および $ySbIdx = 0 \dots numSbY-1$ である各サブブロックインデックス $xSbIdx$ 、 $ySbIdx$ に対しては、 $ibcBuf$ の左上の輝度サンプルに対するその位置を導出する。

$xTL = (xCb + xSbIdx * sbWidth + mv[xSbIdx][ySbIdx][0]) \& (nIbcBufW-1)$

$yTL = (yCb \& (ctbSize-1)) + ySbIdx * sbHeight + mv[xSbIdx][ySbIdx][1]$

$xBR = (xCb + xSbIdx * sbWidth + sbWidth-1 + mv[xSbIdx][ySbIdx][0]) \& (nIbcBufW-1)$

$yBR = (yCb \& (ctbSize-1)) + ySbIdx * sbHeight + sbHeight-1 + mv[xSbIdx][ySbIdx][1]$

もし $(isDecoded[xTL >> 2][yTL >> 2] == 0)$ あるいは $(isDecoded[xBR >> 2][yTL >> 2] == 0)$ あるいは $(isDecoded[xBR >> 2][yBR >> 2] == 0)$ であれば、 $isBVvalid$ は $false$ に等しいと設定される。

【0173】

8.7.5 ピクチャ再構成処理

【0174】

8.7.5.1 一般

【0175】

この処理への入力は以下の通りである。

- ー 現在のピクチャ成分の左上のサンプルに対する現在のブロックの左上のサンプルを規定する位置 $(xCurr, yCurr)$ 、
- ー 現在のブロックの幅および高さをそれぞれ規定する変数 $nCurrSw$ および $nCurrSh$ 、
- ー 現在のブロックの色成分を規定する変数 $cIdx$ 、
- ー 現在のブロックの予測サンプルを規定する $a(nCurrSw) \times (nCurrSh) array\ predSamples$ 、
- ー 現在のブロックの残差サンプル規定する $a(nCurrSw) \times (nCurrSh) array$ 。

【0176】

この処理の出力は以下の通りである。

- ー 再構成されたピクチャサンプルアレイ $recSamples$ 。
- ー IBC 参照アレイ $ibcBuf$

...

$nIbcBufW$ は $ibcBuf$ の幅であることを示し、以下が適用される。

$ibcBuf[(xCurr+i) \& (nIbcBufW-1)][(yCurr+j) \& (ctbSize-1)] = recSamples[xCurr+i][yCurr+j]$

with $i = 0 \dots nCurrSw-1, j = 0 \dots nCurrSh-1$.

【0177】

5.16 実施形態 #16

【0178】

以下の変更を除き、上記実施形態と同じである。

【0179】

【表7】

slice_data() {	ディスクリ プタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>xPrevVPDU = 0</u>	
<u>yPrevVPDU = 0</u>	
<u>if(CtbSizeY == 128)</u>	
<u>reset_ibc_isDecoded(0, 0, 192, CtbSizeY, BufWidth,</u> <u>BufHeight)</u>	
<u>else</u>	
<u>reset_ibc_isDecoded(0, 0, 128*128/CtbSizeY, CtbSizeY,</u> <u>BufWidth, BufHeight)</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
.....	

10

20

【0180】

【表8】

<u>reset_ibc_isDecoded(x0, y0, w, h, BufWidth, BufHeight) {</u>	<u>ディスクリ プタ</u>
<u>if(x0 >= 0)</u>	
<u>for (x = x0 % BufWidth; x < x0 + w; x+=4)</u>	
<u>for (y = y0 % BufHeight; y < y0 + h; y+=4)</u>	
<u>isDecoded[x >> 2][y >> 2] = 0</u>	
<u>}</u>	

30

BufWidthは(CtbSizeY==128)?192:(128*128/CtbSizeY)に等しく、BufHeightはCtbSizeYに等しい。

【0181】

5.17 実施形態#17

【0182】

本明細書では、一部の例における変化は、太字、下線を付したテキストで示される。

【0183】

7.3.7 スライスデータ構文

【0184】

7.3.7.1 一般スライスデータ構文

【0185】

40

50

【表 9】

slice_data() {	ディスク リプタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++,	
CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag &&	
((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end_of_subset_one_bit /* equal to 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]]-1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end_of_brick_one_bit /* equal to 1 */	ae(v)
if(i < NumBricksInCurrSlice-1)	
byte_alignment()	
}	
}	

【0186】

7. 4. 8. 5 符号化ユニット構文

【0187】

以下のすべての条件がtrueである場合、NumHmvpSmrIbcCandをNumHmvpIbcCandに等しく設定し、HmvpSmrIbcCandList[i]をHmvpIbcCandList[i]に等しく設定してi=0..NumHmvpIbcCand-1とすることにより、共有したマージ候補リスト領域に対する履歴に基づく動きベクトル予測子は更新される。

- IsInSmr[x0][y0] is equal to TRUE.
- SmrX[x0][y0] is equal to x0.
- SmrY[x0][y0] is equal to y0.

【0188】

x=x0..x0+cbWidth-1およびy=y0..y0+cbHeight-1に対して、以下の通り割り当てをする。

CbPosX[x][y]=x0 (7-135)

CbPosY[x][y]=y0 (7-136)

CbWidth[x][y]=cbWidth (7-137)

CbHeight[x][y]=cbHeight (7-138)

vSizeをmin(ctbSize, 64)、wIbcBufを(128*128/ctbSize)とする。ibcBufの幅および高さは、wIbcBufおよびctbSizeに従う。

`refreshIbcBuf`が1に等しい場合、以下が適用される。

– `ibcBuf[x % wIbcBuf][y % ctbSize] = -1, for x = x0 . . x0 + wIbcBuf - 1 and y = y0 . . y0 + ctbSize - 1`
– `resetIbcBuf = 0`

(`x0 % vSize`)が0に等しく、かつ、(`y0 % vSize`)が0に等しいとき、但し`x = x0 . . x0 + vSize - 1`であって`y = y0 . . y0 + vSize - 1`であるが、以下が適用される。

`ibcBuf[x % wIbcBuf][y % ctbSize] = -1`
【0189】

8.6.2 IBCブロックの動きベクトル成分の導出処理

10

【0190】

8.6.2.1 一般

【0191】

この処理への入力は以下の通りである。

– 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符号化ブロックの左上のサンプルの輝度位置(`xCb, yCb`)、

– 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数`cbWidth`、

– 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数`cbHeight`。

【0192】

この処理の出力は以下の通りである。

20

– `1 / 16` 分数サンプル精度`mvL`における輝度動きベクトル。

【0193】

輝度動きベクトル`mvL`は、以下のように導出される。

– 8.6.2.2項で規定されたIBC輝度動きベクトル予測の導出処理は、輝度位置(`xCb, yCb`)、変数`cbWidth`および`cbHeight`を入力として呼び出され、その出力は輝度動きベクトル`mvL`である。

– `general_merge_flag[xCb][yCb]`が0の場合、以下が適用される

1. 変数`mvd`は、以下のように導出される。

`mvd[0] = MvdL0[xCb][yCb][0] (8-883)`

30

`mvd[1] = MvdL0[xCb][yCb][1] (8-884)`

2. 8.5.2.14項で規定されるような動きベクトルの丸め処理は、`mvL`に等しく設定された`mvX`セット、`MvShift + 2`に等しく設定された`rightShift`セット、`MvShift + 2`に等しく設定された`leftShift`セットを入力として呼び出され、丸められた`mvL`が出力となる。

3. 輝度動きベクトル`mvL`は、以下のように修正される。

`u[0] = (mvL[0] + mvd[0] + 218) % 218 (8-885)`

`mvL[0] = (u[0] >= 217) ? (u[0] - 218) : u[0] (8-886)`

`u[1] = (mvL[1] + mvd[1] + 218) % 218 (8-887)`

40

`mvL[1] = (u[1] >= 217) ? (u[1] - 218) : u[1] (8-888)`

注1 – 上記で指定した`mvL[0]`および`mvL[1]`の結果値は、常に $-2^{17} \sim 2^{17} - 1$ の範囲に含まれる。

【0194】

8.6.2.6項で規定されるような履歴に基づく動きベクトル予測子リストの更新処理は、輝度動きベクトル`mvL`を使用して呼び出される。

輝度ブロックベクトル`mvL`が以下の制約に従うべきであることは、ビットストリーム適合性の要件である。

– $((yCb + (mvL[1] > 4)) \% wIbcBuf) + cbHeight$ が`c`

50

`tbSize` 以下である。

— $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ の場合、`ibcBuf[(x + (mvL[0] >> 4)) % wIbcBuf][ (y + (mvL[1] >> 4)) % tbSize]` は -1 に等しくない。

【0195】

8.7.5 ピクチャ再構成処理

【0196】

8.7.5.1 一般

【0197】

この処理への入力は以下の通りである。

- 現在のピクチャ成分の左上のサンプルに対する現在のブロックの左上のサンプルを規定する位置 (`xCurr`, `yCurr`)、
- 現在のブロックの幅および高さをそれぞれ規定する変数 `nCurrSw` および `nCurrSh`、
- 現在のブロックの色成分を規定する変数 `cIdx`、
- 現在のブロックの予測サンプルを規定する `an(nCurrSw) × (nCurrSh) array predSamples`、
- 現在のブロックの残差サンプル規定する `an(nCurrSw) × (nCurrSh) array`。

【0198】

この処理の出力は、再構成ピクチャサンプルアレイ `recSamples` および `IBC` バッファアレイ `ibcBuf` である。

【0199】

色成分 `cIdx` の値に基づいて、以下の割り当てが行われる。

- `cIdx` が 0 に等しい場合、`recSamples` は再構成ピクチャサンプルアレイ `SL` に対応し、関数 `clipCidx1` は `Clip1y` に対応する。
- そうでない場合、`cIdx` が 1 に等しければ、`tuCbfcChroma` は `tu_cbf_cb[xCurr][yCurr]` に等しく設定され、`recSamples` は再構成されたクロマサンプル配列 `SCb` に対応し、関数 `clipCidx1` は `Clip1c` に対応する。
- そうでない場合 (`cIdx` が 2 に等しい)、`tuCbfcChroma` を `tu_cbf_cr[xCurr][yCurr]` に等しく設定し、`recSamples` は再構成されたクロマサンプル配列 `SCr` に対応し、関数 `clipCidx1` は `Clip1c` に対応する。

【0200】

`slice_lmcs_enabled_flag` の値によって、以下が適用される。

- `slice_lmcs_enabled_flag` が 0 に等しい場合、位置 (`xCurr`, `yCurr`) における再構成サンプル `recSamples` の $(nCurrSw) × (nCurrSh)$ ブロックは、 $i = 0 \dots nCurrSw - 1$, $j = 0 \dots nCurrSh - 1$ に対しては下記のように導出する。

$recSamples[xCurr+i][yCurr+j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) \quad (8-992)$

- そうでない場合 (`slice_lmcs_enabled_flag` が 1 に等しい)、以下が適用される。

— `cIdx` が 0 に等しい場合、以下が適用される。

— 8.7.5.2 項で規定されるような輝度サンプルのマッピング処理を伴うピクチャ再構成は、輝度位置 (`xCurr`, `yCurr`)、ブロック幅 `nCurrSw` および高さ `nCurrSh`、予測輝度サンプル配列 `preSamples`、および残差輝度サンプル配列 `resSamples` を入力として呼び出され、出力は再構成輝度サンプル配列

10

20

30

40

50

recSamplesとなる。

－それ以外の場合（cIdxが0より大きい場合）は、8.7.5.3項で規定されているクロマサンプルの輝度依存クロマ残差スケーリング処理によるピクチャ再構成が、クロマの位置（xCurr, yCurr）、変換ブロックの幅nCurrSwおよび高さnCurrSh、現在のクロマ変換ブロックの符号化ブロックフラグtuCbFChroma、予測クロマサンプル配列 predSamples、残差クロマサンプル配列 resSamplesを入力として呼び出され、再構成されたクロマサンプル配列 recSamplesを出力とする。

現在の符号化ユニットを復号化した後、以下が適用される。

```
i b c B u f [ ( x C u r r + i ) % w I b c B u f ] [ ( y C u r r + j ) % c t b S i z e ]
```

10

```
= r e c S a m p l e s [ x C u r r + i ] [ y C u r r + j ]
```

```
f o r i = 0 . . n C u r r S w - 1 , j = 0 . . n C u r r S h - 1 .
```

```
[ 0 2 0 1 ]
```

5.18 実施形態 # 18

```
[ 0 2 0 2 ]
```

本明細書では、いくつかの例における変更を、太字、下線、イタリック体で示している。

```
[ 0 2 0 3 ]
```

7.3.7 スライスデータ構文

```
[ 0 2 0 4 ]
```

20

7.3.7.1 一般スライスデータ構文

```
[ 0 2 0 5 ]
```

30

40

50

【表 10】

slice_data() {	ディスクリプタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++,	
CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag &&	
((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end_of_subset_one_bit /* equal to 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]]-1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end_of_brick_one_bit /* equal to 1 */	ae(v)
if(i < NumBricksInCurrSlice-1)	
byte_alignment()	
}	
}	

10

20

【0206】

30

7. 4. 8. 5 符号化ユニット構文

【0207】

以下のすべての条件が `true` である場合、`NumHmvpSmrIbcCand` を `NumHmvpIbcCand` に等しく設定し、`HmvpSmrIbcCandList[i]` を `HmvpIbcCandList[i]` に等しく設定して `i = 0 . . NumHmvpIbcCand - 1` とすることにより、共有したマージ候補リスト領域に対する履歴に基づく動きベクトル予測子は更新される。

- `IsInSmr[x0][y0] is equal to TRUE.`
- `SmrX[x0][y0] is equal to x0.`
- `SmrY[x0][y0] is equal to y0.`

40

【0208】

`x = x0 . . x0 + cbWidth - 1` および `y = y0 . . y0 + cbHeight - 1` に対して、以下の通り割り当てをする。

`CbPosX[x][y] = x0` (7-135)

`CbPosY[x][y] = y0` (7-136)

`CbWidth[x][y] = cbWidth` (7-137)

`CbHeight[x][y] = cbHeight` (7-138)

`vSize` を `min(ctbSize, 64)` に、`wIbcBufY` を $(128 * 128 / \text{CtbSizeY})$ に設定する。

`ibcBufL` は、幅が `wIbcBufY` であり、高さが `CtbSizeY` である配列で

50

ある。

$i b c B u f c b$ および $i b c B u f c r$ は、幅が $w I b c B u f C = (w I b c B u f Y / S u b W i d t h C)$ であり、高さが $(C t b S i z e Y / S u b H e i g h t C)$ であるアレ、すなわち $C t b S i z e C$ である。

$r e s e t I b c B u f$ が 1 に等しい場合、以下が適用される。

– $i b c B u f l [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$
 $, f o r x = x 0 . . x 0 + w I b c B u f Y - 1 \text{ and } y = y 0 . . y 0 + C t b S i z e Y - 1$

– $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$
 $, f o r x = x 0 . . x 0 + w I b c B u f C - 1 \text{ and } y = y 0 . . y 0 + C t b S i z e C - 1$

– $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$, $f o r x = x 0 . . x 0 + w I b c B u f C - 1 \text{ and } y = y 0 . . y 0 + C t b S i z e C - 1$

– $r e s e t I b c B u f = 0$

$(x 0 \% v S i z e Y)$ が 0 と等しく、 $(y 0 \% v S i z e Y)$ が 0 と等しい場合、以下が適用される。

– $i b c B u f l [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$,
 $f o r x = x 0 . . x 0 + v S i z e - 1 \text{ and } y = y 0 . . y 0 + v S i z e - 1$

– $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$, $f o r x = x 0 / S u b W i d t h C . . x 0 / S u b W i d t h C + v S i z e / S u b W i d t h C - 1$
 $\text{and } y = y 0 / S u b H e i g h t C . . y 0 / S u b H e i g h t C + v S i z e / S u b H e i g h t C - 1$

– $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$, $f o r x = x 0 / S u b W i d t h C . . x 0 / S u b W i d t h C + v S i z e / S u b W i d t h C - 1$
 $\text{and } y = y 0 / S u b H e i g h t C . . y 0 / S u b H e i g h t C + v S i z e / S u b H e i g h t C - 1$

【0209】

8.6.2 IBCブロックの動きベクトル成分の導出処理

【0210】

8.6.2.1 一般

【0211】

この処理への入力以下の通りである。

– 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符号化ブロックの左上のサンプルの輝度位置 $(x C b, y C b)$ 、

– 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $c b W i d t h$ 、

– 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $c b H e i g h t$ 。

【0212】

この処理の出力は以下の通りである。

– $1 / 16$ 分数サンプル精度 $m v L$ における輝度動きベクトル。

【0213】

輝度動きベクトル $m v L$ は、以下のように導出される。

– 8.6.2.2 項で規定された IBC 輝度動きベクトル予測の導出処理は、輝度位置 $(x C b, y C b)$ 、変数 $c b W i d t h$ および $c b H e i g h t$ を入力として呼び出され、その出力は輝度動きベクトル $m v L$ である。

– $g e n e r a l_m e r g e_f l a g [x C b] [y C b]$ が 0 と等しい場合、以下が適用される。

4. 変数 $m v d$ は、以下のように導出される。

$m v d [0] = M v d L 0 [x C b] [y C b] [0] \quad (8-883)$

$m v d [1] = M v d L 0 [x C b] [y C b] [1] \quad (8-884)$

5. 8. 5. 2. 1 4 項で規定されるような動きベクトルの丸め処理は、 mvL に等しく設定された mvX セット、 $MvShift+2$ に等しく設定された $rightShift$ セット、 $MvShift+2$ に等しく設定された $leftShift$ セットを入力として呼び出され、丸められた mvL が出力となる。

6. 輝度動きベクトル mvL は、以下のように修正される。

$$u[0] = (mvL[0] + mvd[0] + 2^{18}) \% 2^{18} \quad (8-885)$$
$$mvL[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$$
$$u[1] = (mvL[1] + mvd[1] + 2^{18}) \% 2^{18} \quad (8-887)$$
$$mvL[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$$

10

注1－ 上記で指定した $mvL[0]$ および $mvL[1]$ の結果値は、常に $-2^{17} \sim 2^{17}-1$ の範囲に含まれる。

【0214】

8. 6. 2. 6 項で規定されるような履歴に基づく動きベクトル予測子リストの更新処理は、輝度動きベクトル mvL を使用して呼び出される。

8. 6. 2. 5 項は、 mvL を入力として、 mvC を出力として呼び出される。

輝度ブロックベクトル mvL が以下の制約に従うべきであることは、ビットストリーム適合性の要件である。

－ $((yCb + (mvL[1] > 4)) \% CtbSizeY) + cbHeight$ が $CtbSizeY$ 以下である。

20

－ $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ の場合、 $ibcBufL[(x + (mvL[0] >> 4)) \% wIbcBufY][(y + (mvL[1] >> 4)) \% CtbSizeY]$ は、 -1 に等しくはならない。

－ $treeType$ が $SINGLE_TREE$ に等しい場合、 $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ に対しては、 $ibcBufCb[(x + (mvC[0] >> 5)) \% wIbcBufC][(y + (mvC[1] >> 5)) \% CtbSizeC]$ は、 -1 に等しくはならない。

【0215】

30

8. 6. 3 ibc ブロックの復号化処理

【0216】

8. 6. 3. 1 一般

【0217】

この処理は、 ibc 予測モードで符号化された符号化ユニットを復号化するとき呼び出される。

【0218】

この処理への入力以下の通りである。

－ 現在のピクチャの左上の輝度サンプルに対する現在の符号化ブロックの左上のサンプルを規定する輝度位置 (xCb , yCb)、

－ 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $cbWidth$ 、

－ 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $cbHeight$ 、

－ 現在のブロックの色成分インデックス、

－ 動きベクトル mv 、

－ $an(wIbcBufY) \times (CtbSizeY) array\ ibcBufL$, $an(wIbcBufC) \times (CtbSizeC) array\ ibcBufCb$, $an(wIbcBufC) \times (CtbSizeC) array\ ibcBufCr$ 。

40

【0219】

この処理の出力は以下の通りである。

－ 予測サンプルの配列 $predSamples$ 。

50

$x = xCb \dots xCb + Width - 1$ および $y = yCb \dots yCb + Height - 1$ の場合、以下が適用される

$cIdx$ が 0 に等しい場合

```
predSamples[x][y] = iBcBufL[(x + mv[0] >> 4)
% wIbCbBufY][(y + (mv[1] >> 4)) % CtbSizeY]
```

$cIdx$ が 1 に等しい場合

```
predSamples[x][y] = iBcBufCb[(x + mv[0] >> 5)
% wIbCbBufC][(y + (mv[1] >> 5)) % CtbSizeC]
```

$cIdx$ が 2 に等しい場合

```
predSamples[x][y] = iBcBufCr[(x + mv[0] >> 5)
% wIbCbBufC][(y + (mv[1] >> 5)) % CtbSizeC]
```

【0220】

8.7.5 ピクチャ再構成処理

【0221】

8.7.5.1 一般

【0222】

この処理への入力には以下の通りである。

- ー 現在のピクチャ成分の左上のサンプルに対する現在のブロックの左上のサンプルを規定する位置 ($xCurr$, $yCurr$)、
- ー 現在のブロックの幅および高さをそれぞれ規定する変数 $nCurrSw$ および $nCurrSh$ 、
- ー 現在のブロックの色成分を規定する変数 $cIdx$ 、
- ー 現在のブロックの予測サンプルを規定する $an(nCurrSw) \times (nCurrSh)$ array $predSamples$ 、
- ー 現在のブロックの残差サンプル規定する $an(nCurrSw) \times (nCurrSh)$ array。

【0223】

この処理の出力は、再構成されたピクチャサンプルアレイ $recSamples$ および IBC バッファアレイ $iBcBufL$ 、 $iBcBufCb$ 、 $iBcBufCr$ である。

【0224】

色成分 $cIdx$ の値に基づいて、以下の割り当てが行われる。

- ー $cIdx$ が 0 に等しい場合、 $recSamples$ は再構成ピクチャサンプルアレイ S_L に対応し、関数 $clipCidx1$ は $Clip1_Y$ に対応する。
- ー そうでない場合、 $cIdx$ が 1 に等しければ、 $tuCbfcChroma$ は $tu_cbf_cb[xCurr][yCurr]$ に等しく設定され、 $recSamples$ は再構成されたクロマサンプル配列 S_{Cb} に対応し、関数 $clipCidx1$ は $Clip1_c$ に対応する。
- ー そうでない場合、($cIdx$ が 2 に等しければ、 $tuCbfcChroma$ は $tu_cbf_cb[xCurr][yCurr]$ に等しく設定され、 $recSamples$ は再構成されたクロマサンプル配列 S_{Cb} に対応し、関数 $clipCidx1$ は $Clip1_c$ に対応する。

【0225】

$slice_lmcs_enabled_flag$ の値によって、以下が適用される。

- ー $slice_lmcs_enabled_flag$ が 0 に等しい場合、位置 ($xCurr$, $yCurr$) における再構成サンプル $recSamples$ の ($nCurrSw$) $\times$ ($nCurrSh$) ブロックは、 $i = 0 \dots nCurrSw - 1$, $j = 0 \dots nCurrSh - 1$ に対しては下記のように導出する。

```
recSamples[xCurr+i][yCurr+j] = clipCidx1(
predSamples[i][j] + resSamples[i][j]) (8-92)
```

10

20

30

40

50

－ そうでない場合 (`slice_lmcs_enabled_flag` が 1 に等しい)、以下が適用される。

－ `cIdx` が 0 に等しい場合、以下が適用される。

－ 8.7.5.2 項で規定されるような輝度サンプルのマッピング処理を伴うピクチャ再構成は、輝度位置 (`xCurr`, `yCurr`)、ブロック幅 `nCurrSw` および高さ `nCurrSh`、予測輝度サンプル配列 `preSamples`、および残差輝度サンプル配列 `resSamples` を入力として呼び出され、出力は再構成輝度サンプル配列 `reSamples` となる。

－ それ以外の場合 (`cIdx` が 0 より大きい場合) は、8.7.5.3 項で規定されているクロマサンプルの輝度依存クロマ残差スケーリング処理によるピクチャ再構成が、クロマの位置 (`xCurr`, `yCurr`)、変換ブロックの幅 `nCurrSw` および高さ `nCurrSh`、現在のクロマ変換ブロックの符号化ブロックフラグ `tuCbChroma`、予測クロマサンプル配列 `predSamples`、残差クロマサンプル配列 `resSamples` を入力として呼び出され、再構成されたクロマサンプル配列 `recSamples` を出力とする。

現在の符号化ユニットを復号化した後、以下を適用することができる。

`cIdx` が 0 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_LUMA` に等しい場合、以下が適用される。

```
ibcBufL[(xCurr+i)%wIbcBufY][(yCurr+j)%CtbSizeY]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

`cIdx` が 1 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_CHROMA` に等しい場合、以下が適用される。

```
ibcBufCb[(xCurr+i)%wIbcBufC][(yCurr+j)%CtbSizeC]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

`cIdx` が 2 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_CHROMA` に等しい場合、以下が適用される。

```
ibcBufCr[(xCurr+i)%wIbcBufC][(yCurr+j)%CtbSizeC]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

【0226】

5.19 実施形態 #19

【0227】

本明細書では、一部の例における変化は、太字、下線を付したテキストで示される。

b

【0228】

7.3.7 スライスデータ構文

【0229】

7.3.7.1 一般スライスデータ構文

【0230】

10

20

30

40

50

【表 1 1】

slice_data() {	ディスクリ プタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag && ((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end_of_subset_one_bit /* equal to 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]]-1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end_of_brick_one_bit /* equal to 1 */	ae(v)
if(i < NumBricksInCurrSlice-1)	
byte_alignment()	
}	
}	
}	

10

20

【0 2 3 1】

30

7. 4. 8. 5 符号化ユニット構文

【0 2 3 2】

以下のすべての条件が `true` である場合、`NumHmvpSmrIbcCand` を `NumHmvpIbcCand` に等しく設定し、`HmvpSmrIbcCandList[i]` を `HmvpIbcCandList[i]` に等しく設定して `i = 0 . . NumHmvpIbcCand - 1` とすることにより、共有したマージ候補リスト領域に対する履歴に基づく動きベクトル予測子は更新される。

- `IsInSmr[x0][y0] is equal to TRUE.`
- `SmrX[x0][y0] is equal to x0.`
- `SmrY[x0][y0] is equal to y0.`

40

【0 2 3 3】

`x = x0 . . x0 + cbWidth - 1` および `y = y0 . . y0 + cbHeight - 1` に対して、以下の通り割り当てをする。

`CbPosX[x][y] = x0` (7-135)

`CbPosY[x][y] = y0` (7-136)

`CbWidth[x][y] = cbWidth` (7-137)

`CbHeight[x][y] = cbHeight` (7-138)

`vSize` を `min(ctbSize, 64)` に、`wIbcBufY` を $(128 * 128 / \text{CtbSizeY})$ に設定する。

`ibcBufL` は、幅が `wIbcBufY` であり、高さが `CtbSizeY` である配列で

50

ある。

$i b c B u f c b$ および $i b c B u f c r$ は、幅が $w I b c B u f C = (w I b c B u f Y / S u b W i d t h C)$ であり、高さが $(C t b S i z e Y / S u b H e i g h t C)$ であるアレ、すなわち $C t b S i z e C$ である。

$r e s e t I b c B u f$ が 1 に等しい場合、以下が適用される。

— $i b c B u f l [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$
 , for $x = x0 \dots x0 + w I b c B u f Y - 1$ and $y = y0 \dots y0 + C t b S i z e Y - 1$

— $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$
 , for $x = x0 \dots x0 + w I b c B u f C - 1$ and $y = y0 \dots y0 + C t b S i z e C - 1$

— $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$
 , for $x = x0 \dots x0 + w I b c B u f C - 1$ and $y = y0 \dots y0 + C t b S i z e C - 1$

— $r e s e t I b c B u f = 0$

$(x0 \% v S i z e Y)$ が 0 と等しく、 $(y0 \% v S i z e Y)$ が 0 と等しい場合、以下が適用される。

— $i b c B u f l [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$
 , for $x = x0 \dots x0 + \min(v S i z e, c b W i d t h) - 1$ and $y = y0 \dots y0 + \min(v S i z e, c b H e i g h t) - 1$

— $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$
 , for $x = x0 / S u b W i d t h C \dots x0 / S u b W i d t h C + \min(v S i z e / S u b W i d t h C, c b W i d t h) - 1$ and $y = y0 / S u b H e i g h t C \dots y0 / S u b H e i g h t C + \min(v S i z e / S u b H e i g h t C, c b H e i g h t) - 1$

— $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$
 , for $x = x0 / S u b W i d t h C \dots x0 / S u b W i d t h C + \min(v S i z e / S u b W i d t h C, c b W i d t h) - 1$ and $y = y0 / S u b H e i g h t C \dots y0 / S u b H e i g h t C + \min(v S i z e / S u b H e i g h t C, c b H e i g h t) - 1$

【0234】

8.6.2 IBCブロックの動きベクトル成分の導出処理

【0235】

8.6.2.1 一般

【0236】

この処理への入力には以下の通りである。

- 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符号化ブロックの左上のサンプルの輝度位置 $(x C b, y C b)$ 、
- 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $c b W i d t h$ 、
- 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $c b H e i g h t$ 。

【0237】

この処理の出力は以下の通りである。

- $1 / 16$ 分数サンプル精度 $m v L$ における輝度動きベクトル。

【0238】

輝度動きベクトル $m v L$ は、以下のように導出される。

- 8.6.2.2 項で規定されたIBC輝度動きベクトル予測の導出処理は、輝度位置 $(x C b, y C b)$ 、変数 $c b W i d t h$ および $c b H e i g h t$ を入力として呼び出され、その出力は輝度動きベクトル $m v L$ である。

- $g e n e r a l_m e r g e_f l a g [x C b] [y C b]$ が 0 の場合、以下が適用される

10

20

30

40

50

7. 変数 mv_d は、以下のように導出される。

$$mv_d[0] = MvdL0[xCb][yCb][0] \quad (8-883)$$

$$mv_d[1] = MvdL0[xCb][yCb][1] \quad (8-884)$$

8. 8.5.2.14項で規定されるような動きベクトルの丸め処理は、 mv_L に等しく設定された mv_X セット、 $MvShift+2$ に等しく設定された $rightShift$ セット、 $MvShift+2$ に等しく設定された $leftShift$ セットを入力として呼び出され、丸められた mv_L が出力となる。

9. 輝度動きベクトル mv_L は、以下のように修正される。

$$u[0] = (mv_L[0] + mv_d[0] + 2^{18}) \% 2^{18} \quad (8-885)$$

$$mv_L[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$$

$$u[1] = (mv_L[1] + mv_d[1] + 2^{18}) \% 2^{18} \quad (8-887)$$

$$mv_L[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$$

注1— 上記で指定した $mv_L[0]$ および $mv_L[1]$ の結果値は、常に $-2^{17} \sim 2^{17}-1$ の範囲に含まれる。

【0239】

8.6.2.6項で規定されるような履歴に基づく動きベクトル予測子リストの更新処理は、輝度動きベクトル mv_L を使用して呼び出される。

8.6.2.5項は、 mv_L を入力として、 mv_C を出力として呼び出される。

輝度ブロックベクトル mv_L が以下の制約に従うべきであることは、ビットストリーム適合性の要件である。

— $((yCb + (mv_L[1] > 4)) \% CtbSizeY) + cbHeight$ が $CtbSizeY$ 以下である。

— $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ の場合、 $ibcBufL[(x + (mv_L[0] >> 4)) \% wIbcBufY][(y + (mv_L[1] >> 4)) \% CtbSizeY]$ は、 -1 に等しくはならない。

— $treeType$ が $SINGLE_TREE$ に等しい場合、 $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ に対しては、 $ibcBufCb[(x + (mv_C[0] >> 5)) \% wIbcBufC][(y + (mv_C[1] >> 5)) \% CtbSizeC]$ は、 -1 に等しくはならない。

【0240】

8.6.3 ibc ブロックの復号化処理

【0241】

8.6.3.1 一般

【0242】

この処理は、 ibc 予測モードで符号化された符号化ユニットを復号化するとき呼び出される。

【0243】

この処理への入力には以下の通りである。

- 現在のピクチャの左上の輝度サンプルに対する現在の符号化ブロックの左上のサンプルを規定する輝度位置 (xCb , yCb)、
- 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $cbWidth$ 、
- 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $cbHeight$ 、
- 現在のブロックの色成分インデックスを規定する変数 $cIdx$ 、
- 動きベクトル mv 、
- $an(wIbcBufY) \times (CtbSizeY) \text{ array } ibcBufL$, $an(wIbcBufC) \times (CtbSizeC) \text{ array } ibcBufCb$, $an(wIbcBufC) \times (CtbSizeC) \text{ array } ibcBufCr$.

10

20

30

40

50

この処理の出力は以下の通りである。

－ 予測サンプルの配列 `predSamples`

`x = xCb...xCb+Width-1` および `y = yCb...yCb+Height-1`、
の場合、以下が適用される

`cIdx` が 0 に等しい場合 `predSamples[x][y] = iBcBufL[(x+mv[0]>>4)%wIbCbufY][(y+(mv[1]>>4)%CtbSizeY]`

`cIdx` が 1 に等しい場合 `predSamples[x][y] = iBcBufCb[(x+mv[0]>>5)%wIbCbufC][(y+(mv[1]>>5)%CtbSizeC]`

`cIdx` が 2 に等しい場合

`predSamples[x][y] = iBcBufCr[(x+mv[0]>>5)%wIbCbufC][(y+(mv[1]>>5)%CtbSizeC]`
【0244】

8.7.5 ピクチャ再構成処理

【0245】

8.7.5.1 一般

【0246】

この処理への入力以下の通りである。

－ 現在のピクチャ成分の左上のサンプルに対する現在のブロックの左上のサンプルを規定する位置 (`xCurr`, `yCurr`)、

－ 現在のブロックの幅および高さをそれぞれ規定する変数 `nCurrSw` および `nCurrSh`、

－ 現在のブロックの色成分を規定する変数 `cIdx`、

－ 現在のブロックの予測サンプルを規定する `an(nCurrSw) × (nCurrSh) array predSamples`、

－ 現在のブロックの残差サンプル規定する `an(nCurrSw) × (nCurrSh) array`。

【0247】

この処理の出力は、再構成されたピクチャサンプルアレイ `recSamples` および IBC バッファアレイ `iBcBufL`、`iBcBufCb`、`iBcBufCr` である。

【0248】

色成分 `cIdx` の値に基づいて、以下の割り当てが行われる。

－ `cIdx` が 0 に等しい場合、`recSamples` は再構成ピクチャサンプルアレイ `SL` に対応し、関数 `clipCidx1` は `Clip1Y` に対応する。

－ そうでない場合、`cIdx` が 1 に等しければ、`tuCbfcChroma` は `tu_cbf_cb[xCurr][yCurr]` に等しく設定され、`recSamples` は再構成されたクロマサンプル配列 `Scb` に対応し、関数 `clipCidx1` は `Clip1c` に対応する。

－ そうでない場合、(`cIdx` が 2 に等しければ、`tuCbfcChroma` は `tu_cbf_cb[xCurr][yCurr]` に等しく設定され、`recSamples` は再構成されたクロマサンプル配列 `Scr` に対応し、関数 `clipCidx1` は `Clip1c` に対応する。

【0249】

`slice_lmcs_enabled_flag` の値によって、以下が適用される。

－ `slice_lmcs_enabled_flag` が 0 に等しい場合、位置 (`xCurr`, `yCurr`) における再構成サンプル `recSamples` の $(nCurrSw) \times (nCurrSh)$ ブロックは、 $i = 0 \dots nCurrSw - 1$, $j = 0 \dots nCurrSh - 1$ に対しては下記のように導出する。

`recSamples[xCurr+i][yCurr+j] = clipCidx1(`

10

20

30

40

50

$\text{predSamples}[i][j] + \text{resSamples}[i][j]) \quad (8-92)$

— そうでない場合 (`slice_lmcs_enabled_flag` が 1 に等しい)、以下が適用される。

— `cIdx` が 0 に等しい場合、以下が適用される。

— 8.7.5.2 項で規定されるような輝度サンプルのマッピング処理を伴うピクチャ再構成は、輝度位置 (`xCurr`, `yCurr`)、ブロック幅 `nCurrSw` および高さ `nCurrSh`、予測輝度サンプル配列 `predSamples`、および残差輝度サンプル配列 `resSamples` を入力として呼び出され、出力は再構成輝度サンプル配列 `recSamples` となる。

それ以外の場合 (`cIdx` が 0 より大きい場合) は、8.7.5.3 項で規定されているクロマサンプルの輝度依存クロマ残差スケーリング処理によるピクチャ再構成が、クロマの位置 (`xCurr`, `yCurr`)、変換ブロックの幅 `nCurrSw` および高さ `nCurrSh`、現在のクロマ変換ブロックの符号化ブロックフラグ `tuCbFChroma`、予測クロマサンプル配列 `predSamples`、残差クロマサンプル配列 `resSamples` を入力として呼び出され、再構成されたクロマサンプル配列 `recSamples` を出力とする。

現在の符号化ユニットを復号化した後、以下を適用することができる。

`cIdx` が 0 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_LUMA` に等しい場合、以下が適用される。

```

iBcBufL[(xCurr+i)%wIBcBufY][(yCurr+j)%CtbSizeY] = recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.

```

`cIdx` が 1 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_CHROMA` に等しい場合、以下が適用される。

```

iBcBufCb[(xCurr+i)%wIBcBufC][(yCurr+j)%CtbSizeC] = recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.

```

`cIdx` が 2 に等しく、`treeType` が `SINGLE_TREE` または `DUAL_TREE_CHROMA` に等しい場合、以下が適用される。

```

iBcBufCr[(xCurr+i)%wIBcBufC][(yCurr+j)%CtbSizeC] = recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.

```

[0250]

5.20 実施形態 #20

[0251]

本明細書では、いくつかの例における変更を、太字、下線、イタリック体で示している。

[0252]

7.3.7 スライスデータ構文

[0253]

7.3.7.1 一般スライスデータ構文

[0254]

10

20

30

40

50

【表 1 2】

slice_data() {	ディスクリ プタ
for(i = 0; i < NumBricksInCurrSlice; i++) {	
CtbAddrInBs = FirstCtbAddrBs[SliceBrickIdx[i]]	
for(j = 0; j < NumCtusInBrick[SliceBrickIdx[i]]; j++, CtbAddrInBs++) {	
if((j%BrickWidth[SliceBrickIdx[i]]) == 0) {	
NumHmvpCand = 0	
NumHmvpIbcCand = 0	
<u>resetIbcBuf = 1</u>	
}	
CtbAddrInRs = CtbAddrBsToRs[CtbAddrInBs]	
coding_tree_unit()	
if(entropy_coding_sync_enabled_flag && ((j + 1) % BrickWidth[SliceBrickIdx[i]] == 0)) {	
end_of_subset_one_bit /* equal to 1 */	ae(v)
if(j < NumCtusInBrick[SliceBrickIdx[i]]-1)	
byte_alignment()	
}	
}	
if(!entropy_coding_sync_enabled_flag) {	
end_of_brick_one_bit /* equal to 1 */	ae(v)
if(i < NumBricksInCurrSlice-1)	
byte_alignment()	
}	
}	

【0 2 5 5】

7. 4. 8. 5 符号化ユニット構文

【0 2 5 6】

以下のすべての条件がtrueである場合、NumHmvpSmrIbcCandをNumHmvpIbcCandに等しく設定し、HmvpSmrIbcCandList[i]をHmvpIbcCandList[i]に等しく設定してi=0..NumHmvpIbcCand-1とすることにより、共有したマージ候補リスト領域に対する履歴に基づく動きベクトル予測子は更新される。

- IsInSmr[x0][y0] is equal to TRUE.
- SmrX[x0][y0] is equal to x0.
- SmrY[x0][y0] is equal to y0.

【0 2 5 7】

x=x0..x0+cbWidth-1およびy=y0..y0+cbHeight-1に対して、以下の通り割り当てをする。

CbPosX[x][y]=x0 (7-135)

CbPosY[x][y]=y0 (7-136)

CbWidth[x][y]=cbWidth (7-137)

CbHeight[x][y]=cbHeight (7-138)

vSizeをmin(ctbSize, 64)に、wIbcBufYを(128*128/CtbSizeY)に設定する。

ibcBufLは、幅がwIbcBufYであり、高さがCtbSizeYである配列である。

10

20

30

40

50

$i b c B u f c b$ および $i b c B u f c r$ は、幅が $w I b c B u f C = (w I b c B u f Y / S u b W i d t h C)$ であり、高さが $(C t b S i z e Y / S u b H e i g h t C)$ であるアレ、すなわち $C t b S i z e C$ である。

$r e s e t I b c B u f$ が1に等しい場合、以下が適用される。

— $i b c B u f L [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$,
for $x = x 0 . . x 0 + w I b c B u f Y - 1$ and $y = y 0 . . y 0 + C t b S i z e Y - 1$

— $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$,
for $x = x 0 . . x 0 + w I b c B u f C - 1$ and $y = y 0 . . y 0 + C t b S i z e C - 1$

— $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$,
for $x = x 0 . . x 0 + w I b c B u f C - 1$ and $y = y 0 . . y 0 + C t b S i z e C - 1$

— $r e s e t I b c B u f = 0$

$(x 0 \% v S i z e Y)$ が0と等しく、 $(y 0 \% v S i z e Y)$ が0と等しい場合、以下が適用される。

— $i b c B u f L [x \% w I b c B u f Y] [y \% C t b S i z e Y] = -1$,
for $x = x 0 . . x 0 + \max (v S i z e, c b W i d t h) - 1$ and $y = y 0 . . y 0 + \max (v S i z e, c b H e i g h t) - 1$

— $i b c B u f c b [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$,
for $x = x 0 / S u b W i d t h C . . x 0 / S u b W i d t h C + \max (v S i z e / S u b W i d t h C, c b W i d t h) - 1$ and $y = y 0 / S u b H e i g h t C . . y 0 / S u b H e i g h t C + \max (v S i z e / S u b H e i g h t C, c b H e i g h t) - 1$

— $i b c B u f c r [x \% w I b c B u f C] [y \% C t b S i z e C] = -1$,
for $x = x 0 / S u b W i d t h C . . x 0 / S u b W i d t h C + \max (v S i z e / S u b W i d t h C, c b W i d t h) - 1$ and $y = y 0 / S u b H e i g h t C . . y 0 / S u b H e i g h t C + \max (v S i z e / S u b H e i g h t C, c b H e i g h t) - 1$

【0258】

8.6.2 IBCブロックの動きベクトル成分の導出処理

【0259】

8.6.2.1 一般

【0260】

この処理への入力には以下の通りである。

— 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符号化ブロックの左上のサンプルの輝度位置 $(x C b, y C b)$ 、

— 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $c b W i d t h$ 、

— 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $c b H e i g h t$ 。

【0261】

この処理の出力は以下の通りである。

— $1 / 16$ 分数サンプル精度 $m v L$ における輝度動きベクトル。

【0262】

輝度動きベクトル $m v L$ は、以下のように導出される。

— 8.6.2.2項で規定されたIBC輝度動きベクトル予測の導出処理は、輝度位置 $(x C b, y C b)$ 、変数 $c b W i d t h$ および $c b H e i g h t$ を入力として呼び出され、その出力は輝度動きベクトル $m v L$ である。

— $g e n e r a l_m e r g e_f l a g [x C b] [y C b]$ が0と等しい場合、以下が適用される

10. 変数 $m v d$ は、以下のように導出される。

10

20

30

40

50

$mv d[0] = MvdL0[xCb][yCb][0] \quad (8-883)$

$mv d[1] = MvdL0[xCb][yCb][1] \quad (8-884)$

11. 8. 5. 2. 14項で規定されるような動きベクトルの丸め処理は、 mvL に等しく設定された mvX セット、 $MvShift+2$ に等しく設定された $rightShift$ セット、 $MvShift+2$ に等しく設定された $leftShift$ セットを入力として呼び出され、丸められた mvL が出力となる。

12. 輝度動きベクトル mvL は、以下のように修正される。

$u[0] = (mvL[0] + mv d[0] + 2^{18}) \% 2^{18} \quad (8-885)$

$mvL[0] = (u[0] \geq 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (8-886)$

$u[1] = (mvL[1] + mv d[1] + 2^{18}) \% 2^{18} \quad (8-887)$

$mvL[1] = (u[1] \geq 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (8-888)$

注1－ 上記で指定した $mvL[0]$ および $mvL[1]$ の結果値は、常に $-2^{17} \sim 2^{17}-1$ の範囲に含まれる。

【0263】

8. 6. 2. 6項で規定されるような履歴に基づく動きベクトル予測子リストの更新処理は、輝度動きベクトル mvL を使用して呼び出される。

8. 6. 2. 5項は、 mvL を入力として、 mvC を出力として呼び出される。

輝度ブロックベクトル mvL が以下の制約に従うべきであることは、ビットストリーム適合性の要件である。

－ $((yCb + (mvL[1] > 4)) \% CtbSizeY) + cbHeight$ が $CtbSizeY$ 以下である。

－ $x = xCb \dots xCb + cbWidth - 1$ および $y = yCb \dots yCb + cbHeight - 1$ の場合、 $ibcBufL[(x + (mvL[0] >> 4)) \% wIbcBufY][(y + (mvL[1] >> 4)) \% CtbSizeY]$ は、 -1 に等しくはならない。

【0264】

8. 6. 3 ibc ブロックの復号化処理

【0265】

8. 6. 3. 1 一般

【0266】

この処理は、 ibc 予測モードで符号化された符号化ユニットを復号化するとき呼び出される。

【0267】

この処理への入力には以下の通りである。

- － 現在のピクチャの左上の輝度サンプルに対する現在の符号化ブロックの左上のサンプルを規定する輝度位置(xCb , yCb)、
- － 輝度サンプルにおける現在の符号化ブロックの幅を規定する変数 $cbWidth$ 、
- － 輝度サンプルにおける現在の符号化ブロックの高さを規定する変数 $cbHeight$ 、
- － 現在のブロックの色成分インデックスを規定する変数 $cIdx$ 、
- － 動きベクトル mv

－ $an(wIbcBufY) \times (CtbSizeY)$ array $ibcBufL$,
 $an(wIbcBufC) \times (CtbSizeC)$ array $ibcBufCb$, $an(wIbcBufC) \times (CtbSizeC)$ array $ibcBufCr$.

この処理の出力は以下の通りである。

- － 予測サンプルの配列 $predSamples$ 。

$x = xCb \dots xCb + Width - 1$ および $y = yCb \dots yCb + Height - 1$, の場合、以下が適用される

$cIdx$ が0に等しい場合

10

20

30

40

50

```
predSamples[x][y] = iBcBufL[(x + mv[0] >> 4) % wIbCBufY] [(y + (mv[1] >> 4)) % CtBSizeY]
```

cIdxが1に等しい場合

```
predSamples[x][y] = iBcBufCb[(x + mv[0] >> 5) % wIbCBufC] [(y + (mv[1] >> 5)) % CtBSizeC]
```

cIdxが2に等しい場合

```
predSamples[x][y] = iBcBufCr[(x + mv[0] >> 5) % wIbCBufC] [(y + (mv[1] >> 5)) % CtBSizeC]
```

【0268】

8.7.5 ピクチャ再構成処理

【0269】

8.7.5.1 一般

【0270】

この処理への入力には以下の通りである。

- ー 現在のピクチャ成分の左上のサンプルに対する現在のブロックの左上のサンプルを規定する位置 (xCurr, yCurr)、
- ー 現在のブロックの幅および高さをそれぞれ規定する変数nCurrSwおよびnCurrSh、
- ー 現在のブロックの色成分を規定する変数cIdx、
- ー 現在のブロックの予測サンプルを規定するan(nCurrSw) × (nCurrSh) array predSamples、
- ー 現在のブロックの残差サンプル規定するan(nCurrSw) × (nCurrSh) array。

【0271】

この処理の出力は、再構成されたピクチャサンプルアレイrecSamplesおよびIBCバッファアレイiBcBufL、iBcBufCb、iBcBufCrである。

【0272】

色成分cIdxの値に基づいて、以下の割り当てが行われる。

- ー cIdxが0に等しい場合、recSamplesは再構成ピクチャサンプルアレイSLに対応し、関数clipCidx1はClip1yに対応する。
- ー そうでない場合、cIdxが1に等しければ、tuCbFChromaはtu_cbf_cb[xCurr][yCurr]に等しく設定され、recSamplesは再構成されたクロマサンプル配列Scbに対応し、関数clipCidx1はClip1cに対応する。
- ー そうでない場合、(cIdxが2に等しければ、tuCbFChromaはtu_cbf_cb[xCurr][yCurr]に等しく設定され、recSamplesは再構成されたクロマサンプル配列Scbに対応し、関数clipCidx1はClip1cに対応する。

【0273】

slice_lmcs_enabled_flagの値によって、以下が適用される。

- ー slice_lmcs_enabled_flagが0に等しい場合、位置(xCurr, yCurr)における再構成サンプルrecSamplesの(nCurrSw) × (nCurrSh)ブロックは、i = 0...nCurrSw-1, j = 0...nCurrSh-1に対しては下記のように導出する。

```
recSamples[xCurr+i][yCurr+j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) (8-992)
```

- ー そうでない場合 (slice_lmcs_enabled_flagが1に等しい)、以下が適用される。

- ー cIdxが0に等しい場合、以下が適用される。

10

20

30

40

50

－ 8.7.5.2項で規定されるような輝度サンプルのマッピング処理を伴うピクチャ再構成は、輝度位置(xCurr, yCurr)、ブロック幅nCurrSwおよび高さnCurrSh、予測輝度サンプル配列preSamples、および残差輝度サンプル配列resSamplesを入力として呼び出され、出力は再構成輝度サンプル配列reSamplesとなる。

－ それ以外の場合(cIdxが0より大きい場合)は、8.7.5.3項で規定されているクロマサンプルの輝度依存クロマ残差スケーリング処理によるピクチャ再構成が、クロマの位置(xCurr, yCurr)、変換ブロックの幅nCurrSwおよび高さnCurrSh、現在のクロマ変換ブロックの符号化ブロックフラグtuCbFChroma、予測クロマサンプル配列predSamples、残差クロマサンプル配列resSamplesを入力として呼び出され、再構成されたクロマサンプル配列recSamplesを出力とする。

現在の符号化ユニットを復号化した後、以下を適用することができる。

cIdxが0に等しく、treeTypeがSINGLE_TREEまたはDUAL_TREE_LUMAに等しい場合、以下が適用される。

```
ibcBufL[(xCurr+i)%wIbcBufY][(yCurr+j)%CtbSizeY]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

cIdxが1に等しく、treeTypeがSINGLE_TREEまたはDUAL_TREE_CHROMAに等しい場合、以下が適用される。

```
ibcBufCb[(xCurr+i)%wIbcBufC][(yCurr+j)%CtbSizeC]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

cIdxが2に等しく、treeTypeがSINGLE_TREEまたはDUAL_TREE_CHROMAに等しい場合、以下が適用される。

```
ibcBufCr[(xCurr+i)%wIbcBufC][(yCurr+j)%CtbSizeC]=recSamples[xCurr+i][yCurr+j]
for i=0..nCurrSw-1, j=0..nCurrSh-1.
```

【0274】

図6は、視覚媒体(映像または画像)処理の例示的な方法の一例を示すフローチャートである。方法600は、現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピー符号化モードを使用して現在の映像ブロックのための参照サンプルを記憶するバッファサイズを判定することと(602)、このバッファに記憶された参照サンプルを使用してこの変換を行うことと(604)を含む。

【0275】

以下の項は、方法600の実施形態および他の方法によって実装されるいくつかの例示的な好適な特徴を説明する。追加の例は、本明細書の第4章に記載する。

【0276】

1. 現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピー符号化モードを使用して前記現在の映像ブロックのための参照サンプルを記憶するバッファサイズを判定することと、前記バッファに記憶された前記参照サンプルを使用して前記変換を行うことと、を含む、映像処理方法。

【0277】

2. 前記バッファサイズが所定の定数である、第1項に記載の方法。

【0278】

3. 前記サイズがM×Nであり、MおよびNが整数である、第1項から2項のいずれかに記載の方法。

【0279】

4. M×Nが64×64または128×128または64×128に等しい、第3項に記載の方法。

10

20

30

40

50

【0280】

5. 前記バッファサイズは、前記現在の映像ブロックの符号化ツリーユニットのサイズに等しい、第1項に記載の方法。

【0281】

6. 前記バッファサイズは、前記変換に使用される仮想パイプラインデータユニットのサイズに等しい、第1項に記載の方法。

【0282】

7. 前記バッファサイズは、前記ビットストリーム表現における1つのフィールドに対応する、第1項に記載の方法。

【0283】

8. 前記フィールドは、映像パラメータセット、シーケンスパラメータセットまたはピクチャパラメータセット、またはピクチャヘッダ、スライスヘッダ、またはタイルグループヘッダレベルでのビットストリーム表現に含まれる、第7項に記載の方法。

【0284】

9. 前記バッファの前記サイズが、輝度成分の参照サンプルとクロマ成分の参照サンプルとで異なる、第1～8項のいずれかに記載の方法。

【0285】

10. 前記バッファの前記サイズは、前記現在の映像ブロックのクロマサブサンプリングフォーマットに依存する、第1～8項のいずれかに記載の方法。

【0286】

11. 前記参照サンプルはRGBフォーマットで記憶される、第1～8項のいずれかに記載の方法。

【0287】

12. 前記バッファは、ループフィルタリング前およびループフィルタリング後の再構成サンプルを記憶するために使用される、第1～11項のいずれかに記載の方法。

【0288】

13. ループフィルタリングは、非ブロック化フィルタリング、または適応ループフィルタリング(ALF)、またはサンプル適応オフセット(SAO)フィルタリングを含む、第12項に記載の方法。

【0289】

14. 現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換のため、前記参照サンプルの初期値を使用してイントラブロックコピー符号化モードを使用して前記現在の映像ブロックのための参照サンプルを記憶するバッファを初期化することと、前記バッファに記憶された前記参照サンプルを使用して前記変換を行うことと、を含む、映像処理方法。

【0290】

15. 前記初期値は定数に対応する、第14項に記載の方法。

【0291】

16. 前記初期値は、現在の映像ブロックのビット深度の関数である、第14～15項のいずれかに記載の方法。

【0292】

17. 前記定数が中間グレー値に対応する、第15項に記載の方法。

【0293】

18. 前記初期値は、前回復号化された映像ブロックの画素値に対応する、第14項に記載の方法。

【0294】

19. 前回復号化された映像ブロックは、インループフィルタリング前の復号化されたブロックに対応する、第18項に記載の方法。

【0295】

20. 前記バッファサイズが第1～13項の1つに記載のものである、第14～19

10

20

30

40

50

項のいずれかに記載の方法。

【0296】

21. x および y の数字を使用して、前記バッファ内の画素位置がアドレス指定される、第1～20項のいずれかに記載の方法。

【0297】

22. 前記バッファ内の画素位置を0から $M \times N - 1$ までの範囲の単一の数を使用して、アドレス指定し、 M および N は、バッファの画素幅および画素高さである、第1～20項のいずれかに記載の方法。

【0298】

23. 現在のビットストリーム表現は、前記変換のためのブロックベクトルを含み、
(BVx , BVy) として表される前記ブロックベクトルは、($x - x_0$, $y - y_0$) に等しく、(x_0 , y_0) は、前記現在の映像ブロックの符号化ツリーユニットの左上位置に対応する、第1～20項のいずれかに記載の方法。

10

【0299】

24. 現在のビットストリーム表現は、前記変換のためのブロックベクトルを含み、
(BVx , BVy) として表される前記ブロックベクトルは、($x - x_0 + Tx$, $y - y_0 + Ty$) に等しく、(x_0 , y_0) は、前記現在の映像ブロックの符号化ツリーユニットの左上位置に対応し、 Tx および Ty は、オフセット値である、第1～20項のいずれかに記載の方法。

20

【0300】

25. Tx および Ty が予め規定されたオフセット値である、第24項に記載の方法。

【0301】

26. 前記変換の間、位置(x_0 , y_0) にあり、ブロックベクトル(BVx , BVy) を有する画素に対し、前記バッファにおける対応する参照が、参照位置($x_0 + BVx$, $y_0 + BVy$) に見出される、第1～20項のいずれかに記載の方法。

【0302】

27. 前記参照位置が前記バッファの外にある場合、前記バッファにおける前記参照は、前記バッファの境界でクリッピングすることによって判定される、第26項に記載の方法。

【0303】

28. 前記参照位置が前記バッファの外にある場合、前記バッファにおける前記参照は、所定の値を有すると判定される、第26項に記載の方法。

30

【0304】

29. 前記変換の間、位置(x_0 , y_0) にあり、ブロックベクトル(BVx , BVy) を有する画素に対し、前記バッファにおける対応する参照が、参照位置($(x_0 + BVx) \bmod M$, $(y_0 + BVy) \bmod N$) に見出され、「 $\bmod$ 」は、モデュロ演算であり、 M および N は、前記バッファの x および y 寸法を表す整数である、第1～20項に記載の方法。

【0305】

30. 映像と現在の映像ブロックのビットストリーム表現との間での変換中に、映像境界におけるイントラブロックコピー符号化のための参照サンプルを記憶するバッファをリセットすることと、このバッファに記憶された参照サンプルを使用してこの変換を行うことと、を含む、映像処理方法。

40

【0306】

31. 前記映像境界は、新しいピクチャまたは新しいタイルに対応する、第30項に記載の方法。

【0307】

32. 前記変換は、前記リセットの後、前記バッファを仮想パイプラインデータユニット($VPDU$) の再構成された値で更新することによって行われる、第30項に記載の方法。

50

【0308】

33. 前記変換は、前記リセットの後、前記バッファを符号化ツリーユニットの再構成された値で更新することによって行われる、第30項に記載の方法。

【0309】

34. 前記リセットは、各符号化ツリーユニット行の最初に行われる、項30に記載の方法。

【0310】

35. 前記バッファの前記サイズは、 64×64 の前回復号化されたブロックL個に対応し、Lは整数である、第1項に記載の方法。

【0311】

36. 前記変換中に前記バッファ内のサンプルを読み取るかまたは記憶するために、垂直走査順序が使用される、第1～35項のいずれかに記載の方法。

【0312】

37. 現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピー符号化モードを使用して前記現在の映像ブロックのための参照サンプルを記憶するためのバッファを使用することであって、前記バッファの第1のビット深度は、前記符号化されたデータの第2のビット深度とは異なる、使用することと、前記バッファに記憶された前記参照サンプルを使用して前記変換を行うことと、を含む、映像処理方法。

【0313】

38. 前記第1のビット深度が前記第2のビット深度よりも大きい、第37項に記載の方法。

【0314】

39. 前記第1のビット深度が、変換中に使用される再構成バッファのビット深度と同じである、第37～38項のいずれかに記載の方法。

【0315】

40. 前記第1のビット深度が、前記ビットストリーム表現において値または差分値として信号伝達される、第37～39項のいずれかに記載の方法。

【0316】

41. 前記変換は、クロマおよび輝度成分に対して異なるビット深度を使用する、第37～40項のいずれかに記載の方法。

【0317】

第37～41項までの追加の実施形態および実施例は、第4章の第7項に記載されている。

【0318】

42. イントラブロックコピーモードを使用して現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間で変換を行うことを含み、前記変換中に予測計算に使用される第1の精度は、前記変換中に再構成計算に使用される第2の精度より低い、映像処理方法。

【0319】

43. 前記予測計算は、 $\text{clip} \{ \{ p + [1 \ll (b - 1)] \} \gg b, 0, (1 \ll (\text{bit depth}) - 1) \ll b$ を使用して、再構成サンプル値から予測サンプル値を判定することを含み、pは、再構成サンプル値であり、bは、予め規定されたビットシフト値であり、bit depthは、予測サンプル精度である、第43項に記載の方法。

【0320】

第42～43項までの追加の実施形態および実施例は、第4章の第28～31、および34項に記載されている。

【0321】

44. イントラブロックコピーモードを使用して現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うことを含み、サイズ $nM \times nM$ は、符

10

20

30

40

50

号化ツリーユニットサイズ $M \times M$ に使用され、 n および N は、整数であり、前記現在の映像ブロックは、前記符号化ツリーユニットに位置決めされ、前記参照領域は、前記現在の映像ブロックに対応する符号化ツリーユニットで最も近く、利用可能な $n \times n$ 符号化ツリーユニットである、映像処理方法。

【0322】

第4項の追加の実施形態および実施例は、第4章の第35項に記載されている。

【0323】

45. イントラブロックコピーモードを使用して現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行うことを含み、サイズ $nM \times nM$ は、符号化ツリーユニットサイズ、 $M \times M$ 以外に使用され、 n および N は、整数であり、前記現在の映像ブロックは、前記符号化ツリーユニットに位置決めされ、前記参照領域は、前記現在の映像ブロックに対応する符号化ツリーユニットで最も近く、利用可能な $n \times n - 1$ 符号化ツリーユニットである、映像処理方法。

【0324】

第4項の追加の実施形態および実施例は、第4章の第36項に記載されている。図8および9には、さらなる例示的な実施形態が示されている。

【0325】

46. $M = mW$ 、 $N = H$ であり、 W および H は、現在の映像ブロックの符号化ツリーユニット (CTU) の幅および高さであり、 m は、正の整数である、請求項3に記載の方法。

【0326】

47. $M = W$ かつ $N = nH$ であり、 W および H は、符号化ツリーユニット (CTU) の幅および高さであり、 n は、正の整数である、請求項3に記載の方法。

【0327】

48. $M = mW$ 、 $N = nH$ であり、 W および H は、符号化ツリーユニット (CTU) の幅および高さであり、 m および n は、正の整数である、請求項3に記載の方法。

【0328】

49. n および m が CTU のサイズに依存する、請求項46～48のいずれかに記載の方法。

【0329】

50. 現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、前記映像の成分 X を使用して前記映像の成分 c の前記現在の映像ブロックに対応するブロックベクトルの妥当性を判定することであって、前記成分 X は、前記映像の輝度成分とは異なる、判定することと、前記ブロックベクトルが前記現在の映像ブロックに対して有効であると判定されると前記ブロックベクトルを使用して前記変換を行うことと、を含む映像処理方法。ここで、 (BV_x, BV_y) と表されるブロックベクトルは、 $(x - x_0, y - y_0)$ に等しく、 (x_0, y_0) は、現在の映像ブロックの符号化ツリーユニットの左上位置に対応する。

【0330】

51. 前記成分 c が、前記映像の輝度成分に対応する、項50項に記載の方法。

【0331】

52. 前記現在の映像ブロックがクロマブロックであり、前記映像が4:4:4フォーマットである、第50項に記載の方法。

【0332】

53. 前記映像は、4:2:0形式であり、前記現在の映像ブロックは、位置 (x, y) で始まるクロマブロックであり、前記判定することは、 $isRec(c, ((x + BV_x) >> 5 << 5) + 64 - (((y + BV_y) >> 5) \& 1) * 32 + (x \% 32), ((y + BV_y) >> 5 << 5) + (y \% 32))$ が true である場合には前記ブロックベクトルは無効であると判定することを含む、第50項に記載の方法。

【0333】

10

20

30

40

50

54. 前記映像は、4:2:0形式であり、前記現在の映像ブロックは、位置(x, y)から始まるクロマブロックであり、前記判定することは、 $isRec(c, x + BVx + Chroma_CTU_size, y)$ がtrueである場合には、前記ブロックベクトルは無効であると判定することを含む、第50項に記載の方法

【0334】

55. 映像領域の現在の仮想パイプラインデータユニット(VPDU)の現在の映像ブロックと、現在の映像ブロックのビットストリーム表現との間での変換のため、選択的に、映像領域の第1の行から前処理済みVPDUであるK1および第2の行の前処理済みVPDUであるK2を使用することを判定することと、変換を行うこととあって、この変換は、現在のVPDUの残りをを使用することを除外する、変換を多なうことと、を含む映像処理方法。

10

【0335】

56. K1=1であり、K2=2である、第55項に記載の方法。

【0336】

57. 前記現在の映像ブロックは、前記映像領域の寸法または前記現在のVPDUの寸法に基づいて選択的に処理される、第55～56項に記載の方法。

【0337】

58. 現在の映像ブロックと、前記現在の映像ブロックのビットストリーム表現との間での変換のためにブロックベクトルの妥当性チェックを行うこととあって、前記ブロックベクトルは、イントラブロックコピーモードで使用する、妥当性チェックを行うことと、前記妥当性チェックの結果を使用して前記変換中に前記ブロックベクトルを選択的に使用することと、を含む、映像処理方法。

20

【0338】

59. イントラブロックコピー(IBC)バッファを前記変換中に使用し、前記IBCバッファの幅と高さをWbufおよびHbufとして、前記現在の映像ブロックの寸法がW×Hであり、前記ブロックベクトルは、(BVx, BVy)として表現され、現在の映像ブロックは、WpicおよびHpicの寸法を有する現在のピクチャにあり、符号化ツリーユニットは、幅および高さとしてWctuおよびHctuを有し、前記妥当性チェックは、所定の規則を使用する、第58項に記載の方法。

【0339】

30

60. 前記現在の映像ブロックは、輝度ブロック、クロマブロック、符号化ユニットCU、変換ユニットTU、4×4ブロック、2×2ブロック、または画素座標(X, Y)から始まる親ブロックのサブブロックである、第58～59項のいずれかに記載の方法。

【0340】

61. 前記妥当性チェックは、前記現在のピクチャの境界外にある前記ブロックベクトルを妥当であると見なす、第58～60項のいずれかに記載の方法。

【0341】

62. 前記妥当性チェックは、符号化ツリーユニットの境界外にあるブロックベクトルを妥当であると見なす、第58～60項のいずれかに記載の方法。

【0342】

40

前章の第23～30項では、上記第58～62項の追加例および変形例を述べている。

【0343】

63. 前記変換は、前記現在の映像ブロックから前記ビットストリーム表現を生成することを含む、第1～62項のいずれかに記載の方法。

【0344】

64. 前記変換は、前記ビットストリーム表現から前記現在の映像ブロックの画素値を生成することを含む、第1～62項のいずれかに記載の方法。

【0345】

65. 請求項1～62の1項以上に記載の方法を実装するように構成された処理装置を備える映像エンコーダ装置。

50

【0346】

66. 第1～62項の1項以上に記載の方法を実装するように構成された処理装置を備える、映像デコーダ装置。

【0347】

67. コードが記憶されたコンピュータ可読媒体であって、前記コードは第1～62項のいずれかまたは複数項に記載の方法を実装するための処理装置が実行可能な命令を実施する、コンピュータ可読媒体。

【0348】

図7は、映像／画像処理装置700のハードウェアプラットフォームを示すブロック図である。装置700は、本明細書に記載の方法の1つ以上を実装するために使用してもよい。装置700は、スマートフォン、タブレット、コンピュータ、IoT (Internet of Things) 受信機等により実装されてもよい。装置700は、1つ以上のプロセッサ702と、1つ以上のメモリ704と、映像処理ハードウェア706と、を含んでもよい。1つまたは複数の処理装置702は、本明細書に記載される1つ以上の方法（方法600を含むが、これに限定されない）を実装するように構成されてもよい。メモリ（複数可）704は、本明細書で説明される方法および技術を実装するために使用されるデータおよびコードを記憶するために使用してもよい。映像処理ハードウェア706は、本明細書に記載される技術をハードウェア回路にて実装するために使用してもよい。

10

【0349】

現在の映像ブロックに対応するビットストリーム表現は、ビットの連続したセットである必要はなく、ヘッダ、パラメータセット、およびネットワーク抽象化層 (NAL) パケットにまたがって分散されてもよい。

20

【0350】

セクションA：追加の例示的な実施形態

【0351】

セクションAにおいて、本明細書に記載される技術の一部を実装するために、VVC規格の現在のバージョンを修正し得る別の例示的な実施形態を提示する。

【0352】

本セクションでは、現在のIBC参照バッファ設計におけるいくつかの問題を分析し、これらの問題に対処するための異なる設計を提示する。復号化メモリと混合する代わりに、独立したIBC参照バッファが提案される。現在のアンカと比較して、提案されたスキームは、クラスFで $-0.99\% / -0.71\% / -0.79\%$ のAI/RA/LD-Bの輝度BD率、 $4:2:0$ TGMでは $-2.57\% / -1.81\% / -1.36\%$ で、 6.7% のメモリの減少、あるいは、クラスFでは $-1.31\% / -1.01\% / -0.81\%$ 、 $4:2:0$ TGMでは $-3.23\% / -2.33\% / -1.71\%$ で、 6.7% のメモリの増加を示している。

30

【0353】

A1. 導入

【0354】

イントラブロックコピー、すなわちIBC（または現在のピクチャ参照、即ち前のCPR）符号化モードが採用される。なお、IBC参照サンプルはオンチップメモリに記憶されるべきであり、従って、1つのCTUの限られた参照領域が規定される。バッファのためのエクストラオンチップメモリを制限するために、現在の設計は、 64×64 メモリを再利用して現在のVPDUを復号化するので、IBCをサポートするために必要なメモリは3つの追加の 64×64 ブロックのメモリのみである。CTUの大きさが 128×128 である場合の、現在の参照領域を図2に示す。

40

【0355】

【表 1 3】

- 次の各号に掲げる事項が該当すること：

(yCb + (mvL[1] >> 4)) >> CtbLog2SizeY = yCb >> CtbLog2SizeY (8-972)

(yCb + (mvL[1] >> 4) + cbHeight - 1) >> CtbLog2SizeY = yCb >> CtbLog2SizeY (8-973)

(xCb + (mvL[0] >> 4)) >> CtbLog2SizeY >= (xCb >> CtbLog2SizeY) - 1 (8-974)

(xCb + (mvL[0] >> 4) + cbWidth - 1) >> CtbLog2SizeY <= (xCb >> CtbLog2SizeY) (8-975)

[Ed. (SL): 条件 (8-218) および (8-216) は、6.4.X. で確認された可能性があります。)

- (xCb+(mvL[0] > 4) > CtbLog2SizeY が (xCb > CtbLog2SizeY) - 1 の場合、6.4.X [Ed] 項に記載のブロック可用性の導出プロセスを実行する。(BB):現在の輝度位置(xCurr, yCurr)を(xCb+(mvL[0] > 4)+ CtbSizeY > (CtbLog2SizeY - 1)<(CtbLog2SizeY - 1), ((yCb+(mvL[1] > 4) > (CtbLog2SizeY - 1)> (CtbLog2SizeY - 1)に等しく設定し、出力が FALSE に等しい現在の輝度位置(xCurr, yCurr)を入力として呼び出す。

【0 3 5 6】

このように、合計基準サイズはCTUである。

【0 3 5 7】

A 2. 現在の設計における潜在的な問題

【0 3 5 8】

現在の設計は、現在のVPDUを復号化するために64×64メモリを再使用することを想定し、IBC基準は、それに応じてVPDUメモリの再利用に合わせられる。このような設計は、VPDU復号化メモリをIBCバッファで束ねる。次のような問題が考えられる。

1. より小さなCTUサイズを扱うことが問題となり場合がある。CTUのサイズが32×32であるとする、現在のVPDUを復号化する現在の64×64メモリが、異なるアーキテクチャにおいて32×32レベルのメモリ再利用を効率的にサポートできるかどうかは、不明である。

2. 参照領域は有意に異なる。従って、あまりにも多くのビットストリーム適合性の制約が導入されている。これは、参照領域を効率的に利用し、合法的なビットストリームの生成を回避するために、エンコーダに余分な負担をかける。また、異なるモジュール、例えばマージリストにおいて無効なBVを有する可能性が高まる。これらの無効なBVを扱うことにより、余分なロジックまたは余分な適合性の制約を導入してしまう場合がある。これは、エンコーダまたはデコーダに負担をもたらすだけでなく、BV符号化とMV符号化との間に相違を生じさせる可能性がある。

3. 設計がうまく拡大縮小されていない。VPDU復号化はIBCバッファと混合されるので、現在の1つの128×128CTU設計に対して参照領域を増減させることは容易ではない。を利用することは、より低いまたはより高いプロファイルなどの、後の開発において、より優れた符号化効率とオンチップメモリのトレードオフを有効に利用するための柔軟性を制限する可能性がある。

4. IBC参照バッファのビット深度は、復号化バッファに接続される。通常、スクリーンコンテンツのビット深度は内部復号化のビット深度よりも小さいが、バッファは、大部分が丸められたり量子化されたりするノイズを表すビットを記憶するために、依然と

してメモリを費やす必要がある。より高い復号化ビット深度構成を考える場合、この問題はさらに深刻になる。

【0359】

A3. クリアなIBCバッファ設計

【0360】

上記サブセクションに記載された問題に対処するために、我々は、復号化メモリと混合されない専用のIBCバッファを有することを提案する。

【0361】

128×128CTUの場合、バッファは8ビットのサンプルを含む128×128として規定され、サイズw×hのCU(x, y)が復号化されていると、ループフィルタリング前のバッファの再構成は8ビットに変換され、位置(x%128, y%128)から始まるw×hブロック領域に書き込まれる。モデュロ演算子%は、常に正の数を返す。即ち、x<0であってx%Lを-(x%L)と定義すると、例えば、-3%128=125である。

【0362】

画素(x, y)がIBCモードでBV=(BVx, BVy)で符号化されると仮定すると、IBC参照バッファにおける予測サンプルは、((x+Bvx)%128, (y+Bvy)%128)に位置し、画素値は予測前に10ビットに変換される。

【0363】

バッファを(W, H)と見なす場合、(x, y)から始まるCTUまたはCUを復号化した後、ループフィルタリング前の再構成された画素を(x%W, y%H)から始まるバッファに記憶する。このように、CTUを復号化した後、対応するIBC参照バッファはそれに応じて更新される。このような設定は、CTUのサイズが128×128でない場合に発生することがある。例えば、64×64CTUの場合、現在のバッファサイズは、256×64バッファと見なすことができる。64×64CTUの場合のバッファの状態を図2に示す。

【0364】

図12は、IBC参照バッファステータスを示す図であり、1つのブロックは64×64CTUを示す。

【0365】

このような設計において、IBCバッファはVPDU復号化メモリとは異なるので、すべてのIBC参照バッファを参照として使用できる。

【0366】

IBCバッファのビット深度が8ビットである場合、3つの追加の10ビット64×64バッファを必要とする現在の設計に比べ、オンチップメモリの増加は、(8×4)/(10×3)-100%=6.7%である。

【0367】

ビット深度をさらに低減した場合メモリ要件をさらに低減することができる。例えば、7ビットバッファの場合、オンチップメモリの節約は、100%-(7×4)/(10×3)=6.7%である。

【0368】

この設計では、唯一のビットストリーム適合性の制約は、参照ブロックが現在のタイルの現在のCTU行における再構成された領域内にあるべきである、ということである。

【0369】

各CTU行の開始時に512への初期化が許可される場合、すべてのビットストリーム適合性の制約を除去することができる。

【0370】

A4. 実験結果

【0371】

いくつかの実施形態において、開示された方法は、VTM-4.0ソフトウェアを使用

10

20

30

40

50

して実装されてもよい。

【0372】

10ビットバッファ実装およびCTCの場合、このデコーダは、現在のVTM4.0エンコーダに完全に適合する。すなわち、これは、提案されたデコーダがVTM-4.0 CTCビットストリームを正確に復号化することができることを意味する。

【0373】

7ビットバッファの実装形態では、表1に示す結果が得られる。

【0374】

8ビットバッファの実装形態では、結果を表2に示す。

【0375】

10

20

30

40

50

【表 1 4】

表 1. 7ビットバッファを使用した場合の性能アンカはVTM-4.0で、すべてのシーケンスに対してIBCがオンになっている

	オールイントラ				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1	-0.01%	-0.09%	-0.10%	132%	101%
Class A2	0.05%	0.00%	0.06%	135%	100%
Class B	0.00%	-0.02%	0.01%	135%	100%
Class C	-0.02%	0.01%	0.03%	130%	98%
Class E	-0.13%	-0.16%	-0.04%	135%	99%
全体	-0.02%	-0.05%	0.00%	133%	100%
Class D	0.04%	0.04%	0.12%	127%	107%
Class F	-0.99%	-1.14%	-1.18%	115%	99%
4:2:0 TGM	-2.57%	-2.73%	-2.67%	104%	102%

10

	ランダムアクセス				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1	0.02%	-0.01%	0.01%	109%	100%
Class A2	0.00%	-0.04%	0.03%	111%	100%
Class B	-0.01%	-0.10%	-0.22%	113%	101%
Class C	-0.01%	0.17%	0.12%	115%	100%
Class E					
全体	0.00%	0.00%	-0.04%	112%	100%
Class D	0.05%	0.16%	0.20%	117%	101%
Class F	-0.71%	-0.77%	-0.77%	109%	99%
4:2:0 TGM	-1.81%	-1.65%	-1.64%	107%	101%

20

	低遅延 B				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.01%	0.36%	0.30%	114%	95%
Class C	-0.01%	-0.12%	-0.10%	120%	98%
Class E	0.10%	0.20%	0.18%	107%	99%
全体	0.03%	0.16%	0.13%	114%	97%
Class D	-0.01%	1.07%	0.18%	123%	104%
Class F	-0.79%	-0.89%	-1.01%	110%	100%
4:2:0 TGM	-1.36%	-1.30%	-1.26%	109%	102%

30

40

【0 3 7 6】

50

【表 15】

表 2. 8 ビットバッファでの性能。アンカはVTM-4. 0で、すべてのシーケンスに対してIBCがオンになっている

	オールイントラ				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1	-0.01%	0.02%	-0.10%	129%	102%
Class A2	0.02%	-0.06%	-0.02%	134%	102%
Class B	-0.04%	-0.02%	-0.07%	135%	101%
Class C	-0.03%	0.04%	0.00%	130%	98%
Class E	-0.16%	-0.14%	-0.08%	134%	100%
全体	-0.04%	-0.03%	-0.05%	133%	100%
Class D	0.00%	0.04%	0.02%	126%	101%
Class F	-1.31%	-1.27%	-1.29%	114%	98%
4:2:0 TGM	-3.23%	-3.27%	-3.24%	101%	100%

	ランダムアクセス				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1	-0.01%	-0.08%	0.04%	107%	99%
Class A2	-0.03%	-0.16%	0.06%	110%	99%
Class B	-0.01%	-0.14%	-0.22%	111%	99%
Class C	-0.01%	0.15%	0.09%	115%	100%
Class E					
Overall	-0.01%	-0.05%	-0.03%	111%	99%
Class D	0.01%	0.19%	0.22%	116%	101%
Class F	-1.01%	-0.99%	-1.01%	108%	99%
4:2:0 TGM	-2.33%	-2.14%	-2.19%	105%	100%

	低遅延 B				
	Over VTM-4.0 w/ IBC on				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.00%	0.04%	-0.14%	113%	#NUM!
Class C	-0.05%	-0.28%	-0.15%	119%	98%
Class E	0.04%	-0.16%	0.43%	107%	#NUM!
全体	0.00%	-0.11%	0.00%	113%	#NUM!
Class D	-0.07%	1.14%	0.13%	122%	99%
Class F	-0.81%	-0.92%	-0.96%	111%	99%
4:2:0 TGM	-1.71%	-1.67%	-1.71%	106%	95%

【0377】

図 17 は、本明細書で開示される様々な技術が実装され得る例示的な映像処理システム 1700 を示すブロック図である。様々な実装形態は、システム 1700 のモジュールの一部又は全部を含んでもよい。システム 1700 は、映像コンテンツを受信するための入力ユニット 2802 を含んでもよい。映像コンテンツは、未加工又は非圧縮フォーマット、例えば、8 又は 10 ビットのマルチモジュール画素値で受信されてもよく、又は圧縮又は符号化フォーマットで受信されてもよい。入力ユニット 1702 は、ネットワークインターフェース、周辺バスインターフェース、又は記憶インターフェースを表してもよい。ネットワークインターフェースの例は、イーサネット（登録商標）、パッシブ光ネットワ

ーク（PON）等の有線インターフェース、およびWi-Fi（登録商標）またはセルラーインターフェース等の無線インターフェースを含む。

【0378】

システム1700は、本明細書に記載される様々な符号化又は符号化方法を実装することができる符号化モジュール1704を含んでもよい。符号化モジュール1704は、入力ユニット1702からの映像の平均ビットレートを符号化モジュール1704の出力に低減し、映像の符号化表現を生成してもよい。従って、この符号化技術は、映像圧縮または映像コード変換技術と呼ばれることがある。符号化モジュール1704の出力は、モジュール1706によって表されるように、記憶されてもよいし、接続された通信を介して送信されてもよい。入力ユニット1702において受信された、記憶された又は通信された映像のビットストリーム（又は符号化）表現は、モジュール1708によって使用されて、表示インターフェースユニット1710に送信される画素値又は表示可能な映像を生成してもよい。ビットストリーム表現からユーザが見ることができる映像を生成する処理は、映像伸張（映像展開）と呼ばれることがある。さらに、特定の映像処理動作を「符号化」動作又はツールと呼ぶが、符号化ツール又は動作は、エンコーダ及びそれに対応する、復号化の結果を逆にする復号化ツール又は動作が、デコーダによって行われることが理解されよう。

【0379】

周辺バスインターフェースユニットまたは表示インターフェースユニットの例は、ユニバーサルシリアルバス（USB）または高精細マルチメディアインターフェース（HDMI）（登録商標）またはディスプレイポート等を含んでもよい。ストレージインターフェースの例は、シリアルアドバンスドテクノロジーアタッチメント（SATA）、PCI、IDEインターフェース等を含む。本明細書に記載される技術は、携帯電話、ノートパソコン、スマートフォン、又はデジタルデータ処理及び／又は映像表示を実施可能な他のデバイス等の様々な電子デバイスに実施されてもよい。

【0380】

図18は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態1に関連して説明する。ステップ1802において、処理はイントラブロックコピーモードにおいて、予測のための参照サンプルを記憶するバッファサイズを判定する。ステップ1804において、処理は、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0381】

図19は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態4に関連して説明する。ステップ1902において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定し、このバッファは、ループフィルタリングステップの前に再構成サンプルを記憶するために使用される。ステップ1904において、処理は、バッファに記憶された再構成サンプルを使用してこの変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0382】

図20は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態5に関連して説明する。ステップ2002において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定し、このバッファは、ループフィルタ

リングステップの後に再構成サンプルを記憶するために使用される。ステップ 2 0 0 4 において、処理は、バッファに記憶された再構成サンプルを使用してこの変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0383】

図 2 1 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 6 に関連して説明する。ステップ 2 1 0 2 において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための再構成サンプルを記憶するバッファを判定し、このバッファが、ループフィルタリングステップの前とループフィルタリングステップの後の両方に再構成サンプルを記憶するために使用される。ステップ 2 1 0 4 において、処理は、バッファに記憶された再構成サンプルを使用してこの変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0384】

図 2 2 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 7 に関連して説明する。ステップ 2 2 0 2 において、処理は、バッファを使用してイントラブロックコピーモードにおいて、予測のための参照サンプルを記憶し、バッファの第 1 のビット深度は、ビットストリーム表現において視覚メディアデータを表すために使用される第 2 のビット深度とは異なる。ステップ 2 2 0 4 において、処理は、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0385】

図 2 3 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 8 に関連して説明する。ステップ 2 3 0 2 において、処理は、バッファを初期化してイントラブロックコピーモードにおいて、予測のための参照サンプルを記憶し、バッファは、第 1 の値で初期化される。ステップ 2 3 0 4 において、処理は、バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0386】

図 2 4 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 9 に関連して説明する。ステップ 2 4 0 2 において、処理は、バッファを初期化してイントラブロックコピーモードにおいて、予測のための参照サンプルを記憶し、視覚メディアデータにおける 1 つ以上の映像ブロックの可用性に基づいて、バッファが視覚メディアデータにおける 1 つ以上の画素値で初期化される。ステップ 2 4 0 4 において、処理は、バッファに記憶された参照サンプルを使用して、視覚メディアデータの、1 つ以上の映像ブロックに属さない現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行い、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。

【0387】

図 2 5 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 1 4 c に関連して説明する。ステップ

2502において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定する。ステップ2504において、処理は、バッファに記憶された参照サンプルを使用して変換が行われ、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで変換を行う。ステップ2506において、処理は、参照位置 ($P \bmod M$, $Q \bmod N$)、(但し $\bmod$ はモデュロ演算、 M および N は、バッファの x および y 寸法を表す整数である)に基づいてバッファ内の対応する参照を計算することを含み、参照位置 (P , Q) は、位置 (x_0 , y_0) に空間的に配置され、かつ動き情報に含まれるブロックベクトル (BV_x , BV_y) を有する画素について、ブロックベクトル (BV_x , BV_y) と位置 (x_0 , y_0) を使用して決定される。

10

【0388】

図26は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態14a~14bに関連して説明する。ステップ2602において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定する。ステップ2604において、処理は、バッファに記憶された参照サンプルを使用して変換が行われ、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで変換を行う。ステップ2606において、処理は、参照位置 (P , Q) に基づいてバッファ内の対応する参照を計算し、参照位置 (P , Q) は、位置 (x_0 , y_0) に空間的に配置され、かつ動き情報に含まれるブロックベクトル (BV_x , BV_y) を有する画素について、ブロックベクトル (BV_x , BV_y) と位置 (x_0 , y_0) を使用して決定される。

20

【0389】

図27は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態10に関連して説明する。ステップ2702において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測するための参照サンプルを記憶するバッファを判定し、 x および y の数字を使用して、前記バッファ内の画素位置がアドレス指定される。ステップ2704において、処理は、 x と y の数字に基づいて、バッファに記憶された参照サンプルを使用して変換を行い、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで変換が行われる。

30

【0390】

図28は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態15に関連して説明する。ステップ2802において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードで予測のための参照サンプルを記憶するバッファを判定し、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。ステップ2804において、処理は、参照位置 (P , Q) において、バッファ内の対応する参照を計算し、参照位置 (P , Q) は、現在の映像ブロックの位置 (x_0 , y_0) に空間的に配置され、かつブロックベクトル (BV_x , BV_y) を有する画素について、ブロックベクトル (BV_x , BV_y) と位置 (x_0 , y_0) を使用して決定される。ステップ2806において、処理は、参照位置 (P , Q) がバッファの外にあると判定されると、バッファ内のサンプルを使用して参照位置を再計算する。

40

【0391】

50

図29は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態16に関連して説明する。ステップ2902において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードにおける予測に対する参照サンプルを記憶するバッファを判定し、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。ステップ2904において、処理は、参照位置(P, Q)におけるバッファにおいて、対応する参照を計算し、参照位置(P, Q)は、現在の映像ブロックを含む符号化ツリーユニットの左上位置に対する現在の映像ブロックの位置(x_0 , y_0)に空間的に配置され、かつブロックベクトル(BV_x , BV_y)を有する画素について、ブロックベクトル(BV_x , BV_y)と位置(x_0 , y_0)を使用して決定される。ステップ2906において、処理は、参照位置(P, Q)がバッファの外にあると判定した場合、参照位置の少なくとも一部が予め規定された範囲内にあるように制約する。

【0392】

図30は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態17に関連して説明する。ステップ3002において、処理は、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードにおける予測に対する参照サンプルを記憶するバッファを判定し、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。ステップ3004において、処理は、バッファにおける参照位置(P, Q)における対応する参照を計算し、ブロックベクトル(BV_x , BV_y)および位置(x_0 , y_0)を使用して、現在の映像ブロックを含み、ブロックベクトル(BV_x , BV_y)を有する符号化ツリーユニットの左上位置に対して現在の映像ブロックの位置(x_0 , y_0)に空間的に位置する画素の参照位置(P, Q)を判定する。ステップ3006において、処理は、ブロックベクトル(BV_x , BV_y)がバッファの外にあると判定した場合、バッファの内側のサンプル値のブロックベクトルに基づいて、ブロックベクトル(BV_x , BV_y)をパディングする。

【0393】

図31は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態30に関連して説明する。ステップ3102において、処理は、映像とその映像のビットストリーム表現との間での変換中に、映像境界におけるイントラブロックコピーモードで予測のために参照サンプルを記憶するバッファをリセットする。ステップ3104において、処理は、バッファに記憶された参照サンプルを使用して映像の映像ブロックの変換が行われ、参照ピクチャを参照することなく、映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで変換を行う。

【0394】

図32は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第4章の例示的な実施形態34に関連して説明する。ステップ3202において、処理は、現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間で変換を行う。ステップ3204において、処理は、イントラブロックコピーモードにおいて予測のための参照サンプルを記憶するために使用されるバッファを更新することを含み、このバッファは、後続の映像ブロックと後続の映像ブロックのビットストリーム表現との間での変換のために使用され、後続の映像ブロックと後続の映像ブロックのビットストリーム表現との間での変換は、参照ピクチャを参照することなく、後続の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで行われる。

【0395】

図 3 3 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 3 9 に関連して説明する。ステップ 3 3 0 2 において、処理は、現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換のため、イントラブロックコピーモードにおける予測に対する再構成サンプルを記憶するために使用するバッファを判定し、この変換は、参照ピクチャを参照することなく、現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づくイントラブロックコピーモードで行われる。ステップ 3 3 0 4 において、処理は、バッファに記憶された再構成されたサンプルが変換中にサンプル値を予測するために使用されると判定すると、バッファに記憶された再構成されたサンプルに前処理演算を行うことと、を含む。

10

【0 3 9 6】

図 3 4 は、映像データ処理方法の一例を示すフローチャートである。このフローチャートのステップは、本願の第 4 章の例示的な実施形態 4 2 に関連して説明する。ステップ 3 4 0 2 において、映像領域の現在の仮想パイプラインデータユニット (V P D U) の現在の映像ブロックと、現在の映像ブロックのビットストリーム表現との間での変換のため、選択的に、映像領域の偶数行目の K 1 前処理済み V P D U および / または奇数行目の K 2 前処理済み V P D U を使用するかどうかを判定する。ステップ 3 4 0 4 において、処理は、変換を行い、変換は、現在の V P D U の残りをを使用することを排除し、参照ピクチャを参照することなく、映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づいて、イントラブロックコピーモードで変換が行われる。

20

【0 3 9 7】

本明細書のいくつかの実施形態は、項に基づく形式で提示される。

【0 3 9 8】

1. 視覚メディア処理方法であって、

イントラブロックコピーモードにおいて、予測のための参照サンプルを記憶するバッファサイズを判定することと、

前記バッファに記憶された参照サンプルを使用して、視覚メディアデータの現在の映像ブロックと現在の映像ブロックのビットストリーム表現との間での変換を行うこととを含み、前記変換は、参照ピクチャを参照することなく、前記現在の映像ブロックと同じ映像領域に位置する再構成ブロックに関する動き情報に基づく前記イントラブロックコピーモードで行われる、視覚メディア処理方法。

30

【0 3 9 9】

2. 前記バッファサイズが所定の定数である、第 1 項に記載の方法。

【0 4 0 0】

3. 前記バッファのサイズが $M \times N$ であり、 M および N が整数である、第 1 項から 2 項のいずれか 1 項以上に記載の方法。

【0 4 0 1】

4. $M \times N$ が 64×64 または 128×128 または 64×128 または 96×128 または 128×96 に等しい、第 3 項に記載の方法。

【0 4 0 2】

5. 前記バッファまたはその一部のサイズは、前記現在の映像ブロックまたはその一部を含む符号化ツリーユニットのサイズに等しい、第 1 項に記載の方法。

40

【0 4 0 3】

6. 前記バッファの高さは、前記現在の映像ブロックを含む符号化ツリーユニットの高さに等しい、第 1 項に記載の方法。

【0 4 0 4】

7. 前記バッファサイズは、前記現在の映像ブロックを含む符号化ツリーユニットのサイズに比例対応する、第 1 項に記載の方法。

【0 4 0 5】

8. 整数 m 、 n で表される比例対応は、C T U のサイズが 128×128 の場合、 m

50

= 1、 $n = 1$ であり、CTUのサイズが 64×64 の場合、 $m = 4$ 、 $n = 1$ であり、CTUのサイズが 32×32 の場合、 $m = 16$ 、 $n = 1$ であり、CTUのサイズが 16×16 の場合、 $m = 64$ 、 $n = 1$ となるように、前記現在の映像ブロックを含む符号化ツリーユニット (CTU) のサイズに基づいており、 n は、前記バッファの高さ／前記CTUの高さに等しく、 m は、前記バッファの幅／前記CTUの幅に等しい、第7項に記載の方法。

【0406】

9. 前記バッファの高さと前記バッファの幅との積が、所定の定数である、第1項に記載の方法。

【0407】

10. 前記バッファサイズは、前記変換に使用される仮想パイプラインデータユニットに比例対応する、第1項に記載の方法。

10

【0408】

11. 前記バッファサイズは、前記変換に使用される仮想パイプラインデータユニットのサイズに等しい、第1項に記載の方法。

【0409】

12. 前記バッファサイズは、前記ビットストリーム表現に含まれるフィールドに対応する、第1項に記載の方法。

【0410】

13. 前記フィールドは、映像またはシーケンスまたはピクチャまたはスライスまたはタイルグループレベルでの前記ビットストリーム表現に含まれる、第12項に記載の方法。

20

【0411】

14. 前記フィールドは、映像パラメータセット、シーケンスパラメータセット又はピクチャパラメータセット、又はピクチャヘッダ、スライスヘッダ、又はタイルグループヘッダレベルに含まれる、第13項に記載の方法。

【0412】

15. 前記バッファは、1つの色成分または複数の色成分に関連付けられた参照サンプルを記憶する、第1～14項のいずれか1項以上に記載の方法。

【0413】

16. 2つ以上のバッファが維持される、第1～15項のいずれか1項以上に記載の方法。

30

【0414】

17. 第1のバッファおよび第2のバッファが維持され、第1のサイズを有する前記第1のバッファは、輝度成分に関連付けられた参照サンプルに使用され、第2のサイズを有する前記第2のバッファは、クロマ成分に関連付けられた参照サンプルに使用され、前記第1のバッファは前記第2のバッファとは異なる、第16項に記載の方法。

【0415】

18. 前記第2のサイズは前記第1のサイズに依存する、第17項に記載の方法。

【0416】

19. 前記第2のサイズは前記第1のサイズから独立している、第17項に記載の方法。

40

【0417】

20. 前記第2のサイズは、現在の映像ブロックのクロマサブサンプリングフォーマットに依存する、第17～18項のいずれか1項以上に記載の方法。

【0418】

21. 前記第2のサイズが前記第1のサイズに等しい、第18項に記載の方法。

【0419】

22. 前記参照サンプルはRGBフォーマットで記憶される、第1～21項のいずれかに記載の方法。

【0420】

50

23. 前記バッファサイズは、 64×64 または 128×128 または 64×128 または 128×64 に等しい、第22項に記載の方法。

【0421】

24. 前記バッファサイズは、前記変換に関連付けられた仮想パイプラインデータユニット (VPDU) のサイズおよび／または前記現在の映像ブロックに関連付けられた符号化ツリーブロックのサイズに少なくとも部分的に基づく、第1項に記載の方法

【0422】

25. 前記バッファの前記サイズが前記符号化ツリーブロックの前記サイズに等しい、第24項に記載の方法。

【0423】

26. 前記バッファの高さは、前記現在の映像ブロックに関連付けられた前記符号化ツリーユニットの高さに等しい、第24項に記載の方法。

【0424】

27. 前記バッファサイズは高さを含み、前記高さは、(i) 所定の値と (ii) 前記現在の前記映像ブロックに関連付けられた前記符号化ツリーブロックのサイズとの最小値に等しい、第24項に記載の方法。

【0425】

28. 前記所定の値は64である、第27項に記載の方法。

【0426】

29. 前記バッファの幅は、 $128 * 128 / vsize$ に等しく、 $vsize$ は、前記VPDUのサイズを表す、第24項に記載の方法。

【0427】

30. 前記バッファの高さは $\min(ctbSize, 64)$ に等しく、 $ctbSize$ は前記符号化ツリーブロックのサイズを表し、 $\min(x, y)$ は x および y の最小値を表す、第24項に記載の方法。

【0428】

31. 前記変換は、前記現在の映像ブロックから前記ビットストリーム表現を生成することを含む、第1～30項のいずれかに記載の方法。

【0429】

32. 前記変換は、前記ビットストリーム表現から前記現在の映像ブロックの画素値を生成することを含む、第1～30項のいずれかに記載の方法。

【0430】

33. 請求項1～30の1項以上に記載の方法を実装するように構成された処理装置を備える映像エンコーダ装置。

【0431】

34. 第1～30項の1項以上に記載の方法を実装するように構成された処理装置を備える、映像デコーダ装置。

【0432】

35. コードが記憶されたコンピュータ可読媒体であって、前記コードは第1～30項のいずれか又は複数項に記載の方法を実装するための処理装置が実行可能な命令を実施する、コンピュータ可読媒体。

【0433】

本明細書では、「映像処理」という用語は、映像符号化、映像復号化、映像圧縮、または映像展開を指すことができる。例えば、映像圧縮アルゴリズムは、映像の画素表現から対応するビットストリーム表現への変換、またはその逆の変換中に適用されてもよい。現在の映像ブロックのビットストリーム表現は、例えば、構文によって規定されるように、ビットストリーム内の同じ場所または異なる場所に拡散されるビットに対応していてもよい。例えば、1つのマクロブロックは、変換および符号化された誤り残差値の観点から、且つビットストリームにおけるヘッダおよび他のフィールドにおけるビットを使用して符号化されてもよい。

10

20

30

40

50

【0434】

以上、説明の目的で本開示の技術の特定の実施形態を説明したが、本発明の範囲から逸脱することなく様々な修正が可能であることは、理解されるであろう。従って、本開示の技術は、添付の特許請求の範囲による場合を除き、限定されない。

【0435】

本特許明細書に記載された主題および機能操作の実装形態は、本明細書に開示された構造およびその構造的等価物を含め、様々なシステム、デジタル電子回路、またはコンピュータソフトウェア、ファームウェア、若しくはハードウェアで実施されてもよく、またはそれらの1つ以上の組み合わせで実施してもよい。本明細書に記載された主題の実装形態は、1つ以上のコンピュータプログラム製品、すなわち、データ処理装置によって実行されるため、又はデータ処理装置の操作を制御するために、有形で非可搬性のコンピュータ可読媒体上に符号化されたコンピュータプログラム命令の1つ以上のモジュールとして実装することができる。このコンピュータ可読媒体は、機械可読記憶装置、機械可読記憶基板、メモリデバイス、機械可読伝播信号をもたらす物質の組成物、又はこれらの1つ以上の組み合わせであってもよい。「データ処理ユニット」又は「データ処理装置」という用語は、例えば、プログラマブル処理装置、コンピュータ、又は複数の処理装置若しくはコンピュータを含め、データを処理するためのすべての装置、デバイス、および機械を含む。この装置は、ハードウェアの他に、当該コンピュータプログラムの実行環境を作るコード、例えば、処理装置ファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、又はこれらの1つ以上の組み合わせを構成するコードを含むことができる。

【0436】

コンピュータプログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、又はコードとも呼ばれる）は、コンパイルされた言語又は解釈された言語を含む任意の形式のプログラミング言語で記述することができ、また、それは、スタンドアロンプログラムとして、又はコンピューティング環境で使用するのに適したモジュール、コンポーネント、サブルーチン、又は他のユニットとして含む任意の形式で展開することができる。コンピュータプログラムは、必ずしもファイルシステムにおけるファイルに対応するとは限らない。プログラムは、他のプログラム又はデータを保持するファイルの一部（例えば、マークアップ言語文書に格納された1つ以上のスクリプト）に記録されていてもよいし、当該プログラム専用の単一のファイルに記憶されていてもよいし、複数の調整ファイル（例えば、1つ以上のモジュール、サブプログラム、又はコードの一部を格納するファイル）に記憶されていてもよい。1つのコンピュータプログラムを、1つのサイトに位置する1つのコンピュータ、又は複数のサイトに分散され通信ネットワークによって相互接続される複数のコンピュータで実行させるように展開することも可能である。

【0437】

本明細書に記載された処理およびロジックフローは、入力データ上で動作し、出力を生成することによって機能を実行するための1つ以上のコンピュータプログラムを実行する1つ以上のプログラマブル処理装置によって行うことができる。処理及びロジックフローはまた、特定用途のロジック回路、例えば、FPGA（フィールドプログラマブルゲートアレイ）又はASIC（特定用途向け集積回路）によって行うことができ、装置はまた、特別目的のロジック回路として実装することができる。

【0438】

コンピュータプログラムの実行に適した処理装置は、例えば、汎用及び専用マイクロ処理装置の両方、並びに任意の種類のデジタルコンピュータの任意の1つ以上の処理装置を含む。一般的に、処理装置は、読み出し専用メモリ又はランダムアクセスメモリ又はその両方から命令及びデータを受信する。コンピュータの本質的な要素は、命令を実行するためのプロセッサと、命令およびデータを記憶するための1つ以上の記憶装置とである。一般的に、コンピュータは、データを記憶するための1つ以上の大容量記憶デバイス、例えば、磁気、光磁気ディスク、又は光ディスクを含んでもよく、又はこれらの大容量記憶デ

バイスからデータを受信するか、又はこれらにデータを転送するように動作可能に結合されてもよい。しかしながら、コンピュータは、このようなデバイスを有する必要はない。コンピュータプログラム命令およびデータを記憶するのに適したコンピュータ可読媒体は、あらゆる形式の不揮発性メモリ、媒体、およびメモリデバイスを含み、例えば、E P R O M、E E P R O M、フラッシュメモリデバイス等の半導体メモリデバイスを含む。処理装置及びメモリは、特定用途のロジック回路によって補完されてもよく、又は特定用途のロジック回路に組み込まれてもよい。

【0439】

本明細書は、図面とともに、例示のみを目的とするものであり、例示的とは例を意味することが意図される。本明細書において、「または」の使用は、文脈からそうでないことが明確に示されていない限り、「および／または」を含むことが意図される。

10

【0440】

本特許明細書は多くの詳細を含むが、これらは、任意の発明の範囲又は特許請求の範囲を限定するものと解釈されるべきではなく、むしろ、特定の発明の特定の実施形態に特有であり得る特徴の説明と解釈されるべきである。本特許文献において別個の実施形態の文脈で説明されている特定の特征是、1つの例において組み合わせて実装してもよい。逆に、1つの例の文脈で説明された様々な特征是、複数の実施形態において別個に又は任意の適切なサブコンビネーションで実装してもよい。さらに、特征是、特定の組み合わせで作用するものとして上記に記載され、最初にそのように主張されていてもよいが、主張された組み合わせからの1つ以上の特征是、場合によっては、組み合わせから抜粋されることができ、主張された組み合わせは、サブ組み合わせまたはサブ組み合わせのバリエーションに向けられてもよい。

20

【0441】

同様に、動作は図面において特定の順番で示されているが、これは、所望の結果を達成するために、このような動作が示された特定の順番で又は連続した順番で実行されること、又は示された全ての操作が実行されることを必要とするものと理解されるべきではない。また、本特許明細書に記載されている例における様々なシステムの構成要素の分離は、全ての実施形態においてこのような分離を必要とするものと理解されるべきではない。

【0442】

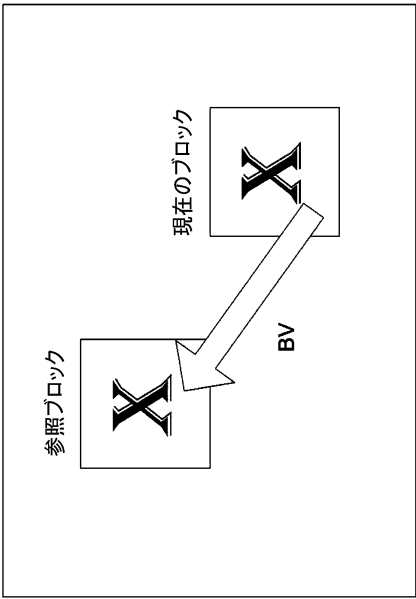
いくつかの実装形態及び例のみが記載されており、この特許文献に記載され図示されている内容に基づいて、他の実施形態、拡張及び変形が可能である。

30

40

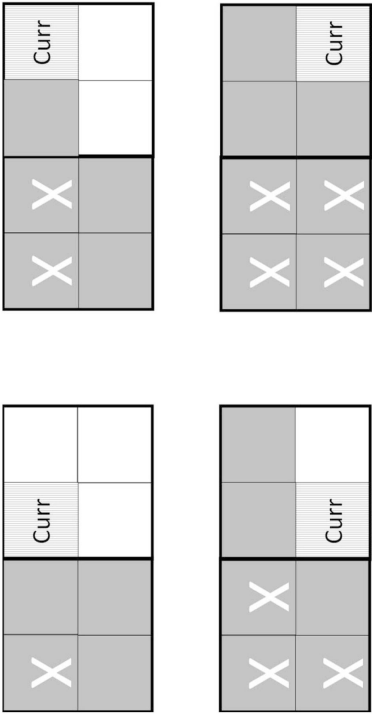
50

【図面】
【図 1】



現在のピクチャ

【図 2】



10

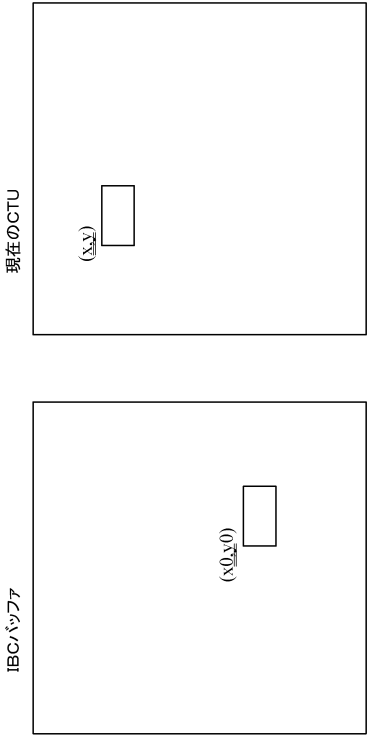
20

30

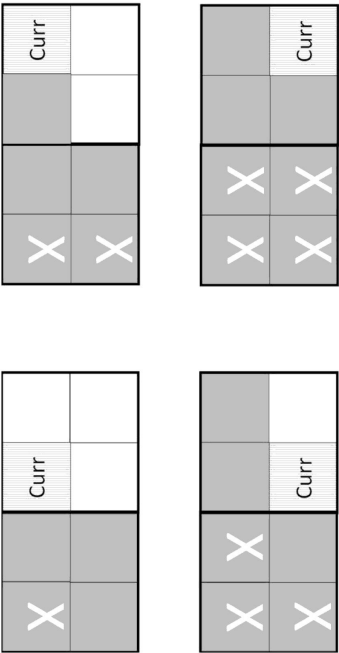
40

50

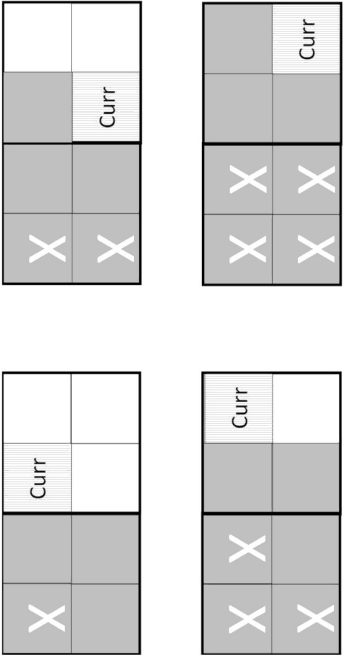
【図 3】



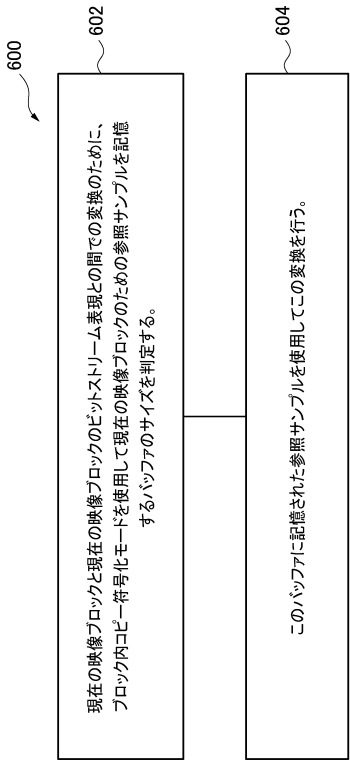
【図 4】



【図 5】



【図 6】



10

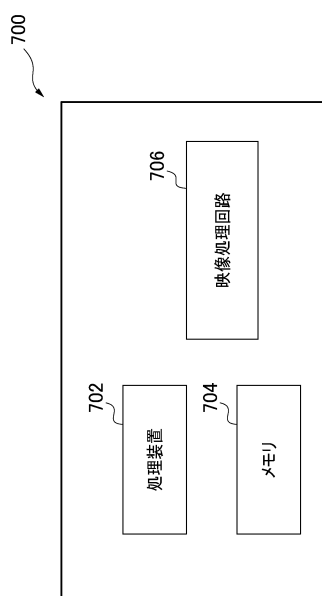
20

30

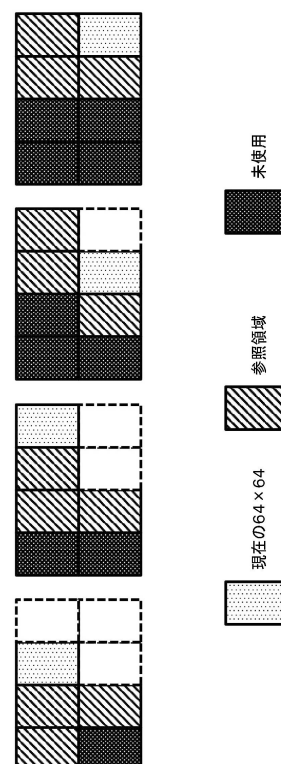
40

50

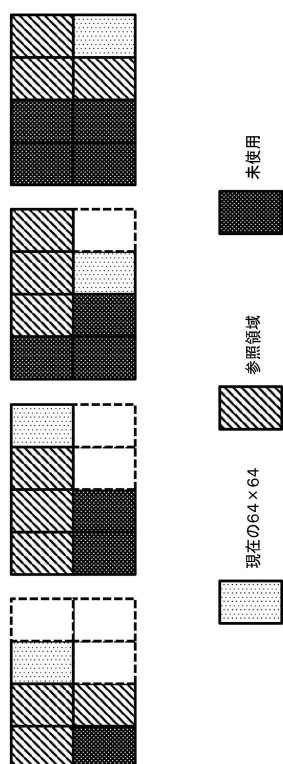
【圖 7】



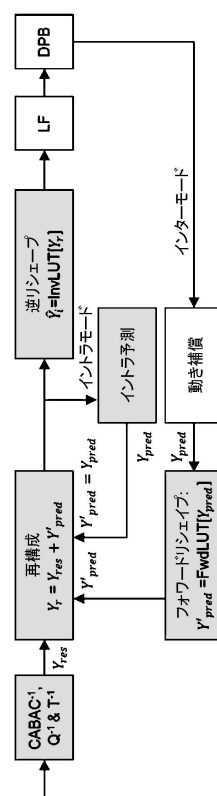
【圖 8】



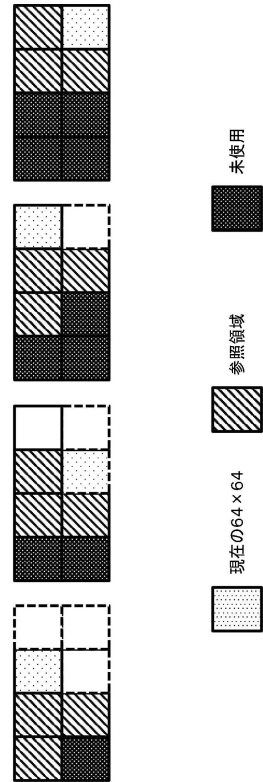
【圖 9】



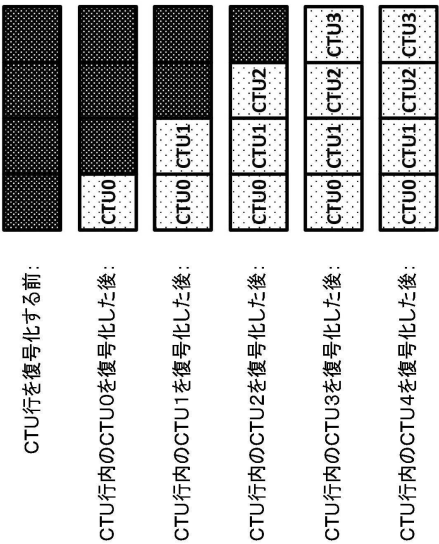
【図 10】



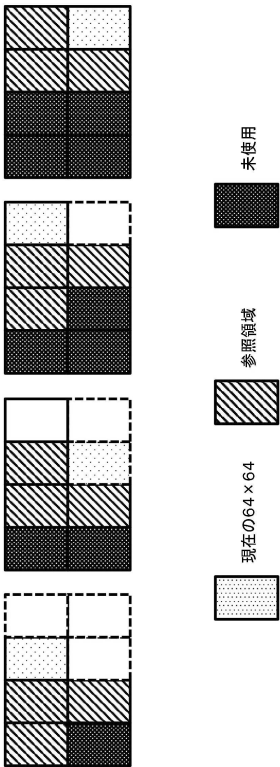
【図 1 1】



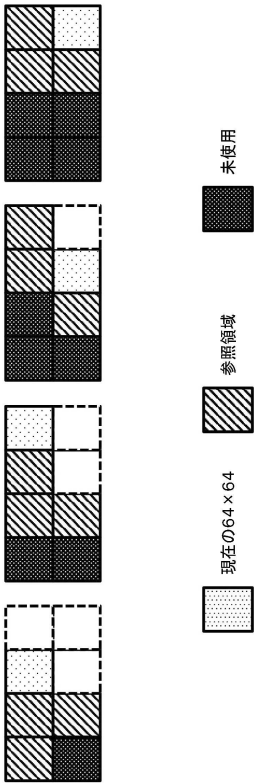
【図 1 2】



【図 1 3】



【図 1 4】



10

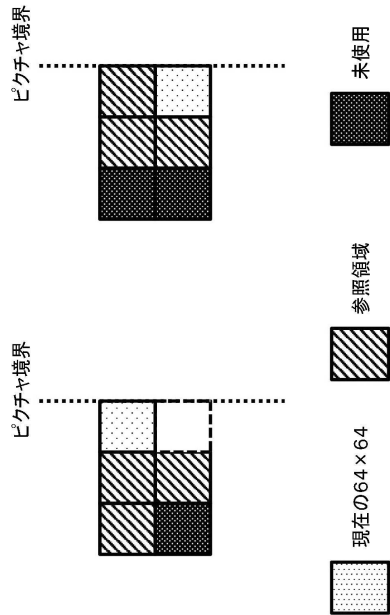
20

30

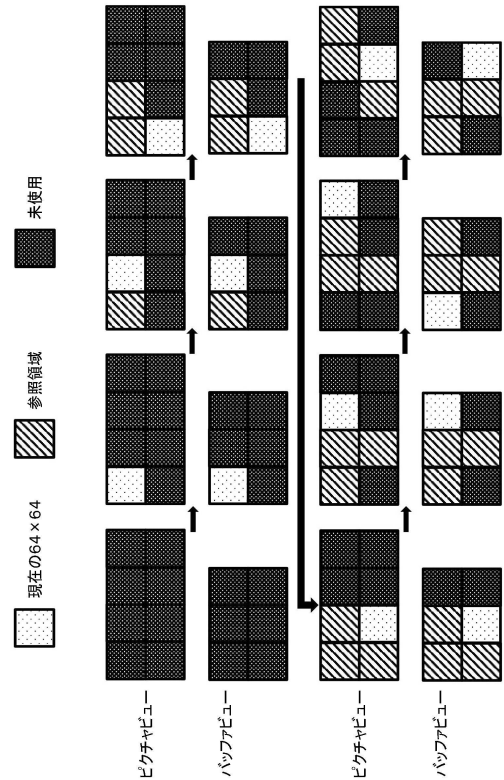
40

50

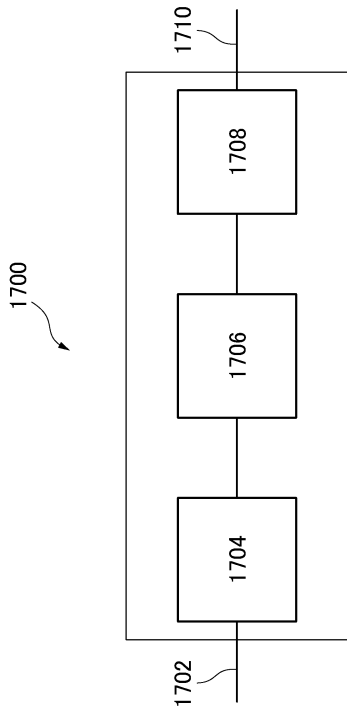
【図 1 5】



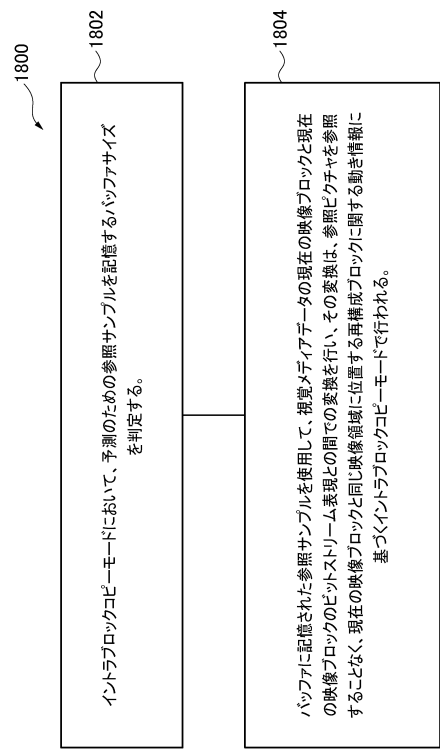
【図 1 6】



【図 1 7】



【図 1 8】



10

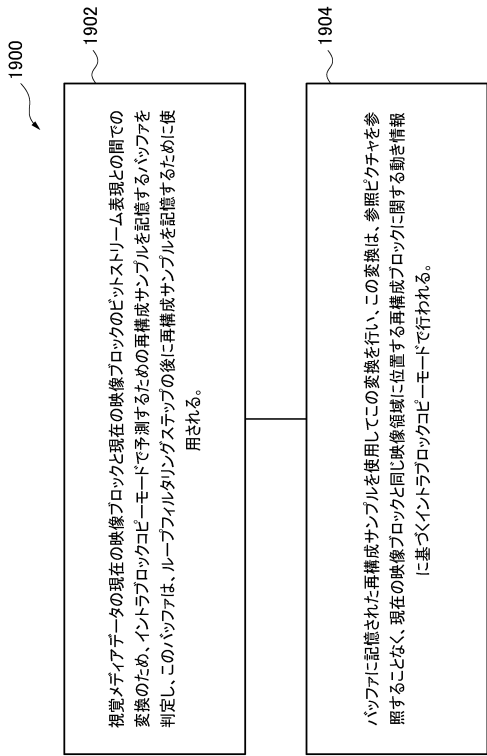
20

30

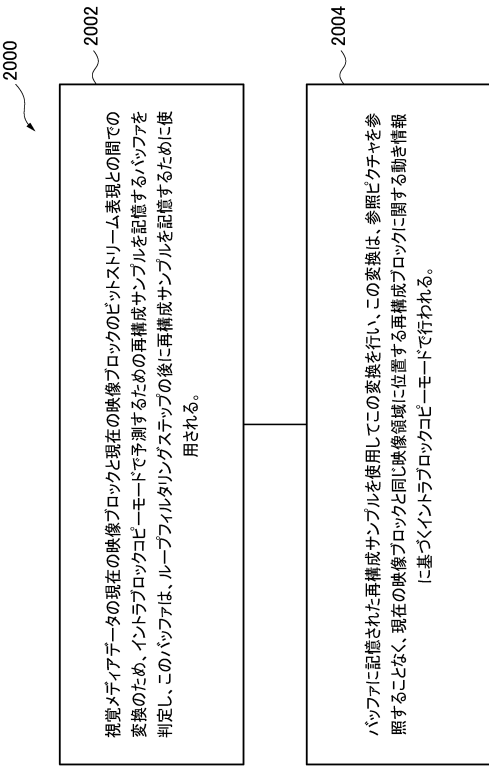
40

50

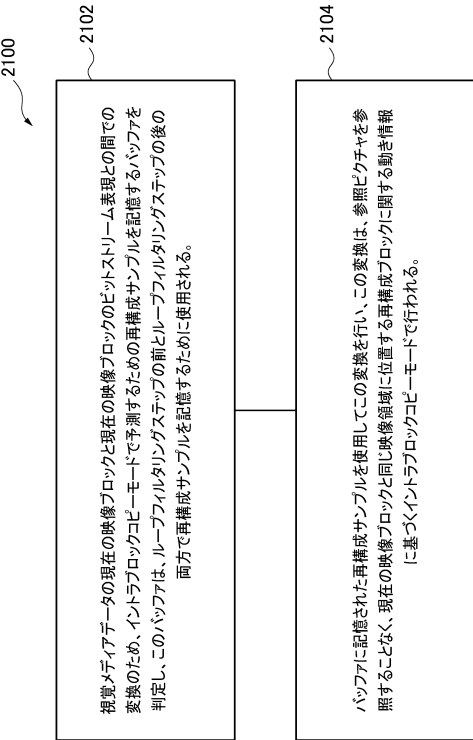
【図 19】



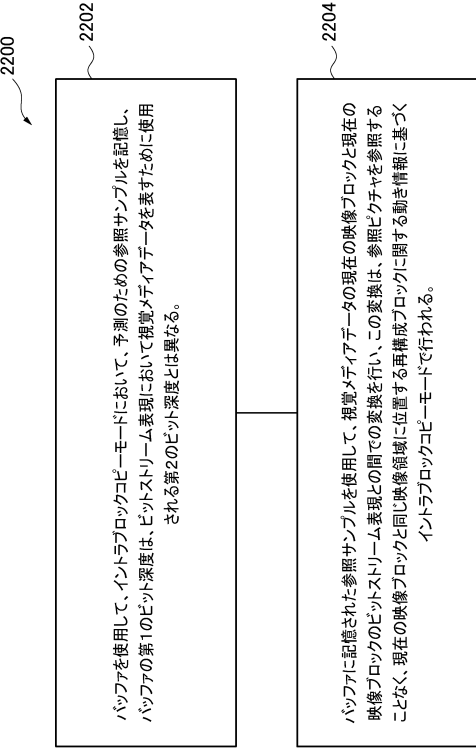
【図 20】



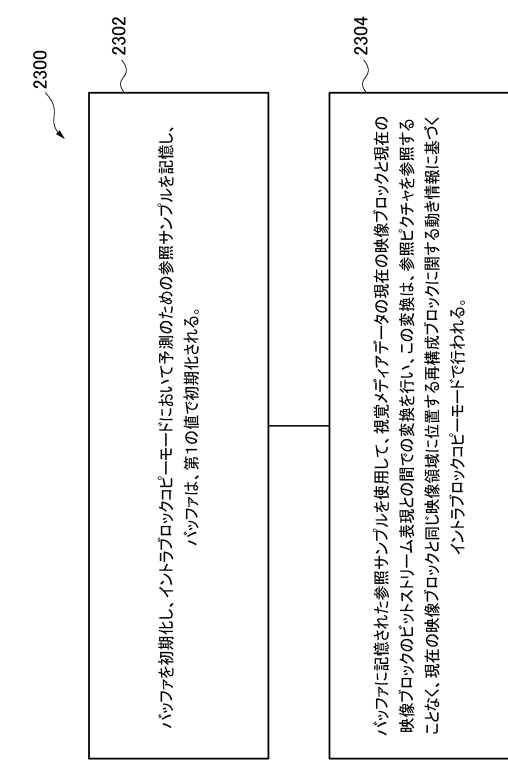
【図 21】



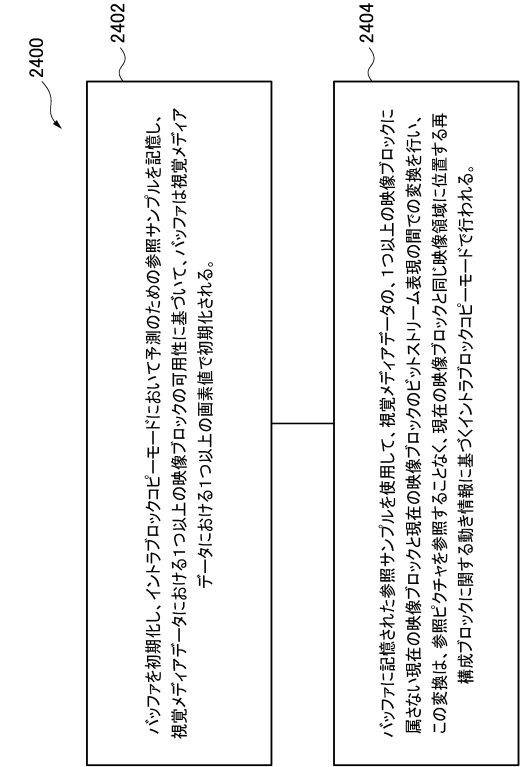
【図 22】



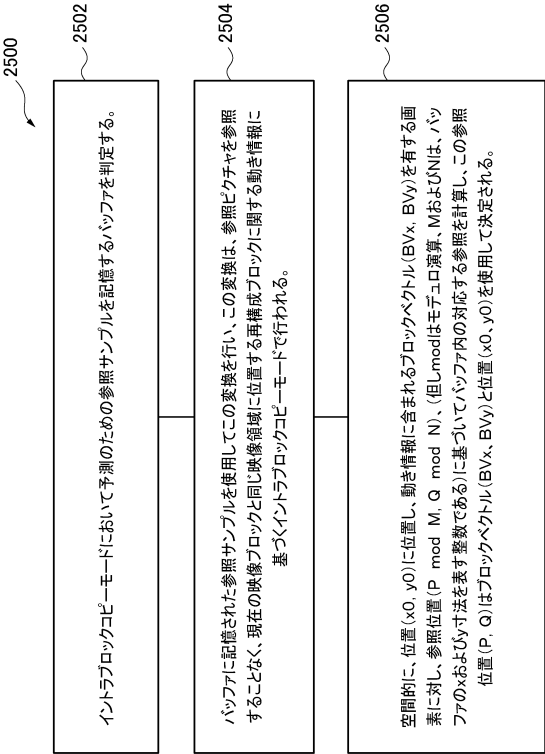
【図 2 3】



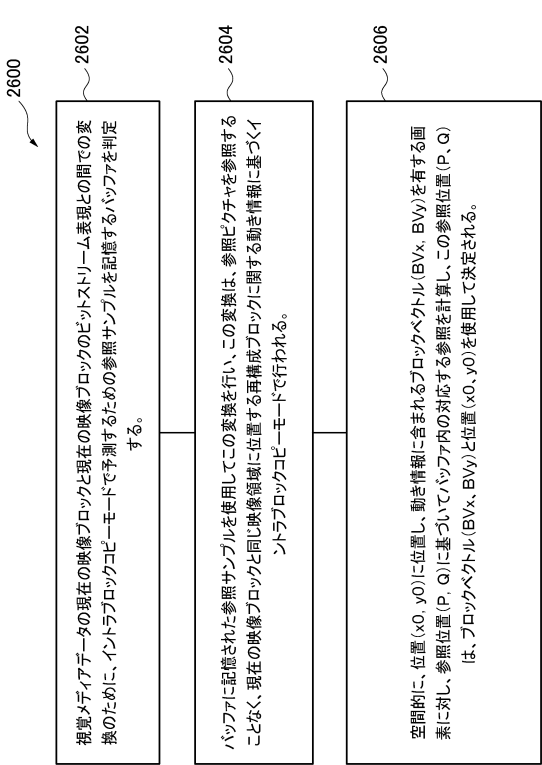
【図 2 4】



【図 2 5】



【図 2 6】



10

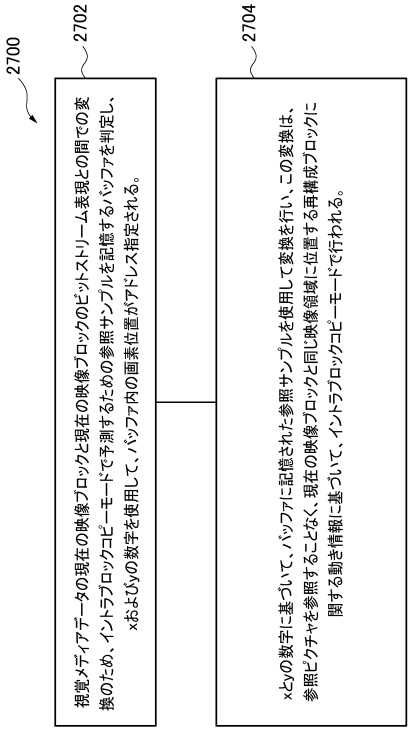
20

30

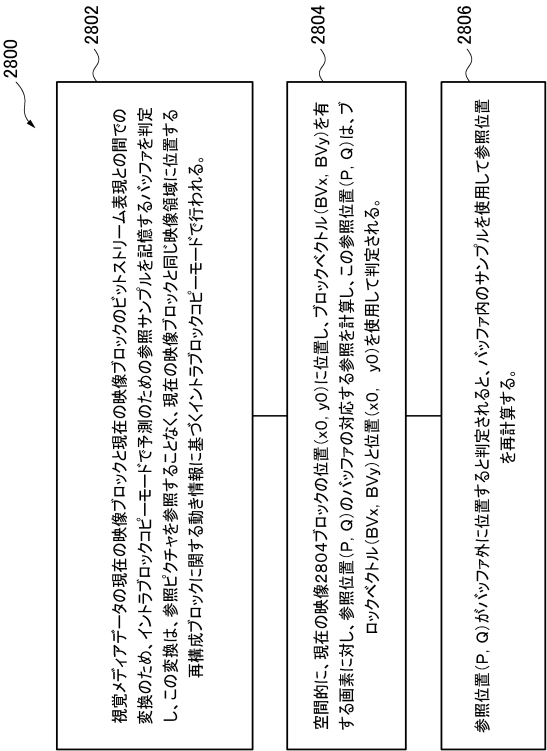
40

50

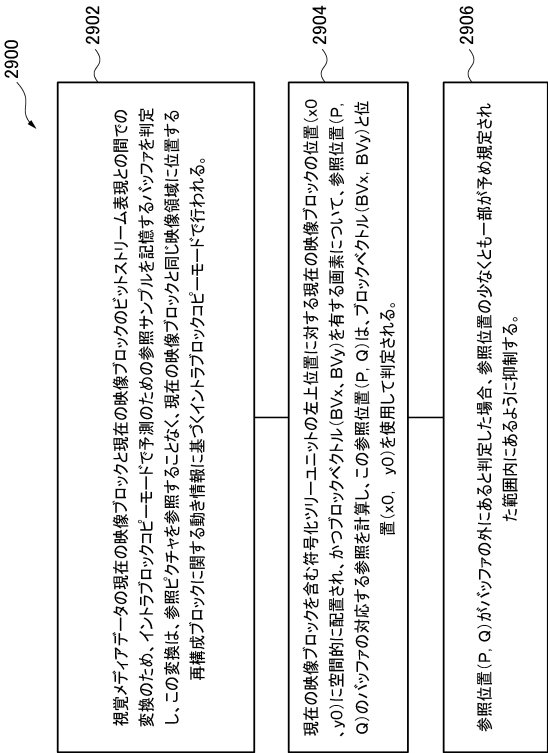
【図 27】



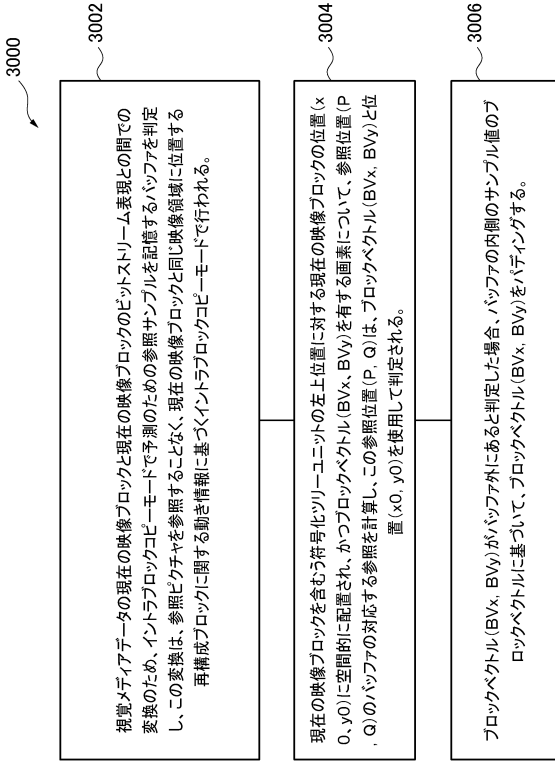
【図 28】



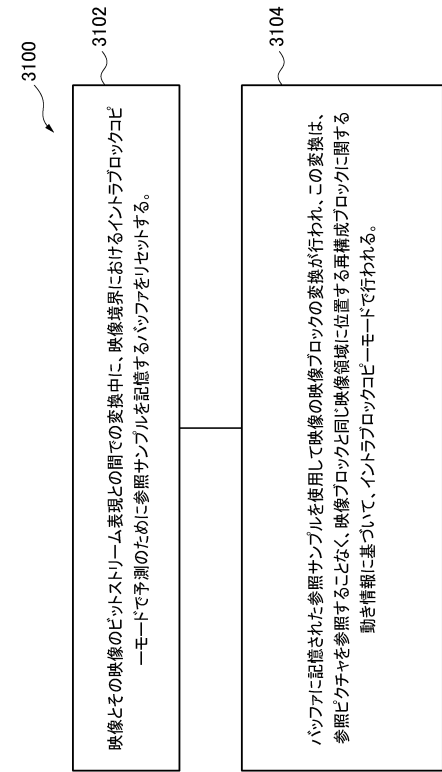
【図 29】



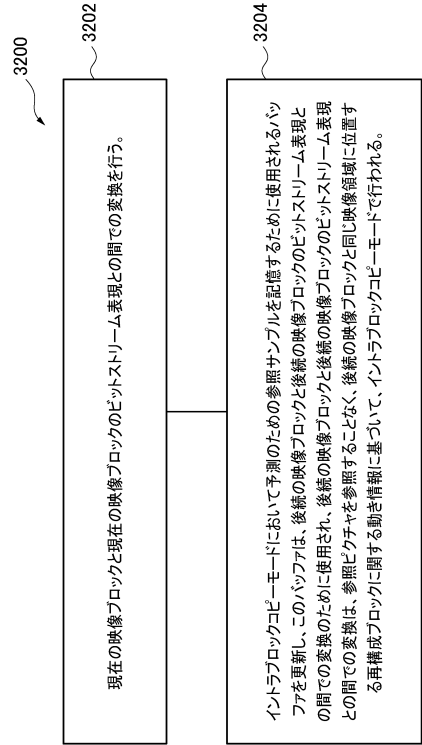
【図 30】



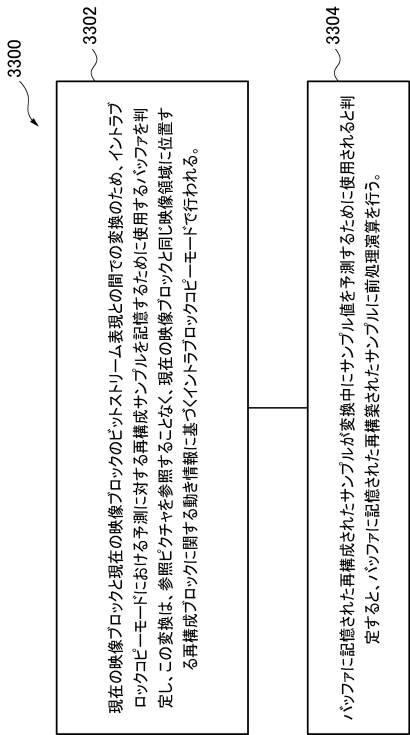
【図 3 1】



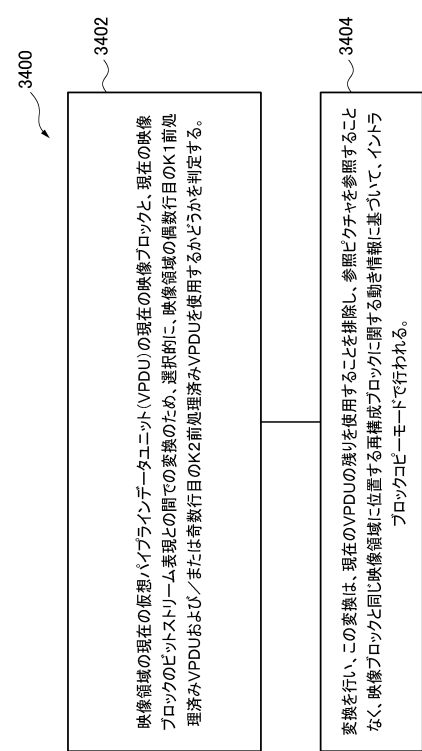
【図 3 2】



【図 3 3】



【図 3 4】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

中国(CN)

BYTEDANCE INC.

アメリカ合衆国 カリフォルニア州 90066, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137 Los Angeles, California 90066 United States of America

(74)代理人 110002000

弁理士法人栄光事務所

(72)発明者 シュー ジジョン

アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー137

(72)発明者 ジャン リー

アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー137

(72)発明者 ジャン カイ

アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー137

(72)発明者 リウ ホンビン

中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, チーチュン ロード, チャイナ サテライト コミュニケーションズ タワー, ナンバー63, ジンリトウティアオ ポストオフィス

(72)発明者 ワン ユエ

中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, チーチュン ロード, チャイナ サテライト コミュニケーションズ タワー, ナンバー63, ジンリトウティアオ ポストオフィス

審査官 鉢呂 健

(56)参考文献 特表2018-521539 (JP, A)

国際公開第2018/190207 (WO, A1)

特表2017-535148 (JP, A)

PHAM VAN, Luong et al., CE8.1.3: Extended CPR reference with 1 buffer line, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9-18 Jan. 2018, [JVET-M0474-v3], JVET-M0474 (version 3), ITU-T, 2019年01月11日, <URL:https://jvet-experts.org/doc_end_user/documents/13_Marrakech/wg11/JVET-M0474-v3.zip>: JVET-M0474_v3.docx: pp.1-9

XU, Xiaozhong et al., CE8: Summary Report on Screen Content Coding, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9-18 Jan. 2019, [JVET-M0028-v3], JVET-M0028 (version 3), ITU-T, 2019年01月10日, <URL:https://jvet-experts.org/doc_end_user/documents/13_Marrakech/wg11/JVET-M0028-v3.zip>: JVET-M0028-v3.docx: pp.4-5

HENG, Brian et al., Non-CE8: Comments on Current Picture Referencing, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9-18 Jan. 2019, [JVET-M0402-v2], JVET-M0402 (version 2), ITU-T, 2019年01月04日, <URL:https://jvet-experts.org/doc_end_user/documents/13_Marrakech/wg11/JVET-M0402-v2.zip>: JVET-M0402-v2.docx: pp.1-6

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00-19/98