

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7197720号
(P7197720)

(45)発行日 令和4年12月27日(2022.12.27)

(24)登録日 令和4年12月19日(2022.12.19)

(51)国際特許分類

H 0 4 N	19/94 (2014.01)	F I	H 0 4 N	19/94
H 0 4 N	19/70 (2014.01)		H 0 4 N	19/70

請求項の数 13 (全99頁)

(21)出願番号	特願2021-548182(P2021-548182)
(86)(22)出願日	令和2年2月24日(2020.2.24)
(65)公表番号	特表2022-520127(P2022-520127)
	A)
(43)公表日	令和4年3月28日(2022.3.28)
(86)国際出願番号	PCT/CN2020/076367
(87)国際公開番号	WO2020/169103
(87)国際公開日	令和2年8月27日(2020.8.27)
審査請求日	令和3年8月17日(2021.8.17)

(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-035 房間
	Room B-0035, 2/F, 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】 パレットモードの使用的指示の独立した符号化

(57)【特許請求の範囲】**【請求項1】**

映像データの処理方法であって、

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間での変換のために、第1の予測モードの使用の第1の指示が規則に従って前記ビットストリームに含まれているかどうかを判定することと、

前記現在の映像ブロックと前記ビットストリームとの間での前記変換を行うことと、を含み、

前記第1の予測モードにおいて、再構成されたサンプルは、1つの代表的な色値のセットによって表現され、代表的な色値の前記セットは、1)パレット予測子、2)エスクープされたサンプル、又は3)前記ビットストリームに含まれたパレット情報のうちの少なくとも1つを含み、

前記規則は、前記第1の指示が前記ビットストリーム内に含まれていないことを規定し、非イントラモードに呼応して偽であると推測される前記第1の指示が前記現在の映像ブロックに対して適用され、

前記規則は、前記現在の映像ブロックのコーディングツリー構造に基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定し、前記コーディングツリー構造は、単一コーディングツリー構造と二重コーディングツリー構造とを含む、

映像データの処理方法。

【請求項2】

前記規則は、現在の映像ブロックのディメンションに基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定する、

請求項1に記載の方法。

【請求項3】

前記規則は、現在の映像ブロックのカラーフォーマットに基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定する、

請求項1～2のいずれか1項に記載の方法。

【請求項4】

前記非イントラモードは第2の予測モードであり、前記第2の予測モードにおいて、前記現在の映像ブロックを含む同じピクチャの再構成サンプル値から予測サンプルを導出する、

請求項1～3のいずれか1項に記載の方法。

【請求項5】

前記非イントラモードは、インター予測モードである、

請求項1～4のいずれか1項に記載の方法。

【請求項6】

前記非イントラモードはスキップモードである、

請求項1～4のいずれか1項に記載の方法。

【請求項7】

前記現在の映像ブロックを含む映像スライスがIスライスであることに呼応して、前記第1の予測モード、第2の予測モード、及びイントラコーディングモードのうちの1つのみを使用して、前記現在の映像ブロックの予測サンプルを生成する、

請求項1～6のいずれか1項に記載の方法。

【請求項8】

第1の予測モードは、イントラコーディングモードから独立している、

請求項1～7のいずれか1項に記載の方法。

【請求項9】

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1～8のいずれか1項に記載の方法。

【請求項10】

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1～9のいずれか1項に記載の方法。

【請求項11】

処理装置と、命令を含む非一時的メモリと、を備える映像データを処理するための装置であって、前記処理装置によって実行されると、前記命令は、前記処理装置に、

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間での変換のために、第1の予測モードの使用の第1の指示が規則に従って前記ビットストリームに含まれているかどうかを判定することと、

前記現在の映像ブロックと前記ビットストリームとの間での前記変換を行うことを行わせ、

前記第1の予測モードにおいて、再構成されたサンプルは、1つの代表的な色値のセットによって表現され、代表的な色値の前記セットは、1)パレット予測子、2)エスケープされたサンプル、又は3)前記ビットストリームに含まれたパレット情報のうちの少なくとも1つを含み、

前記規則は、第1の指示が前記ビットストリーム内に含まれておらず、非イントラモードに呼応して偽であると推測される前記第1の指示が前記現在の映像ブロックに対して適用され、

前記規則は、前記現在の映像ブロックのコーディングツリー構造に基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定し、前記コーディングツリー

10

20

30

40

50

構造は、單一コーディングツリー構造と二重コーディングツリー構造とを含む、
映像データを処理するための装置。

【請求項 1 2】

命令を記憶する非一時的なコンピュータ可読記憶媒体であって、この命令は、処理装置に、

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間での変換のために、第1の予測モードの使用の第1の指示が規則に従ってビットストリームに含まれているかどうかを判定すること、

前記現在の映像ブロックと前記ビットストリームとの間での前記変換を行うことを行わせ、

前記第1の予測モードにおいて、再構成されたサンプルは、1つの代表的な色値のセットによって表現され、代表的な色値の前記セットは、1)パレット予測子、2)エスケープされたサンプル、又は3)前記ビットストリームに含まれたパレット情報のうちの少なくとも1つを含み、

前記規則は、第1の指示が前記ビットストリーム内に含まれていないことを規定し、非インタラモードに呼応して偽であると推測される前記第1の指示が前記現在の映像ブロックに対して適用され、

前記規則は、前記現在の映像ブロックのコーディングツリー構造に基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定し、前記コーディングツリー構造は、単一コーディングツリー構造と二重コーディングツリー構造とを含む、

非一時的なコンピュータ可読記憶媒体。

【請求項 1 3】

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、

映像の現在の映像ブロックに対して、第1の予測モードの使用の第1の指示が、規則に従って前記ビットストリームに含まれるかどうかを判定すること、

前記判定に基づいて前記ビットストリームを生成すること、

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

前記第1の予測モードにおいて、再構成されたサンプルは、1つの代表的な色値のセットによって表現され、代表的な色値の前記セットは、1)パレット予測子、2)エスケープされたサンプル、又は3)前記ビットストリームに含まれたパレット情報のうちの少なくとも1つを含み、

前記規則は、前記第1の指示が前記ビットストリーム内に含まれていないことを規定し、非インタラモードに呼応して偽であると推測される前記第1の指示が前記現在の映像ブロックに対して適用され、

前記規則は、前記現在の映像ブロックのコーディングツリー構造に基づいて、前記第1の指示を前記ビットストリームに含めるかどうかをさらに規定し、前記コーディングツリー構造は、単一コーディングツリー構造と二重コーディングツリー構造とを含む、

方法。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

(関連出願の相互参照)

本願は、2019年2月24日出願の国際特許出願第PCT/CN2019/07599
4号の優先権と利益を主張する、2020年2月24日出願の国際特許出願第PCT/C
N2020/076367号の国内段階である。前述の特許出願はすべて、参照によりそ
の全体が本明細書に組み込まれる。

【0 0 0 2】

本明細書は、映像および画像符号化技術に関する。

【背景技術】

10

20

30

40

50

【 0 0 0 3 】

デジタル映像は、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】**【 0 0 0 4 】**

開示された技術は、パレットモード符号化が使用される映像または画像デコーダまたはエンコーダの実施形態によって使用してもよい。

【 0 0 0 5 】

1つの例示的な態様において、映像処理の方法が開示される。この方法は、映像の映像領域のブロックと映像のビットストリーム表現との間で変換を行うことを含む。このビットストリーム表現は、このブロックのためにパレットモードの使用の第1の指示が信号通知されるかどうかを規定する第1のフォーマット規則と、このブロックのために予測モードの使用の第2の指示に関連するこの第1の指示の位置を規定する第2のフォーマット規則とに従って処理される。

10

【 0 0 0 6 】

1つの例示的な態様において、映像処理の方法が開示される。この方法は、映像における映像領域のブロックと映像のビットストリーム表現との間での変換のために、このブロックの少なくとも1つのパレットモードを含む1つ以上の許容予測モードに基づいて、予測モードを判定することを含む。予測モードに従って、パレットモードの使用の指示を判定する。この方法は、また、1つ以上の許容予測モードに基づいてこの変換を行うことを含む。

20

【 0 0 0 7 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、映像のブロックと映像のビットストリーム表現との間で変換を行うことを含む。ビットストリーム表現は、パレットモードの使用の第1の指示とイントラブロックコピー（IBC）モードの使用の第2の指示とを互いに依存して信号通知することを規定するフォーマット規則に従って処理される。

【 0 0 0 8 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロック寸法に基づいて、このビットストリーム表現におけるパレットモードの使用の指示の存在を判定することと、この判定に基づいてこの変換を行うことと、を含む。

30

【 0 0 0 9 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このビットストリーム表現におけるイントラブロックコピー（IBC）モードの使用の指示の存在を、このブロックの寸法に基づいて判定することと、この判定に基づいてこの変換を行うこととを含む。

【 0 0 1 0 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロックを含む映像領域の第2の指示に基づいて、このブロックに対してパレットモードが許可されているかどうかを判定することと、この判定に基づいてこの変換を行うこととを含む。

40

【 0 0 1 1 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロックを含む映像領域の第2の指示に基づいて、このブロックに対してイントラブロックコピー（IBC）モードが許可されているかどうかを判定することと、この判定に基づいてこの変換を行うこととを含む。

【 0 0 1 2 】

50

別の例示的な態様において、映像処理の方法が開示される。この方法は、パレットモードを使用して、予測モードとは別個に符号化されたパレットモードの変換ユニット、符号化ブロック、または領域を処理することを判定することと、このパレットモードを使用して、この変換ユニット、符号化ブロック、または領域に対してさらに処理を行うことを含む。

【 0 0 1 3 】

別の例示的な態様において、映像処理の方法が開示される。この方法は、現在の映像ブロックに対して、パレットモードの 1 つのパレットエントリに関連付けられたサンプルが、現在の映像ブロックに関連付けられた第 2 のビット深度とは異なる第 1 のビット深度を有することを判定することと、この少なくとも 1 つのパレットエントリに基づいて、現在の映像ブロックのさらなる処理を行うことを含む。10

【 0 0 1 4 】

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、映像のピクチャの現在の映像ブロックと、この映像のビットストリーム表現との間での変換を行うことであって、ビットストリーム表現において、イントラブロックコピーモードがこの変換において使用されるか否かに関する情報が信号通知される、またはこの現在の映像ブロックの符号化条件に基づいて導出される、変換を行うことを含み、このイントラブロックコピーモードは、このピクチャにおける別の映像ブロックからこの現在の映像ブロックを符号化することを含む。

【 0 0 1 5 】

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、映像のピクチャの現在の映像ブロックを変換する間に非ブロック化フィルタを適用するか否かを判定することを含み、ここで、現在の映像ブロックは、現在の映像ブロックの全画素より小さい代表的なサンプル値を使用して現在の映像ブロックを表すパレットモード符号化を使用して符号化され、非ブロック化フィルタを適用すると判定した場合に非ブロック化フィルタが適用されるように変換を行うことを含む。20

【 0 0 1 6 】

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、ある映像のピクチャの現在の映像ブロックとこの映像のビットストリーム表現との間での変換中に使用するために、量子化または逆量子化処理を判定することを含み、ここで、現在の映像ブロックは、現在の映像ブロックの合計画素よりも少ない代表的なサンプル値を使用して現在の映像ブロックを表すパレットモード符号化を使用して符号化され、この量子化または逆量子化処理の判定に基づいてこの変換を行うことを含む。30

【 0 0 1 7 】

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、複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを判定することと、この判定に基づいて、現在の映像ブロックをイントラ符号化されたブロックであると考えることで、最大確率モードのリスト構築処理を行うことと、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パレットまたは表現サンプル値を使用して符号化または復号化される。40

【 0 0 1 8 】

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。

【 0 0 1 9 】

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、複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを判定することと、この判定に基づいて、現在の映像ブロックを非イントラ符号化されたブロックであると考えることで、最大確率モードのリスト構築処理を実行し、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パ50

レットまたは表現サンプル値を使用して符号化または復号化される。

【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、複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを判定することと、この判定に基づいて、現在の映像ブロックを利用不可能なブロックであると考えることで、リスト構築処理を行うことと、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パレットまたは表現サンプル値を使用して符号化または復号化される。

【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、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中に、この現在の映像ブロックがパレット符号化ブロックであることを判定することと、現在の映像ブロックがパレット符号化ブロックであることに基づいて、この変換に使用されるコンテキスト符号化ビンの範囲を判定することと、このコンテキスト符号化ビンの範囲に基づいてこの変換を行うこととを含む。

【0022】

さらに別の例示的な態様において、上述された方法は、処理装置を含む映像エンコーダによって実装されてもよい。

【0023】

さらに別の例示的な態様において、これらの方法は、処理装置実行可能命令の形式で実施されてもよく、コンピュータ可読プログラム媒体に記憶されてもよい。

【0024】

これらの、および他の態様は、本明細書でさらに説明される。

【図面の簡単な説明】

【0025】

【図1】 イントラブロックコピーの例を示す。

【図2】 パレットモードで符号化されたブロックの例を示す。

【図3】 パレットエントリを信号通知するためにパレット予測子を使用する例を示す。

【図4】 水平方向および垂直方向の横断走査の例を示す。

【図5】 パレットインデックスの符号化の例を示す。

【図6】 映像処理装置の例を示すブロック図である。

【図7】 映像エンコーダの実装形態の例を示すブロック図である。

【図8】 映像処理方法の例を示すフローチャートである。

【図9】 フィルタのオン／オフ決定および強／弱フィルタの選択に関する画素の例を示す。

【図10】 4つのモードの2値化の例を示す。

【図11】 4つのモードの2値化の例を示す。

【図12】 67個のイントラモード予測方向の例を示す。

【図13】 近傍の映像ブロックの例を示す。

【図14】 A L F フィルタの形状の例を示す（クロマ：5×5菱形、輝度：7×7菱形）。

【図15】 (a) は垂直方向勾配のサブサンプリングされたラプラシアン計算の例を示す。 (b) は水平勾配のためのサブサンプリングされたラプラシアン計算の例を示す。 (c) は対角勾配のためのサブサンプリングされたラプラシアン計算の例を示す。 (d) は対角勾配のためのサブサンプリングされたラプラシアン計算の例を示す。

【図16】 仮想境界における修正された区分の例を示す。

【図17】 仮想境界における輝度要素のための修正されたA L F フィルタリングの例を示す。

【図18】 E O における画素分類の4つの1-D 3 画素パターンの例を示す。

【図19】 4つのバンドをグループ化してその開始バンド位置で表現することの例を示す。

10

20

30

40

50

【図20】C I I P重み導出に用いられる左上近傍のブロックの例を示す。

【図21】クロマスケーリングアーキテクチャを用いた輝度マッピングの例を示す。

【図22】 4×4 ブロックの走査順序の例を示す。

【図23】 4×4 ブロックのための走査順序の別の例を示す。

【図24】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25】本技術にしたがった映像処理方法を示すフローチャートである。

【図26】本技術にしたがった別の映像処理方法を示すフローチャートである。

【図27】本技術にしたがった別の映像処理方法を示す別のフローチャートである。

【図28】本技術にしたがった別の映像処理方法を示す別のフローチャートである。

10

【図29】本技術にしたがった別の映像処理方法を示す別のフローチャートである。

【図30】本技術にしたがった別の映像処理方法を示す別のフローチャートである。

【図31】本技術にしたがったさらに別の映像処理方法を示す別のフローチャートである。

【発明を実施するための形態】

【0026】

本明細書は、伸張または復号化されたデジタル映像または画像の品質を向上させるために、画像または映像ビットストリームのデコーダによって使用できる様々な技術を提供する。簡潔にするために、本明細書では、用語「映像」は、一連のピクチャ（従来から映像と呼ばれる）および個々の画像の両方を含むように使用される。さらに、映像エンコーダは、さらなる符号化に使用される復号化されたフレームを再構成するために、符号化の処理中にこれらの技術を実装してもよい。

20

【0027】

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つの章に開示された実施形態をその章にのみ限定するものではない。このように、ある章の実施形態は、他の章の実施形態と組み合わせることができる。

【0028】

1. 発明の概要

【0029】

本明細書は、映像符号化技術に関する。具体的には、本発明は、映像符号化において基本色に基づく表現を用いるパレット符号化に関する。HEVCのような既存の映像符号化規格に適用してもよいし、規格（Versatile Video Coding）を確定させるために適用してもよい。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。

30

【0030】

2. 初期の協議

【0031】

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 Model）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）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VVVC規格の策定に取り組んでいる。

40

50

【 0 0 3 2 】

図 7 は映像エンコーダの実装形態の例のブロック図である。図 7 は、エンコーダの実装が、映像エンコーダが映像復号化機能も実行する（次の映像データの符号化に使用するために映像データの圧縮表現を再構成する）フィードバック経路を組み込んでいることを示す。

【 0 0 3 3 】**2 . 1 イントラブロックコピー****【 0 0 3 4 】**

イントラブロックコピー（IBC）、別名、現在のピクチャの参照は、HEVCスクリーンコンテンツ符号化拡張機能（HEVC - SCCC）と現在のVVCテストモデル（VTM - 4 . 0）に採用されている。IBCは、動き補償の概念をインターフレーム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、IBC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。IBC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。IBCは、これらの繰り返しパターン間の冗長性を有效地に除去することができる。HEVC - SCCC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、IBC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、複数のビュー / 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。

10

20

30

【 0 0 3 5 】

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。

【 0 0 3 6 】

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。

40

【 0 0 3 7 】**2 . 2 HEVCスクリーンコンテンツ符号化拡張機能におけるIBC****【 0 0 3 8 】**

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。

【 0 0 3 9 】

変数 offsetX および offsetY は、以下のように導出される。

【 0 0 4 0 】

```
offsetX = ( ChromaArrayType == 0 ) ? 0 : ( mvCLX[0]
] & 0x7 ? 2 : 0 )           ( 8 - 106 )
```

【 0 0 4 1 】

```
offsetY = ( ChromaArrayType == 0 ) ? 0 : ( mvCLX[1]
] & 0x7 ? 2 : 0 )           ( 8 - 107 )
```

50

【0042】

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 $m v L X$ 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。

【0043】

- 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、 $(x C b, y C b)$ と等しく設定された $(x C u r r, y C u r r)$ 、 $(x P b + (m v L X [0] >> 2) - o f f s e t X, y P b + (m v L X [1] >> 2) - o f f s e t Y)$ に等しく設定された近傍の輝度位置 $(x N b Y, y N b Y)$ を入力として呼び出されると、出力はTRUEとなる。

【0044】

- 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、 $(x C b, y C b)$ と等しく設定された $(x C u r r, y C u r r)$ 、 $(x P b + (m v L X [0] >> 2) + n P b W - 1 + o f f s e t X, y P b + (m v L X [1] >> 2) + n P b H - 1 + o f f s e t Y)$ に等しく設定された近傍の輝度位置 $(x N b Y, y N b Y)$ を入力として呼び出されると、出力はTRUEとなる。

10

【0045】

- 次の条件の一方または両方が真であること。

【0046】

- $(m v L X [0] >> 2) + n P b W + x B 1 + o f f s e t X$ の値が 0 以下である。

【0047】

- $(m v L X [1] >> 2) + n P b H + y B 1 + o f f s e t Y$ の値が 0 以下である。

20

【0048】

- 次の条件が真となること。

【0049】

$(x P b + (m v L X [0] >> 2) + n P b S w - 1 + o f f s e t X) / C t b S i z e Y - x C u r r / C t b S i z e Y \leq y C u r r / C t b S i z e Y - (y P b + (m v L X [1] >> 2) + n P b S h - 1 + o f f s e t Y) / C t b S i z e Y$
 $(8 - 108)$

【0050】

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。

30

【0051】

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。

【0052】

2.3 VVC 試験モデルにおけるIBC

【0053】

現在のVVC試験モデル、例えば、VTM-4.0 設計において、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。IBC フラグは、現在のCUの予測モードとして符号化される。このように、各CUに対して、MODE_INTRA、MODE_INTER、およびMODE_IBCという全部で3つの予測モードが存在する。

40

【0054】

2.3.1 IBC マージモード

【0055】

IBC マージモードにおいて、IBC マージ候補リストにおけるエントリを指すインデックスをビットストリームから構文解析する。このIBC マージリストの構築は、以下の

50

ステップのシーケンスに従ってまとめることができる。

【0056】

ステップ1：空間的候補の導出

【0057】

ステップ2：H M V P 候補の挿入

【0058】

ステップ3：対の平均候補の挿入

【0059】

空間的マージ候補の導出において、図面に示す位置にある候補の中から、最大4つのマージ候補が選択されている。導出の順序はA1、B1、B0、A0、B2である。位置A1、B1、B0、A0のいずれかのPUが利用可能でない場合（例えば、別のスライスまたはタイルに属しているため）、またはIBCモードで符号化されていない場合にのみ、位置B2が考慮される。位置A1の候補を加えた後、残りの候補を挿入すると、冗長性チェックを受け、それにより、同じ動き情報を有する候補を確実にリストから排除でき、符号化効率を向上させることができる。計算の複雑性を低減するために、前述の冗長性チェックにおいて、考えられる候補対のすべてを考慮することはしない。代わりに、図面に示された矢印でリンクされた対のみを考慮し、冗長性チェックに使用される対応する候補が同じ動き情報を有していない場合にのみ、その候補をリストに加える。

10

【0060】

空間的候補を挿入した後、IBCマージリストサイズが依然として最大IBCマージリストサイズより小さい場合、H M V P テーブルからのIBC候補を挿入することができる。H M V P 候補の挿入にあたり、冗長性チェックを行う。

20

【0061】

最後に、対の平均候補をIBCマージリストに挿入する。

【0062】

マージ候補によって特定される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、マージ候補は無効なマージ候補と呼ばれる。

【0063】

なお、IBCマージリストに無効なマージ候補を挿入してもよい。

30

【0064】

2.3.2 IBC AMVP モード

【0065】

IBC AMVP モードでは、IBC AMVP リストにおけるエントリを指すAMVPインデックスが、ビットストリームから構文解析される。このIBC AMVP リストの構築は、以下のステップのシーケンスに従ってまとめることができる。

【0066】

ステップ1：空間的候補の導出

【0067】

利用可能な候補が見つかるまで、A0、A1をチェックする。

40

【0068】

利用可能な候補が見つかるまで、B0、B1、B2をチェックする。

【0069】

ステップ2：H M V P 候補の挿入

【0070】

ステップ3：ゼロ候補の挿入

【0071】

空間的候補を挿入した後、IBC AMVP リストサイズが依然として最大IBC AMVP リストサイズより小さい場合、H M V P テーブルからのIBC候補を挿入することができる。

50

【0072】

最後に、IBC AMVPリストにゼロ候補を挿入する。

【0073】

2.4 パレットモード

【0074】

パレットモードの背景にある基本的な考えは、CTUにおけるサンプルを代表的な色値の小さな集合で表現することである。この集合をパレットと呼ぶ。また、エスケープシンボルの後に（場合によっては量子化された）要素値を信号通知することによって、パレットの外側にあるサンプルを示すことができる。これを図2に示す。

【0075】

2.5 HEVCスクリーンコンテンツ符号化拡張機能におけるパレットモード(HEVC-SCC)

【0076】

HEVC-SCCにおけるパレットモードでは、パレットおよびインデックスマップを符号化するために予測方式が用いられる。

【0077】

2.5.1 パレットエントリの符号化

【0078】

パレットエントリを符号化するために、パレット予測子が維持される。SPSにおいて、パレットの最大サイズおよびパレット予測子が信号通知される。HEVC-SCCにおいて、`palette_predictor_initializer_present_flag`がPPSに導入される。このフラグが1である場合、ビットストリームにおいて、パレット予測子を初期化するためのエントリが信号通知される。パレット予測子は、各CTU行、各スライス、および各タイルの始めに初期化される。`palette_predictor_initializer_present_flag`の値によって、`palette_predictor`を0にリセットするか、またはPPSに信号通知されたパレット予測子の初期化エントリを使用してパレット予測子を初期化する。HEVC-SCCでは、PPSレベルでパレット予測子の初期化を明確に無効にするために、サイズ0のパレット予測子初期化モジュールを有効化した。

【0079】

パレット予測子におけるエントリごとに、それが現在のパレットの一部であるかどうかを示すように、再利用フラグが信号通知される。これを図3に示す。再利用フラグは、ゼロのランレンジス符号化を使用して送信される。この後、新しいパレットエントリの数は、次数0の指数ゴロムコードを使用して信号通知される。最後に、新しいパレットエントリのための要素値が信号通知される。

【0080】

2.5.2 パレットインデックスの符号化

【0081】

パレットインデックスは、図4に示すように、水平方向および垂直方向の横断走査を使用して符号化される。`palette_transpose_flag`を使用して、ビットストリームにおける走査順序を明確に信号通知する。以下のサブセクションでは、走査が水平であると仮定する。

【0082】

パレットインデックスは、「INDEX」および「COPY ABOVE」の2つのメインパレットサンプルモードを使用して符号化される。前述のように、エスケープシンボルも「INDEX」モードとして信号送信され、最大パレットサイズに等しいインデックスが割り当てられる。このモードは、最上行を除くフラグを使用して、または前回のモードが「COPY ABOVE」であった場合に信号通知される。「COPY ABOVE」モードでは、上の行のサンプルのパレットインデックスをコピーする。「INDEX」モードにおいて、パレットインデックスは明確に信号通知される。「INDEX」モード

10

20

30

40

50

と「C O P Y _ A B O V E」モードの両方の場合、同じモードを使用して符号化される後続のサンプルの数を規定する実行値を信号通知する。エスケープシンボルが「I N D E X」または「C O P Y _ A B O V E」モードにおける実行の一部である場合、エスケープシンボルごとにエスケープ要素値が信号通知される。パレットインデックスの符号化を図5に示す。

【0083】

この構文順序は、以下のようにして実行される。まず、C Uのためのインデックス値の数が信号通知される。これに続いて、トランケーテッドバイナリ符号化(`truncated binary coding`)を使用して、C U全体の実際のインデックス値を信号通知する。バイパスモードでは、インデックスの数およびインデックス値の両方が符号化される。これにより、インデックス関連バイパスビンがグループ化される。次に、パレットサンプルモード(必要な場合)および実行は、インターリーブ方式で信号通知される。最後に、C U全体のためのエスケープサンプルに対応する要素エスケープ値をグループ化し、バイパスモードで符号化する。

10

【0084】

インデックス値を信号通知した後、追加の構文要素`last_run_type_flag`を信号通知する。この構文要素は、インデックスの数と連動して、ブロックにおける最後の実行に対応する実行値を信号通知する必要をなくす。

【0085】

H E V C - S C Cでは、パレットモードは、4 : 2 : 2、4 : 2 : 0、およびモノクロのクロマフォーマットに対しても有効になる。パレットエントリおよびパレットインデックスの信号通知は、すべてのクロマフォーマットに対してほぼ同じである。非モノクロフォーマットの場合、各パレットエントリは、3つの要素からなる。モノクロフォーマットの場合、各パレットエントリは単一の要素からなる。サブサンプリングされたクロマ方向の場合、クロマサンプルは、2で割り切れる輝度サンプル指数に関連付けられる。C Uのパレットインデックスを再構成した後、1つのサンプルに単一の要素しか関連付けられていない場合、パレットエントリの第1の要素のみが使用される。信号通知における唯一の違いは、エスケープ要素値である。エスケープサンプルごとに、信号通知されるエスケープ要素値の数は、そのサンプルに関連付けられた要素の数によって異なってもよい。

20

【0086】

V V Cにおいて、イントラスライスの符号化にデュアルツリー符号化構造が用いられるので、輝度要素および2つのクロマ要素は、異なるパレットおよびパレットインデックスを有してもよい。また、2つのクロマ要素は、同じパレットおよびパレットインデックスを共有する。

30

【0087】

2.6 V V Cにおける非ブロック化スキーム

【0088】

なお、以下の説明において、 p_{N_M} は、垂直エッジに対してM行目の左側のN番目のサンプル、または水平エッジに対してM列目の上側のN番目のサンプルを表し、 q_{N_M} は、垂直エッジに対してM行目の右側のN番目のサンプル、または水平エッジに対して、M列目の下側のN番目のサンプルを表す。 p_{N_M} および q_{N_M} の例を図9に示す。

40

【0089】

なお、以下の説明において、 p_N は、垂直エッジに対して行の左側のN番目のサンプル、または、水平エッジに対して列の上側のN番目のサンプルを表し、 q_N は、垂直エッジに対して行の右側のN番目のサンプル、又は水平エッジに対して、列の下側のN番目のサンプルを表す。

【0090】

1つの単位として4行に対してフィルタのオン/オフの決定を行う。図9は、フィルタのオン/オフ決定に関する画素を示す。最初の4行のための2つの赤いボックスにおける6つの画素は、4行のためのフィルタのオン/オフを判定するために用いられる。2番

50

目の4行のための2つの赤いボックス内の6つの画素は、第2の4行のためのフィルタのオン／オフを判定するために用いられる。

【0091】

いくつかの実施形態において、まずピクチャの垂直エッジを選別する。そして、垂直エッジフィルタリング処理で修正されたサンプルを入力として、ピクチャの水平エッジをフィルタリングする。各CTUのCTBにおける垂直および水平エッジは、符号化ユニットごとに別個に処理される。符号化ユニットにおける符号化ブロックの垂直エッジは、符号化ブロックの左側のエッジから始まり、符号化ブロックの右側に向かってそれらの幾何学的順にエッジを通って進むようにフィルタリングされる。符号化ユニットにおける符号化ブロックの水平エッジは、符号化ブロックの上側のエッジから始まり、符号化ブロックの下側に向かってそれらの幾何学的順にエッジを通って進むようにフィルタリングされる。

10

【0092】

2.6.1 境界の決定

【0093】

8×8 のブロック境界にフィルタリングを適用する。さらに、それは、（例えば、アフィン動き予測、ATMVPを使用しているため）変換ブロックの境界または符号化サブブロックの境界でなければならない。そのような境界でない場合、フィルタは無効にされる。

【0094】

2.6.2 境界強度計算

【0095】

変換ブロックの境界 / 符号化サブブロックの境界について、それが 8×8 グリッドに位置する場合、それをフィルタリングしてもよく、このエッジのための $bS[xD_i][yD_j]$ ($[xD_i][yD_j]$ は座標を表す) の設定は、以下のように定義される。

20

- サンプル p_0 または q_0 がイントラ予測モードで符号化された符号化ユニットの符号化ブロックにある場合、 $bS[xD_i][yD_j]$ は、2 に等しく設定される。
- あるいは、ブロックエッジが変換ブロックエッジでもあり、サンプル p_0 または q_0 が、1つ以上の非ゼロ変換係数レベルを含む変換イントラブロックにある場合、 $bS[xD_i][yD_j]$ は、1 に等しく設定される。
- あるいは、サンプル p_0 を含む符号化サブブロックの予測モードが、サンプル q_0 を含む符号化サブブロックの予測モードと異なる場合、 $bS[xD_i][yD_j]$ は、1 に等しく設定される。

- あるいは、以下の条件の1つ以上が真である場合、 $bS[xD_i][yD_j]$ を1に等しく設定する。
 - サンプル p_0 を含む符号化サブブロックおよびサンプル q_0 を含む符号化サブブロックは、いずれもIBC予測モードで符号化され、2つの符号化サブブロックの予測に用いられる動きベクトルの水平または垂直要素の絶対差は、 $1/4$ 輝度サンプル単位で4以上である。
 - サンプル p_0 を含む符号化サブブロックの予測のために、サンプル q_0 を含む符号化サブブロックの予測とは異なる参照ピクチャまたは異なる数の動きベクトルが使用される。

30

- 注1 - 2つの符号化サブブロックに使用される参照ピクチャが同じであるかまたは異なるかは、予測を形成するのに参照ピクチャリスト0へのインデックスを使用するか、または参照ピクチャリスト1へのインデックスを使用して形成するかに関わらず、且つ参照ピクチャリスト内のインデックス位置が異なるかどうかに関わらず、どのピクチャが参照されるかによってのみに基づいて判定される。
- 注2 - (xSb, ySb) を含む左上のサンプルを有する符号化サブブロックの予測に使用される動きベクトルの数は、 $PredFlag0[xSb][ySb] + PredFlag1[xSb][ySb]$ に等しい。
- 1つの動きベクトルは、サンプル p_0 を含む符号化サブブロックを予測するために使用され、1つの動きベクトルは、サンプル q_0 を含む符号化サブブロックを予測する

40

50

ために使用され、使用される動きベクトルの水平または垂直要素の絶対差は、 $1 / 4$ 輝度サンプル単位で 4 以上である。

- 2つの動きベクトルおよび 2つの異なる参照ピクチャを使用して、サンプル p_0 を含む符号化サブブロックを予測し、同じ 2つの参照ピクチャの 2つの動きベクトルを使用して、サンプル q_0 を含む符号化サブブロックを予測し、同じ参照ピクチャの 2つの符号化サブブロックの予測に使用される 2つの動きベクトルの水平または垂直要素の絶対差は、 $1 / 4$ 輝度サンプル単位で 4 以上である。

- 同じ参照ピクチャの 2つの動きベクトルを使用して、サンプル p_0 を含む符号化サブブロックを予測し、同じ参照ピクチャの 2つの動きベクトルを使用して、サンプル q_0 を含む符号化サブブロックを予測し、以下の条件の両方が成り立つ。 10

- 2つの符号化サブブロックの予測に使用されるリスト 0 の動きベクトルの水平または垂直要素の間の絶対差は、4 分の 1 輝度サンプルにおいて 4 以上である、または 2つの符号化サブブロックの予測に使用されるリスト 1 の動きベクトルの水平または垂直要素の間の絶対差は、 $1 / 4$ 輝度サンプル単位で 4 以上である。

- サンプル p_0 を含む符号化サブブロックの予測に使用されるリスト 0 動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブブロックの予測に使用されるリスト 1 動きベクトルとの間の絶対差は、 $1 / 4$ 輝度サンプル単位で 4 以上であるか、またはサンプル p_0 を含む符号化サブブロックの予測に使用されるリスト 1 動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブブロックの予測に使用されるリスト 0 動きベクトルとの間の絶対差は、 $1 / 4$ 輝度サンプル単位で 4 以上である。 20

あるいは、変数 $b_S [x D_i] [y D_j]$ を 0 に設定する。

表 2 - 1、表 2 - 2 に B_S の計算規則をまとめる。

【0096】

【表 1】

表 2 - 1 境界強度 ($S P S$ $I B C$ が無効な場合)

優先順位	条件	Y	U	V
5	隣接するブロックのうちの少なくとも 1 つは、イントラである	2	2	2
4	T U 境界および隣接するブロックのうちの少なくとも 1 つは、非ゼロ変換係数を有する	1	1	1
3	隣接するブロックの参照ピクチャまたは MV の数が異なる (单一予測の場合、1、双予測の場合、2)	1	N/A	N/A
2	隣接するブロックに属する同じ参照ピクチャの動きベクトル間の絶対差は、1 つの整数輝度サンプル以上である	1	N/A	N/A
1	上記以外	0	0	0

【0097】

10

20

30

40

50

【表 2】

表 2-2 境界強度 (S P S I B C が有効な場合)

優先順位	条件	Y	U	V
8	隣接するブロックのうちの少なくとも 1 つは、インストラである	2	2	2
7	T U 境界および隣接するブロックのうちの少なくとも 1 つは、非ゼロ変換係数を有する	1	1	1
6	隣接するブロックの予測モードが異なる（例えば、1 つは I B C であり、1 つはインターである）。	1		
5	隣接するブロックに属する動きベクトル間の I B C および絶対差は、いずれも 1 つの整数輝度サンプル以上である。	1	N/A	N/A
4	隣接するブロックの参照ピクチャまたは M V の数が異なる（单一予測の場合、1、双予測の場合、2）	1	N/A	N/A
3	隣接するブロックに属する同じ参照ピクチャの動きベクトル間の絶対差は、1 つの整数輝度サンプル以上である	1	N/A	N/A
1	上記以外	0	0	0

10

20

30

【0098】

2.6.3 輝度要素の非ブロック化決定

【0099】

非ブロック化決定処理は、このサブセクションで説明する。

【0100】

より広く、強い輝度フィルタは、条件 1、条件 2、および条件 3 の全てが T R U E である場合にのみ使用されるフィルタである。

【0101】

条件 1 は、「大ブロック条件」である。この条件は、P 側および Q 側のサンプルが、それぞれ変数 b S i d e P i s L a r g e B l k および b S i d e Q i s L a r g e B l k によって表現される大ブロックに属するかどうかを検出する。b S i d e P i s L a r g e B l k および b S i d e Q i s L a r g e B l k は、以下のように定義される。

```
b S i d e P i s L a r g e B l k = ((e d g e t y p e i s v e r t i c a l a n d p o b e l o n g s t o C U w i t h w i d t h >= 3 2 ) | | (e d g e t y p e i s h o r i z o n t a l a n d p o b e l o n g s t o C U w i t h h e i g h t >= 3 2 )) ? T R U E : F A L S E
```

```
b S i d e Q i s L a r g e B l k = ((e d g e t y p e i s v e r t i c a l a n d q o b e l o n g s t o C U w i t h w i d t h >= 3 2 ) | | (e d g e t y p e i s h o r i z o n t a l a n d q o b e l o n g s t o C U w i t h h e i g h t >= 3 2 )) ? T R U E : F A L S E
```

40

【0102】

b S i d e P i s L a r g e B l k および b S i d e Q i s L a r g e B l k に基づいて、条件 1 を以下のように定義する。

C o n d i t i o n 1 = (b S i d e P i s L a r g e B l k | | b S i d e P i s L a r g e B l k) ? T R U E : F A L S E

次に、条件 1 が真である場合、さらに条件 2 をチェックする。まず、以下の変数を導出する。

- d p 0 , d p 3 , d q 0 , d q 3 をまず H E V C として導出する。

50

```

- ( p 側が 3 2 以上である場合 )
  d p 0 = ( d p 0 + A b s ( p 5 0 - 2 * p 4 0 + p 3 0 ) + 1 ) >> 1
  d p 3 = ( d p 3 + A b s ( p 5 3 - 2 * p 4 3 + p 3 3 ) + 1 ) >> 1
- ( q 側が 3 2 以上の場合 )
  d q 0 = ( d q 0 + A b s ( q 5 0 - 2 * q 4 0 + q 3 0 ) + 1 ) >> 1
  d q 3 = ( d q 3 + A b s ( q 5 3 - 2 * q 4 3 + q 3 3 ) + 1 ) >> 1
C o n d i t i o n 2 = ( d < ) ? T R U E : F A L S E

```

章 2 . 2 . 4 に示すとおり、式中、 $d = d p 0 + d q 0 + d p 3 + d q 3$ である。

条件 1 および条件 2 が有効である場合、いずれかのブロックがサブブロックを使用するかどうかをさらにチェックする。

```

I f ( b S i d e P i s L a r g e B l k )
  I f ( m o d e b l o c k P == S U B B L O C K M O D E )
    S p = 5
  e l s e
    S p = 7
e l s e
  S p = 3
I f ( b S i d e Q i s L a r g e B l k )
  I f ( m o d e b l o c k Q == S U B B L O C K M O D E )
    S q = 5
  e l s e
    S q = 7
e l s e
  S q = 3

```

最後に、条件 1 および条件 2 の両方が有効である場合、提案された非ブロック化方法は、以下のように定義される条件 3（大ブロックの強いフィルタ条件）をチェックする。

条件 3 の S t r o n g F i l t e r C o n d i t i o n において、以下の変数を導出する。
 $d p q$ は H E V C と同様に導出される。

$s p_3 = A b s (p_3 - p_0)$ 、H E V C と同様に導出される。

i f (p 側が 3 2 以上である場合)

```

  i f ( S p == 5 )
    s p 3 = ( s p 3 + A b s ( p 5 - p 3 ) + 1 ) >> 1
  e l s e
    s p 3 = ( s p 3 + A b s ( p 7 - p 3 ) + 1 ) >> 1

```

$s q_3 = A b s (q_0 - q_3)$ は H E V C と同様に導出される

i f (q 側が 3 2 以上である場合)

```

  I f ( S q == 5 )
    s q 3 = ( s q 3 + A b s ( q 5 - q 3 ) + 1 ) >> 1
  e l s e
    s q 3 = ( s q 3 + A b s ( q 7 - q 3 ) + 1 ) >> 1

```

【 0 1 0 3 】

H E V C と同様に、S t r o n g F i l t e r C o n d i t i o n = (d p q i s l e s s t h a n (>> 2), $s p_3 + s q_3$ i s less than ($3 * >> 5$) , and A b s ($p_0 - q_0$) i s less than ($5 * t_c + 1$) >> 1) ? T R U E : F A L S E .

【 0 1 0 4 】

2 . 6 . 4 輝度のためのより強い非ブロック化フィルタ（より大きいブロックのために設計される）

【 0 1 0 5 】

バイリニアフィルタは、境界の両側のサンプルが 1 つの大ブロックに属する場合に用い

10

20

30

40

50

られる。1つの大ブロックに属する1つのサンプルは、垂直エッジの場合、幅32であり、水平エッジの場合、高さ32であるときとして定義される。

【0106】

バイリニアフィルタを以下に示す。

【0107】

次に、上述のHEVC非ブロック化において、ブロックの境界サンプルである、 $i = 0 \sim S_p - 1$ における p_i 、及び、 $j = 0 \sim S_q - 1$ における q_i (p_i 及び q_i は、垂直エッジをフィルタリングする行内の*i*番目のサンプル、または水平エッジをフィルタリングする列内の*i*番目のサンプル) を、以下のように線形補間によって置き換える。

【0108】

【数1】

- $p_i' = (f_i * Middle_{s, t} + (64 - f_i) * P_s + 32) \gg 6$, clipped to $p_i \pm tcPD_i$
- $q_j' = (g_j * Middle_{s, t} + (64 - g_j) * Q_s + 32) \gg 6$, clipped to $q_j \pm tcPD_j$

【0109】

ここで、 $tcPD_i$ および $tcPD_j$ 項は、章2.3.6に記載の位置依存クリッピングであり、 g_j , f_i , $Middle_{s, t}$, P_s および Q_s は、表2-3に示される。

【0110】

10

20

30

40

50

【表 3】

表 2-3 ロングタップ非ブロック化フィルタ

Sp, Sq 7, 7 (p側:7, q側:7)	$f_i = 59 - i * 9, canalsobedescribedasf = \{59, 50, 41, 32, 23, 14, 5\}$ $g_j = 59 - j * 9, canalsobedescribedasg = \{59, 50, 41, 32, 23, 14, 5\}$ $Middle_{7,7} = (2 * (p_o + q_o) + p_1 + q_1 + p_2 + q_2 + p_3 + q_3 + p_4 + q_4 + p_5 + q_5 + p_6 + q_6 + 8) \gg 4$ $P_7 = (p_6 + p_7 + 1) \gg 1, Q_7 = (q_6 + q_7 + 1) \gg 1$
7, 3 (p側:7 q側:3)	$f_i = 59 - i * 9, canalsobedescribedasf = \{59, 50, 41, 32, 23, 14, 5\}$ $g_j = 53 - j * 21, canalsobedescribedasg = \{53, 32, 11\}$ $Middle_{7,3} = (2 * (p_o + q_o) + q_0 + 2 * (q_1 + q_2) + p_1 + q_1 + p_2 + p_3 + p_4 + p_5 + p_6 + 8) \gg 4$ $P_7 = (p_6 + p_7 + 1) \gg 1, Q_3 = (q_2 + q_3 + 1) \gg 1$
3, 7 (p側:3 q側:7)	$g_j = 59 - j * 9, canalsobedescribedasg = \{59, 50, 41, 32, 23, 14, 5\}$ $f_i = 53 - i * 21, canalsobedescribedasf = \{53, 32, 11\}$ $Middle_{3,7} = (2 * (q_o + p_o) + p_0 + 2 * (p_1 + p_2) + q_1 + p_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4$ $Q_7 = (q_6 + q_7 + 1) \gg 1, P_3 = (p_2 + p_3 + 1) \gg 1$
7, 5 (p側:7 q側:5)	$g_j = 58 - j * 13, canalsobedescribedasg = \{58, 45, 32, 19, 6\}$ $f_i = 59 - i * 9, canalsobedescribedasf = \{59, 50, 41, 32, 23, 14, 5\}$ $Middle_{7,5} = (2 * (p_o + q_o + p_1 + q_1) + q_2 + p_2 + q_3 + p_3 + q_4 + p_4 + q_5 + p_5 + 8) \gg 4$ $Q_5 = (q_4 + q_5 + 1) \gg 1, P_7 = (p_6 + p_7 + 1) \gg 1$
5, 7 (p側:5 q側:7)	$g_j = 59 - j * 9, canalsobedescribedasg = \{59, 50, 41, 32, 23, 14, 5\}$ $f_i = 58 - i * 13, canalsobedescribedasf = \{58, 45, 32, 19, 6\}$ $Middle_{5,7} = (2 * (q_o + p_o + p_1 + q_1) + q_2 + p_2 + q_3 + p_3 + q_4 + p_4 + q_5 + p_5 + 8) \gg 4$ $Q_7 = (q_6 + q_7 + 1) \gg 1, P_5 = (p_4 + p_5 + 1) \gg 1$
5, 5 (p側:5 q側:5)	$g_j = 58 - j * 13, canalsobedescribedasg = \{58, 45, 32, 19, 6\}$ $f_i = 58 - i * 13, canalsobedescribedasf = \{58, 45, 32, 19, 6\}$ $Middle_{5,5} = (2 * (q_o + p_o + p_1 + q_1 + q_2 + p_2) + q_3 + p_3 + q_4 + p_4 + 8) \gg 4$ $Q_5 = (q_4 + q_5 + 1) \gg 1, P_5 = (p_4 + p_5 + 1) \gg 1$
5, 3 (p側:5 q側:3)	$g_j = 53 - j * 21, canalsobedescribedasg = \{53, 32, 11\}$ $f_i = 58 - i * 13, canalsobedescribedasf = \{58, 45, 32, 19, 6\}$ $Middle_{5,3} = (q_o + p_o + p_1 + q_1 + q_2 + p_2 + q_3 + p_3 + 4) \gg 3$ $Q_3 = (q_2 + q_3 + 1) \gg 1, P_5 = (p_4 + p_5 + 1) \gg 1$
3, 5 (p側:3 q側:5)	$g_j = 58 - j * 13, canalsobedescribedasg = \{58, 45, 32, 19, 6\}$ $f_i = 53 - i * 21, canalsobedescribedasf = \{53, 32, 11\}$ $Middle_{3,5} = (q_o + p_o + p_1 + q_1 + q_2 + p_2 + q_3 + p_3 + 4) \gg 3$ $Q_5 = (q_4 + q_5 + 1) \gg 1, P_3 = (p_2 + p_3 + 1) \gg 1$

【0 1 1 1】

2 . 6 . 5 クロマの非ブロック化制御

【0 1 1 2】

クロマの強いフィルタは、ブロックの境界の両側に用いられる。ここで、クロマフィルタは、クロマエッジの両側が 8 (クロマ位置) 以上である場合に選択され、3 つの条件付きで、次の決定が満たされる。すなわち、1 番目のものは、大ブロックと同様に、境界強度を決定するためのものである。提案されたフィルタは、クロマサンプルドメインにおいて、ブロックのエッジに直交するブロックの幅または高さが 8 以上である場合に適用できる。第 2 および第 3 のものは、基本的には H E V C 輝度の非ブロック化の決定と同じであり、それぞれオン / オフ決定および強いフィルタの決定となっている。

【0 1 1 3】

第 1 の決定において、表 2-2 に示すように、クロマフィルタリングのために境界強度 (b S) が修正される。表 2-2 の条件を順次チェックする。条件が満たされている場合

10

20

30

40

50

、残りの優先順位の低い条件はスキップされる。

【0114】

大ブロックの境界が検出された場合、 bS が2に等しいか、または bS が1に等しいとき、クロマ非ブロック化が行われる。

【0115】

第2および第3の条件は、基本的には、以下のように、HEVC輝度の強いフィルタの決定と同様である。

【0116】

第2の条件において、その後、HEVC輝度非ブロック化と同様に d を導出する。

第2の条件は、 d がより小さい場合、TRUEとなる。

10

【0117】

第3の条件において、Strong Filter Conditionは、以下のように導出される。

d_{pq} はHEVCと同様に導出される。

$s_{p3} = \text{Abs}(p_3 - p_0)$ 、HEVCと同様に導出される。

$s_{q3} = \text{Abs}(q_0 - q_3)$ はHEVCと同様に導出される。

【0118】

HEVC設計におけるように、Strong Filter Condition = (d_{pq} is less than ($>> 2$), $s_{p3} + s_{q3}$ is less than ($>> 3$), and $\text{Abs}(p_0 - q_0)$ is less than ($5 * tc + 1$) $> > 1$)

20

【0119】

2.6.6 クロマ用の強い非ブロック化フィルタ

【0120】

以下のようなクロマ用の強い非ブロック化フィルタが定義される。

$p_2' = (3 * p_3 + 2 * p_2 + p_1 + p_0 + q_0 + 4) >> 3$

$p_1' = (2 * p_3 + p_2 + 2 * p_1 + p_0 + q_0 + q_1 + 4) >> 3$

$p_0' = (p_3 + p_2 + p_1 + 2 * p_0 + q_0 + q_1 + q_2 + 4) >> 3$

【0121】

提案されたクロマフィルタは、 4×4 クロマサンプルグリッドに対して非ブロック化を行う。

30

【0122】

2.6.7 位置依存クリッピング

【0123】

位置依存クリッピング $tcPD$ は、境界で7個、5個、および3個のサンプルを修正する強く長いフィルタを含む輝度フィルタリング処理の出力サンプルに適用される。量子化誤差分布を仮定して、より高い量子化ノイズを有すると予想されるサンプルに対して、クリッピング値を増加させることが提案され、よって、再構成されたサンプル値の真のサンプル値からのより高い偏差を有することが予想される。

【0124】

非対称フィルタでフィルタリングされた各PまたはQ境界について、章2.3.3における意思決定処理の結果に基づいて、位置依存閾値テーブルが、副情報としてデコーダに提供される2つのテーブル（即ち、 $Tc7$ および $Tc3$ を以下にまとめる）から選択される。

40

```
Tc7 = { 6, 5, 4, 3, 2, 1, 1 } ;
```

```
Tc3 = { 6, 4, 2 } ;
```

```
tcPD = (Sp == 3) ? Tc3 : Tc7 ;
```

```
tcQD = (Sq == 3) ? Tc3 : Tc7 ;
```

【0125】

短い対称フィルタでフィルタリングされるPまたはQ境界に対しては、より小さい位置

50

依存閾値が適用される。

T c 3 = { 3 , 2 , 1 } ;

【 0 1 2 6 】

閾値を定義した後、 t c P および t c Q クリッピング値に従って、 フィルタリングされた p ' i および q ' i サンプル値をクリッピングする。

p ' i = C l i p 3 (p ' i + t c P i , p ' i - t c P i , p ' i) ;

q ' j = C l i p 3 (q ' j + t c Q j , q ' j - t c Q j , q ' j) ;

【 0 1 2 7 】

ここで、 p ' i 、 q ' i はフィルタリングされたサンプル値であり、 p ' i 、 q ' j はクリッピング後の出力サンプル値であり、 t c P i 、 t c P i は V V C t c パラメータ、 t c P D 、 t c Q D から導出されるクリッピング閾値である。関数 C l i p 3 は、 V V C に規定されているような、クリッピング関数である。

【 0 1 2 8 】

2 . 6 . 8 サブブロックの非ブロック化調整

【 0 1 2 9 】

両方のロングフィルタを使用する並列フレンドリな非ブロック化およびサブブロック非ブロック化を可能にするために、ロングフィルタは、ロングフィルタのための輝度制御に示すように、サブブロック非ブロック化 (A F F I N E 、 A T M V P 、または D M V R) を使用する側でのサンプルの修正が、最大で 5 つまでに制限される。さらに、サブブロックの非ブロック化は、 C U または暗黙の T U 境界に近い 8×8 グリッド上のサブブロック境界の修正が、各側において最大 2 つまでのサンプルに制限されるよう調整される。

【 0 1 3 0 】

以下は、 C U 境界と整列されていないサブブロック境界に適用される。

```
If ( mode_b l o c k _ Q == S U B B L O C K M O D E && e d g e != 0 ) {
    i f ( ! ( i m p l i c i t T U && ( e d g e == ( 6 4 / 4 ) ) ) )
        i f ( e d g e == 2 || e d g e == ( o r t h o g o n a l L e n g t h -
2 ) || e d g e == ( 5 6 / 4 ) || e d g e == ( 7 2 / 4 ) )
            S p = S q = 2 ;
        e l s e
            S p = S q = 3 ;
    e l s e
        S p = S q = b S i d e Q i s L a r g e B l k ? 5 : 3
}
```

【 0 1 3 1 】

この場合、 0 に等しいエッジは C U 境界に対応し、 2 に等しいかまたは直交長さ - 2 に等しいエッジは、 C U 境界からのサブブロックの境界 8 サンプルに対応する。ここで、 T U の暗黙的分割が使用される場合、暗黙的 T U は真である。

【 0 1 3 2 】

2 . 6 . 9 輝度 / クロマ用の 4 C T U / 2 C T U 行バッファへの制限

【 0 1 3 3 】

水平エッジが C T U 境界と整列している場合、水平エッジのフィルタリングは、輝度の場合、 S p = 3 、クロマの場合、 S p = 1 、 S q = 1 に制限する。

【 0 1 3 4 】

2 . 7 V V C におけるイントラモード符号化

【 0 1 3 5 】

自然映像に表される任意のエッジ方向をキャプチャするために、 V T M 5 における指向性イントラモードの数は、 H E V C で使用されるように、 3 3 から 6 5 に拡張される。 H E V C にない新しい指向性モードは、図 1 2 に赤い点線の矢印で示されており、平面モードと直流モードは同じままである。これらのより密度の高い指向性イントラ予測モードは、すべてのブロックサイズ、および輝度および彩度イントラ予測の両方に適用される。

10

20

30

40

50

【 0 1 3 6 】

V T M 5において、いくつかの従来の角度イントラ予測モードは、非正方形のブロックのために、広角イントラ予測モードに適応的に置き換えられる。広角イントラ予測については、章 3 . 3 . 1 . 2 を参照されたい。

【 0 1 3 7 】

H E V Cにおいて、すべてのイントラ符号化されたブロックは正方形の形状を有し、その辺の各々の長さは 2 の累乗である。このように、D C モードを使用してイントラ予測子を生成するのに、除算演算を必要としない。V T M 5において、ブロックは長方形であってもよく、一般的な場合、ブロックごとに除算演算を使用することが必要である。D C 予測のための除算演算を回避するために、長辺のみを使用して非正方形のブロックの平均を計算する。

10

【 0 1 3 8 】

最大確率モード（M P M）リスト生成の複雑性を低く維持するために、2つの利用可能な近傍のイントラモードを考慮することによって、6個のM P Mを有するイントラモード符号化方法が使用される。M P Mリストを構築するために、以下の3つの態様を考える。

- i . デフォルトのイントラモード

- i i . 近傍イントラモード

- i i i . 導出イントラモード

【 0 1 3 9 】

M R L 符号化ツールおよびI S P 符号化ツールの適用の有無にかかわらず、イントラブロックに対して統一された 6 - M P Mリストが使用される。M P Mリストは、左上の近傍のブロックのイントラモードに基づいて構成される。ここで、左側のブロックのモードをL e f t とし、上記ブロックのモードをA b o v e とすると、統合M P Mリストは、以下のように構成される（左側および上側のブロックを図 1 3 で示す）。

20

- 近傍のブロックが利用可能でない場合、そのイントラモードはデフォルトでP l a n a r に設定される。

- モードL e f t およびA b o v e の両方が非角度モードである場合、次のようになる。

```
M P M  l i s t  { P l a n a r , D C , V , H , V - 4 , V + 4 }
```

- モードL e f t およびA b o v e の一方が角度モードであり、他方が非角度モードである場合、次のようになる。

MaxモードをL e f t a n d A b o v e でより大きいモードに設定する。

```
M P M  l i s t  { P l a n a r , M a x , D C , M a x - 1 , M a x + 1 , M a x - 2 }
```

- L e f t とA b o v e が共に角度があり、かつそれらが異なる場合、以下のようになる。

MaxモードをL e f t a n d A b o v e でより大きいモードに設定する。

L e f t モードとA b o v e モードとの差が 2 ~ 6 2 の範囲内にある場合

```
M P M  l i s t  { P l a n a r , L e f t , A b o v e , D C , M a x - 1 , M a x + 1 }
```

あるいは、

```
M P M  l i s t  { P l a n a r , L e f t , A b o v e , D C , M a x - 2 , M a x + 2 }
```

- L e f t とA b o v e が両方ともに角度があり、かつ同じである場合、以下のようになる。

```
M P M  l i s t  { P l a n a r , L e f t , L e f t - 1 , L e f t + 1 , D C , L e f t - 2 }
```

【 0 1 4 0 】

また、m p mインデックスコード名の第1のビンは、C A B A C コンテキスト符号化される。現在のイントラブロックがM R L 対応であるか、I S P 対応であるか、または通常

40

50

のイントラブロックであるかに対応して、合計 3 つのコンテキストが使用される。

【 0 1 4 1 】

6 つの M P M リストの生成処理中、プルーニングを使用して重複したモードを除去し、唯一のモードを M P M リストに含めることができるようになる。6 1 個の非 M P M モードのエンタロピー符号化のために、トランケーテッドバイナリコード (T B C) が使用される。

【 0 1 4 2 】

クロマイントラモード符号化の場合、クロマイントラモード符号化の場合、合計 8 つのイントラモードが許可される。これらのモードには、5 つの伝統的なイントラモードと 3 つの構成要素共通の線形モデルモードが含まれる (C C L M 、 L M _ A および L M _ L) 。クロマモード信号通知および導出処理を表 2 - 4 に示す。クロマモード符号化は、対応する輝度ブロックのイントラ予測モードに直接依存する。I スライスにおいて、輝度要素とクロマ要素に対するブロック分割構造の分離が有効化されているため、1 つのクロマブロックは複数の輝度ブロックに対応してもよい。よって、クロマ D M モードの場合、現在のクロマブロックの中心位置を含む、対応する輝度ブロックのイントラ予測モードは直接継承される。

【 0 1 4 3 】

【表 4】

表 2 - 4 c c l m _ が有効な時の輝度モードからのクロマ予測モードの導出

10

20

クロマ予測モード	対応する輝度イントラ予測モード				
	0	50	18	1	X(0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	81	81	81	81	81
5	82	82	82	82	82
6	83	83	83	83	83
7	0	50	18	1	X

30

【 0 1 4 4 】

2 . 8 量子化残差ブロック差動パルスコード変調 (Q R - B D P C M)

【 0 1 4 5 】

J V E T - M 0 4 1 3 において、量子化された残差ブロック差動パルスコード変調 (Q R - B D P C M) が、スクリーンコンテンツを効率的に符号化するために提案されている。

【 0 1 4 6 】

Q R - B D P C M で使用される予測方向は、垂直予測モードおよび水平予測モードであり得る。イントラ予測は、イントラ予測と同様に、予測方向（水平または垂直予測）にサンプルコピーすることで、ブロック全体で予測する。残差を量子化し、量子化された残差とその予測子（水平または垂直）量子化値との間のデルタを符号化する。これは、以下のように説明することができる。サイズ M (行) × N (列) のブロックについて、 $r_{i,j}$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ を、上または左ブロックの境界サンプルからのフィルタリングされていないサンプルを使用して、水平方向（予測ブロックに対して左隣の画素値を 1 ラインずつコピーする）または垂直方向（予測ブロックにおける各ラインに上隣のラインをコピーする）にイントラ予測を行った後の予測残差とする。 $Q(r_{i,j})$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ は、残差 $r_{i,j}$ の量子化バージョンを表し、この場合、残差は、元のブロックと予測ブロック値との間の差である。次に、ブロック D P C M が量子化された残差サンプルに適用され、その結果、要素 $r_{i,j}$ を有する修正された $M \times N$ 個

40

50

の配列 $R \sim$ が得られる。垂直 B D P C M が信号通知されると、以下のようになる。

【0 1 4 7】

【数 2】

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i=0, \quad 0 \leq j \leq (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \end{cases} \quad (2-7-1)$$

10

【0 1 4 8】

水平予測の場合、類似した規則が適用され、残差量子化サンプルは、以下の式によって得られる。

【0 1 4 9】

【数 3】

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), \quad j=0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), \quad 1 \leq j \leq (N-1) \end{cases}$$

20

(2-7-2)

【0 1 5 0】

残差量子化サンプル $r \sim_{i,j}$ はデコーダに送られる。

【0 1 5 1】

デコーダ側では、上記の計算を逆にして、 $Q(r_{i,j})$ 、 $0 \leq i \leq M-1$ 、 $0 \leq j \leq N-1$ を生成する。垂直予測の場合、

【0 1 5 2】

【数 4】

$$Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \quad (2-7-3)$$

30

【0 1 5 3】

水平方向の場合、

【0 1 5 4】

【数 5】

$$Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \quad (2-7-4)$$

40

【0 1 5 5】

逆量子化された残差 $Q^{-1}(Q(r_{i,j}))$ をイントラブロック予測値に加算し、再構成されたサンプル値を生成する。

50

【 0 1 5 6 】

このスキームの主な利点は、逆方向の D P C M を、係数の構文解析中にオンザフライで行うことができ、係数の構文解析中に予測子を追加するだけで済むこと、または、構文解析後に行うことことができるることである。

【 0 1 5 7 】

2 . 9 適応ループフィルタ

【 0 1 5 8 】

V T M 5において、ブロックに基づくフィルタ適応を伴う適応ループフィルタ (A L F) が適用される。輝度要素は、局所勾配の方向および働きに基づいて、 4×4 ブロックごとに 25 個のフィルタのうち 1 つを選択する。

10

【 0 1 5 9 】

2 . 9 . 1 フィルタ形状

【 0 1 6 0 】

V T M 5において、2つの菱形フィルタ形状 (図 14 に示す) が使用される。輝度要素には 7×7 菱形を適用し、クロマ要素には 5×5 菱形を適用する。

【 0 1 6 1 】

2 . 9 . 2 ブロック区分

【 0 1 6 2 】

輝度要素の場合、各 4×4 ブロックを 25 個のクラスのうちの 1 つに分類する。分類インデックス C は、その方向性 D およびアクティビティ A ^ の量子化値に基づいて、以下のように導出される。

20

【 0 1 6 3 】

【 数 6 】

$$C = 5D + \hat{A} \quad (2 - 9 - 1)$$

【 0 1 6 4 】

D および A ^ を計算するために、まず、1 - D ラプラシアンを使用して、水平、垂直および 2 つの対角線方向の勾配を計算する。

30

【 0 1 6 5 】

【 数 7 】

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k, l}, \quad V_{k, l} = |2R(k, l) - R(k, l-1) - R(k, l+1)| \quad (2 - 9 - 2)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k, l}, \quad H_{k, l} = |2R(k, l) - R(k-1, l) - R(k+1, l)| \quad (2 - 9 - 3)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k, l}, \quad D1_{k, l} = |2R(k, l) - R(k-1, l-1) - R(k+1, l+1)| \quad (2 - 9 - 4)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k, l}, \quad D2_{k, l} = |2R(k, l) - R(k-1, l+1) - R(k+1, l-1)| \quad (2 - 9 - 5)$$

40

【 0 1 6 6 】

この場合、i および j は、 4×4 イントラブロックの左上のサンプルの座標を表し、R (i , j) は、座標 (i , j) において再構成されたサンプルを示す。

【 0 1 6 7 】

50

ブロック区分の複雑性を低減するために、サブサンプリングされた 1 - D ラプラシアン計算が適用される。図 15 (a) ~ (d) に示すように、すべての方向の勾配計算に同じサブサンプリング位置を用いる。

【 0 1 6 8 】

そして、水平方向および垂直方向の勾配の D 最大値および D 最小値を以下のように設定する。

【 0 1 6 9 】

【 数 8 】

$$g_{h, v}^{\max} = \max(g_h, g_v), \quad g_{h, v}^{\min} = \min(g_h, g_v) \quad (2-9-6)$$

10

【 0 1 7 0 】

2 つの対角線方向の勾配の最大値および最小値は、以下のように設定される。

【 0 1 7 1 】

【 数 9 】

$$g_{d0, d1}^{\max} = \max(g_{d0}, g_{d1}), \quad g_{d0, d1}^{\min} = \min(g_{d0}, g_{d1}) \quad (2-9-7)$$

20

【 0 1 7 2 】

指向性 D の値を導出するために、これらの値を互いに且つ 2 つの閾値 t_1 および t_2 と比較する。

【 0 1 7 3 】

【 数 1 0 】

ステップ 1. $g_{h, v}^{\max} \leq t_1 \cdot g_{h, v}^{\min}$ と $g_{d0, d1}^{\max} \leq t_1 \cdot g_{d0, d1}^{\min}$ の両方が真である場合、D が 0 に設定される。

30

ステップ 2. $g_{h, v}^{\max}/g_{h, v}^{\min} > g_{d0, d1}^{\max}/g_{d0, d1}^{\min}$ の場合、ステップ 3 から続け、あるいは、ステップ 4 から続ける。

ステップ 3. $g_{h, v}^{\max} > t_2 \cdot g_{h, v}^{\min}$ である場合、D は 2 に設定され、あるいは、D は 1 に設定される。

ステップ 4. $g_{d0, d1}^{\max} > t_2 \cdot g_{d0, d1}^{\min}$ である場合、D は 4 に設定され、あるいは、D は 3 に設定される。

40

【 0 1 7 4 】

アクティビティ値 A は、以下のように計算される。

【 0 1 7 5 】

【 数 1 1 】

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k, l} + H_{k, l}) \quad (2-9-8)$$

【 0 1 7 6 】

A をさらに 0 ~ 4 の範囲に量子化し、量子化された値を A^\wedge とする。

50

【0177】

ピクチャにおけるクロマ要素に対して、分類方法は適用されず、即ち、単一の A L F 係数のセットが各クロマ要素に対して適用される。

【0178】

2.9.3 フィルタ係数およびクリッピング値の幾何学的変換

【0179】

各 4×4 輝度ブロックをフィルタリングする前に、そのブロックに対して計算された勾配値に基づいて、フィルタ係数 $f(k, l)$ および対応するフィルタクリッピング値 $c(k, l)$ に対して、回転または対角線および垂直方向の反転等の幾何学的変換を施す。これは、これらの変換をフィルタ支持領域内のサンプルに適用することに等しい。その考えは、A L F が適用される異なるブロックを、それらの方向性を揃えることによって、より類似させることである。

10

【0180】

対角線、垂直方向の反転および回転を含む 3 つの幾何学的変換を紹介する。

Diagonal : $f_D(k, l) = f(l, k)$, $c_D(k, l) = c(l, k)$,
(2-9-9)

Vertical flip : $f_V(k, l) = f(k, K - l - 1)$, $c_V(k, l) = c(k, K - l - 1)$ (2-9-10)

Rotation : $f_R(k, l) = f(K - l - 1, k)$, $c_R(k, l) = c(K - l - 1, k)$ (2-9-11)

20

【0181】

この場合、K はフィルタのサイズであり、 $0 \leq k, l \leq K - 1$ が係数座標であり、位置 $(0, 0)$ は左上隅にあり、位置 $(K - 1, K - 1)$ は右下隅にある。この変換は、そのブロックに対して計算された勾配値に基づいて、フィルタ係数 $f(k, l)$ およびクリッピング値 $c(k, l)$ に適用される。変換と 4 方向の 4 つの勾配との関係を以下の表にまとめる。

【0182】**【表 5】**

表 2-5 1 つのブロックに対して計算された勾配と変換とのマッピング

30

勾配値	変換
$g_{d2} < g_{d1}$ and $g_h < g_v$	変換なし
$g_{d2} < g_{d1}$ and $g_v < g_h$	対角線
$g_{d1} < g_{d2}$ and $g_h < g_v$	垂直方向の反転
$g_{d1} < g_{d2}$ and $g_v < g_h$	回転

【0183】

2.9.4 フィルタパラメータ信号通知

【0184】

V T M 5において、A L F フィルタパラメータは、適応パラメータセット (A P S) において信号通知される。1つのA P S において、最大 25 組の輝度フィルタ係数およびクリッピング値インデックス、並びに最大 1 組のクロマフィルタ係数およびクリッピング値インデックスを信号通知することができる。ビットオーバーヘッドを低減するために、異なる分類のフィルタ係数をマージすることができる。スライスヘッダにおいて、現在のスライスに使用されるA P S のインデックスが信号通知される。

40

【0185】

A P S から復号化されたクリッピング値インデックスは、クリッピング値の輝度テーブル (L u m a t a b l e) およびクリッピング値のクロマテーブル (C h r o m a t a b l e) を使用してクリッピング値を判定するようになる。これらのクリッ

50

ピング値は、内部ビット深度に依存する。具体的には、クリッピング値の輝度テーブルおよびクロマテーブルは、以下の式によって得られる。

【0186】

【数12】

$$\text{AlfClip}_L = \left\{ \text{round} \left(2^{\frac{B-N-n+1}{N}} \right) \text{ for } n \in [1..N] \right\}, \quad (2-9-12)$$

$$\text{AlfClip}_C = \left\{ \text{round} \left(2^{(B-8)+8\frac{(N-n)}{N-1}} \right) \text{ for } n \in [1..N] \right\} \quad (2-9-13)$$

10

【0187】

ここで、Bは内部ビット深度に等しく、NはVTM5.0における許容されるクリッピング値の数である4に等しい。

【0188】

フィルタリング処理はCTBレベルで制御されてもよい。ALFが輝度CTBに適用されるかどうかを示すために、常に1つのフラグが信号通知される。1つの輝度CTBは、16個の固定フィルタセットのうち1つのフィルタセットを選択し、複数のAPSから1つのフィルタセットを選択することができる。どのフィルタセットが適用されるかを示すように、輝度CTBのためにフィルタセットインデックスが信号通知される。エンコーダおよびデコーダの両方において、16個の固定フィルタセットを予め規定し、ハードコードする。

20

【0189】

フィルタ係数は、128に等しいノルムで量子化される。乗算の複雑性を抑えるために、非中心位置の係数値が $-2^7 \sim 2^7 - 1$ の範囲内に含まれるように、ビットストリーム適合性が適用される。中心位置係数はビットストリームにおいて信号通知されず、128に等しいと見なされる。

【0190】

2.9.5 フィルタリング処理

【0191】

30

デコーダ側において、CTBのためにALFが有効化されると、CU内の各サンプルR(i, j)がフィルタリングされ、その結果、以下に示すように、サンプル値R'(i, j)が得られる。

【0192】

【数13】

$$R'(i, j) = R(i, j) + \left(\left(\sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i+k, j+l) - R(i, j), c(k, l)) + 64 \right) \gg 7 \right) \quad (2-9-14)$$

40

【0193】

ここで、f(k, l)は復号化されたフィルタ係数を表し、K(x, y)はクリッピング関数であり、c(k, l)は復号化されたクリッピングパラメータを表す。変数kおよびlは、-L/2とL/2との間で変化し、ここで、Lはフィルタ長を表す。クリッピング関数 $K(x, y) = \min(y, \max(-y, x))$ であり、これは関数Clip3(-y, y, x)に対応する。

【0194】

2.9.6 行バッファ削減のための仮想境界フィルタリング処理

【0195】

VTM5において、ALFの行バッファ要件を低減するために、水平CTU境界付近の

50

サンプルに対して修正されたブロック区分およびフィルタリングが用いられる。そのため、図 16 に示すように、水平方向 C T U 境界を「N」個のサンプルで移動させることによって、仮想境界を行として定義し、N は、輝度要素に対しては 4 に等しく、クロマ要素に対しては 2 に等しい。

【0196】

図 2 - 11 に示すように、輝度要素に対して修正されたブロック区分を適用する。仮想境界より上の 4×4 ブロックの 1 D ラプラシアン勾配計算のために、仮想境界より上のサンプルのみを使用する。同様に、仮想境界より下の 4×4 ブロックの 1 D ラプラシアン勾配計算のために、仮想境界より下のサンプルのみを使用する。従って、1 D ラプラシアン勾配計算に使用されるサンプルの低減された数を考慮に入れることで、アクティビティ値 A の量子化を拡大縮小する。

10

【0197】

フィルタリング処理のために、仮想境界における対称パディング演算が、輝度要素およびクロマ要素の両方に使用される。図 17 に示すように、フィルタリングされる試料が仮想境界の下に位置する場合、仮想境界の上に位置する近傍のサンプルにパディングを行う。一方、他方の側の対応するサンプルもまた、対称的にパディングされる。

【0198】

2.10 サンプル適応オフセット (Sample Adaptive Offset : S A O)

【0199】

エンコーダが C T B ごとに規定したオフセットを使用して、非ブロック化フィルタ後の再構成された信号にサンプル適応オフセット (S A O) を適用する。H M エンコーダは、まず、現在のスライスに対して S A O 処理を行うか否かを決定する。スライスに S A O を適用した場合、各 C T B は、表 2 - 6 に示すように、5 つの S A O タイプの 1 つに分類される。S A O の概念は、画素をカテゴリに分類し、各カテゴリの画素にオフセットを加えることで歪みを低減することである。S A O 演算は、S A O タイプ 1 - 4 における画素分類のためにエッジ特性を使用するエッジオフセット (E O) と、S A O タイプ 5 における画素分類のために画素強度を使用するバンドオフセット (B O) とを含む。適用可能な C T B は、それぞれ、sao_merge_left_flag、sao_merge_up_flag、S A O タイプ、および 4 つのオフセットを含む S A O パラメータを有する。sao_merge_left_flag が 1 に等しい場合、現在の C T B は、S A O タイプおよび C T B のオフセットを左に再利用する。sao_merge_up_flag が 1 に等しい場合、現在の C T B は、S A O タイプおよび上記 C T B のオフセットを再利用する。

20

【0200】

【表 6】

表 2 - 6 S A O タイプの仕様

S A O タイプ	使用するサンプル適応オフセットタイプ	カテゴリ数
0	なし	0
1	1-D 0° パターンエッジオフセット	4
2	1-D 90° パターンエッジオフセット	4
3	1-D 135° パターンエッジオフセット	4
4	1-D 45° パターンエッジオフセット	4
5	バンドオフセット	4

30

【0201】

50

2.10.1 各 S A O タイプの動作

【0202】

図18に示すように、エッジオフセットは、4つの1-D 3画素パターンを使用して、エッジの方向情報を考慮して現在の画素 p を分類する。左から右へ、 0° 、 90° 、 135° 、 45° 次のようになる。

【0203】

表2-7に従って、各 C T B を5つのカテゴリに分類する。

【0204】

【表7】

表2-7 EOの画素分類規則

カテゴリ	条件	意味
0	以下のいずれも当てはまらない	大きく単調
1	$p < 2$ 近傍	ローカル最小
2	$p < 1$ neighbour && $p == 1$ neighbour	エッジ
3	$p > 1$ neighbour && $p == 1$ neighbour	エッジ
4	$p > 2$ neighbours	ローカル最大

10

20

【0205】

バンドオフセット (B a n d _ O f f s e t、BO) は、画素値の上位5ビットをバンドインデックスとして使用し、1つのC T B 領域におけるすべての画素を32個の均一なバンドに分類する。換言すると、画素強度範囲は、ゼロから最大強度値までの32個の等分したセグメントに分割される（例えば、8ビット画素の場合、255）。図19に示すように、隣り合う4つのバンドをまとめ、それぞれのグループをその最も左側の位置で示す。エンコーダは、各帯域のオフセットを補償することで、すべての位置を探索し、歪みが最大に低減されたグループを得る。

30

【0206】

2.11 統合されたインター予測およびイントラ予測 (C I I P)

【0207】

V T M 5において、C Uがマージモードで符号化される場合、C Uが少なくとも64個の輝度サンプルを含む（すなわち、C U幅×C U高さが64以上である）場合、およびC U幅およびC U高さの両方が128個の輝度サンプル未満である場合、現在のC Uにインター／イントラ予測 (C I I P) 複合モードが適用されるかどうかを示すために、追加のフラグが信号通知される。その名称が示すように、C I I P予測は、インター予測信号とイントラ予測信号とを合成する。通常のマージモードに適用されるのと同じインター予測処理を使用して、C I I PモードP i n t e rにおけるインター予測信号を導出し、平面モードによる通常のイントラ予測処理の後、イントラ予測信号P i n t r aを導出する。そして、重み付け平均を使用してイントラ予測信号とインター予測信号とを合成し、この場合、左上の近傍のブロック（図20に示す）の符号化モードによって、重み値を以下のように算出する。

40

- 上隣が利用可能であり、イントラ符号化されている場合、i s I n t r a T o pを1に設定し、あるいは、i s I n t r a T o pを0に設定する。

- 左隣が利用可能であり、イントラ符号化されている場合、i s I n t r a L e f tを1に設定し、あるいは、i s I n t r a L e f tを0に設定する。

- (i s I n t r a L e f t + i s I n t r a T o p)が2に等しい場合、w tは3に設定される。

50

- あるいは、(isIntraLeft + isIntraLeft)が1に等しい場合、wtは2に設定される。

- あるいは、wtを1に設定する。

CPIP予測は、以下のように形成される。

$\text{PCIP} = ((4 - \text{wt}) * \text{Pinter} + \text{wt} * \text{Pinta} + 2) >> 2 \quad (3-1)$

【0208】

2.12 クロマスケーリングを伴う輝度マッピング(LMCS)

【0209】

VTM5において、ループフィルタの前に新しい処理ブロックとして、クロマスケーリングによる輝度マッピング(LMCS)と呼ばれる符号化ツールが追加される。LMCSは、2つの主な要素を有する。1) 適応型区分線形モデルに基づく輝度要素のインループマッピング、および、2) クロマ要素のために、輝度依存クロマ残差スケーリングを適用する。図21は、デコーダの観点から見たLMCSアーキテクチャを示す。図21中の薄い青色の影が付けられたブロックは、マッピングされたドメインにおいて処理が適用される場所を示し、これらは、逆量子化、逆変換、輝度イントラ予測、および輝度予測と輝度残差との加算を含む。図21中の影が付けられていないブロックは、元の(即ち、マッピングされていない)ドメインにおいて処理が適用される場所を示し、これらは、非ブロック化、ALF、SAO等のループフィルタ、動き補償予測、クロマイントラ予測、クロマ残差とともにクロマ予測を加算し、復号化されたピクチャを参照ピクチャとして記憶することを含む。図21における淡黄色の陰影付きブロックは、輝度信号の前方および後方マッピング並びに輝度依存性クロマスケーリング処理を含む、新しいLMCS機能ブロックである。VVCにおける他のほとんどのツールと同様に、LMCSは、SPSフラグを使用して、シーケンスレベルで有効/無効にすることができる。

【0210】

3. 実施形態が解決しようとする課題の例

【0211】

1つのパレットフラグは、通常、そのエントロピー符号化において異なる制限および分散を有することができる、現在のCUにおいてパレットモードが採用されているかどうかを示すために用いられる。しかしながら、前の映像符号化規格において、どのようにしてパレットフラグをより良好に符号化するかについては、まだ十分に検討されていない。

【0212】

パレットサンプルは、ポストループフィルタ処理によって処理される場合、視覚的アーチファクトを有する可能性がある。

【0213】

非正方形のブロックの場合、パレットの走査順序を改善することができる。

【0214】

4. 実施形態の例

【0215】

以下の詳細な発明は、一般的な概念を説明するための例であると考えられるべきである。これらの発明は狭い意味で解釈されるべきではない。さらに、これらの発明は、任意の方法で組み合わせることができる。

1. 変換ユニット/予測ユニット/符号化ブロック/領域のためのパレットモードの使用の指示は、予測モードとは別個に符号化されてもよい。

a. 一例において、予測モードは、パレットの使用の指示の前に符号化されてもよい。
i. 代替的に、さらに、パレットの使用の指示は、予測モードに基づいて条件付きで信号通知されてもよい。

1. 一例において、予測モードがイントラブロックコピーモードである場合(すなわち、MODE_IBC)、パレットモードの使用の指示の信号通知はスキップされてもよい。代替的に、さらに、現在の予測モードがMODE_IBCである場合、パレットの使用の指示が偽であると推測されてもよい。

10

20

30

40

50

2 . 一例において、予測モードがインターモード（すなわち、MODE_INTERRUPT）である場合、パレットモードの使用の指示の信号通知はスキップされてもよい。代替的に、さらに、現在の予測モードがMODE_INTERRUPTである場合、パレットモードの使用の指示が偽であると推測されてもよい。

3 . 一例において、予測モードがイントラモード（すなわち、MODE_INTERRUPTRA）である場合、パレットモードの使用の指示の信号通知はスキップされてもよい。代替的に、さらに、現在の予測モードがMODE_INTERRUPTRAである場合、パレットモードの使用の指示が偽であると推測されてもよい。

4 . 一例において、予測モードがスキップモードである（すなわち、スキップフラグが1に等しい）場合、パレットモードの使用の指示の信号通知はスキップされてもよい。代替的に、さらに、現在のCUにおいてスキップモードが採用されている場合、パレットモードの使用の指示が偽であると推測されてもよい。10

5 . 一例において、予測モードがイントラモード（例えば、MODE_INTERRUPTA）である場合、パレットモードの使用の指示が信号通知されてもよい。代替的に、さらに、予測モードがインターモードまたはイントラブロックコピーモードである場合、パレットモードの使用の指示の信号通知はスキップされてもよい。

a) 代替的に、さらに、予測モードがイントラモードであり、パルス符号変調（PCM）モードでない場合、パレットモードの使用の指示を信号通知されてもよい。

b) 代替的に、さらに、予測モードがイントラモードである場合、パレットモードの使用の指示は、PCMモードの使用の指示の前に信号通知されてもよい。一例において、パレットモードが適用される場合、PCMモードの使用の信号通知はスキップされてもよい。20

c) 代替的に、さらに、予測モードがインターモードまたはイントラブロックコピーモードである場合、パレットモードの使用の指示の信号通知はスキップされてもよい。

6 . 一例において、予測モードがインターモード（例えば、MODE_INTERRUPTR）である場合、パレットモードの使用の指示が信号通知されてもよい。

a) 代替的に、予測モードがイントラモードである場合、パレットモードの使用の指示の信号通知はスキップされてもよい。

7 . 一例において、予測モードがイントラブロックコピーモードである場合、パレットモードの使用の指示が信号通知されてもよい。代替的に、さらに、予測モードがインターモードまたはイントラモードである場合、パレットモードの使用の指示の信号通知をスキップしてもよい。30

i i . 代替的に、さらに、ピクチャ／スライス／タイルグループのタイプに基づいて、パレットモードの使用の指示を条件付きで信号通知されてもよい。

b . 一例において、予測モードは、パレットモードの使用の指示の後に符号化されてもよい。

c . 一例において、予測モードがINTRAモードまたはINTER_MODEである場合、パレットモードの使用の指示が信号通知されてもよい。

i . 一例において、パレットモードの使用の指示は、スキップフラグ、予測モード、およびPCMモードのフラグの後に符号化されてもよい。40

i i . 一例において、パレットモードの使用の指示は、スキップフラグ、予測モードの後、PCMモードのフラグの前に符号化されてもよい。

i i i . 一例において、現在のブロックがイントラモードで符号化される場合、パレットモードおよびIBCモードの指示がさらに信号通知されてもよい。

1 . 一例において、1つのビットフラグは、パレットまたはIBCモードのいずれかが信号通知されるかどうかを示すように信号通知されてもよい。

2 . 一例において、ビットフラグの信号通知は、ブロック寸法、1つのタイル／タイルグループ／スライス／ピクチャ／シーケンスに対してIBCまたはパレットモードが有効化されるかどうか等の特定の条件下でスキップされてもよい。50

d . 一例において、予測モード（例えば、それがイントラモードであるかまたはインター モードであるか）は、まず符号化され、続いて、それがパレットモードであるかどうかの条件付き信号通知が行われる。

i . 一例において、予測モードがイントラモードである場合、それがパレットモードであるかどうかを示すように、別のフラグをさらに信号通知されてもよい。

1 . 一例において、1つの映像データユニット（例えば、シーケンス／ピクチャ／ タイルグループ／タイル）に対してパレットモードが有効化される場合、「別のフラグ」が信号通知されてもよい。

2 . 一例において、「別のフラグ」は、ブロック寸法の条件下で信号通知されてもよい。 10

3 . 代替的に、さらに、それがパレットモードでない場合、それがPCMモードであるかどうかを示すように、1つのフラグをさらに信号通知されてもよい。

4 . 一例において、「別のフラグ」は近傍のブロックの情報に従ってコンテキスト 符号化されてもよい。代替的に、「別のフラグ」は、1つのコンテキストのみで符号化されたコンテキストでもよい。代替的に、「別のフラグ」は、バイパス符号化されても、すなわちコンテキストなしであってもよい。

i i . 代替的に、予測モードがインターモードである場合、IBCモードであるかどうかを示すように、別のフラグをさらに信号通知されてもよい。

1 . 一例において、1つの映像データユニット（例えば、シーケンス／ピクチャ／ タイルグループ／タイル）に対してIBCモードが有効化されるとき、「別のフラグ」が信号通知されてもよい。 20

2 . 一例において、「別のフラグ」は、ブロック寸法の条件下で信号通知されてもよい。

2 . パレットモードを予測モードの追加候補として追加することが提案される。パレットモードの使用の指示は、例えば、上記の例示的な実施形態1で説明したように、予測モードに基づいて判定／信号通知することができる。いくつかの実施形態において、パレットモードの使用の指示を予測モードとは別個に信号通知する必要がない。

a . 一例において、予測モードは、イントラスライス／Iピクチャ／イントラタイル グループのためのイントラ、イントラブロックコピー、およびパレットモードを含んでもよい。 30

b . 代替的に、予測モードは、イントラスライス／Iピクチャ／イントラタイルグルーピングのためのイントラ、パレットモードを含んでもよい。

c . 一例において、予測モードは、 4×4 ブロックのためのイントラ、イントラブロッ クコピー、およびパレットモードを含んでもよい。

d . 一例において、予測モードは、スライス間／および／またはBピクチャ／タイル グループのためのイントラ、インター、イントラブロックコピー、およびパレットモードを含んでもよい。

e . 一例において、予測モードは、スライス間／Pおよび／またはBピクチャ／タイル グループのためのイントラ、インター、イントラブロックコピーモードを含んでもよい。

f . 代替的に、予測モードは、イントラ、インター、イントラブロックコピー、およ びパレットモードのうちの少なくとも2つを含んでもよい。 40

g . 一例において、インター モードは、 4×4 ブロックの予測モードに含まれなくて もよい。

h . 一例において、ブロックがスキップモードとして符号化されない場合（これはインター モードの特殊な場合である）、予測モードインデックスは、異なるビンを使用してコンテキストベースで符号化されてもよい。いくつかの実施形態において、1つ以上のビンの信号通知は、ブロック寸法（例えば、 4×4 ）または予測モードが無効にされる（例えは、IBCモードが無効にされ、対応するビンがスキップされる）等の条件によってスキップされてもよい。

i . 一例において、4つのモードの2値化は、イントラ（1）、インター（00）

、 I B C (0 1 0) 、およびパレット (0 1 1) として定義される。ここで、3つのピンが使用され、各ビットは1つのピン値に対応する。

i i . 一例において、4つのモードの2値化は、図10に示すように、イントラ (1 0) 、インター (0 0) 、I B C (0 1) 、およびパレット (1 1) として定義される。ここで、2つのピンが使用され、各ビットは1つのピン値に対応する。

i i i . 一例において、現在のスライスがイントラスライスであり、S P S においてI B C が有効化されていない場合、パレットおよびイントラモードの2値化は、パレット (1) およびイントラ (0) のように定義される。ここで、1つのピンが使用される。

i v . 一例において、現在のスライスがイントラスライスでなく、S P S においてI B C が有効化されていない場合、パレット、インターおよびイントラモードの2値化は、i n t r a (1) 、i n t e r (0 0) 、およびP a l e t t e (0 1) として定義される。ここで、2つのピンが使用され、各ビットは1つのピン値に対応する。

v . 一例において、現在のスライスがイントラスライスであり、S P S においてI B C が有効化されている場合、パレットおよびイントラモードの2値化は、I B C (1) 、パレット (0 1) 、イントラ (0 0) のように定義される。ここで、2つのピンが使用され、各ビットは1つのピン値に対応する。

v i . 一例において、4つのモードの2値化は、インター (1) 、イントラ (0 1) 、I B C (0 0 1) 、およびパレット (0 0 0) として定義される。ここで、3つのピンが使用され、各ビットは1つのピン値に対応する。

v i i . 一例において、4つのモードの2値化は、イントラ (1) 、インター (0 1) 、I B C (0 0 1) 、およびパレット (0 0 0) として定義される。ここで、3つのピンが使用され、各ビットは1つのピン値に対応する。

v i i i . 一例において、4つのモードの2値化は、図11に示すように、インター (0) 、イントラ (1 0) 、I B C (1 1 1) 、およびパレット (1 1 0) として定義される。ここで、3つのピンが使用され、各ビットは1つのピン値に対応する。

3 . パレット / I B C モードの使用の指示の信号通知は、他のモードの情報に依存してもよい。

a . 一例において、現在の予測モードがイントラモードであり、I B C モードでない場合、パレットモードの使用の指示が信号通知されてもよい。

b . 一例において、現在の予測モードがイントラモードであり、パレットモードでない場合、I B C モードの使用の指示は、信号通知されてもよい。

4 . モード情報をどのように信号通知するかは、スライス / ピクチャ / タイルグループのタイプに依存してもよい。

a . 一例において、それがスライスイントラタイルグループである場合、それがI B C モードであるかどうかを示すように、1つのフラグを信号通知されてもよい。I B C モードでない場合、パレットモードであるかイントラモードであるかどうかを示すように、別のフラグをさらに信号通知されてもよい。

b . 一例において、それがI - スライス / イントラタイルグループである場合、それがイントラモードであるかどうかを示すように、1つのフラグを信号通知されてもよい。イントラモードでない場合、パレットモードであるかI B C モードであるかどうかを示すように、別のフラグをさらに信号通知されてもよい。

5 . パレットモードの使用の指示は、以下の条件に基づいて信号通知および / または導出されてもよい。

a . 現在のブロックのブロック寸法

i . 一例において、パレットモードの使用の指示は、幅 * 高さが閾値以下（例えば、6 4 * 6 4）のブロックに対してのみ信号通知されてもよい。

i i . 一例において、パレットモードの使用の指示は、幅および高さの両方が閾値（例えば、6 4）以上であるブロックに対してのみ信号通知されてもよい。

i i i . 一例において、パレットモードの使用の指示は、以下のすべての条件が真であるブロックに対してのみ信号通知されてもよい。

10

20

30

40

50

- 1 . 幅および / または高さが閾値、例えば 16 以上である。
- 2 . 幅および / または高さが閾値以下 (例えば、32 または 64)
- i v . 一例において、パレットモードの使用の指示は、幅が高さに等しいブロック (即ち、正方形のブロック) に対してのみ信号通知されてもよい。
- b . 現在のブロックの予測モード
 - c . 現在のブロックの量子化パラメータ
 - d . 近傍のブロックのパレットフラグ
 - e . 近傍のブロックのイントラブロックコピーフラグ
 - f . カラーフォーマットの指示 (例えば、4 : 2 : 0 、 4 : 4 : 4)
 - g . 分離 / 二重符号化ツリー構造
 - h . スライス / タイルグループのタイプおよび / またはピクチャのタイプ
- 6 . IBC モードの使用の指示は、以下の条件に基づいて信号通知および / または導出されてもよい。
- a . 現在のブロックのブロック寸法
 - i . 一例において、IBC モードの使用の指示は、幅または高さの両方が 128 より小さいブロックに対してのみ信号通知されてもよい。
 - b . 現在のブロックの予測モード
 - c . 現在のブロックの量子化パラメータ
 - d . 近傍のブロックのパレットフラグ
 - e . 近傍のブロックのイントラブロックコピーフラグ
 - f . カラーフォーマットの指示 (例えば、4 : 2 : 0 、 4 : 4 : 4)
 - g . 分離 / 二重符号化ツリー構造
 - h . スライス / タイルグループのタイプおよび / またはピクチャのタイプ
- 7 . パレットモードは、非ブロック化決定処理においてイントラモード (例えば、MODE_INTRA) として取り扱われてもよい。
- a . 一例において、 p 側または q 側のサンプルがパレットモードで符号化される場合、境界強度は 2 に設定される。
 - b . 一例において、 p 側および q 側の両方のサンプルがパレットモードで符号化される場合、境界強度は 2 に設定される。
 - c . 代替的に、パレットモードは、非ブロック化決定処理においてインターモード (例えば、MODE_INTER) として扱われてもよい。
- 8 . パレットモードは、非ブロック化決定処理において、別個のモード (例えば、MODE_PLT) として取り扱われてもよい。
- a . 一例において、 p 側および q 側のサンプルがパレットモードで符号化される場合、境界強度は 0 に設定される。
 - i . 代替的に、片側のサンプルがパレットモードで符号化される場合、境界強度は 0 に設定される。
 - b . 一例において、 p 側のサンプルが IBC モードで符号化され、 q 側のサンプルがパレットモードで符号化される場合、境界強度は 1 に設定され、逆もまた同様である。
 - c . 一例において、 p 側のサンプルがイントラモードで符号化され、 q 側のサンプルがパレットモードで符号化される場合、境界強度は 2 に設定され、逆もまた同様である。
- 9 . パレットモードは、非ブロック化処理において変換スキップブロックとして扱われてもよい。
- a . 代替的に、パレットモードは、非ブロック化処理において、 BDP CM ブロックとして扱われてもよい。
- 10 . スライス / タイルグループ / ピクチャレベルフラグに基づいて、 1 つのブロックに対するパレットモードの使用の指示を信号通知および / または導出してもよい。
- a . 一例において、このフラグは、動きベクトル差 (MMVD 、別名、 UMOVE) および / または適応型動きベクトル解像度 (AMVR) モード (例えば、 slice_flag_acmvd_flag) のマージにおいて、わずかな動きベクトル差 (MVD) が許

10

20

30

30

40

50

可されるかどうかを示す。代替的に、さらに、`s l i c e_f r a c m m v d_f l a g` がわずかな MVD が有効化されていることを示す場合、パレットモードの使用的指示の信号通知はスキップされ、パレットモードは無効化されていると推測される。

b . 一例において、このフラグは、スライス / タイルグループ / ピクチャに対してパレットモードが有効化されているかどうかを示す。代替的に、さらに、このようなフラグが、パレットモードが無効であることを示す場合、1つのブロックについてのパレットモードの使用的信号通知をスキップし、パレットモードが無効であると推測する。

11 . スライス / タイルグループ / ピクチャレベルフラグに基づいて、1つのブロックに対するイントラブロックコピーモード (IBC) の使用的指示の信号通知および / または導出されてもよい。

a . 一例において、このフラグは、動きベクトル差 (MMVD、別名、UMVE) および / または適応型動きベクトル解像度 (AMVR) モード (例えば、`s l i c e_f r a c m m v d_f l a g`) とのマージにおいて、わずかな動きベクトル差 (MVD) が許可されるかどうかを示す。代替的に、さらに、`s l i c e_f r a c m m v d_f l a g` がわずかな MVD が有効であることを示す場合、IBC モードの使用的指示の信号通知はスキップされ、IBC モードは無効であると推測される。

b . 一例において、このフラグは、スライス / タイルグループ / ピクチャに対して IBC モードが有効化されているかどうかを示す。代替的に、さらに、このようなフラグが、IBC モードが無効であることを示す場合、1つのブロックに対して IBC モードを使用する信号通知をスキップし、IBC モードが無効であると推測する。

12 . 1つのパレットエントリに関連付けられたサンプルは、元の / 再構成されたサンプルの内部ビット深度および / またはビット深度とは異なるビット深度を有してもよい。

a . 一例において、1に関連付けられたサンプルが N に等しいビット深度を有し得ることを示すと、以下が適用されてもよい。

i . 一例において、N は整数 (例えば、8) であってもよい。

i i . 一例において、N は、元の / 再構成されたサンプルの内部ビット深度および / またはビット深度よりも大きくてもよい。

i i i . 一例において、N は、元の / 再構成されたサンプルの内部ビット深度および / またはビット深度より小さくてもよい。

i v . 一例において、N は、以下に依存してもよい。

- 1 . 現在のブロックのブロック寸法
- 2 . 現在のブロックの量子化パラメータ
- 3 . カラーフォーマットの指示 (例えば、4 : 2 : 0、4 : 4 : 4)
- 4 . 分離 / 二重符号化ツリー構造
- 5 . スライス / タイルグループのタイプおよび / またはピクチャのタイプ
- 6 . パレットエントリ
- 7 . 予測パレットエントリ
- 8 . 色要素のインデックス

b . 一例において、複数のパレットエントリに関連付けられたサンプルは、異なるビット深度を有してもよい。

i . 一例において、C0、C1 は、現在のパレットにおける 2 つのパレットエントリであり、それぞれ b0 および b1 に等しいビット深度を有してもよい。b0 は b1 に等しくなくてもよい。

1 . 一例において、b0 は、元の / 再構成されたサンプルの内部ビット深度および / またはビット深度よりも大きくて / 小さくてもよく、且つ / または b1 は、元の / 再構成されたサンプルの内部ビット深度および / またはビット深度より大きくて / 小さくてもよい。

c . 一例において、パレットモードにおいて、サンプルは、パレットエントリに関連付けられたサンプルのシフトされた値に従って再構成されてもよい。

i . 一例において、サンプルは、パレットエントリにおけるサンプルを M ビットだ

10

20

30

40

50

け左シフトすることによって再構成されてもよい。

i i . 一例において、再構成された値は、 $(C <<< M) + (1 < (M - 1))$ で
あり、この場合、C はパレットエントリである。

i i i . 一例において、サンプルは、パレットエントリにおけるサンプルを M ビッ
ト右シフトすることによって再構成されてもよい。

i v . 一例において、再構成された値は、 $c1ip((C + (1 << (M - 1)))$
 $) >> M, 0, (1 << N) - 1)$ であってもよく、この場合、C はパレットエントリで
あり、N は再構成のビット深度である。

v . 代替的に、さらに、一例において、M は、パレットエントリに関連付けられた
サンプル間のビット深度の差と、再構成されたサンプル / 元のサンプルの内部ビット深度
とに依存してもよい。 10

1 . 一例において、M は、内部ビット深度からパレットエントリにおけるサンプ
ルのビット深度を引いたものに等しくてもよい。

2 . 一例において、M は、パレットエントリにおけるサンプルのビット深度から
内部ビット深度を引いたものに等しくてもよい。

3 . 一例において、M は、元のサンプルのビット深度からパレットエントリにお
けるサンプルのビット深度を引いたものに等しくてもよい。

4 . 一例において、M は、パレットエントリにおけるサンプルのビット深度から
元のサンプルのビット深度を引いたものに等しくてもよい。

5 . 一例において、M は、再構成されたサンプルのビット深度からパレットエン
トリにおけるサンプルのビット深度を引いたものに等しくてもよい。 20

6 . 一例において、M は、パレットエントリにおけるサンプルのビット深度から
再構成されたサンプルのビット深度を引いたものに等しくてもよい。

v i . 一例において、M は、整数（例えば、2）であってもよい。

v i i . 代替的に、さらに、一例において、M は、に依存してもよい。

1 . 現在のブロックのブロック寸法

2 . 現在のブロックの量子化パラメータ

3 . カラーフォーマットの指示（例えば、4 : 2 : 0、4 : 4 : 4）

4 . 分離 / 二重符号化ツリー構造

5 . スライス / タイルグループのタイプおよび / またはピクチャのタイプ

6 . パレットエントリ

7 . 予測パレットエントリ

8 . ブロック / ピクチャ / スライス / タイルにおけるサンプル位置

9 . 色要素のインデックス

v i i i . 一例において、パレットエントリにおけるサンプルに基づくルックアップ
動作は、サンプルの再構成中に使用してもよい。

1 . 一例において、ルックアップテーブルにおける値は、LCU の SPS / VP
S / 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 / LCU 行 / グル
ープにおいて信号通知されてもよい。

2 . 一例において、ルックアップテーブルにおける値は、LCU の SPS / VP
S / 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 / LCU 行 / グル
ープにおいて推測されてもよい。 40

13 . パレット符号化ブロックの信号通知された / 導出された量子化パラメータ (QP)
は、クリップされるなど、エスケープ画素 / サンプルの導出に使用される前に、最初に
修正されてもよい。

a . 一例において、パレット符号化ブロックに適用される QP 範囲は、変換スキップ
モードおよび / または BDCM モードと同様に扱われてもよい。

b . 一例において、パレット符号化ブロックに適用される QP は、 $\max(Qp, 4$
 $+ T)$ に修正されてもよく、この場合、T は、整数値であり、Qp は、このブロックのため
の信号通知されたまたは導出された量子化パラメータである。 50

i . 一例において、Tは、予め規定された閾値であってもよい。

i i . 一例において、Tは、(4 + min_qp_prime_ts_minus_4)に等しくてもよく、min_qp_prime_ts_minus_4が信号通知されてもよい。

14 . エスケープサンプル／シンボルをどのように符号化するかは、量子交換バイパスが有効化されるかどうかにかかわらず、統一されてもよい。

a . 一例において、エスケープサンプルは、固定長で信号通知されてもよい。

b . 一例において、エスケープサンプルは、Nビットを使用して固定長で信号通知されてもよい。

i . 一例において、Nは、整数（例えば、8または10）であってもよく、下記に依存してもよい。 10

1 . LCUのSPS / VPS / 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 / LCU行 / グループにおいて信号通知されるメッセージ

2 . 内部ビット深度

3 . 入力ビット深度

4 . 現在のブロックのブロック寸法

5 . 現在のブロックの量子化パラメータ

6 . カラーフォーマットの指示（例えば、4：2：0、4：4：4）

7 . 分離 / 二重符号化ツリー構造

8 . スライス / タイルグループのタイプおよび / またはピクチャのタイプ 20

c . 一例において、1つのエスケープ画素 / サンプルを信号通知するための符号長は、内部ビット深度に依存してもよい。

i . 代替的に、1つのエスケープ画素 / サンプルを信号通知するためのコード長は、入力ビット深度に依存してもよい。

d . 一例において、1つのエスケープ画素 / サンプルを信号通知するための符号長は、量子化パラメータに依存してもよい。

i . 一例において、1つのエスケープ画素 / サンプルを信号通知するためのコード長は、f (Qp) でもよい。

1 . 一例において、関数 f は、(内部ビット深度 - (Qp - 4) / 6)として定義されてもよい。 30

15 . パレット符号化ブロックおよび非パレット符号化ブロックのための量子化および / または逆量子化処理は、異なる方法で定義されてもよい。

a . 一例において、変換係数または残差のための量子化処理を使用する代わりに、右ビットシフトを使用して、エスケープサンプルを量子化してもよい。

b . 一例において、変換係数または残差のための逆量子化処理を使用する代わりに、左ビットシフトを使用して、エスケープサンプルを逆量子化してもよい。

c . エンコーダ側において、以下を適用してもよい。

i . 一例において、エスケープ画素 / サンプル値は、f (p, Qp)として信号通知されてもよく、この場合、pは、画素 / サンプル値である。

i i . 一例において、関数 f は、 $p > > ((Qp - 4) / 6)$ として定義されてもよく、この場合、pは、画素 / サンプル値であり、Qpは、量子化パラメータである。 40

i i i . 一例において、エスケープ画素 / サンプル値は、 $p > > N$ として信号通知されてもよく、この場合、pは、画素 / サンプル値である。

1 . 一例において、Nは、整数（例えば、2）であってもよく、以下に依存してもよい。

a) LCUのSPS / VPS / 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 / LCU行 / グループにおいて信号通知されるメッセージ

b) 内部ビット深度

c) 入力ビット深度

d) 現在のブロックのブロック寸法 50

- e) 現在のブロックの量子化パラメータ
- f) カラーフォーマットの指示（例えば、4 : 2 : 0、4 : 4 : 4）
- g) 分離 / 二重符号化ツリー構造
- h) スライス / タイルグループのタイプおよび / またはピクチャのタイプ
- d . デコーダ側において、以下を適用してもよい。
 - i . 一例において、エスケープ画素 / サンプル値は、 $f(b_d, p, Q_p)$ として信号通知されてもよい。

1 . 一例において、関数 f は、 $c1ip(0, (1 << (b_d - (Q_p - 4)) / 6) - 1, (p + (1 << (b_d - 1))) >> ((Q_p - 4) / 6))$ として定義されてもよい。 10

i i . 一例において、エスケープ画素 / サンプル値は、 $f(p, Q_p)$ として再構成されてもよく、この場合、 p は、復号化されたエスケープ画素 / サンプル値である。

1 . 一例において、 f は、 $p < ((Q_p - 4) / 6)$ として定義されてもよい。

i i i . 一例において、エスケープ画素 / サンプル値は、 $f(b_d, p, Q_p)$ として再構成されてもよく、この場合、 p は、復号化されたエスケープ画素 / サンプル値である。 20

1 . 一例において、関数 $c1ip$ は、 $c1ip(0, (1 << b_d) - 1, p < ((Q_p - 4) / 6))$ として定義されてもよい。

i v . 上記の例において、クリップ関数 $c1ip(a, i, b)$ は、 $(i < a ? a : (i > b ? b : i))$ として定義されてもよい。 20

v . 上記の例において、クリップ関数 $c1ip(a, i, b)$ は、 $(i <= a ? a : (i >= b ? b : i))$ として定義されてもよい。

v i . 上記例において、 p は画素 / サンプル値であり、 b_d は内部ビット深度または入力ビット深度であり、 Q_p は量子化パラメータである。

16 . パレット符号化ブロックは、最大確率モード (MPM) のリスト構築処理中、1つのイントラブロック（例えば、MODE_INTRA）として扱われてもよい。 30

a . 一例において、MPMリストの構築中に近傍（隣接または非隣接）ブロックのイントラモードをフェッチするとき、近傍のブロック（例えば、左および / または上）がパレットモードで符号化されている場合、それをデフォルトモードの従来のイントラ符号化されたブロック（例えば、MODE_INTRA）として扱ってもよい。 30

i . 一例において、デフォルトモードは、DC / PLANAR / VERT / HORT モードであってもよい。

i i . 一例において、デフォルトモードは、1つのイントラ予測モードであってもよい。

i i i . 一例において、デフォルトモードは、DPS / SPS / VPS / PPS / APS / ピクチャヘッダ / スライスヘッダ / タイルグループヘッダ / 最大符号化ユニット (LCU) / 符号化ユニット (CU) / LCU行 / LCUグループ / TU / PUブロック / 映像符号化ユニットにおいて信号通知されてもよい。

17 . パレット符号化ブロックは、最大確率モード (MPM) のリスト構築処理中、非イントラブロックとして扱われてもよい（例えば、MODE_PLT に等しい予測モードを有するブロックとして扱われてもよい）。 40

a . 一例において、MPMリストの構築中に近傍のブロックのイントラモードをフェッチするとき、1つの近傍のブロック（例えば、左および / または上）がパレットモードで符号化されている場合、それは、インターモードで符号化されたものと同様にまたは類似した方法で取り扱われてもよい。

b . 一例において、MPMリストの構築中に近傍のブロックのイントラモードをフェッチするとき、1つの近傍のブロック（例えば、左および / または上）がパレットモードで符号化されている場合、それをIBCモードで符号化されたものと同様にまたは類似した方法で取り扱われてもよい。

18 . DMモードで符号化されたクロマブロックに対応するパレットモードで符号化さ 50

れた輝度ブロックは、デフォルトのイントラ予測モードを有すると解釈されてもよい。

a . 一例において、パレットモードで符号化された対応する輝度ブロックは、クロマブロックが DM モードで符号化される場合、イントラブロック（例えば、 MODE_INTRA ）またはパレットブロック（例えば、 MODE_PLT ）として扱われてもよい。

b . 一例において、デフォルト予測モードは、 DC / PLANAR / VER / HOR モードであってもよい。

c . 一例において、デフォルト予測モードは、1つのイントラ予測モードであってもよい。

d . 一例において、デフォルト予測モードは、 DPS / SPS / VPS / PPS / APS / ピクチャヘッダ / スライスヘッダ / タイルグループヘッダ / 最大符号化ユニット (LCU) / 符号化ユニット (CU) / LCU 行 / LCU グループ / TU / PU ブロック / 映像符号化ユニットにおいて信号通知されてもよい。 10

19 . パレット符号化ブロックは、履歴に基づく動きベクトル予測 (HMVP) 、マージ (ERGE) および / または高度動きベクトル予測 (AMVP) モードのリスト構築中、利用不可能なブロックとして扱われてもよい。

a . 一例において、利用不可能なブロックは、動き情報を有さないブロックを意味してもよく、またはその動き情報を他のブロックの予測として使用することができない。

b . 一例において、パレットモードで符号化されたブロックは、 HMVP 、 MERGE および / または AMVP モードにおけるリスト構築の過程において、イントラブロック（例えば、 MODE_INTRA ）またはパレットブロック（例えば、 MODE_PLT ）として扱われてもよい。 20

i . 代替的に、一例において、 HMVP 、 MERGE および / または AMVP リストの構築中に近傍のブロックの動き情報をフェッチする場合、パレットモードで符号化された近傍のブロックを無効な参照インデックスを有するブロックとして扱われてもよい。

ii . 代替的に、一例において、 HMVP 、 MERGE および / または AMVP リストの構築中に近傍のブロックの動き情報をフェッチする場合、パレットモードで符号化された近傍のブロックを、参照インデックスが 0 であるインターブロックとして扱われてもよい。

iii . 代替的に、一例において、 HMVP 、 MERGE および / または AMVP モードのリスト構築中に、近傍のブロックの動き情報をフェッチする場合、パレットモードで符号化された近傍のブロックを、動きベクトルがゼロのインターブロックとして扱ってもよい。 30

20 . パレットモードで符号化されたブロックをどのように処理するか（例えば、上述した方法を適用するか否か、および / またはどのように適用するか）は、以下に基づいてもよい。

a . 映像コンテンツ（例えば、スクリーンコンテンツまたは自然コンテンツ）
b . 一例において、デフォルトモードは、 DPS / SPS / VPS / PPS / APS / ピクチャヘッダ / スライスヘッダ / タイルグループヘッダ / 最大符号化ユニット (LCU) / 符号化ユニット (CU) / LCU 行 / LCU グループ / TU / PU ブロック / 映像符号化ユニットにおいて信号通知されてもよい。 40

- c . CU / PU / TU / ブロック / 映像符号化ユニットの位置
- d . 現在のブロックおよび / またはその近傍のブロックのブロック寸法
- e . 現在のブロックおよび / またはその近傍のブロックのブロック形状
- f . カラーフォーマットの指示（例えば、 4 : 2 : 0 、 4 : 4 : 4 、 RGB 、 YUV 等）
- g . 符号化木構造（例えば、デュアルツリーまたはシングルツリー）
- h . スライス / タイルグループのタイプおよび / またはピクチャのタイプ
- i . 色要素（例えば、輝度要素および / またはクロマ要素にのみ適用されてもよい）。
- j . 時間層 ID
- k . 標準のプロファイル / レベル / 層

10

20

30

40

50

21. パレット符号化プロックのためのコンテキスト符号化ビンは、ある範囲内に含まれるよう¹⁰に制限されてもよい。

a. 一例において、カウンタは、コンテキスト符号化されたビンの数を記録するよう¹⁰に、1つのプロックに割り当てられる。カウンタが閾値を超える場合、コンテキスト符号化を使用する代わりにバイパス符号化が適用される。

i. 代替的に、NumColorComp カウンタを割り当てて、各色要素に対してコンテキスト符号化されたビンの数を記録してもよい。NumColorComp は、1つのプロックに符号化される色要素の数である（例えば、YUV フォーマットの1つの CU の場合、NumColorComp は3に設定される）。

ii. 代替的に、カウンタをゼロに初期化してもよく、コンテキストで1つのビン¹⁰を符号化した後、カウンタを1だけ増加させる。

b. 代替的に、カウンタは、ゼロよりも大きいある値（例えば、W * H * K）で初期化されてもよく、コンテキストで1つのビンを符号化した後、カウンタは、1だけ減算される。カウンタがT以下である場合、コンテキスト符号化を使用する代わりにバイパス符号化が適用される。

i. 一例において、Tは、0または1に設定される。

ii. 一例において、Tは、復号化された情報または符号化パスの数等に従って設定される。

c. 一例において、パレット符号化プロックは、コンテキスト符号化ビンの観点から²⁰、TS 符号化プロックまたは非 TS 符号化プロックと比較して、同じまたは異なる閾値を有してもよい。

i. 一例において、パレット符号化プロックのためのコンテキスト符号化ビンの数は、(W * H * T) に設定されてもよく、この場合、W および H は、それぞれ1つのプロックの幅および高さであり、T は整数である。一例において、T は、TS 符号化プロックに使用されるものと同じであるように設定され、例えば、1.75 または 2 である。

ii. 一例において、パレット符号化プロックのコンテキスト符号化ビンの数は、(W * H * NumColorComp * T) に設定されてもよく、この場合、W および H は、それぞれ1つのプロックの幅および高さであり、NumColorComp は、1つのプロックでコード化される色要素の数（例えば、YUV フォーマットの1つの CU あり、NumColorComp は3に設定される）であり、T は、整数である。一例において、T は、TS 符号化プロックに使用されるものと同じであるように設定され、例えば³⁰、1.75 または 2 である。

d. 一例において、パレット符号化プロックの閾値は、コンテキスト符号化ビンの観点から、TS 符号化プロックまたは非 TS 符号化プロックよりも小さくてもよい。

e. 一例において、パレット符号化プロックの閾値は、コンテキスト符号化ビン (context coded bin) の観点から、TS 符号化プロックまたは非 TS 符号化プロックよりも大きくてもよい。

22. パレット符号化プロックは、CIP モードにおいて近傍のイントラプロックを統計する過程において、非イントラプロックとして扱われる（例えば、MODE_PLT に等しい予測モードを有するプロックとして扱われる）。

a. 一例において、CIP モードにおいて近傍のイントラプロックを数える間に近傍のプロックのイントラモードをフェッチするとき、1つの近傍のプロック（例えば、左および / または上）がパレットモードで符号化されている場合、それはインターモードで符号化されたものと同様にまたは類似した方法で取り扱われてもよい。

b. 一例において、CIP モードにおいて近傍のイントラプロックを数える間に近傍のプロックのイントラモードをフェッチするとき、1つの近傍のプロック（例えば、左および / または上）がパレットモードで符号化されている場合、それはIBC モードで符号化されたものと同様にまたは類似した方法で取り扱われてもよい。

c. 代替的に、パレット符号化プロックは、CIP モードにおいて近傍のイントラプロックを統計する過程において、イントラプロックとして扱われてもよい。

10

20

30

40

50

23. パレット符号化サンプルのための前置および／または後置フィルタリング処理をスキップすることが提案される。

- a. 一例において、パレット符号化サンプルはデブロックされなくてもよい。
- b. 一例において、パレット符号化サンプルは、S A O 処理においてオフセットを補償されなくてもよい。
- c. 一例において、パレット符号化サンプルは、A L F 処理においてフィルタリングされなくてもよい。
 - i. 一例において、A L F 処理における分類は、パレット符号化サンプルをスキップしてもよい。
- d. 一例において、L M C S は、パレット符号化サンプルのために無効化されてもよい。

24. パレットモードにおいて、より多くの走査順序を追加することが提案される。

- a. 一例において、次のように定義される逆水平横断走査順序を使用してもよい。
 - i. 一例において、奇数行の走査方向は、左から右であってもよい。
 - ii. 一例において、偶数行の走査方向は、右から左であってもよい。
 - iii. 一例において、 4×4 ブロックの走査順序は、図22に示すとおりであつてもよい。
- b. 一例において、以下のように定義される逆垂直横断走査順序を使用してもよい。
 - i. 一例において、奇数行の走査方向は、上から下へ延びてもよい。
 - ii. 一例において、偶数行の走査方向は、下から上であってもよい。
 - iii. 一例において、 4×4 ブロックの走査順序は、図23に示すとおりであつてもよい。

25. 許容される走査順序の組み合わせは、ブロックの形状に依存してもよい。

- a. 一例において、1つのブロックの幅と高さとの間の比が閾値よりも大きい場合、水平トラバース走査順序および逆水平横断走査順序のみを適用してもよい。
 - i. 一例において、閾値は1に等しい。
 - ii. 一例において、閾値は4に等しい。
- b. 一例において、1つのブロックの高さと幅との間の比が閾値よりも大きい場合、垂直横断および逆垂直横断走査順序のみを適用してもよい。
 - i. 一例において、閾値は1に等しい。
 - ii. 一例において、閾値は4に等しい。

26. Q R - B D P C M 処理において、1つのイントラ予測方向および／または1つの走査方向のみを許可することが提案される。

- a. 一例において、高さより大きい幅を有するブロックでは、垂直方向のみが許容される。
- b. 一例において、幅が高さより小さいブロックでは、水平方向のみが許容される。
- c. 一例において、Q R - B D P C M の方向の指示は、非正方形ブロックについて推測してもよい。
 - i. 一例において、さらに、Q R - B D P C M の方向の指示は、幅が高さよりも大きいブロックの場合、垂直方向に推測されてもよい。
 - ii. 一例において、さらに、Q R - B D P C M の方向の指示は、幅が高さよりも小さいブロックの場合、水平方向に推測されてもよい。

27. ブレット24、25、および26における方法は、 $w * T h$ h または $h * T h$ w を有するブロックにのみ適用されてもよく、この場合、 w および h は、それぞれブロックの幅および高さであり、 $T h$ は閾値である。

- a. 一例において、 $T h$ は整数（例えば、4または8）であり、以下に基づいてよい。
 - i. 映像コンテンツ（例えば、スクリーンコンテンツまたは自然コンテンツ）
 - ii. 一例において、デフォルトモードは、D P S / S P S / V P S / P P S / A P S / ピクチャヘッダ / スライスヘッダ / タイルグループヘッダ / 最大符号化ユニット（

10

20

30

40

50

L C U) / 符号化ユニット (C U) / L C U 行 / L C U グループ / T U / P U ブロック / 映像符号化ユニットにおいて信号通知されてもよい。

i i i . C U / P U / T U / ブロック / 映像符号化ユニットの位置
 i v . 現在のブロックおよび / またはその近傍のブロックのブロック寸法
 v . 現在のブロックおよび / またはその近傍のブロックのブロック形状
 v i . カラーフォーマットの指示 (例えば、4 : 2 : 0 、 4 : 4 : 4 、 R G B 、 Y
 U V 等)
 v i i . 符号化木構造 (例えば、デュアルツリーまたはシングルツリー)
 v i i i . スライス / タイルグループのタイプおよび / またはピクチャのタイプ
 i x . 色要素 (例えば、輝度要素および / またはクロマ要素にのみ適用されてもよ
 い) 。

x . 時間層 I D
 x i . 標準のプロファイル / レベル / 層
【 0 2 1 6 】
 5 . 追加の実施形態
【 0 2 1 7 】
 5 . 1 実施形態 1
【 0 2 1 8 】

この章は、映像のビットストリーム表現がベースラインビットストリーム構文と比較して変更され得る例示的な実施形態を示す。この変更は、太字のイタリック体のテキスト入力項目を使用して強調表示されている。

【 0 2 1 9 】

【表 8】

seq_parameter_set_rbsp() {	ディスクリプタ
sps_max_sub_layers_minus1	u(3)
...	
sps_gbi_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_plt_enabled_flag	u(1)
...	
}	

【 0 2 2 0 】

s p s _ p l t _ e n a b l e d _ f l a g = 1 は、 C V S におけるピクチャの復号化にパレットモードを使用してもよいことを規定する。 s p s _ p l t _ e n a b l e d _ f l a g = 0 は、パレットモードが C V S において使用されないことを規定する。 s p s _ p l t _ e n a b l e d _ f l a g が存在しない場合、 0 に等しいと推測される。

【 0 2 2 1 】

10

20

30

40

50

【表 9】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリプタ
if(tile_group_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] == 0)	
(tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA))	
&& (sps_ibc_enabled_flag sps_plt_enabled_flag)	
pred_mode_scc_flag	ae(v)
if(scc_mode_flag) {	
if(sps_ibc_enabled_flag) {	ae(v)
pred_mode_ibc_flag	
}	
if(sps_plt_enabled_flag) {	
pred_mode_plt_flag	ae(v)
}	
}	
}	
...	
}	

10

20

30

40

【0222】

1に等しい pred_mode_scc_flag は、現在の符号化ユニットがスクリーンコンテンツ符号化モードで符号化されることを規定する。 pred_mode_scc_flag が 0 に等しいことは、現在の符号化ユニットがスクリーンコンテンツ符号化モードで符号化されていないことを規定する。

pred_mode_scc_flag が存在しない場合、それは 0 に等しいと推測される。

pred_mode_plt_flag = 1 は、現在の符号化ユニットがパレットモードで符号化されていることを規定する。 pred_mode_plt_flag = 0 は、現在の符号化ユニットがパレットモードで符号化されていないことを規定する。

pred_mode_plt_flag が存在しない場合、I タイルグループをデコードするとき、sps_plt_enabled_flag の値に等しいと推測され、P または B タイルグループを復号化するとき、それぞれ 0 であると推測される。

pred_mode_scc_flag が 1 と等しく、sps_ibc_enabled_flag が 0 と等しいとき、pred_mode_plt_flag は 1 であると推測される。

pred_mode_ibc_flag = 1 のとき、変数 CuPredMode[x][y] は、 $x = x_0 \dots x_0 + cbWidth - 1$ および $y = y_0 \dots y_0 + cbHeight - 1$ に対して MODE_PLT と等しくなるように設定される。

【0223】

50

【表 1 0】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリプタ
if(tile_group_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] ==0) (tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && (sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] ==0) (tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && sps_plt_enabled_flag && CuPredMode[x0][y0] != MODE_IBC	
pred_mode_plt_flag	ae(v)
}	
...	
}	

10

20

【0 2 2 4】

【表 1 1】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリプタ
if(tile_group_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] ==0) (tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && (sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA && sps_plt_enabled_flag)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

30

40

【0 2 2 5】

`pred_mode_plt_flag = 1` は、現在の符号化ユニットがパレットモードで符号化されていることを規定する。`pred_mode_plt_flag = 0` は、現在の符号化ユニットがパレットモードで符号化されていないことを規定する。

`pred_mode_plt_flag` が存在しない場合、I タイルグループをデコー

50

ドするとき、`sps_plt_enabled_flag` の値に等しいと推測され、P または B タイルグループを復号化するとき、それぞれ 0 であると推測される。

`pred_mode_ibc_flag = 1` のとき、変数 `CuPredMode[x][y]` は、 $x = x_0 \dots x_0 + cbWidth - 1$ および $y = y_0 \dots y_0 + cbHeight - 1$ に対して `MODE_PLT` と等しくなるように設定される。

【0226】

【表12】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリプタ
if(tile_group_type != I sps_ibc_enabled_flag	
sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ae(v)
if((tile_group_type == I && cu_skip_flag[x0][y0] == 0)	
//	
(tile_group_type != I && CuPredMode[x0][y0] ==	
MODE_INTRA)) &&	
(sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] == 0) //	
(tile_group_type != I && CuPredMode[x0][y0] == MODE_INTRA	
&& (sps_ibc_enabled_flag?CuPredMode[x0][y0] != MODE_IBC :TRUE)))	
&&	
sps_plt_enabled_flag	
)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

【0227】

`pred_mode_plt_flag = 1` は、現在の符号化ユニットがパレットモードで符号化されていることを規定する。`pred_mode_plt_flag = 0` は、現在の符号化ユニットがパレットモードで符号化されていないことを規定する。

`pred_mode_plt_flag` が存在しない場合、I タイルグループをデコードするとき、`sps_plt_enabled_flag` の値に等しいと推測され、P または B タイルグループを復号化するとき、それぞれ 0 であると推測される。

`pred_mode_ibc_flag = 1` のとき、変数 `CuPredMode[x][y]` は、 $x = x_0 \dots x_0 + cbWidth - 1$ および $y = y_0 \dots y_0 + cbHeight - 1$ に対して `MODE_PLT` と等しくなるように設定される。

【0228】

10

20

30

40

50

【表 1 3】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	ディスクリプタ
if(tile_group_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_modes(x0, y0, cbWidth, cbHeight)	
...	
}	

10

【0 2 2 9】

【表 1 4】

pred_modes (x0, y0, cbWidth, cbHeight) {	ディスクリプタ
if(tile_group_type == I) {	
if(sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_IBC) {	
if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)	
plt_mode_flag	ae(v)
}	
else {	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA (CuPredMode[x0][y0] != MODE_INTRA && ! sps_ibc_enabled_flag)) {	
if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)	
plt_mode_flag	ae(v)
}	
else {	
if(sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
}	
}	
}	

20

30

【0 2 3 0】

40

50

【表 1 5】

<i>pred_modes (x0, y0, cbWidth, cbHeight) {</i>	ディスクリ プタ
<i>if(tile_group_type == I) {</i>	
<i>if(sps_ibc_enabled_flag sps_plt_enabled_flag)</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] != MODE_INTRA) {</i>	
<i>if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] == MODE_INTRA (CuPredMode[x0][y0] !=</i>	
<i>MODE_INTRA && ! sps_ibc_enabled_flag)) {</i>	
<i>if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>if(sps_ibc_enabled_flag)</i>	
<i>pred_mode_ibc_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>}</i>	
<i>}</i>	

【0 2 3 1】

10

20

30

40

50

【表 1 6】

<i>pred_modes (x0, y0, cbWidth, cbHeight) {</i>	ディスクリ プタ
<i>if(tile_group_type == I) {</i>	
<i>if(sps_ibc_enabled_flag sps_plt_enabled_flag)</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] != MODE_INTRA) {</i>	
<i>if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] == MODE_INTRA (CuPredMode[x0][y0] !=</i>	
<i>MODE_INTRA && ! sps_ibc_enabled_flag)) {</i>	
<i>if(sps_plt_enabled_flag && cbWidth <=64 && cbHeight <= 64)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>if(sps_ibc_enabled_flag)</i>	
<i>pred_mode_ibc_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>}</i>	
<i>}</i>	

10

20

【0 2 3 2】

30

40

50

【表 1 7】

<i>pred_modes (x0, y0, cbWidth, cbHeight) {</i>	ディスクリ プタ
<i>if(tile_group_type == I) {</i>	
<i>if(sps_ibc_enabled_flag sps_plt_enabled_flag)</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] != MODE_INTRA) {</i>	
<i>if(sps_plt_enabled_flag)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>pred_mode_flag</i>	<i>ae(v)</i>
<i>if(CuPredMode[x0][y0] == MODE_INTRA (CuPredMode[x0][y0] !=</i>	
<i>MODE_INTRA && !sps_ibc_enabled_flag)) {</i>	
<i>if(sps_plt_enabled_flag)</i>	
<i>plt_mode_flag</i>	<i>ae(v)</i>
<i>}</i>	
<i>else {</i>	
<i>if(sps_ibc_enabled_flag)</i>	<i>ae(v)</i>
<i>pred_mode_ibc_flag</i>	
<i>}</i>	
<i>}</i>	
<i>}</i>	

10

20

【0 2 3 3】

plt_mode_flag = 1 は、現在の符号化ユニットがパレットモードで符号化されることを規定する。*intra_mode_plt_flag = 0* は、現在の符号化ユニットがパレットモードで符号化されることを規定する。

plt_mode_flag が存在しないとき、偽に等しいと推測される。

pred_mode_scc_flag = 1 のとき、変数 *CuPredMode[x][y]* は、*x = x0 . . x0 + cbWidth - 1* および *y = y0 . . y0 + cbHeight - 1* に対して *MODE_PLT* と等しくなるように設定される。

【0 2 3 4】

pred_mode_flag が 0 に等しい場合、現在の符号化ユニットがインター予測モードまたはIBC予測モードで符号化されることを規定する。*pred_mode_flag* が 1 に等しい場合、現在の符号化ユニットがINTL予測モードまたはPLTモードで符号化されることを規定する。変数 *CuPredMode[x][y]* は、*x = x0 . . x0 + cbWidth - 1 and y = y0 . . y0 + cbHeight - 1* に対しては以下のように導出される。

- *pred_mode_flag* が 0 に等しい場合、*CuPredMode[x][y]* は、*MODE_INTER* に等しく設定される。
- あるいは、(*pred_mode_flag* が 1 に等しい場合)、*CuPredMode[x][y]* を *MODE_INTRA* に等しく設定する。

pred_mode_flag が存在しない場合、Iタイルグループをデコードするとき、1 に等しいと推測され、P またはBタイルグループを復号化するとき、それぞれ 0 に等しいと推測される。

【0 2 3 5】

30

40

50

【表 1 8】

表 9-4 構文要素および関連する 2 値化

構文構造	構文要素	2 値化	
		処理	入力パラメータ
pred_modes()	PLT_mode_flag	FL	cMax=1

【0 2 3 6】

【表 1 9】

表 9-10 コンテキスト符号化ビンを有する構文要素への c t x I n c の割り当て

構文要素	binIdx					
	0	1	2	3	4	>= 5
PLT_mode_flag	0	na	na	na	na	na

【0 2 3 7】

5.2 実施形態 2

【0 2 3 8】

本実施形態では、`modeType`について説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

【0 2 3 9】

変数`modeType`は、符号化ツリーノード内の符号化ユニットに対して、イントラ、IBC、パレット、インターの各符号化モードを使用できるか(`MODE_TYPE_ALL`)、イントラ、パレット、IBCの各符号化モードのみを使用できるか(`MODE_TYPE_INTRA`)、インターの各符号化モードのみを使用できるか(`MODE_TYPE_INTER`)を規定する。

【0 2 4 0】

5.3 実施形態 # 3

【0 2 4 1】

本実施形態では、符号化ユニットの構文について説明する。本実施形態において、`pred_mode_plt_flag`は、`pred_mode_ibc_flag`の後に信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[[]]”の印が付けられる。

【0 2 4 2】

10

20

30

40

50

【表 2 0】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag // sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))[[==]]]	
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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

10

20

30

40

【0 2 4 3】

5.4 実施形態 # 4

【0 2 4 4】

本実施形態では、符号化ユニットの構文について説明する。本実施形態では、`pred_mode_ibc_flag` の後に `pred_mode_plt_flag` が信号通知され、現在の予測モードが `MODE_INTRA` の場合にのみ `pred_mode_plt_flag` が信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには “`[[]]`” の印が付けられる。

【0 2 4 5】

50

【表 2 1】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag])]	
((cbWidth == 4 && cbHeight == 4) // modeType ==	10
MODE_TYPE_INTRA)	
&& !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I	
&& !(cbWidth == 4 && cbHeight == 4) && modeType ==	
MODE_TYPE_ALL)	
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] ==	20
0)))) &&	
sps_ibc_enabled_flag && (cbWidth != 128 cbHeight !=	
128))]]	
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)))) &&	
cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	30
if(cu_skip_flag[x0][y0] == 0 && CuPredMode[x0][y0] ==	
MODE_INTRA &&	
cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag &&	
modeType !=	
MODE_TYPE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

10

20

30

40

【0 2 4 6】

5.5 実施形態 # 5

【0 2 4 7】

本実施形態では、符号化ユニットの構文について説明する。本実施形態では、`pred_mode_ibc_flag` は、`pred_mode_plt_flag` の後に信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには “`[[]]`” の印が付けられる。

【0 2 4 8】

50

【表 2 2】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスク リプタ
if(slice_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	10
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	20
if(CuPredMode[x0][y0] == MODE_INTRA (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_ TYPE_INTER) {	30
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
}	
...	
}	

10

20

30

40

【0 2 4 9】

5. 6 実施形態 # 6

【0 2 5 0】

本実施形態では、符号化ユニットの構文について説明する。本実施形態では、pred_mode_plt_flag の後に pred_mode_ibc_flag が信号通知され、現在の予測モードが MODE_INTRA の場合にのみ pred_mode_plt_flag が信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには “ [] ” の印が付けられる。

【0 2 5 1】

50

【表 2 3】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスク リプタ	
if(slice_type != I sps_ibc_enabled_flag // sps_plt_enabled_flag) {		
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))		10
cu_skip_flag[x0][y0]	ae(v)	
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)		
pred_mode_flag	ae(v)	
if cu_skip_flag[x0][y0] == 0 && (CuPredMode[x0][y0] == MODE_INTRA &&		
cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && modeType != MODE_TYPE_INTER)		20
pred_mode_plt_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))]]		
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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {		30
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)		
pred_mode_ibc_flag	ae(v)	
}		
}		
...		
}		

40

【0 2 5 2】

5. 7 実施形態 # 7

【0 2 5 3】

本実施形態では、符号化ユニットの構文について説明する。本実施形態では、予測モードが MODE_INTRA であるとき、 pred_mode_plt_flag および pred_mode_ibc_flag が信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには “ [[]] ” の印が付けられる。

【0 2 5 4】

50

【表 2 4】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスク リプタ
if(slice_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	10
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	20
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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	30
if(CuPredMode[x0][y0] == MODE_INTRA && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	40

【0 2 5 5】

5 . 8 実施形態 # 8

【0 2 5 6】

本実施形態では、符号化ユニットの構文について説明する。本実施形態において、pred_mode_plt_flag および pred_mode_ibc_flag は、予測モードが MODE_INTRA でない場合に信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには “[]” の印が付けられる。

【0 2 5 7】

【表 2 5】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスク リプタ
if(slice_type != I sps_ibc_enabled_flag / / sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	10
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	20
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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	30
if(CuPredMode[x0][y0] != MODE_INTRA && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

40

【0 2 5 8】

5. 9 実施形態 # 9

【0 2 5 9】

本実施形態では、符号化ユニットの構文について説明する。本実施形態において、予測モードが MODE_INTER であるとき、 pred_mode_plt_flag および pred_mode_ibc_flag が信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[[]]”の印が付けられる。

【0 2 6 0】

50

【表 2 6】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag // sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[x0][y0] == MODE_INTER (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTER && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

10

20

30

40

【0 2 6 1】

5. 1 0 実施形態 # 1 0

【0 2 6 2】

本実施形態では、`pred_mode_plt_flag` の意味を説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

`pred_mode_plt_flag` は、現在の符号化ユニットにおけるパレットモードの使用を規定する。`pred_mode_plt_flag == 1` は、現在の符号化ユニットにおいてパレットモードが適用されていることを示す。`pred_mode_plt_flag == 0` は、現在の符号化ユニットに対してパレットモードが適用されない

50

ことを示す。`pred_mode_p1t_flag` が存在しない場合、`pred_mode_p1t_flag` は 0 に等しいと推測される。

【0263】

5.11 実施形態 # 11

【0264】

本実施形態では、`pred_mode_p1t_flag` の意味を説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

`pred_mode_p1t_flag` は、現在の符号化ユニットにおけるパレットモードの使用を規定する。`pred_mode_p1t_flag == 1` は、現在の符号化ユニットにおいてパレットモードが適用されていることを示す。`pred_mode_p1t_flag == 0` は、現在の符号化ユニットに対してパレットモードが適用されないことを示す。`pred_mode_p1t_flag` が存在しない場合、`pred_mode_p1t_flag` は 0 に等しいと推測される。

`pred_mode_p1t_flag = 1` のとき、変数 `CuP redMode[x][y]` は、 $x = x_0 \dots x_0 + cbWidth - 1$ および $y = y_0 \dots y_0 + cbHeight - 1$ に対して `MODE_PLT` と等しくなるように設定される。

【0265】

5.12 実施形態 # 12

【0266】

本実施形態では、境界強度の導出について説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

【0267】

8.8.3.5 境界フィルタリング強度の導出処理

この処理への入力は以下の通りである。

1つのピクチャサンプルアレイ `recPicture`、

現在のピクチャの左上の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位置 (`xCb`, `yCb`)、

現在の符号化ブロックの幅を規定する変数 `nCbW`、

現在の符号化ブロックの高さを規定する変数 `nCbH`、

垂直 (`EDGE_VER`) エッジまたは水平 (`EDGE_HOR`) エッジのいずれをフィルタリングするかを規定する変数 `edgeType`、

現在の符号化ブロックの色要素を規定する変数 `cidx`、

2次元 (`nCbW`) \times (`nCbH`) 配列の `edgeFlags`。

この処理の出力は、境界フィルタリング強度を規定する 2 次元 (`nCbW`) \times (`nCbH`) アレイ `bS` である。

...

変数 `bS[xDi][yDj]` は、次のように導出される。

あるいは、サンプル `p0` および `q0` の両方が、`intra_bdpcom_flag` が 1 に等しい符号化ブロックに含まれている場合、`bS[xDi][yDj]` は、0 に等しく設定される。

あるいは、サンプル `p0` または `q0` が、イントラ予測モードで符号化された符号化ユニットの符号化ブロックにある場合には、`bS[xDi][yDj]` は、2 に等しく設定される。

あるいは、ブロックエッジも変換ブロックエッジであり、サンプル `p0` または `q0` が、`ciip_flag` が 1 に等しい符号化ブロックにある場合、`bS[xDi][yDj]` は、2 に等しく設定される。

あるいは、ブロックエッジが変換ブロックエッジであり、サンプル `p0` または `q0` が、1 つ以上の非ゼロ変換係数レベルを含む変換イントラブロックにある場合、`bS[xDi][yDj]` は、1 に等しく設定される。

あるいは、ブロックエッジも変換ブロックエッジであり、サンプル `p0` および `q0` が、

10

20

30

40

50

`p r e d _ m o d e _ p l t _ f l a g` が 1 に等しい 2 つの符号化ブロックに含まれている場合、`b S [x D i] [y D j]` は 0 に等しく設定される。

あるいは、サンプル p_0 を含む符号化サブブロックの予測モードが、サンプル q_0 を含む符号化サブブロックの予測モードと異なる場合、`b S [x D i] [y D j]` は、1 に等しく設定される。

あるいは、`c I d x` が 0 であり、且つ、以下の条件の 1 つ以上が真である場合、`b S [x D i] [y D j]` は、1 に等しく設定される。

サンプル p_0 を含む符号化サブブロックおよびサンプル q_0 を含む符号化サブブロックは、いずれも IBC 予測モードで符号化され、2 つの符号化サブブロックの予測に用いられる動きベクトルの水平または垂直要素の絶対差は、1 / 4 輝度サンプル単位で 4 以上である。10

サンプル p_0 を含む符号化サブブロックの予測のために、サンプル q_0 を含む符号化サブブロックの予測とは異なる参照ピクチャまたは異なる数の動きベクトルが使用される。

注 1 - 2 つの符号化サブブロックに使用される参照ピクチャが同じであるかまたは異なるかは、予測を形成するのに参照ピクチャリスト 0 へのインデックスを使用するか、または参照ピクチャリスト 1 へのインデックスを使用して形成するかに関わらず、且つ参照ピクチャリスト内のインデックス位置が異なるかどうかに関わらず、どのピクチャが参照されるかによってのみに基づいて判定される。

注 2 - (`x S b`, `y S b`) を含む左上のサンプルを有する符号化サブブロックの予測に使用される動きベクトルの数は、`P r e d F l a g L 0 [x S b] [y S b] + P r e d F l a g L 1 [x S b] [y S b]` に等しい。20

1 つの動きベクトルは、サンプル p_0 を含む符号化サブブロックを予測するために使用され、1 つの動きベクトルは、サンプル q_0 を含む符号化サブブロックを予測するために使用され、使用される動きベクトルの水平または垂直要素の絶対差は、1 / 4 輝度サンプル単位で 4 以上である。

2 つの動きベクトルおよび 2 つの異なる参照ピクチャを使用して、サンプル p_0 を含む符号化サブブロックを予測し、同じ 2 つの参照ピクチャの 2 つの動きベクトルを使用して、サンプル q_0 を含む符号化サブブロックを予測し、同じ参照ピクチャの 2 つの符号化サブブロックの予測に使用される 2 つの動きベクトルの水平または垂直要素の絶対差は、1 / 4 輝度サンプル単位で 4 以上である。30

同じ参照ピクチャの 2 つの動きベクトルを使用して、サンプル p_0 を含む符号化サブブロックを予測し、同じ参照ピクチャの 2 つの動きベクトルを使用して、サンプル q_0 を含む符号化サブブロックを予測し、以下の条件の両方が成り立つ。

2 つの符号化サブブロックの予測に使用されるリスト 0 の動きベクトルの水平または垂直要素の間の絶対差は、1 / 4 輝度サンプルにおいて 4 以上である、または 2 つの符号化サブブロックの予測に使用されるリスト 1 の動きベクトルの水平または垂直要素の間の絶対差は、4 分の 1 輝度サンプル単位で 4 以上である。

サンプル p_0 を含む符号化サブブロックの予測に使用されるリスト 0 動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブブロックの予測に使用されるリスト 1 動きベクトルとの間の絶対差は、1 / 4 輝度サンプル単位で 4 以上であるか、またはサンプル p_0 を含む符号化サブブロックの予測に使用されるリスト 1 動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブブロックの予測に使用されるリスト 0 動きベクトルとの間の絶対差は、1 / 4 載度サンプル単位で 4 以上である。40

あるいは、変数 `b S [x D i] [y D j]` を 0 に設定する。

【0 2 6 8】

5 . 1 3 a 実施形態 # 1 3 a

【0 2 6 9】

本実施形態では、境界強度の導出について説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

【0 2 7 0】

8 . 8 . 3 . 5 境界フィルタリング強度の導出処理

この処理への入力は以下の通りである。

1つのピクチャサンプルアレイ `recPicture`、

現在のピクチャの左上の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位置 (`xCb`, `yCb`)

現在の符号化ブロックの幅を規定する変数 `nCbW`、

現在の符号化ブロックの高さを規定する変数 `nCbH`、

垂直 (`EDGE_VER`) エッジまたは水平 (`EDGE_HOR`) エッジのいずれをフィルタリングするかを規定する変数 `edgeType`、

- 現在の符号化ブロックの色要素を規定する変数 `cidx`、

2次元 (`nCbW`) × (`nCbH`) 配列の `edgeFlags`。

この処理の出力は、境界フィルタリング強度を規定する2次元 (`nCbW`) × (`nCbH`) アレイ `bS` である。

..

変数 `bS[xDi][yDj]` は、次のように導出される。

`cidx` が 0 に等しく、サンプル `p0` および `q0` の両方が、`intra_bdpcom_flag` が 1 に等しい符号化ブロックに含まれている場合、`bS[xDi][yDj]` は 0 に等しく設定される。

あるいは、サンプル `p0` または `q0` が、イントラ予測モードで符号化された符号化ユニットの符号化ブロックにある場合には、`bS[xDi][yDj]` は、2 に等しく設定される。

あるいは、ブロックエッジも変換ブロックエッジであり、サンプル `p0` または `q0` が、`ciip_flag` が 1 に等しい符号化ブロックにある場合、`bS[xDi][yDj]` は、2 に等しく設定される。

あるいは、ブロックエッジが変換ブロックエッジでもあり、サンプル `p0` または `q0` が、1 つ以上の非ゼロ変換係数レベルを含む変換イントラブロックにある場合、`bS[xDi][yDj]` は、1 に等しく設定される。

あるいは、ブロックエッジも変換ブロックエッジであり、サンプル `p0` または `q0` が、`pred_mode_pct_flag` が 1 に等しい符号化ブロックに含まれている場合には、`bS[xDi][yDj]` は、0 に等しく設定される。

あるいは、サンプル `p0` を含む符号化サブブロックの予測モードが、サンプル `q0` を含む符号化サブブロックの予測モードと異なる場合、`bS[xDi][yDj]` は、1 に等しく設定される。

あるいは、`cidx` が 0 であり、且つ、以下の条件の 1 つ以上が真である場合、`bS[xDi][yDj]` は、1 に等しく設定される。

サンプル `p0` を含む符号化サブブロックおよびサンプル `q0` を含む符号化サブブロックは、いずれも IBC 予測モードで符号化され、2 つの符号化サブブロックの予測に用いられる動きベクトルの水平または垂直要素の絶対差は、1 / 4 輝度サンプル単位で 4 以上である。

サンプル `p0` を含む符号化サブブロックの予測のために、サンプル `q0` を含む符号化サブブロックの予測とは異なる参照ピクチャまたは異なる数の動きベクトルが使用される。

注 1 - 2 つの符号化サブブロックに使用される参照ピクチャが同じであるかまたは異なるかは、予測を形成するのに参照ピクチャリスト 0 へのインデックスを使用するか、または参照ピクチャリスト 1 へのインデックスを使用して形成するかに関わらず、且つ参照ピクチャリスト内のインデックス位置が異なるかどうかに関わらず、どのピクチャが参照されるかによってのみに基づいて判定される。

注 2 - (`xSb`, `ySb`) を含む左上のサンプルを有する符号化サブブロックの予測に使用される動きベクトルの数は、`PredFlagL0[xSb][ySb] + PredFlagL1[xSb][ySb]` に等しい。

1 つの動きベクトルは、サンプル `p0` を含む符号化サブブロックを予測するために使用

10

20

30

40

50

され、1つの動きベクトルは、サンプル q_0 を含む符号化サブプロックを予測するために使用され、使用される動きベクトルの水平または垂直要素の絶対差は、1 / 4 輝度サンプル単位で4以上である。

2つの動きベクトルおよび2つの異なる参照ピクチャを使用して、サンプル p_0 を含む符号化サブプロックを予測し、同じ2つの参照ピクチャの2つの動きベクトルを使用して、サンプル q_0 を含む符号化サブプロックを予測し、同じ参照ピクチャの2つの符号化サブプロックの予測に使用される2つの動きベクトルの水平または垂直要素の絶対差は、1 / 4 載度サンプル単位で4以上である。

同じ参照ピクチャの2つの動きベクトルを使用して、サンプル p_0 を含む符号化サブプロックを予測し、同じ参照ピクチャの2つの動きベクトルを使用して、サンプル q_0 を含む符号化サブプロックを予測し、以下の条件の両方が成り立つ。10

2つの符号化サブプロックの予測に使用されるリスト0の動きベクトルの水平または垂直要素の間の絶対差は、1 / 4 載度サンプルにおいて4以上である、または2つの符号化サブプロックの予測に使用されるリスト1の動きベクトルの水平または垂直要素の間の絶対差は、4分の1輝度サンプル単位で4以上である。

サンプル p_0 を含む符号化サブプロックの予測に使用されるリスト0動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブプロックの予測に使用されるリスト1動きベクトルとの間の絶対差は、1 / 4 載度サンプル単位で4以上であるか、またはサンプル p_0 を含む符号化サブプロックの予測に使用されるリスト1動きベクトルの水平または垂直要素と、サンプル q_0 を含む符号化サブプロックの予測に使用されるリスト0動きベクトルとの間の絶対差は、1 / 4 載度サンプル単位で4以上である。20

あるいは、変数 $bS[xDi][yDj]$ を0に設定する。

【0 2 7 1】

5.13 b 実施形態 # 1 3 b

【0 2 7 2】

本実施形態では、エスケープサンプルの符号化および再構成について説明する。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには「[[]]」の印が付けられる。20

【0 2 7 3】

【表27】

<code>palette_coding(x0, y0, cbWidth, cbHeight, startComp, numComps) {</code>	ディスクリプタ
<code>...</code>	
<code>/* Parsing escape values */</code>	
<code>if(palette_escape_val_present_flag) {</code>	
<code> for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)</code>	
<code> for(sPos = 0; sPos < cbWidth* cbHeight; sPos++) {</code>	
<code> xC = TraverseScanOrder[cbWidth][cbHeight][sPos][0]</code>	
<code> yC = TraverseScanOrder[cbWidth][cbHeight][sPos][1]</code>	
<code> if(PaletteIndexMap[cIdx][xC][yC] == (MaxPaletteIndex -</code>	
<code>1)) {</code>	
<code> palette_escape_val</code>	<code>[[ae(v)]] u(v)</code>
<code> PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val</code>	
<code> }</code>	
<code> }</code>	
<code> }</code>	
<code>}</code>	

【0 2 7 4】

10

20

30

40

50

パレットモードの復号化処理

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ブロックの左上のサンプルを規定する位置 (x_{Cb} , y_{Cb})、

変数 $startComp$ は、パレットテーブルの最初の色要素を規定し、

現在のブロックの色要素を規定する変数 $cidx$ 、

現在のブロックの幅および高さをそれぞれ規定する 2 つの変数 $nCbW$ および $nCbH$ 。

この処理の出力は、配列 $recSamples[x][y]$ であり、 $x = 0 \dots nCbW - 1$, $y = 0 \dots nCbH - 1$ は、ブロックの再構成サンプル値を規定する。

$cidx$ の値に基づいて、変数 $nSubWidth$ および $nSubHeight$ は、以下のように導出される。 10

$cidx$ が 0 に等しい場合、 $nSubWidth$ は 1 に設定され、 $nSubHeight$ は 1 に設定される。

あるいは、 $nSubWidth$ を $SubWidthC$ に設定し、 $nSubHeight$ を $SubHeightC$ に設定する。

位置 (x_{Cb} , y_{Cb}) にある再構成サンプル配列 $recSamples$ の ($nCbW \times nCbH$) ブロックは、 $x = 0 \dots nCbW - 1$ および $y = 0 \dots nCbH - 1$ である $recSamples[x][y]$ で表され、 $0 \sim nCbW - 1$ の範囲の各 x と $0 \sim nCbH - 1$ の範囲の各 y に対する $recSamples[x][y]$ の値は、以下のように導出される。 20

変数 x_L および y_L は、以下のように導出される。

```
x_L = palette_transpose_flag ? x * nSubHeight : x * nSubWidth(8-69)
y_L = palette_transpose_flag ? y * nSubWidth : y * nSubHeight(8-70)
```

変数 $bIsEscapeSample$ は、以下のように導出される。

$PaletteIndexMap[x_{Cb} + x_L][y_{Cb} + y_L]$ が $MaxPaletteIndex$ に等しく、 $palette_escape_val_present_flag$ が 1 に等しい場合、 $bIsEscapeSample$ は 1 に設定される。

あるいは、 $bIsEscapeSample$ を 0 に設定する。 30

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

```
recSamples[x][y] = CurrentPaletteEntries[cIdx][PaletteIndexMap[x_{Cb} + x_L][y_{Cb} + y_L]](8-71)
```

あるいは、 $cu_transquant_bypass_flag$ が 1 に等しい場合には、以下が適用される。

```
recSamples[x][y] = PaletteEscapeVal[cIdx][x_{Cb} + x_L][y_{Cb} + y_L](8-72)
```

あるいは、($bIsEscapeSample$ が 1 に等しく、 $cu_transquant_bypass_flag$ が 0 に等しい場合)、以下の順序付けられたステップが適用される。 40

第 8.7.1 項で規定される量子化パラメータの導出処理は、現在のピクチャの左上サンプルに対する現在のブロックの左上サンプルを規定する位置 (x_{Cb} , y_{Cb}) を使用して呼び出される。

量子化パラメータ qP は、以下のように導出される。

$cidx$ が 0 に等しい場合

$qP = Max(0, Qp'Y)(8-73)$

あるいは、 $cidx$ が 1 に等しい場合、

$qP = Max(0, Qp'Cb)(8-74)$

あるいは、($cidx$ が 2 に等しい場合)、

10

20

30

40

50

```

qP = Max(0, Qp'Cr) (8-75)
変数bitDepthは、以下のように導出される。
bitDepth = (cIdx == 0) ? BitDepthY : BitDepthC
(8-76)
[[The list levelScale[]は、k=0..5.]]でlevel
Scale[k] = {40, 45, 51, 57, 64, 72}、として規定される。

```

以下が適用される

```

[[tmpVal = (PaletteEscapeVal[cIdx][xCb+XL]
[yCb+YL]*levelScale[qP%6]) << (qP/6)+32) >
> 6 (8-77)
recSamples[x][y] = Clip3(0, (1<<bitDepth)
-1, tmpVal) (8-78)]
recSamples[x][y] = Clip3(0, (1<<bitDepth)-1,
PaletteEscapeVal[cIdx][xCb+XL][yCb+YL])
(8-78)

```

以下の条件の1つが真であるとき、

cIdx is equal が0に等しく、and numCompsが1に等しい
cIdxが3に等しい

変数PredictorPaletteSize[startComp]および配列PredictorPaletteEntriesは、以下のように導出または修正される。

```

for(i=0; i) CurrentPaletteSize[startComp];
i++)
for(cIdx=startComp; cIdx) (startComp+numComps);
cIdx++)

```

```

newPredictorPaletteEntries[cIdx][i] = CurrentPaletteEntries[cIdx][i]
newPredictorPaletteSize = CurrentPaletteSize[startComp]

```

```

for(i=0; i) PredictorPaletteSize && newPredictorPaletteSize) PaletteMaxPredictorSize;
i++)
if(!PalettePredictorEntryReuseFlags[i]) {

```

```

for(cIdx=startComp; cIdx) (startComp+numComps);
cIdx++) (8-79)
newPredictorPaletteEntries[cIdx][newPredictorPaletteSize] =

```

```

PredictorPaletteEntries[cIdx][i]
newPredictorPaletteSize++
}
for(cIdx=startComp; cIdx) (startComp+numComps);
cIdx++)

```

```

for(i=0; i) newPredictorPaletteSize; i++)
PredictorPaletteEntries[cIdx][i] = newPredictorPaletteEntries[cIdx][i]
PredictorPaletteSize[startComp] = newPredictorPaletteSize

```

PredictorPaletteSize[startComp]の値が0からPaletteMaxPredictorSizeまでの範囲内にあることが、ビットストリーム適合性の要件である。

5.14 実施形態 # 14

【0276】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには「[[]]」の印が付けられる。

【0277】

8.4.5.3 パレットモードの復号化処理

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ブロックの左上のサンプルを規定する位置 (x_{Cb} , y_{Cb})、
- 変数 $startComp$ は、パレットテーブルの最初の色要素を規定し、10
- 現在のブロックの色要素を規定する変数 $cidx$ 、
- 現在のブロックの幅および高さをそれぞれ規定する 2 つの変数 $nCbW$ および $nCbH$ 。

この処理の出力は、配列 $recSamples[x][y]$ であり、 $x = 0 \dots nCbW - 1$, $y = 0 \dots nCbH - 1$ は、ブロックの再構成サンプル値を規定する。

$cidx$ の値に基づいて、変数 $nSubWidth$ および $nSubHeight$ は、以下のように導出される。

- $cidx$ が 0 に等しい場合、 $nSubWidth$ は 1 に設定され、 $nSubHeight$ は 1 に設定される。
- あるいは、 $nSubWidth$ を $SubWidthC$ に設定し、 $nSubHeight$ を $SubHeightC$ に設定する。20

位置 (x_{Cb} , y_{Cb}) にある再構成サンプル配列 $recSamples$ の ($nCbW \times nCbH$) ブロックは、 $x = 0 \dots nCTbW - 1$ および $y = 0 \dots nCbH - 1$ である $recSamples[x][y]$ で表され、 $0 \sim nCbW - 1$ の範囲の各 x と $0 \sim nCbH - 1$ の範囲の各 y に対する $recSamples[x][y]$ の値は、以下のように導出される。

- 変数 x_L および y_L は、以下のように導出される。
 $x_L = paletteTransposeFlag ? x * nSubHeight : x * nSubWidth$ (8-234)
 $y_L = paletteTransposeFlag ? y * nSubWidth : y * nSubHeight$ (8-235) 30

- 変数 $bIsEscapeSample$ は、以下のように導出される。
- $PaletteIndexMap[x_{Cb} + x_L][y_{Cb} + y_L]$ が $MaxPaletteIndex$ に等しく、 $paletteEscape_val_present_flag$ が 1 に等しい場合、 $bIsEscapeSample$ は 1 に設定される。

- あるいは、 $bIsEscapeSample$ を 0 に設定する。
- $bIsEscapeSample$ が 0 に等しい場合、以下が適用される。

$recSamples[x][y] = CurrentPaletteEntries[cIdx][PaletteIndexMap[x_{Cb} + x_L][y_{Cb} + y_L]]$ (8-236)

- あるいは、 $cu_transquant_bypass_flag$ が 1 に等しい場合、以下が適用される。

$recSamples[x][y] = PaletteEscapeVal[cIdx][x_{Cb} + x_L][y_{Cb} + y_L]$ (8-237)

- あるいは、($bIsEscapeSample$ が 1 に等しく、 $cu_transquant_bypass_flag$ が 0 に等しい場合)、以下の順序付けられたステップが適用される。

1. 第 8.7.1 項で規定される量子化パラメータの導出処理は、現在のピクチャの左上サンプルに対する現在のブロックの左上サンプルを規定する位置 (x_{Cb} , y_{Cb}) を使用して呼び出される。

10

20

30

40

50

[E d . (B B) : Q P は、イントラ C U 復号化処理の最初に既に導出されているので、この従属項内でそれらを再び導出する必要はない。これは H E V C v 4 S C C にあるように思われるが、このような冗長性は排除できる。ご確認のこと。]

2 . 量子化パラメータ q P は、以下のように導出される。

- c I d x が 0 に等しい場合

$$q P = \text{Max} (Q p \text{PrimeTsMin}, Q p' Y) \quad (8 - 238)$$

- あるいは、c I d x が 1 に等しい場合、

$$q P = \text{Max} (Q p \text{PrimeTsMin}, Q p' C b) \quad (8 - 239)$$

- あるいは、(c I d x が 2 に等しい場合)、

$$q P = \text{Max} (Q p \text{PrimeTsMin}, Q p' C r) \quad (8 - 240)$$

ここで、min_q p_prime_ts_minus 4 は、変換スキップモードにおける最小許容量子化パラメータを以下のように規定する。

$$Q p \text{PrimeTsMin} = 4 + \text{min}_q p_prime_ts_minus 4$$

3 . 変数 bitDepth は、以下のように導出される。

$$\text{bitDepth} = (c I d x == 0) ? \text{BitDepthY} : \text{BitDepthC} \quad (8 - 241)$$

4 . list_levelScale [] は、k = 0 .. 5 の時、levelScale [k] = { 40, 45, 51, 57, 64, 72 } として規定される

[E d . (B B) : 非パレット C U の場合、levelScale は rectNonT sFlag に依存する。ここでも適用されますか？]

5 . 以下が適用される

$$\text{tmpVal} = (\text{PaletteEscapeVal}[c I d x][x C b + x L][y C b + y L] * \text{levelScale}[q P \% 6]) << (q P / 6) + 32) >> 6 \quad (8 - 242)$$

$$\text{recSamples}[x][y] = \text{Clip3}(0, (1 << \text{bitDepth}) - 1, \text{tmpVal}) \quad (8 - 243)$$

以下の条件の 1 つが真であるとき、

- c I d x は equal to 0 and numComps は equal to 1 ;

- c I d x が 3 に等しい

変数 PredictorPaletteSize [startComp] および配列 PredictorPaletteEntries は、以下のように導出または修正される。

$$\text{for} (i = 0 ; i) \text{CurrentPaletteSize}[startComp]; i++)$$

$$\text{for} (c I d x = startComp; c I d x) (startComp + numComps); c I d x ++)$$

$$\text{newPredictorPaletteEntries}[c I d x][i] = \text{currentPaletteEntries}[c I d x][i]$$

$$\text{newPredictorPaletteSize} = \text{currentPaletteSize}[startComp]$$

$$\text{for} (i = 0 ; i) \text{PredictorPaletteSize} \&& \text{newPredictorPaletteSize}) \text{PaletteMaxPredictorSize}; i++)$$

$$\text{if} (! \text{PalettePredictorEntryReuseFlags}[i]) \{ \text{for} (c I d x = startComp; c I d x) (startComp + numComps); c I d x ++) \quad (8 - 244)$$

$$\text{newPredictorPaletteEntries}[c I d x][\text{newPredictorPaletteSize}] = \text{PredictorPaletteEntries}[c I d x][i]$$

$$\text{newPredictorPaletteSize}++$$

10

20

30

40

50

```

}
for (cIdx = startComp; cIdx < (startComp + numComps); cIdx++)
    for (i = 0; i < newPredictorPaletteSize; i++)
        PredictorPaletteEntries[cIdx][i] = newPredictorPaletteEntries[cIdx][i];
PredictorPaletteSize[startComp] = newPredictorPaletteSize;
PredictorPaletteSize[startComp] の値が 0 から PredictorPaletteEntries までの範囲内にあることが、ビットストリーム適合性の要件である。 10

```

【0278】

5.15 実施形態 # 15

【0279】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには「[]」の印が付けられる。

【0280】

8.4.2 輝度イントラ予測モードのための導出処理

...

- あるいは、skip_intra_flag[xPb][yPb] および DimFlag[xPb][yPb] はいずれも 0 に等しい場合)、IntraPredModeY[xPb][yPb] は、以下の順序ステップによって導出される。 20

1. 近傍位置(xNbA, yNbA) および (xNbB, yNbB) は、それぞれ、(xPb - 1, yPb) および (xPb, yPb - 1) に等しく設定される。

2. X を A または B のいずれかに置き換える場合、変数 `c and IntraPredModeX` は、以下のように導出される。

- 6.4.1 項で規定される z 走査順序のブロックの可用性導出処理は、入力として位置(xCurr, yCurr) を (xPb, yPb) に等しく、近傍位置(xNbY, yNbY) を (xNbX, yNbX) に等しく設定して呼び出され、その出力は `availableX` に割り当てられる。 30

- 候補イントラ予測モード `c and IntraPredModeX` は、以下のように導出される。

- `availableX` が FALSE に等しい場合、`c and IntraPredModeX` は INTRA_DC に等しく設定される。

[[- あるいは、CuPredMode[xNbX][yNbX] が MODE_INTRA または pcm_flag[xNbX][yNbX] が 1 に等しい、または `c and IntraPredModeX` が INTRA_DC,] に等しく設定される]

- あるいは、CuPredMode[xNbX][yNbX] が MODE_INT に等しくなく、pcm_flag[xNbX][yNbX] が 1 に等しいか、または palette_mode_flag が 1 に等しい場合、`c and IntraPredModeX` は INTRA_DC と等しく設定される。 40

- あるいは、X が B に等しく、yPb_1 が ((yPb > CtbLog2SizeY) CtbLog2SizeY) 未満である場合、`c and IntraPredModeB` は INTRA_DC に等しく設定される。

- あるいは、IntraPredModeY[xNbX][yNbX] が 34 より大きい場合、`c and IntraPredModeX` は INTRA_DC に等しく設定される。

...

【0281】

5.16 実施形態 # 16

10

20

30

40

50

【0282】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[]”の印が付けられる。

【0283】**8.4.2 輝度イントラ予測モードのための導出処理**

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を規定する輝度位置 (x_{Cb} , y_{Cb})、
 - 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 $cbWidth$
 - 載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 $cbHeight$
- この処理において、輝度イントラ予測モード $IntraPredModeY[x_{Cb}][y_{Cb}]$ が導出される。

1. X を A または B のいずれかに置き換える場合、変数 $candidateIntraPredModeX$ は、以下のように導出される。

- 6.4.X 項 [Ed. (BB)] : 近傍ブロックの可用性チェック処理 tbd で規定されるブロックの可用性導出処理は、入力として、(x_{Cb} , y_{Cb}) に等しく設定された位置 (x_{curr} , y_{curr}) と、(x_{NbX} , y_{NbX}) に等しく設定した近傍位置 (x_{NbY} , y_{NbY}) で呼び出され、出力を $availableX$ に割り当てる。

- 候補イントラ予測モード $candidateIntraPredModeX$ は、以下のように導出される。

- 以下の条件の 1 つ以上が真である場合、 $candidateIntraPredModeX$ を $INTRA_PLANAR$ に等しく設定する。

- 変数 $availableX$ は $FALSE$ に等しい。
- $CandidateMode[x_{NbX}][y_{NbX}]$ は、 $MODE_INTRA$ と等しくない。
- $pred_mode_flag$ は 1 に等しい。
- $intra_mip_flag[x_{NbX}][y_{NbX}]$ は 1 に等しい。
- X が B に等しく、 $y_{Cb} - 1$ が ($(y_{Cb} >> CtbLog2SizeY) << CtbLog2SizeY$) 未満である。

- あるいは、 $candidateIntraPredModeX$ を $IntraPredModeY[x_{NbX}][y_{NbX}]$ に等しく設定する。

...

($x = x_{Cb} \dots x_{Cb} + cbWidth - 1$, $y = y_{Cb} \dots y_{Cb} + cbHeight - 1$ の場合、変数 $IntraPredModeY[x][y]$) は、 $IntraPredModeY[x_{Cb}][y_{Cb}]$ と等しく設定される。

【0284】**5.17 実施形態 #17****【0285】**

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[]”の印が付けられる。

【0286】**8.4.3 載度イントラ予測モードのための導出処理**

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を規定する輝度位置 (x_{Cb} , y_{Cb})、
 - 載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 $cbWidth$ 、
 - 載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 $cbHeight$ 。
- この処理において、輝度イントラ予測モード $IntraPredModeY[x_{Cb}][y_{Cb}]$ が導出される。

2. X を A または B のいずれかに置き換える場合、変数 $candidateIntraPred$

10

20

30

40

50

Mode X は、以下のように導出される。

- 6.4.X 項 [Ed. (BB)] : 近傍ブロックの可用性導出処理 tbd] で規定されたブロックの可用性導出処理は、入力として、(xCb, yCb) に等しく設定された位置 (xCurr, yCurr) と、(xNbX, yNbX) に等しく設定した近傍位置 (xNbY, yNbY) で呼び出され、出力を availableX に割り当てる。

- 候補イントラ予測モード candIntraPredModeX は、以下のように導出される。

- 以下の条件の 1つ以上が真である場合、candIntraPredModeX を [[INTRA_PLANAR]] INTRA_DC に等しく設定する。

- 変数 availableX は FALSE に等しい。
- CuPredMode [xNbX] [yNbX] は、MODE_INTRA と等しくない。

- intra_mip_flag [xNbX] [yNbX] は 1 に等しい。
- X が B に等しく、yCb_1 が ((yCb >> CtbLog2SizeY) << CtbLog2SizeY) 未満である。

- あるいは、candIntraPredModeX を IntraPredModeY [xNbX] [yNbX] に等しく設定する。

...

- ($x = x_{Cb} \dots x_{Cb} + cbWidth - 1$, $y = y_{Cb} \dots y_{Cb} + cbHeight - 1$ の場合、変数 IntraPredModeY [x] [y]) は、IntraPredModeY [xCb] [yCb] と等しく設定される。

【0287】

5.18 実施形態 #18

【0288】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには「[[]]」の印が付けられる。

【0289】

8.4.3 輝度イントラ予測モードのための導出処理

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 。

この処理において、輝度イントラ予測モード IntraPredModeY [xCb] [yCb] が導出される。

3. X を A または B のいずれかに置き換える場合、変数 candIntraPredModeX は、以下のように導出される。

- 6.4.X 項 [Ed. (BB)] : 近傍ブロックの可用性導出処理 tbd] で規定されるブロックの可用性導出処理は、入力として、(xCb, yCb) に等しく設定された位置 (xCurr, yCurr) と、(xNbX, yNbX) に等しく設定した近傍位置 (xNbY, yNbY) で呼び出され、出力を availableX に割り当てる。

- 候補イントラ予測モード candIntraPredModeX は、以下のように導出される。

- 以下の条件の 1つ以上が真である場合、candIntraPredModeX を [[INTRA_PLANAR]] INTRA_DC に等しく設定する。

- 変数 availableX は FALSE に等しい。
- CuPredMode [xNbX] [yNbX] は、MODE_INTRA と等しくない。

- intra_mip_flag [xNbX] [yNbX] は 1 に等しい。
- pred_mode_p1t_flag は 1 に等しい。

10

20

30

40

50

- XがBに等しく、y C b_1が((y C b >> C t b L o g 2 S i z e Y)<< C t b L o g 2 S i z e Y)未満である。

- あるいは、c a n d I n t r a P r e d M o d e XをI n t r a P r e d M o d e Y [x N b X] [y N b X]に等しく設定する。

...

(x = x C b . . x C b + c b W i d t h - 1、y = y C b . . y C b + c b H e i g h t - 1の場合、変数I n t r a P r e d M o d e Y [x] [y])は、I n t r a P r e d M o d e Y [x C b] [y C b]と等しく設定される。

【0290】

5.19 実施形態 # 19

10

【0291】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[]”の印が付けられる。

【0292】

20

30

40

50

【表 2 8】

Coding_unit(x0, y0, cbWidth, cbHeight, treeTypeCurr, isInSCIPURegion, SCIPUConsMode) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeTypeCurr != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) SCIPUConsMode == MODE_NON_INTER) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && SCIPUConsMode == MODE_ALL)	
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) && SCIPUConsMode != MODE_INTER)	
pred_mode_ibc_flag	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && SCIPUConsMode != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(isInSCIPURegion && SCIPUConsMode == MODE_ALL && CuPredMode[x0][y0] != MODE_INTER) {	
treeType = DUAL_TREE_LUMA	
} else {	
treeType = treeTypeCurr	
}	
...	
}	

【0293】

5.20 実施形態 #20

【0294】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには「[[]]」の印が付けられる。

【0295】

10

20

30

40

50

【表 2 9】

Coding_unit(x0, y0, cbWidth, cbHeight, treeTypeCurr, isInSCIPURegion, SCIPUConsMode) {	ディスクリ プタ
if(slice_type != I sps_ibc_enabled_flag // sps_plt_enabled_flag) {	
if(treeTypeCurr != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) SCIPUConsMode == MODE_NON_INTER) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && SCIPUConsMode == MODE_ALL)	
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) && SCIPUConsMode != MODE_INTER)	
pred_mode_ibc_flag	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && SCIPUConsMode != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(isInSCIPURegion && SCIPUConsMode == MODE_ALL && CuPredMode[x0][y0] != MODE_INTER) {	
treeType = DUAL_TREE_LUMA	
} else {	
treeType = treeTypeCurr	
}	
...	
}	

10

20

30

【0 2 9 6】

5 . 2 1 実施形態 # 2 1

【0 2 9 7】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“ []] ”の印が付けられる。

40

【0 2 9 8】

50

【表30】

Coding_unit(x0, y0, cbWidth, cbHeight, treeTypeCurr, isInSCIPURegion, SCIPUConsMode) {	ディスクリ プタ
<pre> if(slice_type != I sps_ibc_enabled_flag // sps_plt_enabled_flag) { if(treeTypeCurr != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) SCIPUConsMode == MODE_NON_INTER) && !sps_ibc_enabled_flag)) </pre>	
<pre> cu_skip_flag[x0][y0] if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && SCIPUConsMode == MODE_ALL) </pre>	ae(v)
<pre> pred_mode_flag 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) && SCIPUConsMode != MODE_INTER) </pre>	ae(v)
<pre> pred_mode_ibc_flag if(((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && SCIPUConsMode != MODE_INTER) </pre>	ae(v)
<pre> pred_mode_plt_flag } if(isInSCIPURegion && SCIPUConsMode == MODE_ALL && CuPredMode[x0][y0] != MODE_INTER) { treeType = DUAL_TREE_LUMA } else { treeType = treeTypeCurr } ... } </pre>	ae(v)

〔 0 2 9 9 〕

5.2.2 實施形態 # 2.2

【 0 3 0 0 】

本実施形態では、符号化ユニットの構文について説明する。本実施形態において、pred_mode_flagは、pred_mode_ibc_flagの後に信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[[]]”の印が付けられる。

(0 3 0 1)

【表 3 1】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag / / sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([[cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	
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))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) {	
if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

【0 3 0 2】

5 . 2 3 実施形態 # 2 3

【0 3 0 3】

新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“ []] ”の印が付けられる。

【0 3 0 4】

10

20

30

40

50

【表 3 2】

Coding_unit(x0,y0,cbWidth,cbHeight,treeTypeCurr, isInSCIPURegion, SCIPUConsMode) {	ディスク リプタ
if(slice_type != I sps_ibc_enabled_flag sps_plt_enabled_flag) {	
if(treeTypeCurr != DUAL_TREE_CHROMA &&!(((cbWidth == 4 && cbHeight == 4) SCIPUConsMode == MODE_NON_INTER) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && SCIPUConsMode == MODE_ALL)	
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) && SCIPUConsMode != MODE_INTER)	
pred_mode_ibc_flag	ae(v)
if(((slice_type == I (cbWidth == 4 && cbHeight == 4) / / pred_mode_ibc_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !pred_mode_ibc_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && SCIPUConsMode != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(isInSCIPURegion && SCIPUConsMode == MODE_ALL && CuPredMode[x0][y0] != MODE_INTER) {	
treeType = DUAL_TREE_LUMA	
} else {	
treeType = treeTypeCurr	
}	
...	
}	

【0305】

5.2.4 実施形態 # 2.4

【0306】

本実施形態では、符号化ユニットの構文について説明する。本実施形態において、`pred_mode_plt_flag`は、`pred_mode_ibc_flag`の後に信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“[[]]”の印が付けられる。

【0307】

10

20

30

40

50

【表 3 3】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, treeType, modeType) {	ディスクリプタ
if(slice_type != I sps_ibc_enabled_flag / / sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !([[cbWidth == 4 && cbHeight == 4 && !sps_ibc_enabled_flag)]] ((cbWidth == 4 && cbHeight == 4) // modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
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))]]	
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)))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_TYPE_INTER) { if(sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if((((slice_type == I (cbWidth == 4 && cbHeight == 4) pred_mode_ibc_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && pred_mode_ibc_flag && CuPredMode[x0][y0] != MODE_INTRA)) && cbWidth <= 64 && cbHeight <= 64 && sps_plt_enabled_flag && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
...	
}	

【0308】

40

5.2.5 実施形態 # 25

【0309】

30

本実施形態では、符号化ユニットの構文について説明する。本実施形態では、現在の予測モードが MODE_PLT であるとき、パレット構文が信号通知される。新規に追加されたテキストは太字のイタリック体で表され、削除されたテキストには“ [] ”の印が付けられる。

【0310】

50

【表 3 4】

7. 3. 7. 5 符号化ユニット構文

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	ディスク リプタ
chType = treeType == DUAL_TREE_CHROMA?1 :0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if((slice_type == I && cu_skip_flag[x0][y0] ==0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
}	
if([[CuPredMode[chType][x0][y0] == MODE_INTRA]] CuPredMode[chType][x0][y0] == MODE_PLT) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(treeType == DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
} else {	
...	
}	
...	
}	

10

20

30

40

【0 3 1 1】

5. 2 6 実施形態 # 2 6

【0 3 1 2】

本実施形態では、クロマイントラ予測モードの導出処理について説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

【0 3 1 3】

クロマイントラ予測モードのための導出処理

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クロマ符号化ブロックの左上の

50

サンプルを規定する輝度位置 (x_{Cb} , y_{Cb})、

- 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 $cbHeight$ 。
この処理において、クロマイントラ予測モード $IntraPredModeC[x_{Cb}] [y_{Cb}]$ が導出される。

対応する輝度イントラ予測モード $lumaIntraPredMode$ は、以下のように導出される。

- $intra_mip_flag[x_{Cb}] [y_{Cb}]$ が 1 に等しい場合、 $lumaIntraPredMode$ は $INTRA_PLANAR$ に等しく設定される。
- あるいは、 $CuPredMode[0][x_{Cb}] [y_{Cb}]$ が $MODE_IBC$ または $MODE_PLT$ に等しい場合、 $lumaIntraPredMode$ は $INTRA_DC$ に等しく設定される。
- あるいは、 $lumaIntraPredMode$ は、 $IntraPredModeY[x_{Cb} + cbWidth / 2] [y_{Cb} + cbHeight / 2]$ に等しく設定される。
- ...

【0314】

5.27 実施形態 #27

【0315】

本実施形態では、輝度サンプルのマッピング処理を行うピクチャ再構成処理について説明する。新規に追加されたテキストは、太字のイタリック体で示されている。

【0316】

輝度サンプルのためのマッピング処理を伴うピクチャ再構成。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サンプルに対する現在のブロックの左上のサンプルの位置 (x_{Curr} , y_{Curr})、
- ブロック幅を規定する変数 $nCurSw$ 、
- ブロックの高さを規定する変数 $nCurSh$ 、
- 現在のブロックの輝度予測サンプルを規定する $an(nCurSw) \times (nCurSh) array_predSamples$ 、
- 現在のブロックの輝度残差サンプルを規定する $an(nCurSw) \times (nCurSh) array_resSamples$ 。

この処理の出力は、再構成された輝度ピクチャサンプルアレイ $recSamples$ である。

マッピングされた予測輝度サンプル $predMapSamples$ の $(nCurSw) \times (nCurSh)$ アレイは、以下のように導出される。

- 以下の条件の 1 つが真である場合、 $predMapSamples[i][j]$ は、 $i = 0 .. nCurSw - 1$, $j = 0 .. nCurSh - 1$ の場合、 $predSamples[i][j]$ に等しく設定される。
- $CuPredMode[0][xCurr][yCurr]$ は、 $MODE_INTRA$ に等しい。

- $CuPredMode[0][xCurr][yCurr]$ は、 $MODE_IBC$ に等しい。

- $CuPredMode[0][xCurr][yCurr]$ は、 $MODE_PLT$ に等しい。

- $CuPredMode[0][xCurr][yCurr]$ は $MODE_INTER$ に等しく、 $ciip_flag[xCurr][yCurr]$ は 1 に等しい。

- あるいは、 $(CuPredMode[0][xCurr][yCurr]$ は $MODE_INTER$ に等しく、 $ciip_flag[xCurr][yCurr]$ は 0 に等しい場合)、以下が適用される。

...

10

20

30

40

50


```

    }
}
}

```

【 0 3 1 9 】

図6は、映像処理装置600のブロック図である。装置600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600は、スマートフォン、タブレット、コンピュータ、IOT(モノのインターネット)受信機等により実施されてもよい。装置600は、1つ以上の処理装置602と、1つ以上のメモリ604と、映像処理ハードウェア606と、を含んでもよい。処理装置(単数または複数)602は、本明細書に記載される1つ以上の方法を実装するように構成されてもよい。メモリ(単数または複数)604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606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。

【 0 3 2 0 】

図8は、映像を処理する方法800のフローチャートである。この方法800は、パレットモードを使用して、予測モードとは別個に符号化されたパレットモードの変換ユニット、符号化ブロック、又は領域を処理することを判定すること(805)と、このパレットモードを使用して、この変換ユニット、符号化ブロック、又は領域に対してさらに処理を行うこと(810)を含む。

【 0 3 2 1 】

方法800を参照し、パレットモード符号化のいくつかの例およびその使用について、本明細書の第4章に記載する。

【 0 3 2 2 】

方法800を参照すると、映像ブロックは、パレットモード符号化に関するビットストリーム生成規則を使用することによってビット効率を達成し得る映像ビットストリームにおいて符号化されてもよい。

【 0 3 2 3 】

この方法は、パレットモードの使用を指示する前に、予測モードを符号化することを含むことができる。

【 0 3 2 4 】

この方法は、予測モードに基づいてパレットモードの使用を条件付きで信号通知することを含むことができる。

【 0 3 2 5 】

この方法は、予測モードがイントラブロックコピーモードであり、パレットモードの使用指示の信号通知をスキップすることを含むことができる。

【 0 3 2 6 】

この方法は、現在の予測モードがイントラブロックコピーモードであることに基づいて、パレットモードの使用指示が偽であると判定されることを含むことができる。

【 0 3 2 7 】

この方法は、予測モードがインターモードであり、パレットモードの使用指示の信号通知がスキップされることを含むことができる。

【 0 3 2 8 】

この方法は、現在の予測モードがインターモードであることに基づいて、パレットモードの使用指示が偽であると判定されることを含むことができる。

【 0 3 2 9 】

この方法は、予測モードがイントラモードであり、パレットモードの使用指示の信号通知がスキップされることを含むことができる。

【 0 3 3 0 】

この方法は、現在の予測モードがイントラモードであることに基づいて、パレットモードの使用指示が偽であると判定されることを含むことができる。

10

20

30

40

50

【 0 3 3 1 】

この方法は、予測モードがイントラモードであり、パレットモードの使用指示の信号通知がスキップされることを含むことができる。

【 0 3 3 2 】

この方法は、予測モードがイントラロックコピーモードであり、パレットモードの使用指示の信号通知を行うことを含むことができる。

【 0 3 3 3 】

この方法は、ピクチャ、スライス、またはタイルグループのタイプに基づいて、パレットモードの使用指示を信号通知することを含むことができる。

【 0 3 3 4 】

この方法は、パレットモードを予測モードの候補として追加することを含むことができる。

【 0 3 3 5 】

この方法は、予測モードが、イントラモード、イントラロックコピーモード、イントラスライス用パレットモードまたは、インターライズ、Iピクチャ、Pピクチャ、Bピクチャ、またはイントラタイルグループのうちの1つ以上を含むことができる。

【 0 3 3 6 】

この方法は、予測モードが、イントラモード、インターモード、イントラロックコピーモード、またはパレットモードのうちの2つ以上を含むことができる。

【 0 3 3 7 】

この方法は、パレットモードの使用が信号通知によって示されるか、または条件に基づいて導出されることを含むことができる。

【 0 3 3 8 】

この方法は、現在のブロックのブロック寸法、現在のブロックの予測モード、現在のブロックの量子化パラメータ (Q P)、近傍のブロックのパレットフラグ、近傍のブロックのイントラブロックコピーフラグ、カラーフォーマットの指示、別個のまたはデュアル符号化ツリー構造、またはスライスタイプ、グループタイプまたはピクチャタイプのうちの1つ以上を含むことができる。

【 0 3 3 9 】

この方法は、スライスレベルフラグ、タイルグループレベルフラグ、またはピクチャレベルフラグに基づいて、パレットモードの使用を信号通知または導出することを含むことができる。

【 0 3 4 0 】

この方法は、スライスレベルフラグ、タイルグループレベルフラグ、またはピクチャレベルフラグに基づいて、イントラブロックコピーモードの使用の指示を信号通知または導出することを含むことができる。

【 0 3 4 1 】

前の章で開示した項目 6 ~ 9 を参照し、いくつかの実施形態において、以下の解決策を用いることが好ましい。

【 0 3 4 2 】

1つの解決策は、映像のピクチャの現在の映像ブロックと、この映像のビットストリーム表現との間での変換を行うことを含み、イントラブロックコピーモードがこの変換において使用されるか否かに関する情報が、ビットストリーム表現において信号通知される、または、この現在の映像ブロックの符号化条件に基づいて導出され、ここで、このイントラブロックコピーモードは、このピクチャにおける別の映像ブロックからのこの現在の映像ブロックを符号化することを含む、映像処理方法を含んでもよい。様々な実施形態において以下の特徴を実装することができる。

【 0 3 4 3 】

- 符号化条件は、現在の映像ブロックのブロック寸法を含む。

【 0 3 4 4 】

10

20

30

40

50

- 符号化条件は、現在の映像ブロックの予測モードまたは前記現在の映像ブロックの変換に用いられる量子化パラメータを含む。

【0345】

前章で開示した項目13～15を参照し、いくつかの実施形態において、以下の解決策を実装することが好ましい。

【0346】

1つの解決策は、映像のピクチャの現在の映像ブロックを変換する間に非ブロック化フィルタを適用するか否かを判定することと、ここで、現在の映像ブロックは、現在の映像ブロックの全画素より小さい代表的なサンプル値を使用して現在の映像ブロックを表すパレットモード符号化を使用して符号化され、非ブロック化フィルタを適用すると判定した場合に非ブロック化フィルタが適用されるように変換を行うことについての方法を含んでもよい。

10

【0347】

別の解決策は、ある映像のピクチャの現在の映像ブロックとこの映像のビットストリーム表現との間での変換中に使用するために、量子化または逆量子化処理を判定することと、ここで、現在の映像ブロックは、現在の映像ブロックの合計画素よりも少ない代表的なサンプル値を使用して現在の映像ブロックを表すパレットモード符号化を使用して符号化され、この量子化または逆量子化処理の判定に基づいて、この変換を行うことと、を含む映像処理方法を含んでもよい。追加の特徴は、以下を含んでもよい。

20

【0348】

- 現在の映像ブロックに対して判定された量子化または逆量子化処理は、パレット符号化モードとは異なるように符号化された別の映像ブロックに適用される別の量子化または別の逆量子化処理とは異なる。

【0349】

- この変換は、現在の映像ブロックをビットストリーム表現に符号化することを含む。

【0350】

- この変換は、ビットストリーム表現を復号化して映像の現在の映像ブロックを生成することを含む。

【0351】

- この判定は、イントラ符号化された他の映像ブロックの変換に用いられる別の決定処理と同じ決定処理を用いる。

30

【0352】

開示された技術は、向上した符号化ツリー構造を使用して圧縮効率を向上させるために、映像エンコーダまたはデコーダにおいて実施され得ることが理解される。

【0353】

前の章の項目16～21を参照し、いくつかの解決策は以下の通りである。

【0354】

複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを判定することと、この判定に基づいて、現在の映像ブロックをイントラ符号化されたブロックであると考えることで、最大確率モードのリスト構築処理を行うことと、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パレットまたは表現サンプル値を使用して符号化または復号化されることを含む、映像処理方法。

40

【0355】

上記方法において、リスト構築処理は、近傍のパレット符号化ブロックをデフォルトモードのイントラブロックとして扱う。

【0356】

複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを

50

判定することと、この判定に基づいて、現在の映像ブロックを非イントラ符号化されたブロックであると考えることで、最大確率モードのリスト構築処理を実行し、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パレットまたは表現サンプル値を使用して符号化または復号化される、映像処理方法。

【0357】

上記方法において、前記リスト構築処理は、近傍のパレット符号化ブロックのイントラモードをフェッチする場合、近傍のパレット符号化ブロックをインター符号化ブロックとして扱う。

【0358】

複数の映像ブロックを含む映像の現在の映像ブロックとこの映像のビットストリーム表現との間での変換のために、現在の映像ブロックがパレット符号化ブロックであることを判定することと、この判定に基づいて、現在の映像ブロックを利用不可能なブロックであると考えることで、リスト構築処理を行うことと、このリスト構築処理の結果に基づいてこの変換を行うことと、を含み、このパレット符号化ブロックは、パレットまたは表現サンプル値を使用して符号化または復号化される、映像処理方法。

【0359】

上記方法において、前記リスト構築処理は、履歴に基づく動きベクトル予測である。

【0360】

上記方法において、前記リスト構築処理は、M E R G E または高度動きベクトル予測モードである。

【0361】

上記方法において、前記判定することは、前記映像のコンテンツに基づいて判定することをさらに含む。

【0362】

上記方法において、前記判定は、ビットストリーム表現におけるフィールドに対応する。

【0363】

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中に、この現在の映像ブロックがパレット符号化ブロックであることを判定することと、現在の映像ブロックがパレット符号化ブロックであることを基づいて、この変換に使用されるコンテキスト符号化ビンの範囲を判定することと、このコンテキスト符号化ビンの範囲に基づいてこの変換を行うことを含む、映像処理方法。

【0364】

上記方法において、現在の映像ブロックのうち、範囲外にあるビンを、バイパス符号化技術を使用して符号化するか、または変換中にバイパス復号化技術を使用して復号化する。

【0365】

前記変換は、前記ビデオを前記ビットストリーム表現に符号化することを含む、上記記載の方法。

【0366】

上記方法において、前記変換は、前記ビットストリーム表現を復号化して前記映像を生成することを含む。

【0367】

図24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2400を示すブロック図である。様々な実装形態は、システム2400のモジュールの一部又は全部を含んでもよい。システム2400は、映像コンテンツを受信するための入力ユニット2402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1902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（P O N）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

10

20

30

40

50

－インターフェース等の無線インターフェースを含む。

【 0 3 6 8 】

システム 2400 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 2404 を含んでもよい。符号化モジュール 2404 は、入力ユニット 2402 からの映像の平均ビットレートを符号化モジュール 2404 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 2404 の出力は、モジュール 2406 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 2402 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 2408 によって使用されて、表示インターフェースユニット 2410 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。

【 0 3 6 9 】

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（U S B）または高精細マルチメディアインターフェース（H D M I（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（S A T A）、P C I、I D E 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。

【 0 3 7 0 】

図 25 は、本技術にしたがった映像処理方法 2500 を示すフローチャートである。この方法 2500 は、動作 2510 において、映像の映像領域のブロックと映像のビットストリーム表現との間で変換を行うことを含む。このビットストリーム表現は、このブロックのためにパレットモードの使用の第 1 の指示が信号通知されるかどうかを規定する第 1 のフォーマット規則と、このブロックのために予測モードの使用の第 2 の指示に関連するこの第 1 の指示の位置を規定する第 2 のフォーマット規則とに従って処理される。

【 0 3 7 1 】

いくつかの実施形態において、映像領域は、映像の変換ユニット、符号化ユニット、予測ユニット、または領域を含む。いくつかの実施形態において、予測モードの使用の第 2 の指示は、ビットストリーム表現におけるパレットモードの使用の第 1 の指示の前に位置決めされる。

【 0 3 7 2 】

いくつかの実施形態において、パレットモードの使用の第 1 の指示は、予測モードの使用の第 2 の指示に基づいて、ビットストリーム表現において条件付きで含まれる。いくつかの実施形態において、予測モードの使用の第 2 の指示がイントラブロックコピー（I B C）予測モードを示す場合、ビットストリーム表現においてパレットモードの使用の第 1 の指示をスキップする。いくつかの実施形態において、予測モードの使用の第 2 の指示がインター予測モードを示す場合、ビットストリーム表現においてパレットモードの使用の第 1 の指示をスキップする。いくつかの実施形態において、予測モードの使用の第 2 の指示がイントラ予測モードを示す場合、ビットストリーム表現においてパレットモードの使用の第 1 の指示をスキップする。いくつかの実施形態において、予測モードの使用の第 2 の指示がスキップモードを示す場合、ビットストリーム表現においてパレットモードの使用の第 1 の指示をスキップする。いくつかの実施形態において、ビットストリーム表現におけるパレットモードの使用の第 1 の指示をスキップすることは、パレットモードが使用されていないことを示す。

【 0 3 7 3 】

いくつかの実施形態において、パレットモードの使用の第1の指示は、予測モードの使用の第2の指示がIBC予測モードを示す場合、ビットストリームにおいて符号化される。いくつかの実施形態において、パレットモードの使用の第1の指示は、予測モードの使用の第2の指示がイントラ予測モードを示す場合、ビットストリームにおいて符号化される。いくつかの実施形態において、予測モードは、パルス符号変調(PCM)モードではない。いくつかの実施形態において、パレットモードの使用の第1の指示は、ビットストリーム表現におけるPCMモードの使用の指示の前に符号化される。いくつかの実施形態において、ビットストリーム表現において、PCMモードの使用の指示をスキップする。いくつかの実施形態において、IBCモードの指示はビットストリーム表現に符号化される。いくつかの実施形態において、イントラ予測モードが使用される場合、ビットストリーム表現におけるフラグは、パレットモードまたはIBCモードがビットストリーム表現において信号通知されるかどうかを示す。いくつかの実施形態において、フラグはブロックの状態に基づいてスキップされ、この状態は、ブロック寸法、ブロックに関連付けられた領域に対してIBCモードが有効化されるかどうか、或いはブロックに関連づけられた領域に対してパレットモードが有効化されるかどうかを含む。

10

【 0 3 7 4 】

いくつかの実施形態において、パレットモードの使用の第1の指示は、予測モードの使用の第2の指示がインター予測モードを示す場合、ビットストリームにおいて符号化される。いくつかの実施形態において、パレットモードの使用の第1の指示は、スキップモードの指示、予測モードの指示、またはPCMモードの使用の指示のうちの少なくとも1つの後に符号化される。いくつかの実施形態において、パレットモードの使用の第1の指示は、スキップモードまたは予測モードの指示の後に符号化され、PCMモードの使用の指示の前に符号化される。

20

【 0 3 7 5 】

いくつかの実施形態において、パレットモードの使用の第1の指示は、ビットストリーム表現における予測モードの使用の第2の指示の前に位置決めされる。いくつかの実施形態において、パレットモードの使用の第1の指示は、予測モードの使用の第2の指示の後に位置決めされ、予測モードの使用の第2の指示は、ビットストリーム表現におけるイントラ予測モードまたはインター予測モードを示す。いくつかの実施形態において、パレットモードの使用の第1の指示は、ピクチャ、スライス、またはタイルグループタイプに基づいて信号通知される。いくつかの実施形態において、パレットモードの使用の第1の指示は、ブロックに対して前記パレットモードが有効化されることを示す第1のフラグを含むことができる。いくつかの実施形態において、パレットモードの使用の第1の指示は、パレットモードがシーケンスレベル、ピクチャレベル、タイルグループレベル、またはタイルレベルで有効であることを示す第1のフラグに基づいて、条件付きでビットストリーム表現に含まれる。いくつかの実施形態において、ブロックに対してパレットモードが無効化されるとき、ブロックのPCMモードを示す別のフラグがビットストリーム表現に含まれる。いくつかの実施形態において、第1のフラグは現在のブロックの1つ以上の近傍ブロックの情報に基づいてコンテキスト符号化される。いくつかの実施形態において、第1のフラグは現在のブロックの1つ以上の近傍ブロックからのコンテキスト情報なしに符号化される。

30

【 0 3 7 6 】

いくつかの実施形態において、予測モードの使用の第2の指示は、予測モードを示す第2のフラグを含む。いくつかの実施形態において、ビットストリーム表現における第2のフラグが、予測モードがインターモードであることを示す場合、ビットストリーム表現は、イントラブロックコピーモードが有効化されるかどうかを示す第3のフラグをさらに含む。いくつかの実施形態において、ビットストリーム表現における第2のフラグが、予測モードがイントラモードであることを示す場合、ビットストリーム表現は、イントラブロックコピーモードが有効化されるかどうかを示す第3のフラグをさらに含む。いくつかの

40

50

実施形態において、第3のフラグは、ブロック寸法に基づいてビットストリーム表現に条件付きで含まれる。

【0377】

いくつかの実施形態において、ブロックは符号化ユニットであり、ビットストリーム表現における第2のフラグは予測モードがイントラモードであることを示す。いくつかの実施形態において、第1のフラグは、ブロック寸法に基づいてビットストリーム表現に条件付きで含まれる。

【0378】

図26は、本技術にしたがった映像処理方法2600を示すフローチャートである。この方法2600は、動作2610において、映像における映像領域のブロックと映像のビットストリーム表現との間での変換のために、このブロックの少なくとも1つのパレットモードを含む1つ以上の許容予測モードに基づいて、予測モードを判定することを含む。予測モードに従って、パレットモードの使用の指示を判定する。方法2600は、動作2620において、その判定に基づいて変換を行うことを含む。

10

【0379】

いくつかの実施形態において、1つ以上の許容予測モードは、イントラモードを含む。いくつかの実施形態において、1つ以上の許容予測モードは1つのイントラブロックコピー（IBC）モードを含む。いくつかの実施形態において、1つ以上の許容予測モードは、インターモードを含む。

【0380】

いくつかの実施形態において、映像領域は、イントラスライス、イントラピクチャまたはイントラタイルグループを含む。いくつかの実施形態において、1つ以上の許容予測モードは、イントラモード、イントラブロックコピーモードおよびパレットモードを含む。

20

【0381】

いくつかの実施形態において、映像領域は、インタースライス、インターピクチャ、インターフィルグループ、Pスライス、Bスライス、PピクチャまたはBピクチャを含む。いくつかの実施形態において、1つ以上の許容予測モードは、イントラモード、イントラブロックコピーモード、パレットモードおよびインターモードを含む。

【0382】

いくつかの実施形態において、ブロック寸法は 4×4 であってもよい。いくつかの実施形態において、1つ以上の許容予測モードは、ブロック寸法が 4×4 である場合、インターモードを除外する。

30

【0383】

いくつかの実施形態において、ビットストリーム表現は、ブロックがスキップモードで符号化されていない場合、1つ以上の許容予測モードを表す少なくとも1つの予測モードインデックスを含み、予測モードインデックスは、1つ以上のバイナリビンを使用して表される。

【0384】

いくつかの実施形態において、予測モードインデックスは、3つのバイナリビンを使用して表され、第1のビン値「1」は、イントラモードを示し、第1のビン値「0」および第2のビン値「0」は、インターモードを示し、第1のビン値「0」、第2のビン値「1」および第3のビン値「0」は、IBCモードを示し、第1のビン値「0」、第2のビン値「1」および第3のビン値「1」は、パレットモードを示す。

40

【0385】

いくつかの実施形態において、予測モードインデックスは、2つのバイナリビンを使用して表され、第1のビン値「1」および第2のビン値「0」は、イントラモードを示し、第1のビン値「0」および第2のビン値「0」は、インターモードを示し、第1のビン値「0」および第2のビン値「1」は、IBCモードを示し、第1のビン値「1」および第2のビン値「1」は、パレットモードを示す。

【0386】

50

いくつかの実施形態において、予測モードインデックスは、映像の現在のスライスがイントラスライスであり、IBCモードが無効である場合、1つのバイナリビンを使用して表され、第1のビン値「0」はイントラモードを表し、第2のビン値「1」はパレットモードを表す。

【0387】

いくつかの実施形態において、予測モードインデックスは、映像の現在のスライスがイントラスライスでなく、IBCモードが無効である場合、2つのバイナリビンを使用して表され、第1のビン値「1」はイントラモードを表し、第1のビン値「0」および第2のビン値「0」はインター モードを表し、第1のビン値「0」および第2のビン値「1」はパレットモードを示す。いくつかの実施形態において、予測モードインデックスは、映像の現在のスライスがイントラスライスであり、IBCモードが有効化される場合、2つのバイナリビンを使用して表され、第1のビン値「1」はIBCモードを表し、第1のビン値「0」および第2のビン値「1」はパレットモードを表し、第1のビン値「0」および第2のビン値「0」はイントラモードを示す。いくつかの実施形態において、IBCモードの使用指示は、ビットストリーム表現のシーケンスパラメータセット(SPS)において信号通知される。

【0388】

いくつかの実施形態において、予測モードインデックスは、3つのバイナリビンを使用して表され、ここで、第1のビン値「1」はインター モードを表し、第1のビン値「0」および第2のビン値「1」はイントラモードを表し、第1のビン値「0」、第2のビン値「0」および第3のビン値「1」はIBCモードを表し、第1のビン値「0」、第2のビン値「0」および第3のビン値「0」はパレットモードを示す。

【0389】

いくつかの実施形態において、予測モードインデックスは、3つのバイナリビンを使用して表され、ここで、第1のビン値「1」はイントラモードを表し、第1のビン値「0」および第2のビン値「1」はインター モードを表し、第1のビン値「0」、第2のビン値「0」および第3のビン値「1」はIBCモードを表し、第1のビン値「0」、第2のビン値「0」および第3のビン値「0」はパレットモードを示す。

【0390】

いくつかの実施形態において、予測モードインデックスは、3つのバイナリビンを使用して表され、ここで、第1のビン値「0」は、インター モードを示し、第1のビン値「1」および第2のビン値「0」は、イントラモードを示し、第1のビン値「1」、第2のビン値「1」および第3のビン値「1」は、IBCモードを示し、第1のビン値「1」、第2のビン値「1」および第3のビン値「0」は、パレットモードを示す。

【0391】

いくつかの実施形態において、1つの条件が満たされた場合、1つ以上のバイナリビンの信号通知はビットストリーム表現においてスキップされる。いくつかの実施形態において、条件はブロック寸法を含む。いくつかの実施形態において、この条件は、無効にされている予測モードを含み、ビットストリーム表現において、この予測モードに対応するバイナリビンをスキップする。

【0392】

図27は、本技術にしたがった映像処理方法2700を示すフローチャートである。この方法2700は、動作2710において、映像のブロックと映像のビットストリーム表現との間で変換を行うことを含む。ビットストリーム表現は、パレットモードの使用の第1の指示とイントラブロックコピー(IBC)モードの使用の第2の指示とを互いに依存して信号通知することを規定するフォーマット規則に従って処理される。

【0393】

いくつかの実施形態において、このフォーマット規則は、ブロックの予測モードがIBCモードでない第1の予測モードに等しい場合、第1の指示がビットストリーム表現で信号通知されることを規定する。いくつかの実施形態において、このフォーマット規則は、

10

20

30

40

50

ブロックの予測モードがパレットモードでない第1の予測モードに等しい場合、第2の指示がビットストリーム表現で信号通知されることを規定する。いくつかの実施形態において、第1の予測モードはイントラモードである。

【0394】

図28は、本技術にしたがった映像処理方法2800を示すフローチャートである。この方法2800は、動作2810において、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロックの寸法に基づいて、このビットストリーム表現におけるパレットモードの使用の指示の存在を判定することを含む。方法2800は、動作2820において、その判定に基づいて変換を行うことを含む。

【0395】

図29は、本技術にしたがった映像処理方法2900を示すフローチャートである。この方法2900は、動作2910において、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このビットストリーム表現におけるイントラブロックコピー（IBC）モードの使用の指示の存在を、このブロックの寸法に基づいて判定することを含む。方法2900は、動作2920において、その判定に基づいて変換を行うことを含む。いくつかの実施形態において、ブロック寸法は、ブロックにおけるサンプルの数量、ブロックの幅またはブロックの高さのうち少なくとも1つを含むことができる。

【0396】

いくつかの実施形態において、ブロックの幅が閾値以下である場合、この指示はビットストリーム表現で信号通知される。いくつかの実施形態において、ブロックの高さが閾値以下である場合、この指示はビットストリーム表現で信号通知される。いくつかの実施形態において、閾値は64であってもよい。

10

【0397】

いくつかの実施形態において、ブロックの幅および高さが閾値よりも大きい場合、この指示はビットストリーム表現で信号通知される。いくつかの実施形態において、閾値は4であってもよい。いくつかの実施形態において、ブロックにおけるサンプルの数が閾値よりも大きい場合、この指示はビットストリーム表現で信号通知される。いくつかの実施形態において、閾値は16であってもよい。いくつかの実施形態において、ブロックの幅がブロックの高さに等しい場合、この指示はビットストリーム表現で信号通知される。

20

【0398】

いくつかの実施形態において、(1)ブロックの幅が第1の閾値よりも大きい、(2)ブロックの高さが第2の閾値よりも大きい、または(3)ブロックにおけるサンプルの数が第3の閾値以下である場合、この指示はビットストリーム表現に含まれていない。いくつかの実施形態において、第1の閾値および第2の閾値は64である。いくつかの実施形態において、第3閾値は16であってもよい。

30

【0399】

いくつかの実施形態において、この判定は、このブロックに関連付けられた特徴にさらに基づいて行われる。いくつかの実施形態において、特徴はブロックの予測モードを含む。いくつかの実施形態において、特徴はブロックの量子化パラメータを含む。いくつかの実施形態において、特徴はブロックの近傍のブロックのパレットフラグを含む。いくつかの実施形態において、特徴はブロックの近傍のブロックのIBCフラグを含む。いくつかの実施形態において、特徴はブロックのカラーフォーマットの指示を含む。いくつかの実施形態において、特徴はブロックの符号化木構造を含む。いくつかの実施形態において、特徴はブロックのスライスグループタイプ、タイルグループタイプまたはピクチャタイプを含む。

40

【0400】

図30は、本技術にしたがった映像処理方法3000を示すフローチャートである。方法3000は、動作3010において、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロックを含む映像領域の第2の指示に基づいて、このブロックに対してパレットモードが許可されているかどうかを判定することを含む。方法

50

3000は、また、動作3020において、その判定に基づいて変換を行うことを含む。

【0401】

いくつかの実施形態において、映像領域は、スライス、タイルグループまたはピクチャを含むことができる。いくつかの実施形態において、ビットストリーム表現は、第2の指示がわずかな動きベクトル差が有効化されることを示す場合、パレットモードが許可されるかどうかの明確な指示を除外する。いくつかの実施形態において、第2の指示は、ビットストリーム表現に存在するフラグとして表される。いくつかの実施形態において、第2の指示は映像領域に対してパレットモードが有効化されるかどうかを示す。いくつかの実施形態において、ビットストリーム表現は、第2の指示が映像領域に対してパレットモードが無効化されていることを示す場合、パレットモードが許可されているかどうかを明確に示すことを含まない。いくつかの実施形態において、ビットストリーム表現がパレットモードを許可するかどうかを明確に示すものを含まない場合、ブロックに対してパレットモードは許可されない。

10

【0402】

図31は、本技術に従った映像処理の方法3100を示すフローチャートである。方法3100は、動作3110において、映像のブロックとこの映像のビットストリーム表現との間での変換のために、このブロックを含む映像領域の第2の指示に基づいて、このブロックに対してイントラブロックコピー（IBC）モードが許可されているかどうかを判定することを含む。方法3100は、また、動作3120において、その判定に基づいて変換を行うことを含む。

20

【0403】

いくつかの実施形態において、映像領域は、スライス、タイルグループまたはピクチャを含むことができる。いくつかの実施形態において、ビットストリーム表現は、第2の指示がわずかな動きベクトル差が有効化されることを示す場合、IBCモードが許可されるかどうかの明確な指示を除外する。いくつかの実施形態において、第2の指示は、ビットストリーム表現に存在するフラグとして表される。いくつかの実施形態において、第2の指示は映像領域に対してIBCモードが有効化されるかどうかを示す。いくつかの実施形態において、ビットストリーム表現は、第2の指示が映像領域に対してIBCモードが無効化されていることを示す場合、IBCモードが許可されているかどうかを明確に示すことを含まない。いくつかの実施形態において、ビットストリーム表現がIBCモードを許可するかどうかを明確に示すものを含まない場合、ブロックに対してIBCモードは許可されない。

30

【0404】

いくつかの実施形態において、変換はビットストリーム表現から現在のブロックを生成する。いくつかの実施形態において、変換は現在のブロックからビットストリーム表現を生成する。

【0405】

開示される技術のいくつかの実施形態は、映像処理ツールまたはモードを有効化するように決定または判定することを含む。一例において、映像処理ツールまたはモードが有効化される場合、エンコーダは、1つの映像ブロックを処理する際にこのツールまたはモードを使用するまたは実装するが、このツールまたはモードの使用に基づいて、結果として得られるビットストリームを必ずしも修正しなくてもよい。すなわち、映像のブロックから映像のビットストリーム表現への変換は、決定または判定に基づいて映像処理ツールまたはモードが有効化される場合に、この映像処理ツールまたはモードを使用する。別の例において、映像処理ツールまたはモードが有効化される場合、デコーダは、ビットストリームが映像処理ツールまたはモードに基づいて修正されたことを知って、ビットストリームを処理する。すなわち、決定または判定に基づいて有効化された映像処理ツールまたはモードを使用して、映像のビットストリーム表現から映像のブロックへの変換を行う。

40

【0406】

開示される技術のいくつかの実施形態は、映像処理ツールまたはモードを無効化するよ

50

うに決定または判定することを含む。一例において、映像処理ツールまたはモードが無効にされている場合、エンコーダは、映像のブロックを映像のビットストリーム表現に変換する際に、このツールまたはモードを使用しない。別の例において、映像処理ツールまたはモードが無効にされている場合、デコーダは、決定または判定に基づいて有効化された映像処理ツールまたはモードを使用してビットストリームが修正されていないことを知つて、ビットストリームを処理する。

【0407】

本明細書に記載された開示された、およびその他の解決策、実施例、実施形態、モジュール、および機能動作の実装形態は、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、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。開示された、およびその他の実施形態は、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装されるため、またはデータ処理装置の動作を制御するために、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施することができる。このコンピュータ可読媒体は、機械可読記憶デバイス、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、またはこれらの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、または複数の処理装置、若しくはコンピュータを含む、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、またはこれらの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。伝播信号は、人工的に生成した信号、例えば、機械で生成した電気、光、または電磁信号であり、適切な受信装置に送信するための情報を符号化するために生成される。

【0408】

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、またはコードとも呼ばれる）は、コンパイルされた言語また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、また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、要素、サブルーチン、また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、他のプログラムまた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、また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、また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。

【0409】

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。処理およ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）または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。

【0410】

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、汎用およ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

10

20

30

40

50

を含む。一般的に、処理装置は、リードオンリーメモリまたはランダムアクセスメモリまたはその両方から命令およ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処理装置と、命令およびデータを記憶するための1つ以上のメモリデバイスである。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、または光ディスクを含んでもよく、また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、また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、EEPROM、EEPROM、フラッシュ記憶装置、磁気ディスク、例えば内部ハードディスクまたはリムーバブルディスク、光磁気ディスク、およびCD-ROMおよびDVD-ROMディスク等の半導体記憶装置を含む。処理装置およびメモリは、特定用途のロジック回路によって補完されてもよく、または特定用途のロジック回路に組み込まれてもよい。

【0411】

本特許明細書は多くの詳細を含むが、これらは、任意の主題の範囲また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、特定の技術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コンテキストで説明されている特定の特徴は、1つの例において組み合わせて実装してもよい。逆に、1つの例のコンテキストで説明された様々な特徴は、複数の実施形態において別個にまた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、特徴は、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もよいが、主張された組み合わせからの1つ以上の特徴は、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、サブコンビネーションまたはサブコンビネーションのバリエーションに向けられてもよい。

【0412】

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または連続した順番で行われること、または示された全ての動作が行わ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。

【0413】

いくつかの実装形態およ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コンテンツに基づいて、他の実施形態、拡張および変形が可能である。

10

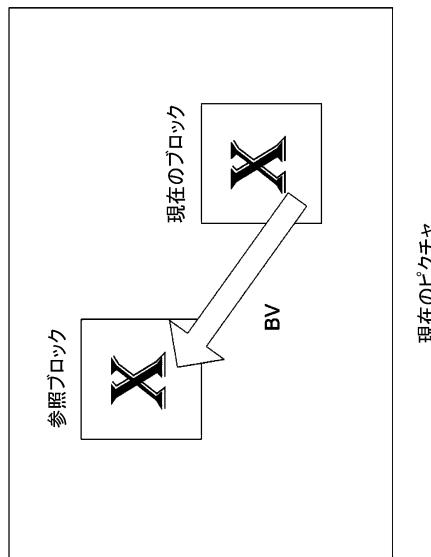
20

30

40

50

【図面】
【図 1】



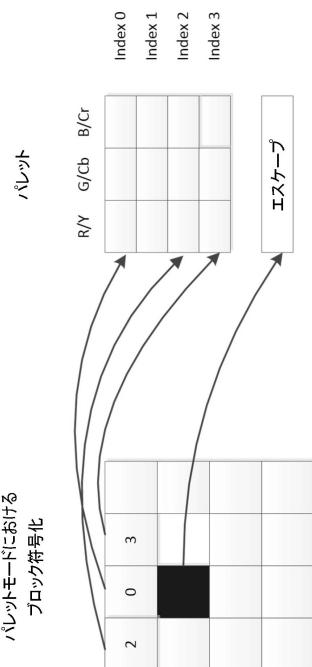
【図 3】

前のハーフレット 現在のハーフレット

Index	G/Y	B/Cb	R/Cr	Predフラグ	Index	G/Y	B/Cb	R/Cr
0	G0	B0	R0	1	0	G0	B0	R0
1	G1	B1	R1	0				
2	G2	B2	R2	1	1	G2	B2	R2
3	G3	B3	R3	1	2	G3	B3	R3
4	G4	B4	R4	0				
5	G5	B5	R5	0				

再利用ハーフエントリ(3) 信号通知された新しいハーフエントリ(2)

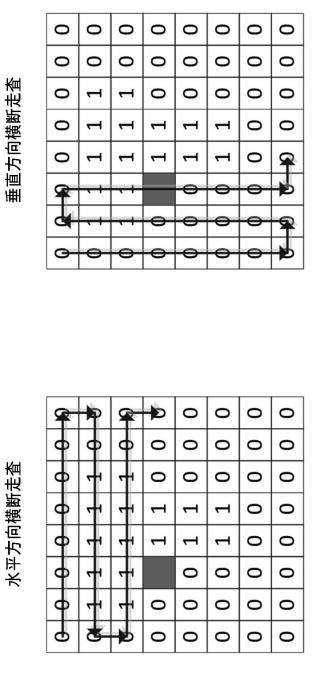
【図 2】



10

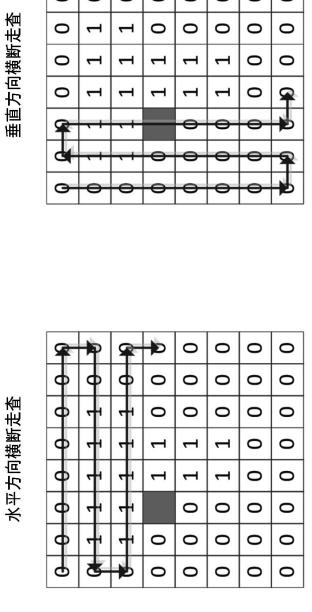
20

【図 4】



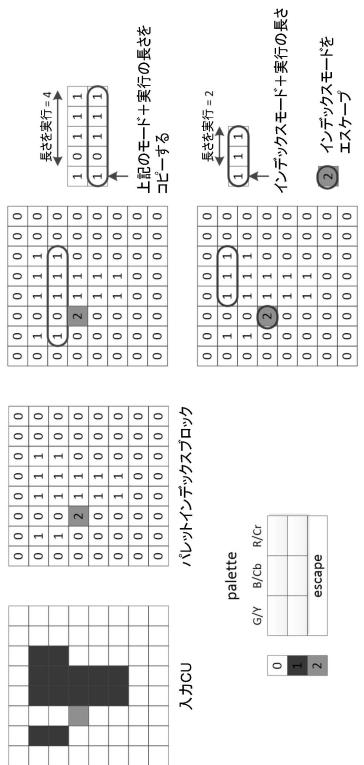
30

40

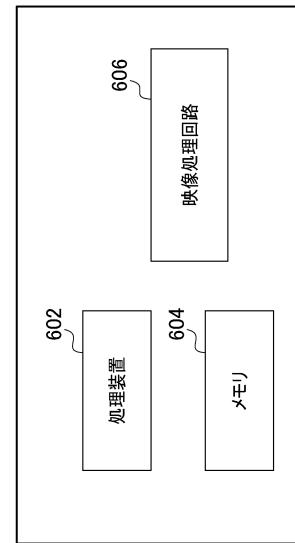


50

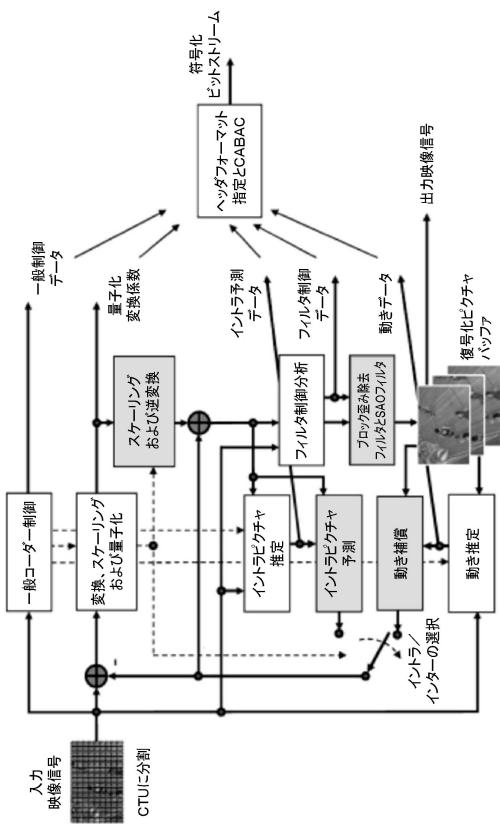
【図5】



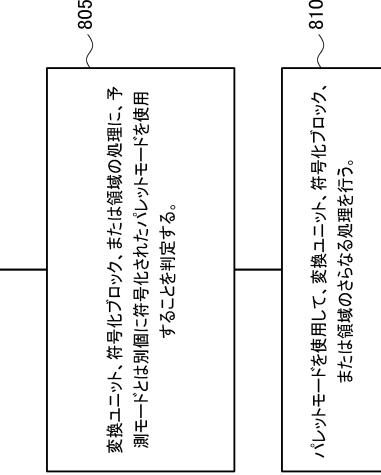
【図6】



【図7】



【図8】



変換ユニット、符号化ブロック、または領域の処理に、予測モードとは別個に符号化されたパレットモードを使用することを判定する。

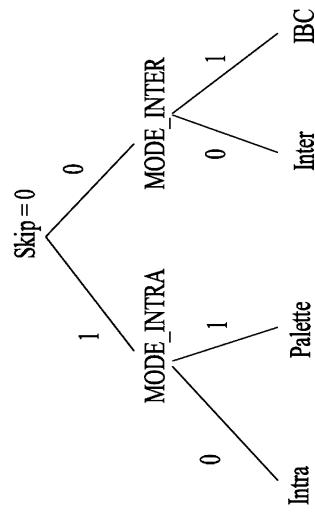
レジットモードを使用して、変換ユニット、符号化ブロック、
または領域のさらなる処理を行う。

【図 9】

【図 10】

最初の4行	p3 ₀	p2 ₀	p1 ₀	p0 ₀	q0 ₀	q1 ₀	q2 ₀	q3 ₀
p3 ₁	p2 ₁	p1 ₁	p0 ₁	q0 ₁	q1 ₁	q2 ₁	q3 ₁	
p3 ₂	p2 ₂	p1 ₂	p0 ₂	q0 ₂	q1 ₂	q2 ₂	q3 ₂	
p3 ₃	p2 ₃	p1 ₃	p0 ₃	q0 ₃	q1 ₃	q2 ₃	q3 ₃	
2番目4行	p3 ₄	p2 ₄	p1 ₄	p0 ₄	q0 ₄	q1 ₄	q2 ₄	q3 ₄
p3 ₅	p2 ₅	p1 ₅	p0 ₅	q0 ₅	q1 ₅	q2 ₅	q3 ₅	
p3 ₆	p2 ₆	p1 ₆	p0 ₆	q0 ₆	q1 ₆	q2 ₆	q3 ₆	
p3 ₇	p2 ₇	p1 ₇	p0 ₇	q0 ₇	q1 ₇	q2 ₇	q3 ₇	

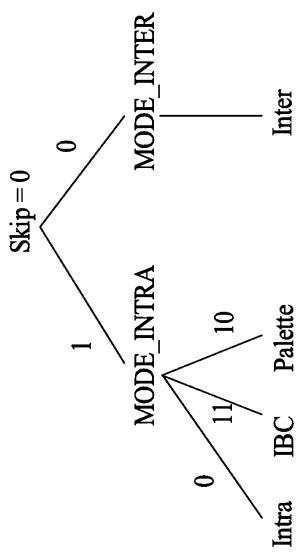
10



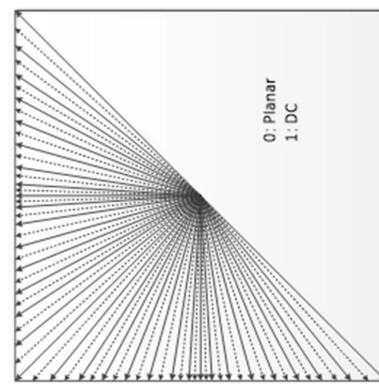
20

【図 11】

【図 12】



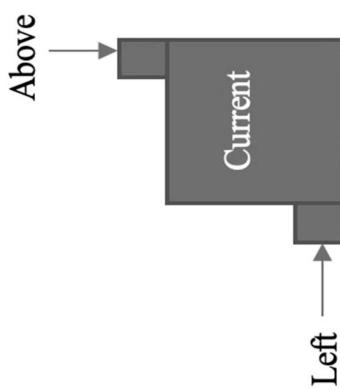
30



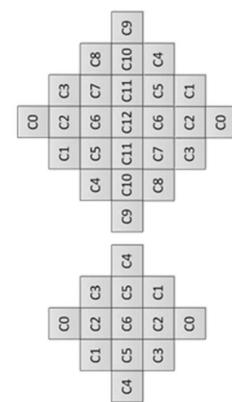
40

50

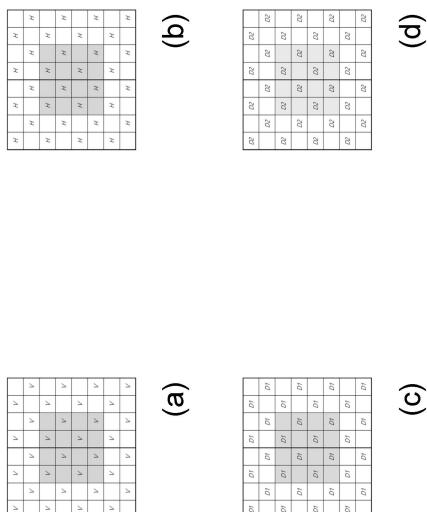
【図13】



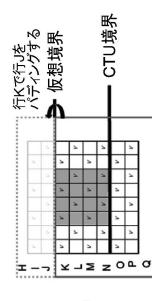
【図14】



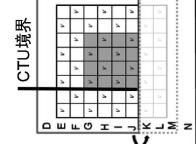
【図15】



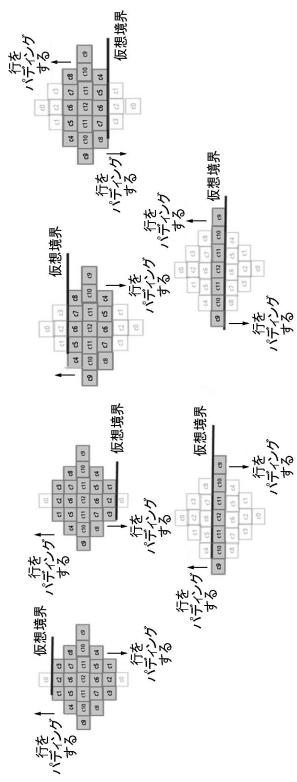
【 図 1 6 】



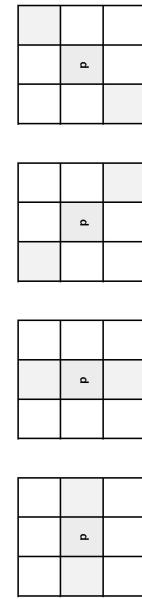
40



【図 17】

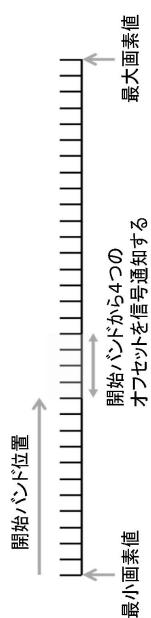


【図 18】

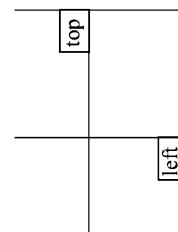


10

【図 19】



【図 20】

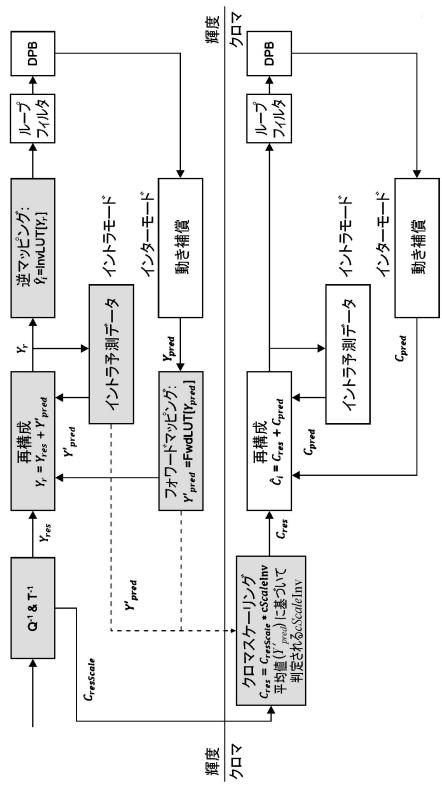


30

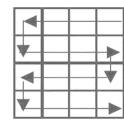
40

50

【図 2 1】



【図 2 2】



10

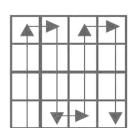
20

30

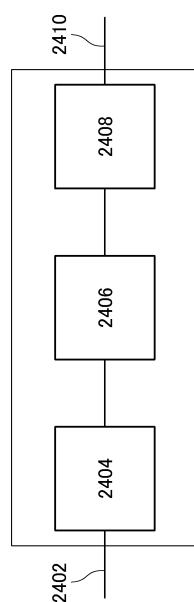
40

50

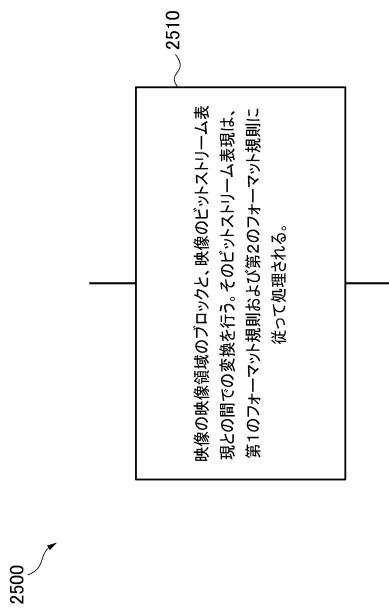
【図 2 3】



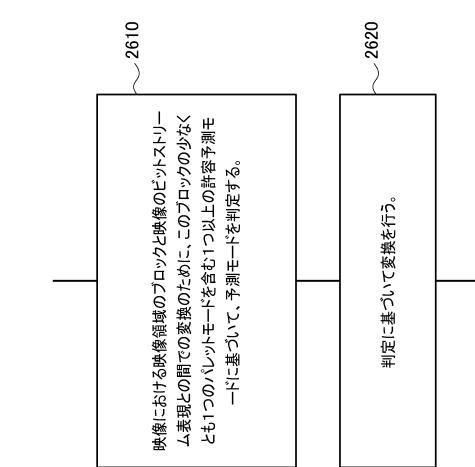
【図 2 4】



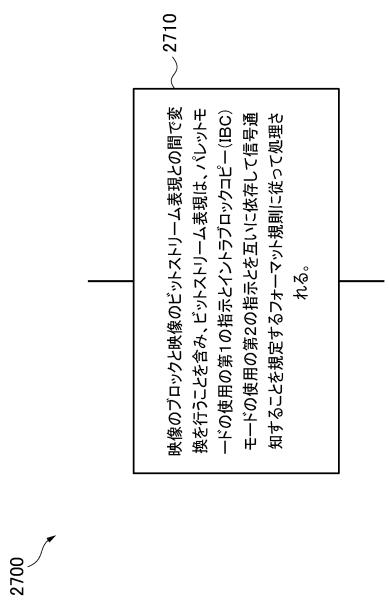
【図 2 5】



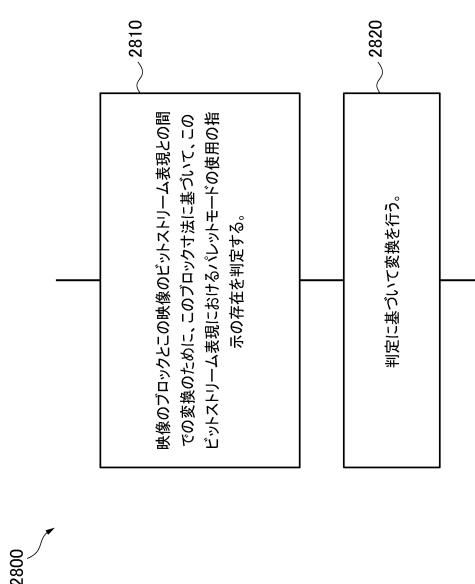
【図 2 6】



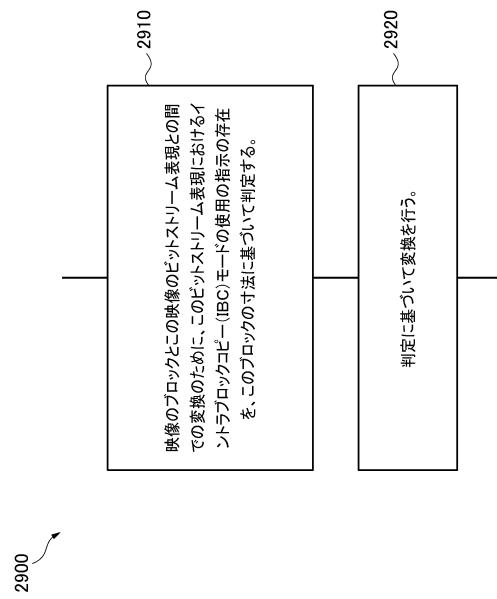
【図 2 7】



【図 2 8】

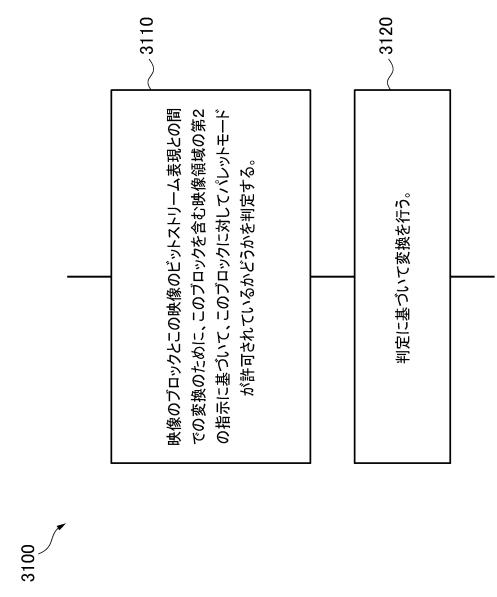


【図 29】



10

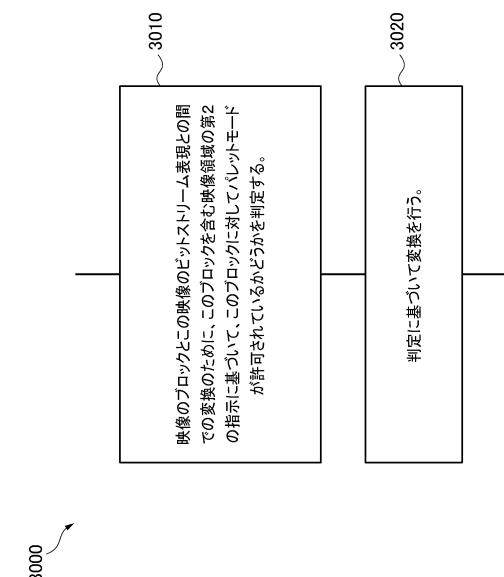
【図 31】



30

40

【図 30】



50

フロントページの続き

B Y T E D A N C E I N C .

アメリカ合衆国 カリフォルニア州 90066 , ロサンゼルス , ウエスト ジェファーソン ブールバード 12655 , シックスス フロア , 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 ジュ ウェイジャ

アメリカ合衆国 90066 カリフォルニア州 , ロサンゼルス , ウエスト ジェファーソン ブールバード 12655 , シックスス 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 , ウエスト ジェファーソン ブールバード 12655 , シックスス フロア , スイート ナンバー137

(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 , ウエスト ジェファーソン ブールバード 12655 , シックスス 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 , ウエスト ジェファーソン ブールバード 12655 , シックスス 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 , デーチュン 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 , デーチュン 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 岩井 健二

(56)参考文献 国際公開第2016/115343 (WO , A2)

Rajan Joshi, et al. , High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 6 , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1 /SC 29/WG 11 , JCTVC-W1005-v4 , 23rd Meeting: San Diego, USA , 2016年06月 , pp.52-53,62-64,106-108,113-115

Roman Chernyak, Sergey Ikonin, and Jianle Chen , CE15-related: Separate Palette Coding for Luma and Chroma components , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-L0427-v1 , 12th Meeting: Macao, CN , 2018年10月 , pp.1-4

Yu-Chen Sun, et al. , CE8: Separate palette coding for luma and chroma (Test8.2.5) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-M0052v2 , 13th Meeting: Marrakech, MA , 2019年01月 , pp.1-6

Weijia Zhu, et al. , CE8-related: Palette Mode Coding , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-N0258-v1 , 14th Meeting: Geneva, CH , 2019年03月 , pp.1-6

(58)調査した分野 (Int.Cl. , DB名)

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