

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7680119号
(P7680119)

(45)発行日 令和7年5月20日(2025.5.20)

(24)登録日 令和7年5月12日(2025.5.12)

(51)国際特許分類		F I	
H O 4 N	19/11 (2014.01)	H O 4 N	19/11
H O 4 N	19/159 (2014.01)	H O 4 N	19/159
H O 4 N	19/186 (2014.01)	H O 4 N	19/186
H O 4 N	19/176 (2014.01)	H O 4 N	19/176
H O 4 N	19/593 (2014.01)	H O 4 N	19/593
請求項の数 18 外国語出願 (全99頁)			
(21)出願番号	特願2024-128(P2024-128)	(73)特許権者	520476341
(22)出願日	令和6年1月4日(2024.1.4)		北京字節跳動網絡技術有限公司
(62)分割の表示	特願2022-506683(P2022-506683) の分割		Beijing Bytedance Network Technology Co., Ltd.
原出願日	令和2年8月3日(2020.8.3)		中華人民共和國100041北京市石景山区実興大街30号院3号楼2層B-0035房間
(65)公開番号	特開2024-41836(P2024-41836A)		Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan District Beijing 100041 China
(43)公開日	令和6年3月27日(2024.3.27)		
審査請求日	令和6年1月31日(2024.1.31)	(73)特許権者	520477474
(31)優先権主張番号	PCT/CN2019/099158		バイトダンス インコーポレイテッド
(32)優先日	令和1年8月3日(2019.8.3)		最終頁に続く
(33)優先権主張国・地域又は機関	中国(CN)		

(54)【発明の名称】 ビデオ・データ処理方法、装置、及び記憶媒体

(57)【特許請求の範囲】

【請求項1】

ビデオ・データを処理する方法であって：

ビデオのビデオ・ユニットの第1のクロマ・ブロックと前記ビデオのビットストリームとの間の第1の変換の際に、前記第1のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第1のルマ・ブロックのコーディング・モードが第1のコーディング・モードである場合に、前記第1のクロマ・ブロックに対するクロマ・イントラ予測モードの導出プロセスにおいて、デフォルト・イントラ予測モードが使用されることを決定するステップ；

前記デフォルト・イントラ予測モードに基づいて、前記第1のクロマ・ブロックに対する前記クロマ・イントラ予測モードを導出するステップ；及び

前記クロマ・イントラ予測モードに基づいて前記第1の変換を実行するステップ；

を含み、当該方法は、更に：

前記ビデオの第2のクロマ・ブロックと前記ビットストリームとの間の第2の変換の際に、前記第2のクロマ・ブロックはクロス・コンポーネント線型モデル・イントラ予測モードでコーディングされるという第1判定を行うステップ；

前記第2のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第2のルマ・ブロックのコーディング・モードが第2のコーディング・モードである場合に、前記第2のクロマ・ブロックに対する二次変換プロセスにおいて所定のイントラ予測モードが使用されるという第2判定を行うステップ；

前記所定のイントラ予測モードに基づいて、前記二次変換プロセスに対する変換セットを決定するステップ；及び

前記変換セットに基づいて前記第2の変換を実行するステップ；
を含み、前記二次変換プロセスの際に：

符号化の場合に、量子化の前に、前記第2のクロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の場合に、インバース一次変換を適用する前に、前記第2のクロマ・ブロックの逆量子化の出力にインバース二次変換が適用される、方法。

【請求項2】

請求項1に記載の方法において、前記第1のルマ・ブロックの第1のコーディング・モードが、行列ベクトル乗算を使用して前記第1のルマ・ブロックの予測サンプルを生成するマトリクス・ベースのイントラ予測（MIP）モードである場合に、前記デフォルト・イントラ予測モードは、平面イントラ予測モードとして決定される、方法。

10

【請求項3】

請求項1に記載の方法において、前記第1のルマ・ブロックの第1のコーディング・モードが、ブロック・ベクトルにより決定されるものと同じビデオ領域のサンプル値のブロックから導出される、前記第1のルマ・ブロックの予測サンプルを生成するイントラ・ブロック・コピー（IBC）モードである場合に、前記デフォルト・イントラ予測モードは、DCイントラ予測モードとして決定される、方法。

【請求項4】

20

請求項1に記載の方法において、前記第1のルマ・ブロックの第1のコーディング・モードが、代表サンプル値のパレットを使用して前記第1のルマ・ブロックを表現又は再構成することを可能にするパレット・モードである場合に、前記デフォルト・イントラ予測モードは、DCイントラ予測モードとして決定される、方法。

【請求項5】

請求項1に記載の方法において、前記第1のルマ・ブロックのコーディング・モードが、マトリクス・ベースのイントラ予測（MIP）モードではなく、イントラ・ブロック・コピー（IBC）モードでもなく、パレット・モードでもない場合に、前記第1のルマ・ブロックのルマ・イントラ予測モードは、前記第1のクロマ・ブロックに対する前記クロマ・イントラ予測モードの導出プロセスにおいて使用される、方法。

30

【請求項6】

請求項1に記載の方法において、前記クロマ・イントラ予測モードは、クロス・コンポーネント線型モデル予測モードが前記第1のクロマ・ブロックに適用されるかどうかを示す第1のフラグ（cclm_mode_flag）、
INTRA_LT_CCLM、INTRA_L_CCLM及びINTRA_T_CCLMクロマ・イントラ予測モードのうちの何れが、前記第1のクロマ・ブロックに適用されるかを示す第2のフラグ（cclm_mode_idx）、

前記第1のクロマ・ブロックのクロマ・サンプルに対するイントラ予測モードを示す第3のフラグ（intra_chroma_pred_mode）、及び

lumaIntraPredModeとして示される変数であって、前記第1のルマ・ブロックが前記第1のコーディング・モードでコーディングされる場合には前記デフォルト・イントラ予測モードに等しく、或いは、前記第1のルマ・ブロックが前記第1のコーディング・モードでコーディングされない場合には前記第1のルマ・ブロックのルマ・イントラ予測モードに等しい変数

40

を用いて導出される、方法。

【請求項7】

請求項6に記載の方法において、前記クロマ・イントラ予測モードは、cclm_mode_flag、cclm_mode_idx、intra_chroma_pred_mode及びlumaIntraPredModeを用いて、以下のテーブル1：

テーブル1

50

【表 1】

cclm_mode_flag	cclm_mode_idx	intra_chroma_pred_mode	lumaIntraPredMode				
			0	50	18	1	X (0 <= X <= 66)
0	-	0	66	0	0	0	0
0	-	1	50	66	50	50	50
0	-	2	18	18	66	18	18
0	-	3	1	1	1	66	1
0	-	4	0	50	18	1	X
1	0	-	81	81	81	81	81
1	1	-	82	82	82	82	82
1	2	-	83	83	83	83	83

に示されるようにして導出される、方法。

【請求項 8】

請求項7に記載の方法において、前記ビデオ・ユニットのクロマ・フォーマットが4:2:2である場合に、テーブル1におけるクロマ・イントラ予測モードXは、所定のマッピング・テーブルに従ってクロマ・イントラ予測モードYにマッピングされ、前記テーブル1におけるクロマ・イントラ予測モードXは、以後、前記クロマ・イントラ予測モードYに等しく設定される、方法。

【請求項 9】

請求項1に記載の方法において、前記デフォルト・イントラ予測モードは、インター・コーディングされるブロックには指定されない、方法。

【請求項 10】

請求項1に記載の方法において、前記変換セットは所定の変換セット選択テーブルから選択され、IntraPredModeとして示される前記変換セット選択テーブルに対するインデックスは、前記第2のルマ・ブロックが前記第2のコーディング・モードでコーディングされる場合には所定のイントラ予測モードに等しく、或いは、前記第2のルマ・ブロックが前記第2のコーディング・モードでコーディングされない場合には前記第2のルマ・ブロックのルマ・イントラ予測モードに等しい、方法。

【請求項 11】

請求項10に記載の方法において、前記所定の変換セット選択テーブルは：
所定の変換セット選択テーブル

【表 2】

IntraPredMode	transform set index
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode	1

であり、IntraPredModeは [-14,80] のレンジを有する、方法。

【請求項 12】

請求項1に記載の方法において、前記二次変換が適用される前記第2のクロマ・ブロックの領域内に、最後の非ゼロ係数が位置していない場合、前記二次変換は、前記第2のクロ

マ・ブロックに対してディセーブルにされる、方法。

【請求項 13】

請求項1に記載の方法において、前記二次変換は、低周波ノン・セパラブル変換（LFNST）ツールに対応している、方法。

【請求項 14】

請求項1-13のうちの何れか一項に記載の方法において、前記第1の変換は、前記第1のクロマ・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み、前記第2の変換は、前記第2のクロマ・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、方法。

【請求項 15】

請求項1-13のうちの何れか一項に記載の方法において、前記第1の変換は、前記第1のクロマ・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み、前記第2の変換は、前記第2のクロマ・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、方法。

【請求項 16】

プロセッサと命令を有する非一時的なメモリとを含むビデオ・データを処理する装置であって、命令は、プロセッサにより実行されると、前記プロセッサに：

ビデオのビデオ・ユニットの第1のクロマ・ブロックと前記ビデオのビットストリームとの間の第1の変換の際に、前記第1のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第1のルマ・ブロックのコーディング・モードが第1のコーディング・モードである場合に、前記第1のクロマ・ブロックに対するクロマ・イントラ予測モードの導出プロセスにおいて、デフォルト・イントラ予測モードが使用されることを決定するステップ；

前記デフォルト・イントラ予測モードに基づいて、前記第1のクロマ・ブロックに対する前記クロマ・イントラ予測モードを導出するステップ；及び

前記クロマ・イントラ予測モードに基づいて前記第1の変換を実行するステップ；

を実行させ、前記命令は、前記プロセッサにより実行されると、更に、前記プロセッサに：

前記ビデオの第2のクロマ・ブロックと前記ビットストリームとの間の第2の変換の際に、前記第2のクロマ・ブロックはクロス・コンポーネント線型モデル・イントラ予測モードでコーディングされるという第1判定を行うステップ；

前記第2のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第2のルマ・ブロックのコーディング・モードが第2のコーディング・モードである場合に、前記第2のクロマ・ブロックに対する二次変換プロセスにおいて所定のイントラ予測モードが使用されるという第2判定を行うステップ；

前記所定のイントラ予測モードに基づいて、前記二次変換プロセスに対する変換セットを決定するステップ；及び

前記変換セットに基づいて前記第2の変換を実行するステップ；

を実行させ、前記二次変換プロセスの際に：

符号化の場合に、量子化の前に、前記第2のクロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の場合に、インバース一次変換を適用する前に、前記第2のクロマ・ブロックの逆量子化の出力にインバース二次変換が適用される、装置。

【請求項 17】

命令を記憶する非一時的なコンピュータ読み取り可能な記憶媒体であって、命令は、プロセッサに：

ビデオのビデオ・ユニットの第1のクロマ・ブロックと前記ビデオのビットストリームとの間の第1の変換の際に、前記第1のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第1のルマ・ブロックのコーディング・モードが第1のコーディング・モードである場合に、前記第1のクロマ・ブロックに対するクロマ・イントラ予測モードの導出プロセスにおいて、デフォルト・イントラ予測モードが使用されることを決定するステップ；

前記デフォルト・イントラ予測モードに基づいて、前記第1のクロマ・ブロックに対する前記クロマ・イントラ予測モードを導出するステップ；及び

前記クロマ・イントラ予測モードに基づいて前記第1の変換を実行するステップ；
を実行させ、前記命令は、更に、前記プロセッサに：

前記ビデオの第2のクロマ・ブロックと前記ビットストリームとの間の第2の変換の際に、前記第2のクロマ・ブロックはクロス・コンポーネント線型モデル・イントラ予測モードでコーディングされるという第1判定を行うステップ；

前記第2のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第2のルマ・ブロックのコーディング・モードが第2のコーディング・モードである場合に、前記第2のクロマ・ブロックに対する二次変換プロセスにおいて所定のイントラ予測モードが使用されるという第2判定を行うステップ；

前記所定のイントラ予測モードに基づいて、前記二次変換プロセスに対する変換セットを決定するステップ；及び

前記変換セットに基づいて前記第2の変換を実行するステップ；
を実行させ、前記二次変換プロセスの際に：

符号化の場合に、量子化の前に、前記第2のクロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の場合に、インバース一次変換を適用する前に、前記第2のクロマ・ブロックの逆量子化の出力にインバース二次変換が適用される、記憶媒体。

【請求項 18】

ビデオのビットストリームを記憶する方法であって：

ビデオのビデオ・ユニットの第1のクロマ・ブロックに関し、前記第1のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第1のルマ・ブロックのコーディング・モードが第1のコーディング・モードである場合に、前記第1のクロマ・ブロックに対するクロマ・イントラ予測モードの導出プロセスにおいて、デフォルト・イントラ予測モードが使用されることを決定するステップ；

前記デフォルト・イントラ予測モードに基づいて、前記第1のクロマ・ブロックに対する前記クロマ・イントラ予測モードを導出するステップ；及び

前記クロマ・イントラ予測モードに基づいて前記ビットストリームを生成するステップ；及び

前記ビットストリームを非一時的なコンピュータ読み取り可能な記憶媒体に記憶するステップ；

を含み、前記方法は、更に：

前記ビデオの第2のクロマ・ブロックはクロス・コンポーネント線型モデル・イントラ予測モードでコーディングされるという第1判定を行うステップ；

前記第2のクロマ・ブロックの中央におけるクロマ・サンプルに対応する位置にあるルマ・サンプルをカバーする第2のルマ・ブロックのコーディング・モードが第2のコーディング・モードである場合に、前記第2のクロマ・ブロックに対する二次変換プロセスにおいて所定のイントラ予測モードが使用されるという第2判定を行うステップ；

前記所定のイントラ予測モードに基づいて、前記二次変換プロセスに対する変換セットを決定するステップ；及び

前記変換セットに基づいて前記ビットストリームを生成するステップ；

を含み、前記二次変換プロセスの際に：

符号化を行うために、量子化の前に、前記第2のクロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用される、方法。

【発明の詳細な説明】

【背景技術】

【0001】

関連出願の相互参照

パリ条約に関して適用可能な特許法及び／又は規則に基づいて、本願は2019年8月3日

10

20

30

40

50

付で出願された国際特許出願第PCT/CN2019/099158号に対する優先権及び利益を主張するように適時に行われている。法律に基づく全ての目的に関し、前述の出願の開示全体は、本願の開示の一部として参照により援用される。

【0002】

技術分野

本特許文献はビデオ処理技術、デバイス、及びシステムに関連する。

【0003】

背景

ビデオ圧縮における進歩にもかかわらず、デジタル・ビデオは、インターネットその他のデジタル通信ネットワークにおいて最大の帯域幅を依然として占めている。ビデオを受信及び表示することが可能な接続ユーザー・デバイスの台数が増加するにつれて、デジタル・ビデオの利用に対する帯域幅の需要は増加し続けるであろうということが予想される。

【発明の概要】

【0004】

デバイス、システム、及び方法は、デジタル・ビデオ・コーディング、具体的にはビデオ・コーディングにおける残差コーディングのためのコンテキスト・モデリングに関連している。説明される方法は、既存のビデオ・コーディング規格（例えば、高効率ビデオ・コーディング（High Efficiency Video Coding, HEVC）及び将来のビデオ・コーディング規格又はビデオ・コーデックの両方に適用される可能性がある。

【0005】

代表的な一態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオのビデオ・ユニットの現在のビデオ・ブロックと、ビデオのコーディングされた表現との間の変換に関し、特定のイントラ予測モードを使用してコーディングされたビデオ・ユニットのデフォルト・イントラ予測モードを決定し、その結果、現在のビデオ・ブロックの予測ブロックが、ある方向に沿って現在のビデオ・ブロックの隣接するピクセルを外挿することなく生成されるステップと、その決定に基づいて変換を実行するステップとを含む。

【0006】

別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ルールを使用して、ビデオのクロマ・ブロックの所定の位置をカバーするビデオのルマ・ブロックについての決定を行うステップと、決定に基づいて、ビデオとビデオのコーディングされた表現との間の変換を実行するステップとを含み、クロマ・ブロックは、イントラ予測モードを使用して、コーディングされた表現で表現される。

【0007】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ルールを使用して、ビデオのクロマ・ブロックのイントラ予測モードを、クロマ・ブロックに対応するルマ・ブロックのコーディング・モードに基づいて導出するステップと、導出されたイントラ予測モードに基づいて、クロマ・ブロックとビデオのコーディングされた表現との間の変換を実行するステップとを含み、ルールは、ルマ・ブロックのコーディング・モードが、ある方向に沿ってルマ・ブロックの隣接するピクセルを外挿することなく、ルマ・ブロックの予測ブロックが生成される特定のイントラ予測モードである場合に、デフォルトのイントラ予測モードを使用することを規定している。

【0008】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオのクロマ・ブロックに関し、非ノーマル・クロマ・イントラ予測モードがビデオのクロマ・ブロックに適用されるかどうかの第1判定を行うステップと、クロマ・ブロックに対応するルマ・ブロックに関し、ルマ・イントラ予測モードがルマ・ブロックに適用される第2判定を行うステップと、ルマ・イント

ラ予測モードに基づいてクロマ・ブロックに変換セット又は変換行列が適用される第3判定を行うステップと、第3判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行するステップとを含み、非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む。

【0009】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオのクロマ・ブロックに関し、クロマ・ブロックに対応するルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用してコーディングされる第1判定を行うステップと、

第1判定に基づいて、ルールに従ってクロマ・ブロックに対する変換セット又は変換行列の第2判定を行うステップと、第2判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行するステップとを含み、ルールは、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用してコーディングされることに起因して、クロマ・ブロックに関連する1つ以上のデフォルト・モード又はデフォルト変換セットは、クロマ・ブロックが非ノーマル・クロマ・イントラ予測モードを使用してコーディングされる場合に、変換セット又は変換行列を決定することを規定しており、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックをコーディングすることを含み、非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む。

【0010】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオの現在のビデオ・ブロックとビデオのコーディングされた表現との間の変換に関し、1つの色成分の現在のビデオ・ブロックに適用される第2変換ツールの適用可能性を、1) 別の色成分の対応するブロックのコーディング・モード、又は2) 現在のビデオ・ブロックのコーディング・モードのうちの少なくとも1つに基づいて決定するステップと、決定に基づいて変換を実行するステップとを含み、二次変換ツールを使用して：符号化の間に、量子化の前に、現在のビデオ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は復号化の間に、インバース一次変換を適用する前に、現在のビデオ・ブロックの逆量子化の出力にインバース二次変換が適用される。

【0011】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定された位置をカバーしているルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うステップと、第1判定に基づいて、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用して符号化されることに起因して、予め決定されたイントラ予測モードをクロマ・ブロックに適用する第2判定を行うステップと、第2判定に従って、ビデオとビデオのコーディングされた表現との変換を実行するステップとを含み、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックを符号化することを含む。

【0012】

更に別の代表的な態様において、開示される技術は、ビデオ処理方法を提供するために使用することが可能である。本方法は、ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定された位置をカバーしているルマ・ブロックが、ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うステップと、第1判定に基づいて、ルマ・ブロックのノーマル・ルマ・イントラ予測モードに基づいて、クロマ・イントラ予測モードを導出する第2判定を行うステップと、第2判定に従って、ビデオとビデオのコーデ

イングされた表現との変換を実行するステップとを含み、ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用して、ルマ・ブロックを符号化することを含む。

【0013】

更に別の代表的な一態様において、上記の方法は、プロセッサで実行することが可能なコードの形態で具現化され、コンピュータ読み取り可能なプログラム媒体に記憶される。

【0014】

更に別の代表的な一態様において、上記の方法を実行するように構成された又は動作することが可能なデバイスが開示される。デバイスは、この方法を実装するようにプログラムされたプロセッサを含んでもよい。

【0015】

更に別の代表的な一態様において、ビデオ・デコーダ装置は本願で説明される方法を実装することができる。

【0016】

開示された技術の上記及びその他の態様及び特徴は、図面、明細書及び特許請求の範囲において、より詳細に記載されている。

【図面の簡単な説明】

【0017】

【図1】例示的なエンコーダのブロック図を示す。

【0018】

【図2】67個のイントラ予測モードの例を示す。

【0019】

【図3】4×4ブロックに対するALWIPの例を示す。

【0020】

【図4】8×8ブロックに対するALWIPの例を示す。

【0021】

【図5】8×4ブロックに対するALWIPの例を示す。

【0022】

【図6】16×16ブロックに対するALWIPの例を示す。

【0023】

【図7】予測ブロックに隣接する4つの参照ラインの例を示す。

【0024】

【図8】4×8及び8×4ブロックの分割例を示す。

【0025】

【図9】4×8、8×4、4×4を除く全てのブロックの分割例を示す。

【0026】

【図10】JEMにおける二次変換の例を示す。

【0027】

【図11】提案される縮小二次変換 (reduced secondary transform, RST) の例を示す。

【0028】

【図12】フォワード及びインバース縮小変換の例を示す。

【0029】

【図13】16×48行列を用いたフォワードRST 8×8プロセスの例を示す。

【0030】

【図14】8×8行列に対するゼロ・アウト領域の例を示す。

【0031】

【図15】サブ・ブロック変換モードSBT-V及びSBT-Hの例を示す。

【0032】

【図16】4×4コーディング・グループに対する対角線右上向きスキャン順序の例を示す

10

20

30

40

50

。

【0033】

【図17】サイズ4×4のコーディング・グループによる8×8ブロックに対する対角線右上向きスキャン順序の例を示す。

【0034】

【図18】確率モデルを選択するために使用されるテンプレートの例を示す。

【0035】

【図19】依存量子化 (dependent quantization) に使用される2つのスカラー量子化器の例を示す。

【0036】

【図20】提案される依存量子化プロセスに対する状態遷移と量子化器選択の例を示す。

【0037】

【図21】4つのコーディング・グループを有する8×8ブロックの例である。

【0038】

【図22A】ビデオ処理の例示的な方法のフローチャートを示す。

【図22B】ビデオ処理の例示的な方法のフローチャートを示す。

【図22C】ビデオ処理の例示的な方法のフローチャートを示す。

【図22D】ビデオ処理の例示的な方法のフローチャートを示す。

【図22E】ビデオ処理の例示的な方法のフローチャートを示す。

【図22F】ビデオ処理の例示的な方法のフローチャートを示す。

【0039】

【図23A】本件で説明されるビジュアル・メディア復号化又はビジュアル・メディア符号化技術を実装するためのハードウェア・プラットフォーム例のブロック図である。

【図23B】本件で説明されるビジュアル・メディア復号化又はビジュアル・メディア符号化技術を実装するためのハードウェア・プラットフォーム例のブロック図である。

【0040】

【図24A】デュアル・ツリー・パーティションの下での1つのクロマ・ブロックと、それに対応するルマ・ブロックのうちの前者の例を示す。

【図24B】デュアル・ツリー・パーティションの下での1つのクロマ・ブロックと、それに対応するルマ・ブロックの後者の例を示す。

【0041】

【図25】対応するルマ・ブロックからのDM導出のための‘CR’位置の例を示す。

【0042】

【図26A】ビデオ処理の例示的な方法のフローチャートを示す。

【図26B】ビデオ処理の例示的な方法のフローチャートを示す。

【図26C】ビデオ処理の例示的な方法のフローチャートを示す。

【0043】

【図27A】ビデオ処理の例示的な方法のフローチャートを示す。

【図27B】ビデオ処理の例示的な方法のフローチャートを示す。

【図27C】ビデオ処理の例示的な方法のフローチャートを示す。

【図27D】ビデオ処理の例示的な方法のフローチャートを示す。

【図27E】ビデオ処理の例示的な方法のフローチャートを示す。

【発明を実施するための形態】

【0044】

開示される技術の実施形態は、圧縮パフォーマンスを改善するために、既存のビデオ・コーディング規格（例えば、HEVC, H.265）及び将来の規格に適用される可能性がある。セクション見出しは、本明細書においては説明の可読性を向上させるために使用されており、如何なる方法によっても、本説明又は実施形態（及び／又は実装）を個々のセクションだけに限定してはいない。

【0045】

10

20

30

40

50

2 ビデオ・コーディング・イントロダクション

より高い解像度のビデオの益々増える要請に起因して、ビデオ・コーディング方法及び技術は現代技術の至る所に存在する。ビデオ・コーデックは、典型的には、デジタル・ビデオを圧縮又は解凍する電子回路又はソフトウェアを含み、より高いコーディング効率を提供するために絶えず改良されている。ビデオ・コーデックは、圧縮されていないビデオを圧縮されたフォーマットへ、又は逆向きに変換する。ビデオ品質と、ビデオを表現するために使用されるデータ量（ビット・レートによって決定される）と、符号化及び復号化アルゴリズムの複雑さと、データ損失及びエラーに対する感度と、編集の容易さと、ランダム・アクセスと、エンド・ツー・エンド遅延（レイテンシ）との間には複雑な関係が存在する。圧縮されたフォーマットは、通常、標準的なビデオ圧縮規格、例えば、高効率ビデオ・コーディング（HEVC）規格（H.265又はMPEG-H Part 2としても知られている）、ファイナライズされる予定の汎用ビデオ・コーディング（Versatile Video Coding, VVC）規格、又は他の現在及び／又は将来のビデオ・コーディング規格に準拠している。

【0046】

ビデオ・コーディング規格は、周知のITU-T及びISO/IEC規格の開発を通じて主に発展してきた。ITU-TはH.261とH.263を作成し、ISO/IECはMPEG-1とMPEG-4 Visualを作成し、その2つの組織は共同してH.262/MPEG-2ビデオとH.264/MPEG-4アドバンスド・ビデオ・コーディング（AVC）とH.265/HEVC規格とを作成した。H.262以来、ビデオ・コーディング規格はハイブリッド・ビデオ・コーディング構造に基づいており、ここでは時間的予測と変換コーディングが使用される。HEVCを越える将来のビデオ・コーディング技術を探求するため、2015年に共同ビデオ探索チーム（Joint Video Exploration Team, JVET）がVCEGとMPEGにより共同で設立された。それ以来、多くの新しい方法がJVETによって採用されており、共同探索モデル（Joint Exploration Model, JEM）[3][4]と名付けられる参照ソフトウェアに組み込まれている。2018年4月には、VCEG（Q6/16）とISO/IEC JTC1 SC29/WG11（MPEG）の間で共同ビデオ・エキスパート・チーム（JVET）が発足し、HEVCと比較して50%のビットレート低減を目指すVVC規格に取り組んだ。

【0047】

2.1 典型的なビデオ・コーデックのコーディング・フロー

図1は、VVCのエンコーダ・ブロック図の例を示し、3つのループ内フィルタリング・ブロック、即ち：デブロッキング・フィルタ（DF）、サンプル適応オフセット（SAO）、及びALFを含む。予め定義されたフィルタを使用するDFとは異なり、SAO及びALFはそれぞれ、現在のピクチャのオリジナル・サンプルを使用して、オリジナル・サンプルと再構成されたサンプルとの間の平均二乗誤差を、オフセットを追加することによって、及び有限インパルス応答（FIR）フィルタを適用することによって減少させ、コーディングされたサイド情報は、オフセットとフィルタ係数をシグナリングする。ALFは、各ピクチャの最終処理ステージに位置し、以前のステージで生成されたアーチファクトを捕捉して修復しようとするツールと見なすことができる。

【0048】

2.2 VVCにおけるイントラ・コーディング

2.2.1 67個のイントラ予測モードによるイントラ・モード・コーディング

自然なビデオで提示される任意のエッジ方向を捕らえるために、方向イントラ・モードの数は、HEVCで使用されているような33個から、65個へ拡張される。追加の方向モードは、図2において点線の矢印として描かれており、平面モード及びDCモードは同じままである。これらのより高密度な方向性イントラ予測モードは、全てのブロック・サイズに対して、そしてルマ及びクロマ・イントラ予測の両方に対して適用される。

【0049】

従来の角度イントラ予測方向は、図2に示すように時計回り方向に45度から-135度までに対して定義される。VTM2では、幾つかの従来の角度イントラ予測モードは、非正方ブ

10

20

30

40

50

ロックに対する広角イントラ予測モードで適応的に置換される。置換されるモードは、オリジナルの方法を用いてシグナリングされ、解析後に広角モードのインデックスに再マッピングされる。イントラ予測モードの総数は変化せず、即ち67であり、イントラ・モード・コーディングは変わらない。

【0050】

HEVCでは、イントラ・コーディングされるブロックは全て正方形であり、各辺の長さは2のべき乗である。従って、DCモードを使用してイントラ予測子を生成するために分割処理は必要とされない。VVV2では、ブロックは長方形の形状を有する可能性があり、一般的なケースではブロックごとに分割処理を使用する必要がある。DC予測のための分割処理を回避するために、長辺のみが、非正方形ブロックに対する平均を計算するために使用

10

【0051】

67個のイントラ予測モードに加えて、非正方形ブロック（WAIP）及びポジション依存イントラ予測結合（PDPC）法の広角イントラ予測が、特定のブロックに対して更にインネーブルにされる。PDPCは、以下のイントラ・モード：平面、DC、水平、垂直、左下の角度モードとその8つの隣接する角度モード、及び右上の角度モードとその8つの隣接する角度モードに対して、シグナリングなしに適用される。

【0052】

2.2.2 アフィン線型加重イントラ予測（ALWIP又はマトリクス・ベースのイントラ予測）

アフィン線型加重イントラ予測（ALWIP，マトリクス・ベースのイントラ予測（Matrix based intra prediction, MIP）としても知られている）はJVET-N0217で提案されている。

20

【0053】

2.2.2.1 行列ベクトル乗算による縮小予測信号の生成

まず、隣接する参照サンプルは平均化によってダウン・サンプリングされ、縮小参照信号 $bdry_{red}$ を生成する。次いで、行列ベクトル積を計算してオフセットを加えることによって、縮小予測信号 $pred_{red}$ が計算される：

【0054】

【数1】

$$pred_{red} = A \cdot bdry_{red} + b$$

30

【0055】

ここで、 A は、 $W=H=4$ である場合は4列、他の全ての場合は8列と、 $W_{red} \cdot H_{red}$ 行とを有する行列である。 b はサイズが $W_{red} \cdot H_{red}$ であるベクトルである。

【0056】

2.2.2.2 ALWIPプロセス全体の説明

平均化、行列ベクトル乗算、及び線型補間の全体的なプロセスが、図3-6で様々な形状に関して示されている。残りの形状は、図示のケースのうちの何れかと同様に扱われることに留意されたい。

40

【0057】

1. 4×4 ブロックであるとする、ALWIPは境界の各軸に沿って2つの平均をとる。結果の4つの入力サンプルは、行列ベクトル乗算に入る。行列は集合 S_0 から取られる。オフセットを加えた後、これは16個の最終予測サンプルを生じる。線型補間は予測信号を生成するためには不要である。従って、サンプル当たり合計 $(4 \cdot 16) / (4 \cdot 4) = 4$ 回の乗算が実行される。

【0058】

2. 8×8 ブロックであるとする、ALWIPは境界の各軸に沿って4つの平均をとる。結果の8つの入力サンプルは、行列ベクトル乗算に入る。行列は集合 S_1 から取られる。これは予測ブロックの奇数位置における16個のサンプルを生じる。従って、サンプル当たり合

50

計 $(8 \cdot 16) / (8 \cdot 8) = 2$ 回の乗算が実行される。オフセットを加えた後、これらのサンプルは、縮小された上・境界を使用することによって、垂直方向に補間される。元の左・境界を使用することによって、水平補間が続く。

【0059】

3. 8×4 ブロックであるとする、ALWIPは境界の水平軸に沿って4つの平均と、左境界における4つの元の境界値をとる。結果の8つの入力サンプルは、行列ベクトル乗算に入る。行列は集合 S_1 から取られる。これは予測ブロックの水平方向の奇数位置と垂直方向の各位置とで16個のサンプルを生じる。従って、サンプル当たり合計 $(8 \cdot 16) / (8 \cdot 4) = 4$ 回の乗算が実行される。オフセットを加えた後、これらのサンプルは、元の左・境界を使用することによって、水平方向に補間される。

10

【0060】

4. 16×16 ブロックであるとする、ALWIPは境界の各軸に沿って4つの平均をとる。結果の8つの入力サンプルは、行列ベクトル乗算に入る。行列は集合 S_2 から取られる。これは予測ブロックの奇数位置で64個のサンプルを生じる。従って、サンプル当たり合計 $(8 \cdot 64) / (16 \cdot 16) = 2$ 回の乗算が実行される。オフセットを加えた後、これらのサンプルは、上・境界の8つの平均を使用することによって、垂直方向に補間される。元の左・境界を使用することによって、水平補間が続く。この場合、補間プロセスは、如何なる乗算も追加しない。従って、ALWIP予測を計算するためには、全体として、サンプル当たり2回の乗算が必要とされる。

【0061】

20

より大きな形状についても、手順は本質的には同じであり、サンプル当たりの乗算回数は4未満であることを確認することは容易である。

【0062】

$W \times 8$ ブロックの場合 ($W > 8$)、水平補間のみ必要となり、なぜならサンプルは奇数の水平位置であって各垂直位置で与えられているからである。

【0063】

最後に、 $W \times 4$ ブロックの場合 ($W > 8$)、 A_kbe を、ダウン・サンプリングしたブロックの水平軸に沿った奇数エントリに対応する全ての行を除外することによって生じる行列とする。従って、出力サイズは32であり、再び、水平補間のみが実行されるように残る。

【0064】

30

転置されるケースは相応に取り扱われる。

イントラ予測ブロックのMPMリストを作成する場合において、隣接ブロックがMIPモードでコーディングされるならば、MIPモードは、次のテーブルを使用することによってイントラ予測モードにマッピングされるであろう。ここで、隣接するルマ・ブロックの幅、高さをそれぞれ $widthNeig$ 、 $hightNeig$ とすると、 $MipSizeId$ は次のようにして導出される：

- ・ $widthNeig$ と $hightNeig$ が4に等しい場合、 $MipSizeId$ は0に等しく設定される。
- ・ それ以外の場合において、 $widthNeig$ と $hightNeig$ の両方が8以下であるならば、 $MipSizeId$ は1に等しく設定される。
- ・ それ以外の場合、 $MipSizeId$ は2に等しく設定される。

40

【0065】

【数2】

テーブル8-4 - MIPとイントラ予測モードの間のマッピングの仕様

IntraPredModeY[xNbX][yNbX]	MipSizeId		
	0	1	2
0	0	0	1
1	18	1	1
2	18	0	1
3	0	1	1
4	18	0	18
5	0	22	0
6	12	18	1
7	0	18	0
8	18	1	1
9	2	0	50
10	18	1	0
11	12	0	
12	18	1	
13	18	0	
14	1	44	
15	18	0	
16	18	50	
17	0	1	
18	0	0	
19	50		
20	0		
21	50		
22	0		
23	56		
24	0		
25	50		
26	66		
27	50		
28	56		
29	50		
30	50		
31	1		
32	50		
33	50		
34	50		

10

20

2.2.2.3 シンタックス及びセマンティクス

7.3.6.5 コーディング・ユニット・シンタックス

30

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(tile_group_type != I sps_ibc_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)
}	

40

50

if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(abs(Log2(cbWidth) - Log2(cbHeight)) <= 2)	
intra_lwip_flag[x0][y0]	ae(v)
if(intra_lwip_flag[x0][y0]) {	
intra_lwip_mpm_flag[x0][y0]	ae(v)
if(intra_lwip_mpm_flag[x0][y0])	
intra_lwip_mpm_idx[x0][y0]	ae(v)
else	
intra_lwip_mpm_remainder[x0][y0]	ae(v)
} else {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
...	
}	
}	

10

20

30

40

【 0 0 6 6 】

2.2.3 多重参照ライン (MRL)

多重参照ライン (Multiple reference line, MRL) イントラ予測は、イントラ予測のために、より多くの参照ラインを使用する。図7において、4つの参照ラインの例が描かれており、ここで、セグメントA及びFのサンプルは、再構成された隣接するサンプルから取ってくるのではなく、それぞれセグメントB及びEからの最も近いサンプルでパディングされる。HEVCイントラ・ピクチャ予測は、最も近い参照ライン (即ち参照ライン0) を使用している。MRLでは、2つの追加ライン (参照ライン1及び参照ライン3) が使用される。

50

【0067】

選択された参照ラインのインデックス (mrl_idx) は、シグナリングされて、イントラ予測子を生成するために使用される。0より大きな参照ライン・インデックスの場合、追加の参照ライン・モードをMPMリストに含めるだけであり、残りのモードを使用せずにMPMインデックスをシグナリングするだけである。参照ライン・インデックスは、イントラ予測モードの前にシグナリングされ、プレーナ (Planar) 及びDCモードは、ゼロでない参照ライン・インデックスがシグナリングされた場合に、イントラ予測モードから除外される。

【0068】

MRLは、CTU内側のブロックの第1ラインについてはディセーブルにされて、現在のCTUライン外側にある拡張された参照サンプルを使用することを禁止する。また、追加ラインが使用される場合、PDPCはディセーブルにされる。

2.2.4 イントラ・サブ・ブロック・パーティショニング (ISP)

JVET-M0102では、ISPが提案されており、これはテーブル1に示されるように、ルマ・イントラ予測ブロックを、ブロック・サイズのディメンジョンに依存して垂直に又は水平に、2つ又は4つのサブ・パーティションに分割する。図8及び図9は、2つの可能性の例を示す。全てのサブ・パーティションは、少なくとも16サンプルを有するという条件を満たしている。ブロック・サイズが $4 \times N$ 又は $N \times 4$ ($N \geq 8$) である場合、許容される場合には、 $1 \times N$ 又は $N \times 1$ のサブ・パーティションが存在してもよい。

【0069】

【数3】

テーブル1: ブロック・サイズに応じたサブ・パーティションの数 (最大変換サイズ by maxTBSIZE で示される)

分割方向	ブロック・サイズ	サブ・パーティション数
N/A	最小変換サイズ	非分割
4x8: 水平 8x4: 垂直	4×8 及び 8×4	2
シグナリングされる	4×8 でも 8×4 でもなく、 $W \leq \text{maxTBSIZE}$ 及び $H \leq \text{maxTBSIZE}$ である場合	4
Horizontal	上記ケースではなく、 $H > \text{maxTBSIZE}$ である場合	4
Vertical	上記ケースではなく、 $H > \text{maxTBSIZE}$ である場合	4

【0070】

【数4】

10

20

30

40

50

これら各サブ・パーティションについて、残差信号は、エンコーダから送信された係数をエントロピー復号化し、それらをインバート量子化、及びインバート変換することによって生成される。次いで、サブ・パーティションは、イントラ予測され、最終的に、残差信号を予測信号に加えることによって、対応する再構成されたサンプルが取得される。従って、各サブ・パーティションの再構成された値は、次のものの予測を生成すること（プロセスを反復することになるであろう）、等々に利用することが可能である。全てのサブ・パーティションは同じイントラ・モードを共有する。

テーブル 2 : `predModeIntra` に依存する `trTypeHor` 及び `trTypeVer` の仕様

<code>predModeIntra</code>	<code>trTypeHor</code>	<code>trTypeVer</code>
INTRA_PLANAR, INTRA_ANGULAR31, INTRA_ANGULAR32, INTRA_ANGULAR34, INTRA_ANGULAR36, INTRA_ANGULAR37	(<code>nTbW</code> $\geq$ 4 && <code>nTbW</code> $\leq$ 16) ? DST- VII : DCT-II	(<code>nTbH</code> $\geq$ 4 && <code>nTbH</code> $\leq$ 16) ? DST-VII : DCT-II
INTRA_ANGULAR33, INTRA_ANGULAR35	DCT-II	DCT-II
INTRA_ANGULAR2, INTRA_ANGULAR4,...,INTRA_ANGULAR28, INTRA_ANGULAR30, INTRA_ANGULAR39, INTRA_ANGULAR41,...,INTRA_ANGULAR63, INTRA_ANGULAR65	(<code>nTbW</code> $\geq$ 4 && <code>nTbW</code> $\leq$ 16) ? DST- VII : DCT-II	DCT-II
INTRA_ANGULAR3, INTRA_ANGULAR5,..., INTRA_ANGULAR27, INTRA_ANGULAR29, INTRA_ANGULAR38, INTRA_ANGULAR40,...,INTRA_ANGULAR64, INTRA_ANGULAR66	DCT-II	(<code>nTbH</code> $\geq$ 4 && <code>nTbH</code> $\leq$ 16) ? DST-VII : DCT-II

2.2.4.1 シンタックス及びセマンティクス

7.3.7.5 コーディング・ユニット・シンタックス

<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	Descriptor
<code>if(slice_type != I sps_abc_enabled_flag) {</code>	
<code>if(treeType != DUAL_TREE_CHROMA)</code>	
<code>cu_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(cu_skip_flag[x0][y0] == 0 && slice_type != I)</code>	
<code>pred_mode_flag</code>	<code>ae(v)</code>
<code>if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && sps_abc_enabled_flag)</code>	
<code>pred_mode_abc_flag</code>	<code>ae(v)</code>
<code>}</code>	
<code>if(CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code>if(sps_pcm_enabled_flag && cbWidth $\geq$ MinIpcmCbSizeY && cbWidth $\leq$ MaxIpcmCbSizeY && cbHeight $\geq$ MinIpcmCbSizeY && cbHeight $\leq$ MaxIpcmCbSizeY)</code>	
<code>pcm_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(pcm_flag[x0][y0]) {</code>	
<code>while(!byte_aligned())</code>	
<code>pcm_alignment_zero_bit</code>	<code>f(1)</code>
<code>pcm_sample(cbWidth, cbHeight, treeType)</code>	

10

20

30

40

50

<code>} else {</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>if((y0 % CtbSizeY) > 0)</code>	
<code> intra_luma_ref_idx[x0][y0]</code>	<code>ae(v)</code>
<code>if((intra_luma_ref_idx[x0][y0] == 0 &&</code>	
<code> (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) &&</code>	
<code> (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))</code>	
<code> intra_subpartitions_mode_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_subpartitions_mode_flag[x0][y0] == 1 &&</code>	
<code> cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)</code>	
<code> intra_subpartitions_split_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_luma_ref_idx[x0][y0] == 0 &&</code>	
<code> intra_subpartitions_mode_flag[x0][y0] == 0)</code>	
<code> intra_luma_mpm_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_luma_mpm_flag[x0][y0])</code>	
<code> intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
<code>else</code>	
<code> intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)</code>	
<code> intra_chroma_pred_mode[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */</code>	
<code>...</code>	
<code>}</code>	
<code>...</code>	
<code>}</code>	

10

20

intra_subpartitions_mode_flag[x0][y0] (1 に等しいもの) は、現在のイントラ・コーディング・ユニットが、NumIntraSubPartitions[x0][y0]個の矩形変換ブロック・サブパーティションにパーティション化されることを指定する。

intra_subpartitions_mode_flag[x0][y0] (0 に等しいもの) は、現在のイントラ・コーディング・ユニットが、矩形変換ブロック・サブパーティションにパーティション化されないことを指定する。
intra_subpartitions_mode_flag[x0][y0] が存在しない場合、それは 0 に等しいと推定される。

intra_subpartitions_split_flag[x0][y0] は、イントラ・サブ・パーティション・スプリット・タイプが水平であるか又は垂直であるかを指定する。

intra_subpartitions_split_flag[x0][y0] が存在しない場合、それは次のように推定される：

30

- **cbHeight** が **MaxTbSizeY** より大きい場合、**intra_subpartitions_split_flag**[x0][y0] は 0 に等しいと推定される。
- それ以外の場合 (**cbWidth** が **MaxTbSizeY** より大きい場合) 、
intra_subpartitions_split_flag[x0][y0] は 1 に等しいと推定される。

変数 **IntraSubPartitionsSplitType** は、テーブル 7-9 に示すように、現在のルマ・コーディング・ブロックに使用される分割のタイプを指定する。**IntraSubPartitionsSplitType** は次のように導出される：

- **intra_subpartitions_mode_flag**[x0][y0] が 0 に等しい場合、**IntraSubPartitionsSplitType** は 0 に等しく設定される。
- それ以外の場合、**IntraSubPartitionsSplitType** は、1 + **intra_subpartitions_split_flag**[x0][y0] に等しく設定される。

40

テーブル7-9 - IntraSubPartitionsSplitTypeに対する関連名称

IntraSubPartitionsSplitType	Name of IntraSubPartitionsSplitType
0	ISP_NO_SPLIT
1	ISP_HOR_SPLIT
2	ISP_VER_SPLIT

変数 NumIntraSubPartitions は、イントラ・ルマ・コーディング・ブロックが分割される変換ブロック・サブパーティションの数を指定する。NumIntraSubPartitions は次のように導出される：

- IntraSubPartitionsSplitType が ISP_NO_SPLIT に等しい場合、NumIntraSubPartitions は 1 に等しく設定される。
- それ以外の場合、以下のうちの 1 つの条件が真であるならば、NumIntraSubPartitions は 2 に等しく設定される：
 - cbWidth は 4 に等しく且つ cbHeight は 8 に等しい、
 - cbWidth は 8 に等しく且つ cbHeight は 4 に等しい。
- それ以外の場合、NumIntraSubPartitions は 4 に等しく設定される。

【 0 0 7 1 】

【 数 5 】

10

20

30

40

50

2.3 クロマ・イントラ・モード・コーディング

クロマ・イントラ・モード・コーディングの場合、クロス・コンポーネント線型モデル (cross-component linear model, CCLM) がイネーブルにされているか否かに応じて、合計 8 個又は 5 個のイントラ・モードがクロマ・イントラ・モード・コーディングに許容される。これらのモードは、5 つの伝統的なイントラ・モードと 3 つのクロス・コンポーネント線型モデル・モードを含む。クロマ DM モードは、対応するルマ・イントラ予測モードを使用する。I スライスでは、ルマ及びクロマ成分に対して別個のブロック・パーティショニング構造がイネーブルにされるので、1 つのクロマ・ブロックが複数のルマ・ブロックに対応する場合がある。従って、クロマ DM モードの場合、現在のクロマ・ブロックの中央位置をカバーする対応するルマ・ブロックのイントラ予測モードは、直接的にそのまま継承される。

10

テーブル 8-2 — sps_cclm_enabled_flag が 0 に等しい場合における

intra_chroma_pred_mode[xCb][yCb] 及び

IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2] に依存する

IntraPredModeC[xCb][yCb] の仕様

intra_chroma_pred_mode[xCb][yCb]	IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2]				
	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 (DM)	0	50	18	1	X

20

テーブル 8-3 — sps_cclm_enabled_flag が 1 に等しい場合における

intra_chroma_pred_mode[xCb][yCb] 及び

IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2] に依存する

IntraPredModeC[xCb][yCb] の仕様

intra_chroma_pred_mode[xCb][yCb]	IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2]				
	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 (DM)	0	50	18	1	X

30

2.4 VVC における変換コーディング

2.4.1 VVC における多重変換セット (MTS)

2.4.1.1 明示的な多重変換セット (MTS)

【 0 0 7 2 】

VTM4では、 64×64 のサイズに及ぶ大きなブロック・サイズの変換がイネーブルにされており、これは高解像度ビデオ、例えば1080pや4Kシーケンスに主に有用である。高周波変換係数は、サイズ（幅又は高さ、又は幅及び高さの両方）が64に等しい変換ブロックに対して、低周波係数のみが保持されるようにゼロ化される。例えば、 $M \times N$ 変換ブロックの場合、Mをブロック幅とし、Nをブロック高さとし、Mが64に等しい場合に、変換係数のうちの左32列のみが保持される。同様に、Nが64に等しい場合、変換係数の上位32行のみが保持される。大きなブロックに対して変換スキップ・モードが使用される場合、如何なる値もゼロ化することなく、ブロック全体が使用される。

HEVCで採用されているDCT-IIに加えて、多重変換選択 (Multiple Transform Sele

40

50

ction, MTS) 方式が、インター及びイントラ・コーディングされたブロック双方を残差コーディングするために使用される。これはDCT8/DST7から選択される複数の変換を使用する。新たに導入される変換行列はDST-VIIとDCT-VIIIである。以下のテーブル4は、選択されるDST/DCTの基底関数を示す。

【0073】

【数6】

テーブル4：VVCで使用される変換行列の基底関数

変換タイプ	基底関数 $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

【0074】

変換行列の直交性を保持するために、変換行列はHEVCの変換行列よりも精密に量子化される。水平変換の後及び垂直変換の後に、変換された係数の中間値を16ビットの範囲内に保つため、全ての係数は10ビットを有するべきである

【0075】

MTSスキームを制御するために、イントラ及びインターそれぞれについて、別々のイネーブル化フラグがSPSレベルで指定される。MTSがSPSでイネーブルにされると、MTSが適用されるか否かを指定するために、CUレベル・フラグがシグナリングされる。ここで、MTSはルマについてのみ適用される。MTS CUレベル・フラグは、以下の条件が充足される場合にシグナリングされる。

【0076】

- 幅及び高さ双方が32以下であること

【0077】

- CBFフラグが1に等しいこと

MTS CUフラグがゼロに等しい場合、DCT2は双方向に適用される。しかしながら、MTS CUフラグが1に等しい場合、2つの他のフラグが追加的にシグナリングされて、水平及び垂直方向それぞれについて変換タイプを指定する。変換及びシグナリング・マッピングはテーブル5にされるとおりである。変換行列の精度に関しては、8ビットの一次変換コアが使用される。従って、4点DCT-2及びDST-7、8点、16点、及び32点DCT-2を含む、HEVCで使用される全ての変換コアは同一に保たれる。また、64点DCT-2、4点DCT-8、8点、16点、32点DST-7及びDCT-8を含む他の変換コアは、8ビット一次変換コアを使用する。

【0078】

【数7】

10

20

30

40

50

テーブル 5 : 水平及び垂直方向に関する **tu_mts_idx** の復号化される値及び対応する変換行列のマッピング

tu_mts_idx の ビン・スト リング	tu_mts_idx	イントラ／インター	
		水平	垂直
0	0	DCT2	
1 0	1	DST7	DST7
1 1 0	2	DCT8	DST7
1 1 1 0	3	DST7	DCT8
1 1 1 1	4	DCT8	DCT8

10

【 0 0 7 9 】

大きなサイズのDST-7及びDCT-8の複雑さを低減させるため、32に等しいサイズ（幅又は高さ、又は幅及び高さの双方）を有するDST-7及びDCT-8ブロックに関し、高周波変換係数はゼロ化される。16×16の低周波領域内の係数のみが保持される。

【 0 0 8 0 】

異なる変換が適用される場合に加えて、VVCは、HEVCにおけるTSの概念に類似する変換スキップ（TS）と呼ばれるモードもサポートしている。TSはMTSの特殊なケースとして取り扱われる。

20

【 0 0 8 1 】

2.4.2 JVET-N0193で提案される縮小二次変換（RST）

2.4.2.1 JEMにおけるノン・セパラブル二次変換（NSST）

JEMでは、フォワード一次変換と量子化（エンコーダ側）の間、逆量子化とインバート一次変換（デコーダ側）の間で、二次変換が適用される。図10に示すように、4×4（又は8×8）の二次変換はブロック・サイズに依存して実行される。例えば、4×4二次変換が小さなブロック（即ち、min（width, height） < 8）に適用され、8×8ブロック毎に、8×8二次変換がより大きなブロック（即ち、min（width, height） > 4）に適用される。

30

【 0 0 8 2 】

一例として以下のような入力を使用してノン・セパラブル変換（non-separable transform）の適用を説明する。ノン・セパラブル変換を適用するために、まず、

【 0 0 8 3 】

【数 8】

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

40

【 0 0 8 4 】

【数 9】

という 4x4 入力ブロック X が、 $\vec{X}$ として表現される：

【 0 0 8 5 】

【数 1 0】

50

$$\vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T$$

【 0 0 8 6 】

【数 1 1】

ノン・セパラブル変換は、 $\vec{F} = T \cdot \vec{X}$ として計算され、ここで $\vec{F}$ は変換係数ベクトルであり、 T は 16×16 変換行列である。 16×1 係数ベクトル $\vec{F}$ は、次いで、そのブロックのスキニング順序（水平、垂直、又は対角）を使用して、 4×4 ブロックとして再編成される。より小さなインデックスの係数は、 4×4 係数ブロックの中で、より小さなスキニング・インデックスとともに配置されるであろう。合計で 35 個の変換セットが存在し、変換セットあたり 3 個のノン・セパラブル変換行列（カーネル）が使用される。イントラ予測モードから変換セットへのマッピングは、予め定められている。各々の変換セットに対して、選択されたノン・セパラブル二次変換（non-separable secondary transform, NSST）候補は、明示的にシグナリングされる二次変換インデックスによって更に指定される。インデックスは、変換係数の後、イントラ CU 毎に一度ビット・ストリームでシグナリングされる。

10

【 0 0 8 7 】

2.4.2.2 JVET-N0193における縮小二次変換（RST）

RST（低周波ノン・セパラブル変換（Low Frequency Non-Separable Transform, LFNST））がJVET-K0099で導入され、（35個の変換セットではなく）4つの変換セットのマッピングがJVET-L0133で導入されている。このJVET-N0193では、 16×64 行列（更に、 16×48 行列に縮小される）及び 16×16 行列が使用される。表記の便宜上、 16×64 （ 16×48 に縮小される）変換は $RST_{8 \times 8}$ 、 16×16 変換は $RST_{4 \times 4}$ として表記される。図11はRSTの一例を示す。

20

【 0 0 8 8 】

2.4.2.2.1 RST計算

縮小変換（RT）の主要なアイデアは、 N 次元ベクトルを別空間内の R 次元ベクトルにマッピングすることであり、ここで R/N （ $R < N$ ）は縮小因子である。

【 0 0 8 9 】

【数 1 2】

RT行列は次のような $R \times N$ 行列である：

$$T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \cdots & t_{RN} \end{bmatrix}$$

30

40

【 0 0 9 0 】

ここで、変換行列の R 個の行は N 次元空間の R 個の基底である。RTに対するインバート変換行列は、そのフォワード変換の転置（transpose）である。フォワード及びインバートRTは図12に描かれている。

【 0 0 9 1 】

この寄稿では、4という縮小因子（ $1/4$ サイズ）の $RST_{8 \times 8}$ が適用される。従って、従来の 8×8 ノン・セパラブル変換行列サイズである 64×64 の代わりに、 16×64 のダイレクト・マトリクス（direct matrix）が使用される。言い換えれば、 64×16 インバートRST行列が、 8×8 の左上領域でコア（一次）変換係数を生成するために、デコーダ側で使用される。フォワード $RST_{8 \times 8}$ は 16×64 （又は 8×8 ブロックに対しては 8×64 ）行列を使

50

用し、その結果、所与の 8×8 領域内の左上の 4×4 領域においてのみ非ゼロ係数を生成する。言い換えれば、RSTが適用されると、左上の 4×4 領域を除く 8×8 領域はゼロ係数のみを有することになる。RST 4×4 、 16×16 (4×4 ブロックに対しては 8×16) の場合、ダイレクト・マトリクス乗算が適用される。

【0092】

インバートRSTは、以下の2つの条件が満たされる場合に、条件付きで適用される：

【0093】

○ ブロック・サイズは、所与の閾値以上であること ($W \geq 4$ & $H \geq 4$)

【0094】

○ 変換スキップ・モード・フラグはゼロに等しいこと

10

【0095】

変換係数ブロックの幅 (W) と高さ (H) の両方が4より大きい場合、RST 8×8 は、変換係数ブロックの左上の 8×8 領域に適用される。それ以外の場合、RST 4×4 が変換係数ブロックの左上の $\min(8, W) \times \min(8, H)$ の領域に適用される。

【0096】

RSTインデックスが0に等しい場合、RSTは適用されない。それ以外の場合、RSTが適用され、そのカーネルは、RSTインデックスを用いて選択される。RSTの選択方法とRSTインデックスのコーディングについては後述する。

【0097】

更に、RSTは、イントラ及びインター・スライスの両方においてイントラCUに、またルマ及びクロマの両方に適用される。デュアル・ツリーがイネーブルにされている場合、ルマとクロマに対するRSTインデックスは別々にシグナリングされる。インター・スライスの場合（デュアル・ツリーはディセーブルにされている）、単一のRSTインデックスがシグナリングされ、ルマとクロマの両方に使用される。

20

【0098】

2.4.2.2.2 RSTの制約

ISPモードが選択される場合、RSTはディセーブルにされ、RSTインデックスはシグナリングされず、なぜなら、たとえRSTが適切な全てのパーティション・ブロックに適用されたとしても、パフォーマンスの改善はわずかであるに過ぎないからである。更に、ISPで予測された残差に対してRSTをディセーブルにすることは、符号化の複雑さを減らす可能性

30

【0099】

2.4.2.2.3 RST選択

RST行列は4つの変換セットから選択され、それら各々は2つの変換から構成される。どの変換セットが適用されるかは、以下のように、イントラ予測モードから決定される：

【0100】

(1) 3つのCCLMモードのうちの1つが指定される場合、変換セット0が選択される。

【0101】

【数13】

40

- (2) それ以外の場合、変換セットの選択は、以下のテーブルに従って実行される：
変換セット選択テーブル

IntraPredMode	Tr. set index
IntraPredMode < 0	1
0 ≤ IntraPredMode ≤ 1	0
2 ≤ IntraPredMode ≤ 12	1
13 ≤ IntraPredMode ≤ 23	2
24 ≤ IntraPredMode ≤ 44	3
45 ≤ IntraPredMode ≤ 55	2
56 ≤ IntraPredMode	1

10

【0102】

IntraPredModeと表記される、上記テーブルにアクセスするためのインデックスは、[-14,83]というレンジを有し、それは広角イントラ予測に使用される変換モード・インデックスである。

【0103】

後に、CCLMモードでコーディングされたクロマ・ブロックに対する低周波数非ノン・セパラブル変換（LFNST）（RSTとしても知られている）セットの選択は、変数IntraPredMode_CCLMに基づいて修正され、ここで、IntraPredMode_CCLMは[-14,80]というレンジを有する。IntraPredMode_CCLMは、現在のクロマ・ブロックのディメンジョンとコロケートド（co-located）ルマ・イントラ予測モードとによって決定される。

20

【0104】

デュアル・ツリーがイネーブルにされている場合、現在のクロマ・ブロックにおける左上のクロマ・サンプルの対応するルマ・サンプルをカバーするブロック（例えば、PU）は、コロケートド（又は同等位置の）ルマ・ブロックとして定義される。具体例は、図24a及び24bに示されており、同等位置の場所はTLによって示される。

【0105】

2.4.2.2.4 縮小ディメンジョンのRST行列

更なる簡略化に関し、同じ変換セットの設定を用いて16×64行列の代わりに16×48行列が適用され、それらの各々は、左上の8×8ブロックの中の3つの4×4ブロックから（右下の4×4ブロックを除いている）、48個のデータを取り込む（図13に示されている）。

30

【0106】

2.4.2.2.5 RSTシグナリング

フォワードRST8x8は16×48行列を使用し、その結果、最初の3つの4×4領域内の中で左上の4×4領域のみにおいて非ゼロの係数を生成する。言い換えれば、RST8x8が適用される場合に、左上4×4（RST8x8によるもの）と右下4×4（一次変換によるもの）の領域のみが非ゼロの係数を有する可能性がある。その結果、右上の4×4及び左下の4×4ブロック領域（図14に示されており、「ゼロ・アウト（zero-out）」領域と呼ばれる）で何らかの非ゼロ要素が検出された場合、RSTインデックスはコーディングされず、なぜならその場合はRSTが適用されなかったことを意味するからである。そのようなケースでは、RSTインデックスはゼロであると推定される。

40

【0107】

2.4.2.2.6 1つのCGにおけるゼロ・アウト領域

通常、4×4のサブ・ブロックにインバートRSTを適用する前に、4×4のサブ・ブロック内の何らかの係数は非ゼロである可能性がある。しかしながら、場合によっては、インバートRSTがサブ・ブロックに適用される前に、4×4サブ・ブロックの幾つかの係数はゼロでなければならないように制限される。

50

【 0 1 0 8 】

nonZeroSizeを変数であるとする。インバートRSTの前に、1-Dアレイに再配列された場合にnonZeroSizeより小さくないインデックスを有する如何なる係数も、ゼロでなければならないことが要求される。

【 0 1 0 9 】

nonZeroSizeが16に等しい場合、左上の4×4サブ・ブロックの係数にゼロ・アウト制約はない。

【 0 1 1 0 】

【 数 1 4 】

JVET-N0193 では、現在のブロック・サイズが4×4又は8×8である場合、nonZeroSizeは8に等しく設定される（即ち、図14に示されるようなレンジ[8, 15]内のスキニング・インデックスを有する係数は0であるものとする）。その他のブロック・ディメンジョンでは、nonZeroSizeは16に等しく設定される。

2.4.2.2.7 ワーキング・ドラフトにおける RST の説明

7.3.2.3 シーケンス・パラメータ・セット RBSP シンタックス

seq_parameter_set_rbsp() {	Descriptor
.....	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
...	
sps_st_enabled_flag	u(1)
...	
}	

7.3.7.11 残差コーディング・シンタックス

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3)) && n >= 8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
abs_level_gt1_flag [n]	ae(v)
...	

10

20

30

40

50

7.3.7.5 コーディング・ユニット・シンタックス

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descript or
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
if(Min(cbWidth, cbHeight) >= 4 && sps_st_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0) {	
st_idx[x0][y0]	ae(v)
}	
}	
}	
}	
}	

sps_st_enabled_flag (1に等しいもの) は、st_idx がイントラ・コーディング・ユニットに対する残差コーディング・シンタックスに存在する可能性があることを指定する。**sps_st_enabled_flag** (0に等しいもの) は、st_idx がイントラ・コーディング・ユニットに対する残差コーディング・シンタックスに存在しないことを指定する。

10

20

30

40

50

st_idx[x0][y0]は、選択された変換セットにおける 2つの候補カーネルの間でどの二次変換カーネルが適用されるかを指定する。**st_idx[x0][y0]** (0に等しいもの) は、二次変換が適用されないことを指定する。アレイ・インデックス **x0,y0** は、ピクチャの左上サンプルに対する、想定される変換ブロックの左上サンプルの位置 (x0, y0) を指定する。

st_idx[x0][y0]が存在しない場合、**st_idx[x0][y0]**は 0 に等しいと推定される。

st_idx を送信するかどうかは、CU 内の全ての TU における非ゼロ係数の数に依存することに留意すべきであり (例えば、単一ツリーの場合、3 ブロック (即ち、Y, Cb, Cr) における非ゼロ係数の数 ; デュアル・ツリーの場合で、ルマがコーディングされ、ルマ・ブロックにおける非ゼロ係数の数であり ; デュアル・ツリーの場合で、クロマがコーディングされる場合、2つのクロマ・ブロックにおける非ゼロ係数の数である)。更に、閾値はパーティショニング構造に依存し、(treeType == SINGLE_TREE) ? 2 : 1) である。

10

st_idx のビンはコンテキスト・コーディングされる。より具体的には、以下が適用される :

テーブル 9-9 – シンタックス要素及び関連する二進化

シンタックス構造	シンタックス要素	二進化	
		プロセス	入力パラメータ
.....			
	st_idx[][]	TR	cMax = 2, cRiceParam = 0

テーブル 9-15 コンテキスト・コーディングされるビンを利用した **ctxInc** のシンタックス要素に対する割り当て

20

シンタックス要素	binIdx					
	0	1	2	3	4	>= 5
.....						
st_idx[][]	0,1,4,5 (条項 9.5.4.2.8)	2,3,6,7 (条項 9.5.4.2.8)	na	na	na	na
.....						

9.5.4.2.8 シンタックス要素 **st_idx** に対する **ctxInc** の導出プロセス

このプロセスに対する入力、色成分インデックス **cIdx** と、**cIdx** に依存する現在のピクチャの左上サンプルに対する、現在のルマ又はクロマ・コーディング・ブロックの左上サンプルを指定するルマ又はクロマ位置 (x0, y0) と、ツリー・タイプ **treeType** と、条項 8.4.2 で規定されているようなルマ・イントラ予測モード **IntraPredModeY[x0][y0]** と、条項 7.4.7.5 で規定されているようなクロマ・サンプルのイントラ予測モードを指定するシンタックス要素 **intra_chroma_pred_mode[x0][y0]** と、多重変換選択インデックス **tu_mts_idx[x0][y0]** である。このプロセスの出力は変数 **ctxInc** である。

30

変数 **intraModeCtx** は次のように導出される :

cIdx が 0 に等しい場合、**intraModeCtx** は次のように導出される :

intraModeCtx = (**IntraPredModeY[x0][y0]** <= 1) ? 1 : 0

それ以外の場合 (**cIdx** は 0 より大きい) 、**intraModeCtx** は以下のように導出される :

intraModeCtx = (**intra_chroma_pred_mode[x0][y0]** >= 4) ? 1 : 0

変数 **mtsCtx** は以下のように導出される :

mtsCtx = (**tu_mts_idx[x0][y0]** == 0 && **treeType** != SINGLE_TREE) ? 1 : 0

変数 **ctxInc** は以下のように導出される :

ctxInc = (**binIdx** << 1) + **intraModeCtx** + (**mtsCtx** << 2)

40

50

2.4.2.2.8 RST 使用法の概要

RST は、1 つのブロック内の非ゼロ係数の数が、単独及び別個のツリーに対してそれぞれ、2 及び 1 より大きい場合にのみイネーブルにされることが可能である。更に、RST がイネーブルにされる場合、RST 適用コーディング・グループ (CG) に対する非ゼロ係数の位置についての以下の制限も要求される。

テーブル 1 : RST の使用法

ブロック・サイズ	RST タイプ	RST が適用される CGs の数	RST が適用される何れの CG が、非ゼロ係数を有する可能性があるか	RST が適用される CG における非ゼロ係数の可能性のある場所 (1 つの CG に対する nonZeroSize)
4x4	RST4x4 (16x16)	1 (左上 4x4)	左上 4x4	対角線スキャン順序で最初の 8 個 (図 16 における 0.7 : 右上方向スキャン順序 (例えば CG として 4x4) ,nonZeroSize = 8
4x8/8x4	RST4x4 (16x16)	1 (左上 4x4)	左上 4x4	全て, nonZeroSize = 16
4xN 及び Nx4 (N>8)	RST4x4 (16x16)	2 (4xN: 上端 4x8; Nx4: 左端 4x8)	4xN: 上端 4x8; Nx4: 左端 4x8	全て, nonZeroSize = 16
8x8	RST8x8 (16x48)	3 (フォワード RST の後に 1 つの CG のみが非ゼロ係数を有する可能性がある)	左上 4x4	対角線スキャン順序で最初の 8 個 (図 16 における 0.7 : 右上方向スキャン順序 (例えば CG として 4x4) ,nonZeroSize = 8
その他 (W*H, W>8, H>8)	RST8x8 (16x48)	3 (フォワード RST の後に 1 つの CG のみが非ゼロ係数を有する可能性がある)	左上 4x4	all, nonZeroSize = 16

【 0 1 1 1 】

2.4.3 サブ・ブロック変換

1 に等しい cu_cbf を有するインター予測された CU の場合、残差ブロック全体又は残差ブロックのサブ・パートが復号化されるかどうかを示すために、cu_sbt_flag がシグナリングされてもよい。前者の場合、CU の変換タイプを決定するために、インター MTS 情報が更に解析される。後者の場合、残差ブロックの一部は推定適応変換でコーディングされ、残差ブロックの他の部分はゼロ・アウト化される。SBT は複合インター・イントラ・モードには適用されない。

【 0 1 1 2 】

【 数 1 5 】

サブ・ブロック変換では、SBT-V及びSBT-H（クロマTBは常にDCT-2を使用する）のルマ変換ブロックに、位置依存変換が適用される。SBT-HとSBT-Vの2つのポジションは異なるコア変換に関連付けられる。より具体的には、各SBTポジションに対する水平及び垂直変換は図3に規定されている。例えばSBT-Vポジション0に対する水平及び垂直変換はそれぞれDCT-8及びDST-7である。残差TUの一方の側が32より大きい場合、対応する変換はDCT-2として設定される。従って、サブ・ブロック変換は、残差ブロックのTUタイリング、cbf、水平及び垂直変換を一緒に指定し、これは、ブロックの主要な残差がブロックの一方の側にあるケースに対するシンタックス・ショートカット（syntax shortcut）と考えられてもよい。

2.4.3.1 シンタックス要素

7.3.7.5 コーディング・ユニット・シンタックス

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(slice_type != I sps_ibc_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I)	
pred_mode_flag	ae(v)
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
...	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
...	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	
}	
}	

10

20

30

40

50

cu_sbt_flag (1に等しいもの)は、現在のコーディング・ユニットに対して、サブブロック変換が使用されることを指定する。**cu_sbt_flag** (0に等しいもの)は、現在のコーディング・ユニットに対して、サブブロック変換が使用されないことを指定する。**cu_sbt_flag**が存在しない場合、その値は0に等しいと推定される。

NOTEー：サブブロック変換が使用される場合、コーディング・ユニットは2つの変換ユニットに分割され；一方の変換ユニットは残差データを有し、他方は残差データを有しない。

cu_sbt_quad_flag (1に等しいもの)は、現在のコーディング・ユニットに対して、サブブロック変換が、現在のコーディング・ユニットの1/4サイズの変換ユニットを含むことを指定する。
cu_sbt_quad_flag (0に等しいもの)は、現在のコーディング・ユニットに対して、サブブロック変換が、現在のコーディング・ユニットの1/2サイズの変換ユニットを含むことを指定する。
cu_sbt_quad_flagが存在しない場合、その値は0に等しいと推定される。

10

cu_sbt_horizontal_flag (1に等しいもの)は、現在のコーディング・ユニットが2つの変換ユニットに水平に分割されることを指定する。**cu_sbt_horizontal_flag[x0][y0]** (0に等しいもの)は、現在のコーディング・ユニットが2つの変換ユニットに垂直に分割されることを指定する。
cu_sbt_horizontal_flagが存在しない場合、その値は以下のように導出される：

- ー **cu_sbt_quad_flag**が1に等しい場合、**cu_sbt_horizontal_flag**はallowSbtHorQに等しく設定される。
- ー それ以外の場合 (**cu_sbt_quad_flag**は0に等しい)、**cu_sbt_horizontal_flag**はallowSbtHorHに等しく設定される。

20

cu_sbt_pos_flag (1に等しいもの)は、現在のコーディング・ユニットにおける第1変換ユニットの **tu_cbf_luma**, **tu_cbf_cb** 及び **tu_cbf_cr** は、ビットストリームに存在しないことを指定する。
cu_sbt_pos_flag (0に等しいもの)は、現在のコーディング・ユニットにおける第2変換ユニットの **tu_cbf_luma**, **tu_cbf_cb** 及び **tu_cbf_cr** は、ビットストリームに存在しないことを指定する。変数 **SbtNumFourthsTb0** は以下のように導出される：

$$\text{sbtMinNumFourths} = \text{cu_sbt_quad_flag} ? 1 : 2 \quad (7-117)$$

$$\text{SbtNumFourthsTb0} = \text{cu_sbt_pos_flag} ? (4 - \text{sbtMinNumFourths}) : \text{sbtMinNumFourths} \quad (7-118)$$

sps_sbt_max_size_64_flag (0に等しいもの)は、サブブロック変換を許可するための最大CU幅と高さが32ルマ・サンプルであることを指定する。**sps_sbt_max_size_64_flag** (1に等しいもの)は、サブブロック変換を許可するための最大CU幅と高さが64ルマ・サンプルであることを指定する。

30

$$\text{MaxSbtSize} = \text{sps_sbt_max_size_64_flag} ? 64 : 32 \quad (7-33)$$

【0113】

2.4.4 量子化残差ドメイン・ブロック差分パルス・コード変調コーディング (QR-BDPCM)

JVET-N0413では、量子化残差ドメインBDPCM (以下、RBDPCMと表す) が提案されている。イントラ予測と同様に、予測方向にサンプル・コピーを行うことによって (水平又は垂直予測)、ブロック全体に関してイントラ予測が行われる。残差は量子化され、量子化された残差とその予測子 (水平又は垂直) 量子化値との間のデルタがコーディングされる。

40

【0114】

【数16】

サイズが M (行) $\times$ N (列) であるブロックに関し、 $r_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ が、上又は左ブロック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（ライン毎に予測ブロックにわたって左隣接ピクセル値をコピーして）水平に又は（予測されるブロックの各ラインに上隣接ラインをコピーして）垂直にイントラ予測を実行した後の予測残差であるとする。 $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ を、残差 $r_{i,j}$ の量子化されたバージョンとし、ここで、残差は元のブロックと予測されたブロックの値の間の差分である。次いで、ブロック DPCM が量子化された残差サンプルに適用され、要素 $\tilde{r}_{i,j}$ を有する修正された $M \times N$ アレイ $\tilde{R}$ という結果が得られる。垂直 BDPCM がシグナリングされる場合は以下ようになる：

10

【 0 1 1 5 】

【数 1 7】

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

【 0 1 1 6 】

水平予測の場合、同様なルールが適用され、残差量子化サンプルは以下によって取得される：

20

【 0 1 1 7 】

【数 1 8】

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

【 0 1 1 8 】

【数 1 9】

残差量子化サンプル $\tilde{r}_{i,j}$ はデコーダに送られる。

30

【 0 1 1 9 】

【数 2 0】

デコーダ側では、 $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ を生成するために上記の計算が逆に行われる。垂直予測の場合は次のとおりである。

【 0 1 2 0 】

【数 2 1】

40

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

【 0 1 2 1 】

水平の場合は次のとおりである。

【 0 1 2 2 】

【数 2 2】

50

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

【0 1 2 3】

【数 2 3】

インバート量子化残差 $Q^{-1}(Q(r_{i,j}))$ を、イントラ・ブロック予測値に加算して、再構成されたサンプル値を生成する。

10

【0 1 2 4】

QR-BDPCMが選択される場合、適用される変換は存在しない。

【0 1 2 5】

2.5 係数のエントロピー・コーディング

2.5.1 変換適用ブロックの係数コーディング

HEVCでは、コーディング・ブロックの変換係数は、オーバーラップしない係数グループ（又はサブブロック）を用いてコーディングされ、各CGは、コーディング・ブロックの 4×4 ブロックの係数を含む。コーディング・ブロック内のCGとCG内の変換係数は、予め定義されたスキャン順序に従ってコーディングされる。

20

【0 1 2 6】

コーディング・ブロック内のCGとCG内の変換係数は、予め定義されたスキャン順序に従ってコーディングされる。CGとCG内の係数の両方とも、対角線右上向きスキャン順序に従う。 4×4 ブロック及び 8×8 スキャン順序の例がそれぞれ図16及び図17に描かれている。

【0 1 2 7】

コーディングの順序は、反転したスキャン順序（即ち、図17ではCG3からCG0への復号化）であり、あるブロックを復号化する場合、最後の非ゼロ係数の座標が最初に復号化されることに留意されたい。

【0 1 2 8】

少なくとも1つの非ゼロ変換係数を有するCGの変換係数レベルのコーディングは、複数のスキャン・パスに分離されてもよい。第1パスでは、第1ビン（bin0で示され、significant_coeff_flagとも言及され、これは係数の大きさが0より大きいことを示す）がコーディングされる。次に、第2／第3ビン（それぞれbin1及びbin2により指定され、coeff_abs_greater1_flag及びcoeff_abs_greater2_flagとも言及される）をコーディングするコンテキストに対する2つのスキャン・パスが適用されてもよい。最終的に、必要に応じて、係数レベルの符号情報及び残りの値（coeff_abs_level_remainingとも呼ばれる）をコーディングするための2つ以上のスキャン・パスが呼び出される。最初の3つのスキャン・パスのビンのみが、レギュラー・モードでコーディングされ、それらのビンは以下の説明ではレギュラー・ビンと呼ばれることに留意されたい。

30

40

【0 1 2 9】

VVC 3では、各々のCGについて、レギュラー・コーディングされたビンとバイパス・コーディングされたビンはコーディング順序の中で分離され；まず、サブブロックに対する全てのレギュラー・コーディングされたビンが伝送され、その後にバイパス・コーディングされたビンが伝送される。サブブロックの変換係数レベルは、スキャン・ポジションにおける5つのパスで以下のようにしてコーディングされる：

【0 1 3 0】

○ パス1：有意性（sig_flag）、1より大フラグ（gt1_flag）、パリティ（par_level_flag）、2より大フラグ（gt2_flag）のコーディングが、コーディング順序で処理される。sig_flagが1に等しい場合、まず、gt1_flagがコーディングされる（それは、絶対レベル

50

が1より大きいかどうかを指定する)。gt1_flagが1に等しい場合、par_flagが追加的にコーディングされる（それは、絶対レベル・マイナス2のパリティを指定する）。

【0131】

○ パス2：残りの絶対レベル（リメインダ（remainder））のコーディングが、1に等しいgt2_flag又は1に等しいgt1_flagを有する全てのスキャン・ポジションに対して処理される。非バイナリ・シンタックス要素は、Golomb-Riceコードで二進化され、結果のピンは算術符号化エンジンのバイパス・モードでコーディングされる。

【0132】

○ パス3：（レギュラー・コーディングされたピンの限界に到達したことに起因して）第1パスでsig_flagがコーディングされていない係数の絶対レベル（absLevel）が、Golomb-Riceコードを使用して算術符号化エンジンのバイパス・モードで完全にコーディングされる。

【0133】

○ パス4：1に等しいsig_coeff_flagを有する全てのスキャン・ポジションに対する符号（sign_flag）をコーディングする。

【0134】

4x4サブブロックに関して、32より多くないレギュラー・コーディングされたピン（sig_flag, par_flag, gt1_flag及びgt2_flag）が符号化又は復号化されることが保証される。2x2クロマ・サブブロックの場合、レギュラー・コーディングされたピンの数は8に制限される。

【0135】

（パス3における）非バイナリ・シンタックス要素リメインダをコーディングのためのライス・パラメータ（ricePar）が、HEVCと同様に導出される。各サブブロックの始めに、riceParは0に等しく設定される。シンタックス要素リメインダをコーディングした後に、ライス・パラメータは予め定義された式に従って修正される。（パス4における）非バイナリ・シンタックス要素absLevelをコーディングするために、ローカル・テンプレート中の絶対値の合計sumAbsが決定される。変数ricePar及びposZeroは、テーブル・ルックアップによって、依存量子化及びsumAbsに基づいて決定される。中間変数codeValueは以下のように導出される：

【0136】

○ absLevel[k]が0に等しい場合、codeValueはposZeroに等しく設定される；

【0137】

○ それ以外の場合、absLevel[k]がposZero以下であるならば、codeValueはabsLevel[k]-1に等しく設定される；

【0138】

○ それ以外の場合（absLevel[k]はposZeroより大きい）、codeValueはabsLevel[k]に等しく設定される。

【0139】

codeValueの値は、ライス・パラメータriceParとともにGolomb-Riceコードを使用してコーディングされる。

【0140】

2.5.1.1 係数コーディングのためのコンテキスト・モデリング

変換係数レベルの絶対値に関連するシンタックス要素の確率モデルの選択は、ローカルな近隣における絶対値レベルの値又は部分的に再構成された絶対値レベルの値に依存する。使用されるテンプレートは図18に示されている。

【0141】

選択される確率モデルは、ローカルな近隣における絶対値レベル（又は部分的に再構成された絶対値レベル）の合計と、ローカルな近隣における0より大きな絶対値レベルの数（1に等しいsig_coeff_flagsの数によって与えられる）に依存する。コンテキスト・モデリング及び二値化は、ローカルな近隣に対する以下の尺度に依存する：

【0142】

○ numSig：ローカルな近隣における非ゼロ・レベルの数；

【0143】

○ sumAbs1：ローカルな近隣における第1パスの後の部分的に再構成された絶対値レベル (absLevel1) の合計；

【0144】

○ sumAbs：ローカルな近隣における再構成された絶対値レベルの合計；

【0145】

○ ダイアゴナル・ポジション (d)：変換ブロック内の現在のスキャン・ポジションの水平及び垂直座標の合計。

【0146】

numSig, sumAbs1,及びdの値に基づいて、sig_flag, par_flag, gt1_flag,及びgt2_flagをコーディングするための確率モデルが選択される。abs_remainderを二進化するためのライス・パラメータは、sumAbs及びnumSigの値に基づいて選択される。

【0147】

2.5.1.2 依存量子化 (DQ)

更に、同じHEVCスカラー量子化が、依存スケール量子化 (dependent scale quantization) と呼ばれる新たな概念とともに使用される。依存スカラー量子化とは、変換係数の許容可能な再構成値のセットが、再構成順序において現在の変換係数レベルに先行する変換係数レベルの値に依存しているアプローチを指す。このアプローチの主な効果は、HEVCで使用されるような従来の独立スカラー量子化と比較して、許容可能な再構成ベクトルが、N次元ベクトル空間 (Nは変換ブロックにおける変換係数の数を表す) において、より高密度に詰め込まれることである。これは、N次元単位体積あたりの許容可能な再構成ベクトルの所与の平均数に対して、入力ベクトルと最も近い再構成ベクトルとの間の平均逸脱 (average distortion) が低減されることを意味する。依存スカラー量子化のアプローチは：(a) 異なる再構成レベルを有する2つのスカラー量子化器を定義すること、及び (b) 2つのスカラー量子化器の間の切り替えのためのプロセスを定義することによって実現される。

【0148】

Q0とQ1で示される2つのスカラー量子化器が図19に示されている。利用可能な再構成レベルの位置は、量子化ステップ・サイズΔによって一意に指定される。使用されるスカラー量子化器 (Q0又はQ1) は、ビットストリームで明示的にはシグナリングされない。その代わりに、現在の変換係数に使用される量子化器は、コーディング／再構成順序において現在の変換係数に先行する変換係数レベルのパリティによって決定される。

【0149】

【数24】

10

20

30

40

50

図 20 に示すように、2 つのスカラ量子化器 (Q0 及び Q1) の間の切り替えは、4 つの状態を有する状態マシンによって実現される。状態は、0, 1, 2, 3 という 4 つの異なる値をとることができる。状態は、コーディング／再構成順序において現在の変換係数に先行する変換係数レベルのパリティによって一意に決定される。変換ブロックに対する逆量子化の始めに、状態は 0 に等しく設定される。変換係数は、スキャニング順序（即ち、それらがエントロピー復号化されるのと同じ順序）で再構成される。現在の変換係数が再構成された後に、状態は図 20 に示されるように更新され、ここで、k は変換係数レベルの値を示す。

2.5.1.3 シンタックス及びセマンティクス

7.3.7.11 残差コーディング・シンタックス

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if((tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbWidth > 4)	
log2TbWidth = 4	
else	
log2TbWidth = Min(log2TbWidth, 5)	
if(tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbHeight > 4)	
log2TbHeight = 4	
else	
log2TbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth < 2 && cIdx == 0) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2 && cIdx == 0) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
QState = 0	

10

20

30

40

50

for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
remBinsPass1 = ((log2SbW + log2SbH) < 4 ? 8 : 32)	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = -1	
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) &&	
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
abs_level_gt1_flag [n]	ae(v)
remBinsPass1--	
if(abs_level_gt1_flag[n]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gt3_flag [n]	ae(v)
remBinsPass1--	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] +	
abs_level_gt1_flag[n] + 2 * abs_level_gt3_flag[n]	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
if(remBinsPass1 < 4)	
firstPosMode1 = n - 1	
}	
for(n = numSbCoeff - 1; n >= firstPosMode1; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(abs_level_gt3_flag[n])	
abs_remainder [n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
}	
for(n = firstPosMode1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
dec_abs_level [n]	ae(v)
if(AbsLevel[xC][yC] > 0)	
firstSigScanPosSb = n	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
if(dep_quant_enabled_flag !sign_data_hiding_enabled_flag)	

10

20

30

40

50

signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if((AbsLevel[xC][yC] > 0) && (!signHidden (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ae(v)
}	
if(dep_quant_enabled_flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(AbsLevel[xC][yC] > 0)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff_sign_flag[n])	
QState = QStateTransTable[QState][par_level_flag[n]]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(AbsLevel[xC][yC] > 0) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
}	

10

20

30

【 0 1 5 0 】

2.5.2 TSコーディングされたブロック及びQR-BDPCMコーディングされたブロックの係数コーディング

QR-BDPCMはTSコーディングされたブロックのコンテキスト・モデリング方法に従う。

【 0 1 5 1 】

TS残差に関する修正された変換係数レベル・コーディング

レギュラー残差コーディングの場合と比較して、TSに対する残差コーディングは以下の変更を含む：

【 0 1 5 2 】

(1) 最後のx/yポジションのシグナリングはない。

【 0 1 5 3 】

(2) 全ての先行するフラグが0に等しい場合、最後のサブブロックを除いてすべてのサブブロックについてコーディングされるcoded_sub_block_flag。

【 0 1 5 4 】

(3) 縮小テンプレートをを用いるsig_coeff_flagコンテキスト・モデリング

40

50

【 0 1 5 5 】

(4) abs_level_gt1_flag及びpar_level_flagに対するシングル・コンテキスト・モデル

【 0 1 5 6 】

(5) 符号フラグ、5,7,9より大きな追加フラグに対するコンテキスト・モデリング

【 0 1 5 7 】

(6) リメインダ二進化のための修正されたライス・パラメータ導出

(7) コンテキスト・コーディングされたビンのサンプル当たりの数に対する制限、1ブロック内でサンプル当たり2つのビン

【 0 1 5 8 】

【 数 2 5 】

10

20

30

40

50

2.5.2.1 シンタックス及びセマンティクス

7.3.6.10 変換ユニット・シンタックス

	Descriptor
transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	
...	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)	
transform_skip_flag [x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (tbWidth <= 32) && (tbHeight <= 32) && (!transform_skip_flag[x0][y0]))	
tu_mts_idx [x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_coding_ts(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
if(tu_cbf_cb[x0][y0])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 1)	
if(tu_cbf_cr[x0][y0])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 2)	
}	

10

20

	Descriptor
residual_ts_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	
log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
numSbCoeff = 1 << (log2SbSize << 1)	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) - 1	
/* Loop over subblocks from top-left (DC) subblock to the last one */	
inferSbCbf = 1	
MaxCcbs = 2 * (1 << log2TbWidth) * (1 << log2TbHeight)	
for(i = 0; i <= lastSubBlock; i++) {	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize][i][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize][i][1]	
if((i != lastSubBlock !inferSbCbf)	
coded_sub_block_flag [xS][yS]	ae(v)
MaxCcbs--	
if(coded_sub_block_flag[xS][yS] && i < lastSubBlock)	
inferSbCbf = 0	
}	
/* First scan pass */	
inferSbSigCoeffFlag = 1	
for(n = (i == 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(coded_sub_block_flag[xS][yS] && (n == numSbCoeff - 1 !inferSbSigCoeffFlag)) {	
sig_coeff_flag [xC][yC]	ae(v)
MaxCcbs--	
if(sig_coeff_flag[xC][yC])	
inferSbSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
coeff_sign_flag [n]	ae(v)

30

40

50

abs_level_gtx_flag[n][0]	ae(v)
MaxCcbs = MaxCcbs - 2	
if(abs_level_gtx_flag[n][0]) {	
par_level_flag[n]	ae(v)
MaxCcbs--	
}	
AbsLevelPassX[xC][yC] =	
sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag[n][0]	
}	
/* Greater than X scan passes (numGtXFlags=5) */	
for(i = 1; i <= 5 - 1 && abs_level_gtx_flag[n][i - 1]; i++) {	
for(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
abs_level_gtx_flag[n][i]	ae(v)
MaxCcbs--	
AbsLevelPassX[xC][yC] += 2 * abs_level_gtx_flag[n][i]	
}	
}	
/* remainder scan pass */	
for(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(abs_level_gtx_flag[n][numGtXFlags - 1])	
abs_remainder[n]	ae(v)
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (1 - 2 * coeff_sign_flag[n]) *	
(AbsLevelPassX[xC][yC] + abs_remainder[n])	
}	
}	
}	
}	

10

20

コンテキスト・コーディングされたビンの数は、各 CG に対してサンプル当たりビン 2 つより大きくないように制限される。

テーブル9-15 — コンテキスト・コーディングされたビンによるctxIncのシンタックス要素に対する割り当て

シンタックス要素	binIdx					
	0	1	2	3	4	>= 5
last_sig_coeff_x_prefi x	0..23 (clause 9.5.4.2.4)					
last_sig_coeff_y_prefi x	0..23 (clause 9.5.4.2.4)					
last_sig_coeff_x_suffi x	bypass	bypass	bypass	bypass	bypass	bypass
last_sig_coeff_y_suffi x	bypass	bypass	bypass	bypass	bypass	bypass
coded_sub_block_flag [][]	(MaxCcbs > 0) ? (0..7 (clause 9.5.4.2.6)) : bypass	na	na	na	na	na
sig_coeff_flag[][]	(MaxCcbs > 0) ? (0..93 (clause 9.5.4.2.8)) : bypass	na	na	na	na	na

30

40

50

テーブル9-15 — コンテキスト・コーディングされたビンによるctxIncのシンタックス要素に対する割り当て

シンタックス要素	binIdx					
	0	1	2	3	4	>= 5
par_level_flag[]	(MaxCcbs > 0) ? (0..33 (clause 9.5.4.2.9)) : bypass	na	na	na	na	na
abs_level_gtx_flag[][i]	0..70 (clause 9.5.4.2.9)	na	na	na	na	na
abs_remainder[]	bypass	bypass	bypass	bypass	bypass	bypass
dec_abs_level[]	bypass	bypass	bypass	bypass	bypass	bypass
coeff_sign_flag[] transform_skip_flag[x0][y0] == 0	bypass	na	na	na	na	na
coeff_sign_flag[] transform_skip_flag[x0][y0] == 1	0	na	na	na	na	na

10

【 0 1 5 9 】

20

2.5 クロマダイレクトモード

ダイレクト・モード (Direct Mode, DM)では、クロマ・イントラ予測モードを導出するためにコロケーテッド (同等位置の) ルマ・ブロックの予測モードが使用される。

【 0 1 6 0 】

まず、イントラ予測モードlumaIntraPredModeが導出される：

- ・コロケーテッド・ルマ・ブロックがMIPモードでコーディングされる場合、lumaIntraPredModeは平面モードに等しく設定される。

- ・それ以外の場合、コロケーテッド・ルマ・ブロックがIBCモード又はパレット・モードでコーディングされるならば、lumaIntraPredModeはDCモードに等しく設定される。

30

- ・それ以外の場合、lumaIntraPredModeは、クロマ・ブロックの中央の対応するルマ・サンプルをカバーするコロケーテッド・ルマ・ブロックのイントラ予測モードに等しく設定される。一例が図25に示されている。

【 0 1 6 1 】

【数 2 6】

40

50

次に、以下のテーブルで太字のイタリック体で強調されているように、イントラ・クロマ予測モード（IntraPredModeC と記される）は、lumaIntraPredMode に従って導出される。4 に等しい intra_chroma_pred_mode は DM モードを指すことに留意されたい。

テーブル 8-2 - cclm_mode_flag, cclm_mode_idx, intra_chroma_pred_mode and lumaIntraPredMode に依存する IntraPredModeC[xCb][yCb] の仕様

cclm_mode_flag	cclm_mode_idx	intra_chroma_pred_mode	lumaIntraPredMode				
			0	50	18	1	X (0 <= X <= 6)
0	-	0	66	0	0	0	0
0	-	1	50	66	50	50	50
0	-	2	18	18	66	18	18
0	-	3	1	1	1	66	1
0	-	<i>4</i>	<i>0</i>	<i>50</i>	<i>18</i>	<i>1</i>	<i>X</i>
1	0	-	81	81	81	81	81
1	1	-	82	82	82	82	82
1	2	-	83	83	83	83	83

10

【 0 1 6 2 】

20

【数 2 7】

最後に、ピクチャのカラー・フォーマットが 4:2:2 である場合、IntraPredModeC は、DM モードに関する以下のテーブルに従って更に修正される。

chroma_format_idc が 2 に等しい場合におけるクロマ・イントラ予測モード X からモード Y への 4:2:2 マッピング・プロセスの仕様

mode X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mode Y	0	1	61	62	63	64	65	66	2	3	4	6	8	10	12	13	14	16
mode X	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
mode Y	18	20	22	23	24	26	28	30	32	33	34	35	36	37	38	39	40	41
mode X	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
mode Y	42	43	44	44	44	45	46	46	46	47	48	48	48	49	50	51	52	52
mode X	54	55	56	57	58	59	60	61	62	63	64	65	66					
mode Y	52	53	54	54	54	55	56	56	56	57	58	59	60					

30

【 0 1 6 3 】

【数 2 8】

40

50

詳細なドラフトは以下のように規定される：

8.4.3 クロマ・イントラ予測モードの導出プロセス

このプロセスに対する入力は、次のとおりである：

- ー 現在のピクチャの左上ルマ・サンプルに対する、現在のクロマ・コーディング・ブロックの左上サンプルを指定するルマ位置(x_{Cb} , y_{Cb}),
- ー ルマ・サンプルの現在のコーディング・ブロックの幅を指定する変数 $cbWidth$,
- ー ルマ・サンプルの現在のコーディング・ブロックの高さを指定する変数 $cbHeight$.

このプロセスでは、クロマ・イントラ予測モード $IntraPredModeC[x_{Cb}][y_{Cb}]$ が導出される。

対応するルマ・イントラ予測モード $lumaIntraPredMode$ は次のように導出される：

- ー $intra_mip_flag[x_{Cb}][y_{Cb}]$ が1に等しい場合、 $lumaIntraPredMode$ は INTRA_PLANAR に等しく設定される。
- ー それ以外の場合、 $CuPredMode[0][x_{Cb}][y_{Cb}]$ が MODE_IBC 又は MODE_PLT に等しいならば、 $lumaIntraPredMode$ は INTRA_DC に等しく設定される。
- ー それ以外の場合、 $lumaIntraPredMode$ は $IntraPredModeY[x_{Cb} + cbWidth / 2][y_{Cb} + cbHeight / 2]$ に等しく設定される。

クロマ・イントラ予測モード $IntraPredModeC[x_{Cb}][y_{Cb}]$ は、テーブル8-2に規定されているような $cclm_mode_mode_flag$, $cclm_mode_idx$, $intra_chroma_pred_mode$ 及び $lumaIntraPredMode$ を使用して導出される。

テーブル8-2 - $cclm_mode_flag$, $cclm_mode_idx$, $intra_chroma_pred_mode$ and $lumaIntraPredMode$ に依存する $IntraPredModeC[x_{Cb}][y_{Cb}]$ の仕様

$cclm_mode_flag$	$cclm_mode_idx$	$intra_chroma_pred_mode$	$lumaIntraPredMode$				
			0	50	18	1	X ($0 \leq X \leq 6$)
0	-	0	66	0	0	0	0
0	-	1	50	66	50	50	50
0	-	2	18	18	66	18	18
0	-	3	1	1	1	66	1
0	-	4	0	50	18	1	X
1	0	-	81	81	81	81	81
1	1	-	82	82	82	82	82
1	2	-	83	83	83	83	83

$chroma_format_idc$ が2に等しい場合、クロマ・イントラ予測モードYは、テーブル8-3で規定されているようなテーブル8-2におけるクロマ・イントラ予測モードXを使用して導出され、クロマ・イントラ予測モードXはその後にクロマ・イントラ予測モードYに等しく設定される。

テーブル8-3 - $chroma_format_idc$ が2に等しい場合におけるクロマ・イントラ予測モードXからモードYへの4:2:2マッピング・プロセスの仕様

mode X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mode Y	0	1	61	62	63	64	65	66	2	3	4	6	8	10	12	13	14	16
mode X	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
mode Y	18	20	22	23	24	26	28	30	32	33	34	35	36	37	38	39	40	41
mode X	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
mode Y	42	43	44	44	44	45	46	46	46	47	48	48	48	49	50	51	52	52
mode X	54	55	56	57	58	59	60	61	62	63	64	65	66					
mode Y	52	53	54	54	54	55	56	56	56	57	58	59	60					

3 既存の実装の欠点

現行の設計は以下の問題を有する：

【0165】

(1) クロマ成分に対する4つの事前に定義された変換セットは、ルマ成分の変換セットと同じである。更に、同じイントラ予測モードを利用するルマ及びクロマ・ブロックは、同じ変換セットを使用する。しかしながら、典型的には、ルマ成分と比較してクロマ信号はより滑らかである。同じセットを使用することは、最適ではない場合がある。

【0166】

(2) RSTは、全てのCGではなく特定のCGに限って適用される。しかしながら、RSTインデックスをシグナリングすることに関する決定は、全ブロックにおける非ゼロ係数の数に依存する。RSTが適用されるCGの全ての係数がゼロである場合、RSTインデックスをシグナリングする必要性はない。しかしながら、現行の設計はそれでもインデックスをシグナリングする場合があります、不必要なビットを浪費している。

【0167】

(3) RSTインデックスは、残差コーディングの後にシグナリングされ、なぜなら非ゼロ係数がどの程度多いか、非ゼロ係数が所定の場所に存在するかどうかを記録する必要があるからである（例えば、セクション2.3.2.2.7のnumZeroOutSigCoeff, numSigCoeff）。このような設計は、解析プロセスをより複雑にする。

【0168】

(4) RSTインデックスは、コンテキスト・コーディングされ、コンテキスト・モデリングは、コーディングされたルマ／クロマ・イントラ予測モード、及びMTSインデックスに依存する。このような設計は、イントラ予測モードの再構成の観点から言えば解析の遅延を導入してしまう。また、8つのコンテキストが導入されており、これはハードウェア実装の負担となる可能性がある。

【0169】

(a) DMとCCLMは同じコンテキスト・インデックス・オフセットを共有しているが、それらは2つの異なるクロマ・イントラ予測法であるので意味をなさない。

【0170】

(5) 非TS残差コーディングの現行の設計は、まず、係数情報をコーディングし、次いでRSTのインデックス（即ち、RSTを使用するか又は使用しないか、使用する場合には、どの行列が選択されるか）のコーディングに続く。このような設計では、残差のエントロピー・コーディングにおいて、RSTのオン／オフ情報を考慮することができない。

【0171】

(6) RSTは、一次変換が適用された変換ブロックの左上領域に常に適用される。しかしながら、異なる一次変換を基礎とする場合、エネルギーが変換ブロックの左上領域に集中することは、常には正しくない。

【0172】

(7) RST関連情報をシグナリングするかどうかの決定は、デュアル・ツリー及びシングル・ツリーのコーディング構造に対して異なる方法で行われる。

【0173】

(8) CU内に1つより多いTUがある場合（例えば、CUサイズは128×128である）、RST関連情報を解析するかどうかは、全てのTUを復号化した後でのみ決定することができる。例えば、128×128 CUの場合、最初のPBは最後のPBの後に来るLFNSTインデックスを待つことなく処理することはできない。これは、（CABACが分離可能である場合に）全体的な64×64ベースのデコーダ・パイプラインを必然的に壊さないが、特定数のデコーダ・パイプライン・ステージに対してデータ・バッファリングを4倍増加させる。これはコストがかかる。

【0174】

(9) JVET-00219では、コロケートッド・ルマ・ブロックのイントラ予測モードは、CCLMモード符号化クロマ・ブロックのLFNST変換セットを決定するために使用される。

10

20

30

40

50

しかしながら、コロケーテッド・ルマ・ブロックは、非イントラ予測モード（即ち、DC/平面/角度 予測方向を使用するような従来のイントラ予測方法ではない）でコーディングされる可能性がある。例えば、デュアル・ツリーの場合、コロケーテッド・ルマ・ブロックは、パレット・モード、IBCモード等でコーディングされる可能性がある。この場合、コロケーテッド・ルマ・ブロックのイントラ予測モードは未定である。従って、クロマ・ブロックに対する二次変換セットを導出する方法は不明である。

【0175】

（10）サブ条項8.4.4で定義されたクロマ・イントラ予測ブロックの導出は、対応する左上ルマ・サンプルをカバーするルマ・ブロックがMIP/IBC/パレット・モードでコーディングされるかどうかをチェックする。それが真であれば、デフォルト・モードが与えられる。それ以外の場合は、中央ルマ・サンプルのイントラ予測モードが使用される。それは2つの問題を生じさせる：

a. （例えば、図24（a）及び図24（b）のTL、現在のクロマ・ブロックの左上クロマ・サンプルの対応するルマ・サンプルである）左上ルマ・サンプルをカバーするコーディング・ブロックがMIPモードでコーディングされ、（例えば、図25のCRである）中央のルマ・サンプルをカバーするコーディング・ブロックがノーマル・イントラ・モードでコーディングされる場合、このケースでは、デフォルト・モードが使用されてクロマ・イントラ予測モードに設定され、それは、現在のクロマ・ブロックとCRをカバーするルマ・ブロックとの間の相関を遮断する。より低いコーディング・パフォーマンスが生じることになる。

b. （例えば、図24（a）及び図24（b）のTL、現在のクロマ・ブロックの左上クロマ・サンプルの対応するルマ・サンプルである）左上ルマ・サンプルをカバーするコーディング・ブロックがイントラ・モードでコーディングされ、（例えば、図25のCRである）中央のルマ・サンプルをカバーするコーディング・ブロックがIBC/パレット/MIPでコーディングされる場合、このケースでは、CRをカバーするコーディング・ブロックからのイントラ予測モードが、クロマDMモードを導出するために使用される。しかしながら、MIP/IBC/パレット・モードに関連するこのようなイントラ予測モードの定義はない。

【0176】

4 残差コーディングのコンテキスト・モデリングのための方法例

ここで開示される技術の実施形態は、既存の実装の欠点を克服し、それによって、より高いコーディング効率を有するビデオ・コーディングを提供する。開示される技術に基づく残差コーディングのためのコンテキスト・モデリングのための方法は、現存する及び将来の双方のビデオ・コーディング規格を強化する可能性があり、種々の実装について説明される以下の実施例において説明される。以下に提供される開示される技術の実施例は、一般的な概念を説明しており、限定として解釈されるようには意図されていない。実施例において、明示的に別意を指定しない限り、これらの実施例で説明される種々の特徴は組み合わせることが可能である。

【0177】

以下の説明において、“ブロック”とは、コーディング・ユニット（CU）、変換ユニット（TU）、又はビデオ・データの何らかの矩形領域を指す可能性がある。“現在のブロック”とは、復号化／符号化されたコーディング・ユニット（CU）、現在復号化／符号化されつつある変換ユニット（TU）、又はビデオ・データの何らかの復号化／符号化されつつある矩形コーディング領域を指す可能性がある。“CU”又は“TU”は、“コーディング・ブロック”及び“変換ブロック”としても知られている可能性がある。

【0178】

これらの実施例において、RSTはJVET-N0193における設計の如何なるバリエーションであってもよい。RSTは、二次変換を1つのブロックに適用するか、又はある変換を、変換スキップ（TS）コーディングされたブロックに適用することが可能な如何なる技術（例えば、TSコーディングされたブロックに適用されるJVET-N0193で提案されたRST）であってもよい。

【0179】

以下“ノーマル・イントラ予測モード”とは、DCモード、平面モード、及び角度イントラ予測モード（例えば、広角イントラ予測モードを更に含むことが可能であるもの）のような、隣接するピクセルをある方向から外挿（extrapolating）することによって予測信号が生成される従来のイントラ予測方法を指すために使用される。ノーマル・イントラ予測モードを使用せずにコーディングされるブロックについては、ブロックは、少なくとも1つのコーディング方法、例えば、IBC、MIP、パレット、BDPCMイントラ予測モードでコーディングされる可能性がある。

【0180】

更に、‘ゼロ・アウト領域’又は‘ゼロ・アウトCG’は、二次変換プロセスで使用される縮小された変換サイズに起因して、常にゼロ係数を有するこれらの領域/CGを示してもよい。例えば、二次変換サイズが 16×32 であり、CGサイズが 4×4 である場合、それは最初の2つのCGに適用されるであろうが、最初のCGだけが非ゼロ係数を有する可能性があり、2番目の 4×4 CGもゼロ・アウトCGと言及される。

RSTにおける変換行列の選択

1. RSTが適用されるサブ領域は、ブロックの左上部分ではないサブ領域である可能性がある。

a. 一例では、RSTは、ブロックの右上、右下、左下、又は中央のサブ領域に適用される可能性がある。

b. RSTが適用されるサブ領域は、イントラ予測モード及び／又は一次変換行列（例えば、DCT-II、DST-VII、恒等変換）に依存する可能性がある。

2. RSTで使用される変換セット及び／又は変換行列の選択は、色成分に依存する可能性がある。

a. 一例では、変換行列の1セットがルマ（又はG）成分に対して使用され、1セットがクロマ成分（又はB/R）成分に対して使用されてもよい。

b. 一例では、各色成分は、1セットに対応してもよい。

c. 一例では、少なくとも1つの行列は、異なる色成分に対する2つ又は複数のセットの何れにおいても相違する。

3. RSTで使用される変換セット及び／又は変換行列の選択は、イントラ予測方法（例えば、CCLM、多重参照ライン・ベースのイントラ予測法、行列ベースのイントラ予測法）に依存する可能性がある。

a. 一例では、変換行列の1セットがCCLMコーディングされたブロックに対して使用され、他のものが非CCLMコーディング・ブロックに対して使用されてもよい。

b. 一例では、変換行列の1セットがノーマルなイントラ予測コーディングされたブロックに対して使用され、他のものが多重参照ラインが有効なブロック（即ち、イントラ予測のために隣接するラインを使用しないもの）に対して使用されてもよい。

c. 一例では、変換行列の1セットがジョイント・クロマ残差コーディングを用いたブロックに対して使用され、他のものはジョイント・クロマ残差コーディングが適用されないブロックに対して使用されてもよい。

d. 一例では、少なくとも1つの行列は、異なるイントラ予測方法に対する2つ又は複数のセットの何れにおいても相違する。

e. 代替的に、RSTは、特定のイントラ予測方向及び／又は特定のコーディング・ツール、例えば、CCLM、及び／又はジョイント・クロマ残差コーディング、及び／又は特定の色成分（例えば、クロマ）でコーディングされたブロックに対してディセーブルにされてもよい。

4. デフォルトのイントラ予測モードは、MIP、IBC、パレットのようなノーマル・イントラ予測モードでコーディングされていないブロック（例えば、CU/PU/CB/PB）に割り当てられる可能性がある。

a. デフォルト・モードは、コーディング方法（MIP/IBC/パレット）に依存する可能性がある。

10

20

30

40

50

b. デフォルト・モードは、シグナリングされてもよいし、又はオン・ザ・フライで導出されてもよい。

c. デフォルト・モードは、クロマ導出モード（DM）の導出で利用されてもよい。

d. デフォルト・モードは、他のブロックのイントラ予測モードを予測するために使用される可能性がある。例えば、現在のブロックの後続ブロックをコーディングするため最確モード（most-probable-mode, MPM）リストの導出に利用されてもよい。

e. ある色成分（例えばルマ）のブロックに割り当てられるデフォルト・モードは、別の色成分（例えばクロマ）の変換セット又は変換インデックスの導出に利用されてもよい。

f. 代替的に、更に、デフォルト・モードは、予測モード（例えば、イントラ／インター／IBC）と一緒に記憶されてもよい。

g. デフォルト・モードは、インター・コーディングされたブロックには割り当てられない可能性がある。

5. クロマ・イントラ予測モード導出プロセスの全ての動作において、クロマ・コーディング・ブロックの同一の予め決定された位置をカバーする1つのルマ・コーディング・ブロック（対応するルマ・ブロックとして知られている場合がある）の情報を使用することが提案される。例えば、クロマ・コーディング・ブロックの同じ所定の位置をカバーする同じルマ・コーディング・ブロックは、ルマ・コーディング・ブロックの予測モード（又はIBC/MIP/パレットのようなコーディング方法）をチェックするために使用され、ルマ・コーディング・ブロックのイントラ予測モードを取って来るために使用される。

a. 一例では、同じ所定の位置が、現在のクロマ・ブロックの中央クロマ・サンプルの対応するルマ・サンプル（例えば、図25のCR）に関連付けられるように定義される。

b. 一例では、同じ所定の位置が、現在のクロマ・ブロックの左上クロマ・サンプルの対応するルマ・サンプル（例えば、図24bのTL）に関連付けられるように定義される。

c. 代替的に、更に、同じ所定の位置をカバーするコーディング・ブロックがIBC/パレット/MIPモード/何らかの他の非ノーマル・イントラ予測モードでコーディングされる場合、デフォルト・イントラ予測モードが、クロマ・イントラ予測モードを導出するために使用されてもよい。それ以外の場合、同じ所定の位置をカバーするコーディング・ブロックの復号化されたイントラ予測モードが、クロマ・イントラ予測モードを導出するために使用されてもよい。

6. 非ノーマル・イントラ予測モード（例えば、CCLM）でコーディングされたクロマ・ブロックに関し、対応するルマ・ブロックが非ノーマル・イントラ予測モード（例えば、IBC/パレット/MIP）でコーディングされる場合、RST又はその他のコーディング・ツールで使用される変換セット／変換マトリクスは、1つ以上のデフォルト・モード又はデフォルト変換セットから導出されてもよい。

a. デフォルトの変換セット／変換行列を使用するか、或いは対応するルマ・ブロックのルマ・イントラ予測モードから変換セット／変換行列を導出するかは、対応するルマ・ブロックのコーディング・モードに依存してもよい。

i. 一例では、対応するルマ・ブロックがノーマル・イントラ予測モード及び／又はBDPCMでコーディングされる場合、変換セット／変換行列は、対応するルマ・ブロックに関連するルマ・イントラ予測モード（例えば、発明項目5で定義されるルマ・ブロック又は同等位置のルマ・ブロック）に従って導出されてもよい。

ii. 一例では、対応するルマ・ブロックが非ノーマル・イントラ予測モード（例えば、IBC／パレット及び／又はBDPCM）でコーディングされる場合、デフォルト変換セット（例えば、K（例えば、K=0）に等しいセット・インデックスを有するもの）が使用されてもよい。

b. ある色成分のブロックに対してRSTをイネーブルにするかどうかは、別の

10

20

30

40

50

色成分における1つ以上の対応するブロックのコーディング方法及び／又は現在のブロックのコーディング方法に依存してもよい。

i. クロマ・ブロックに対してRSTをイネーブルにするかどうかは、1つ以上の対応するルマ・ブロックのコーディング方法及び／又は現在のクロマ・ブロックのコーディング方法に依存してもよい。

ii. 一例では、対応するルマ・ブロックがノーマル・イントラ予測モード及び／又はBDPCMでコーディングされている場合、RSTはクロマ・ブロックに対してイネーブルにされてもよい。

iii. 一例では、対応するルマ・ブロックが非ノーマル・イントラ予測モード（例えば、IBC／パレット及び／又はBDPCM）でコーディングされている場合、RSTはクロマ・ブロックに対してディセーブルにされてもよい。

10

1) 代替的に、更に、対応するルマ・ブロックが非ノーマル・イントラ予測モード（例えば、IBC／パレット及び／又はBDPCM）でコーディングされている場合、RSTは、非ノーマル・イントラ予測モード（例えば、CCLM）でコーディングされたクロマ・ブロックに対してディセーブルにされてもよい。

c. コロケートッド・ルマ・ブロックがノーマル・イントラ予測モードでコーディングされていない場合、所定のイントラ予測モードが割り当てられてもよく、それは次いでRSTで使用される変換セット及び／又は変換行列の選択に使用される（例えば、セクション2.4.2.2.3に示される変換セット選択テーブルに従う）。

i. 一例では、コロケートッド・ルマ・ブロックがIBC（イントラ・ブロック・コピー）モードでコーディングされる場合、第1の所定のイントラ予測モード（例えば、DC又は平面モード）が割り当てられてもよい。

20

ii. 一例では、コロケートッド・ルマ・ブロックがパレット・モードでコーディングされる場合、第2の所定のイントラ予測モード（例えば、DC又は平面モード）が割り当てられてもよい。

iii. 一例では、コロケートッド・ルマ・ブロックがMIPモードでコーディングされる場合、第3の所定のイントラ予測モード（例えば、平面又はDCモード）が割り当てられてもよい。

iv. 一例では、コロケートッド・ルマ・ブロックがブロック・ベースのデルタ・コード変調（Block-based Delta Pulse Code Modulation, BDPCM）モードでコーディングされる場合、第4の所定のイントラ予測モード（例えば、平面又はDCモード）が割り当てられてもよい。

30

v. 一例では、所定のイントラ予測モードの各々について、それは、DCモード、平面モード、垂直モード、水平モード、45度モード、又は135度モードであってもよい。

vi. 代替的に、コロケートッド・ルマ・ブロックがMIPモードでコーディングされる場合、MIPモードは、コロケートッド・ルマ・ブロックのディメンジョン及びMIPモードに従って、例えばセクション2.2.2.2 “MIPとイントラ予測モードの間のマッピングの仕様” におけるテーブルを使用することによって、イントラ予測モードにマッピングされてもよい。

40

vii. 所定のモードは、DCモード及び平面モードのような候補から適応的に選択されてもよい。

1) 所定のモードは、エンコーダからデコーダへシグナリングされてもよい。

2) 所定のモードは、エンコーダからデコーダへ導出されてもよい。

a. 例えば、デコーダが、現在のピクチャはスクリーン・コンテンツ・ピクチャであると判断した場合に、所定のモードはDCモードであってもよく、それ以外の場合に、所定のモードは平面モードであってもよい。

viii. 所定のモードは、クロマDM導出プロセス（例えば、サブ条項8.4.

50

3におけるlumaIntraPredMode)に使用されるものと同じ方法で定義されてもよい。

d. コロケータッド・ルマ・ブロックがノーマル・イントラ予測モード又は／及びBDPCMモードでコーディングされる場合、クロマ・イントラ予測モードは、DMモード（セクション2.7で規定されるようなもの）と同様に、同等位置のルマ・ブロックのイントラ予測モードに依存して導出されてもよい。次いで、導出されるクロマ・イントラ予測モードは、RSTで使用される変換セット及び／又は変換行列の選択に使用される（例えば、セクション2.6に示されている変換セット選択テーブルに従う）。

e. 上記の項目において、コロケータッド・ルマ・ブロックは、図24bのTLや図25のCRのように、特定のルマ位置をカバーするコーディング・ブロックであってもよい。

10

7. RSTで使用される変換セット及び／又は変換行列の選択は、一次変換に依存する可能性がある。

a. 一例では、1つのブロックに適用される一次変換が恒等変換である場合（例えば、TSモードが1つのブロックに適用される）、RSTで使用される変換セット及び／又は変換行列は、他の種類の一次変換とは異なる可能性がある。

b. 一例では、1つのブロックに適用される水平及び垂直1-D一次変換が同じ基底（例えば、双方ともDCT-II）である場合、RSTで使用される変換セット及び／又は変換行列は、異なる方向（垂直又は水平）に対する異なる基底による一次変換とは異なる可能性がある。

RSTサイド情報のシグナリング及び残差コーディング

20

8. RSTのサイド情報（例えば、st_idx）をシグナリングするかどうか、及び／如何にして行うかは、（スキヤニング順序で）ブロック内で最後の非ゼロ係数に依存する可能性がある。

a. 一例では、最後の非ゼロ係数が、RSTが適用されたCG内に位置する場合に限り、RSTはイネーブルにされてもよく、RSTのインデックスがシグナリングされてもよい。

b. 一例では、最後の非ゼロ係数が、RSTが適用されたCG内に位置していない場合、RSTはディセーブルにされ、RSTのシグナリングはスキップされる。

9. RSTのサイド情報（例えば、st_idx）をシグナリングするかどうか、及び／如何にして行うかは、CU内の全ての利用可能な色成分ではなく、特定の色成分の係数に依存する可能性がある。

30

a. 一例では、RSTのサイド情報をシグナリングするかどうか、及び／如何にして行うかを決定するために、サイド情報のみが使用される可能性がある。

i. 代替的に、更に、上記の方法はブロックのディメンジョンが特定の条件を充足している場合にのみ適用される。

1) 条件は $W < T1$ 又は $H < T2$ である。

2) 例えば、 $T1=T2=4$ である。従って、 4×4 CUの場合、ルマ・ブロック・サイズは 4×4 であり、 $4:2:0$ フォーマットにおける2つのクロマ・ブロックは 2×2 であり、この場合、ルマ情報のみが使用される可能性がある。

ii. 代替的に、更に、上記の方法は現在のパーティション・タイプ・ツリーがシングル・ツリーである場合にのみ適用される。

40

b. 1つの色成分の情報を使用するか、又は全ての色成分の情報を使用するかは、ブロックのディメンジョン／コーディングされた情報に依存する可能性がある。

7. RSTのサイド情報（例えば、st_idx）をシグナリングするかどうか、及び／如何にして行うかは、ブロック全体ではなく、1つのブロックの部分的な領域内の係数に依存する可能性がある。

a. 一例では、部分的な領域は、RSTが適用されるCGとして定義されてもよい。

b. 一例では、部分的な領域は、ブロックのスキヤニング順序又は逆のスキヤニング順序における最初の又は最後のM個の（例えば、 $M=1$ 、又は2）CGとして定義されてもよい。

50

i. 一例では、Mはブロック・ディメンジョンに依存する可能性がある。
 ii. 一例では、ブロック・サイズが $4 \times N$ 及び／又は $N \times 4$ ($N > 8$)である場合に、Mは2に設定される。

iii. 一例では、ブロック・サイズが 4×8 及び／又は 8×4 及び／又は $W \times H$ ($W \geq 8$, $H \geq 8$)である場合に、Mは1に設定される。

c. 一例では、ディメンジョン $W \times H$ を有するブロックの情報（例えば、ブロックの非ゼロ係数の数）は、RSTの使用及び／又はRST関連情報のシグナリングを決定するために考慮に入れることが許容されない可能性がある。

i. 例えば、 $W < T1$ 又は $H < T2$ である場合に、ブロックの非ゼロ係数の数はカウントされない可能性がある。例えば、 $T1 = T2 = 4$ である。

d. 一例では、部分的な領域は、ディメンジョン $W \times H$ を有する現在のブロックの左上 $M \times N$ 領域として定義される可能性がある。

i. 一例では、MはWより小さい可能性があり、及び／又はNはHより小さい可能性がある。

ii. 一例では、M及びNは固定数である可能性があり、例えば、 $M = N = 4$ である。

iii. 一例では、M及び／又はNは、W及び／又はHに依存する可能性がある。

iv. 一例では、M及び／又はNは、最大許容変換サイズに依存する可能性がある。

1) 例えば、Wが8より大きく、Hが4に等しい場合、 $M = 8$ 及び $N = 4$ である。

2) 例えば、Hが8より大きく、Wが4に等しい場合、 $M = 4$ 及び $N = 8$ である。

3) 例えば、上記の2つの条件が何れも満たさない場合、 $M = 4$ 及び $N = 4$ である。

v. 代替的に、更に、これらの方法は、7.cの条件が満たされていないような、特定のブロック寸法に対してのみ適用されてもよい。

e. 一例では、部分的な領域は、全てのブロックに対して同一であってもよい。

i. 代替的に、これはブロック・ディメンジョン及び／又はコーディングされた情報に基づいて変更されてもよい。

f. 一例では、部分的な領域は、スキャニング順序インデックスの所与のレンジに依存する可能性がある。

i. 一例では、部分的な領域は、ディメンジョン $W \times H$ を有する現在のブロックの係数スキャン順序（例えば、逆の復号化順序）に基づいて、両端を含む $[dxS, idxE]$ 内でそれらのスキャニング順序インデックスを伴う特定のレンジ内に位置する係数をカバーするものであってもよい。

1) 一例では、 $idxS$ は0に等しい。

2) 一例では、 $idxE$ は $W \times H - 1$ より小さい可能性がある。

3) 一例では、 $idxE$ は固定数である可能性がある。例えば、 $idxE = 15$ である。

4) 一例では、 $idxE$ はW及び／又はHに依存する可能性がある。

a. 例えば、Wが8より大きく、Hが4に等しい場合、 $idxE = 31$ である。

b. 例えば、Hが8より大きく、Wが4に等しい場合、 $idxE = 31$ である。

c. 例えば、Wが8に等しく、Hが8に等しい場合、 $idxE = 7$ である。

d. 例えば、Wが4に等しく、Hが4に等しい場合、 $idxE = 7$ である。

10

20

30

40

50

e. 例えば、上記の2つの条件a)とb)の何れも満たされない場合、 $IdxE=15$ である。

f. 例えば、上記の2つの条件a),b),c)及びd)の何れも満たされない場合、 $IdxE=15$ である。

g. 例えば、上記の2つの条件c)とd)の何れも満たされない場合、 $IdxE=15$ である。

ii. 代替的に、更に、これらの方法は7.cの条件が満たされていないような、特定のブロック寸法に対してのみ適用されてもよい。

g. 一例では、これは部分的な領域内の非ゼロ係数の位置に依存する可能性がある。

h. 一例では、これは部分的な領域内の非ゼロ係数のエネルギー（2乗和又は絶対値の和など）に依存する可能性がある。

i. 一例では、これはブロック全体ではなく、1ブロックの部分的な領域内の非ゼロ係数の数に依存する可能性がある。

i. 代替的に、これはCU内の1つ以上のブロックの部分的な領域内の非ゼロ係数の数に依存する可能性がある。

ii. 1ブロックの部分的な領域内の非ゼロ係数の数が閾値より小さい場合、RSTのサイド情報のシグナリングはスキップされてもよい。

iii. 一例では、閾値はNであるように固定される（例えば、 $N=1$ 又は2）。

iv. 一例では、閾値は、スライス・タイプ／ピクチャ・タイプ／パーティション・ツリー・タイプ（デュアル又はシングル）／ビデオ・コンテンツ（スクリーン・コンテンツ又はカメラでキャプチャされたコンテンツ）に依存してもよい。

v. 一例では、閾値は、4:2:0又は4:4:4のようなカラー・フォーマット、及び／又はY又はCb/Crのような色成分に依存してもよい。

11. RSTが適用される可能性のあるCGに非ゼロ係数が存在しない場合、RSTはディセーブルにされるものとする。

a. 一例では、RSTが1ブロックに適用される場合、RSTが適用される少なくとも1つのCGは、少なくとも1つの非ゼロ係数を含まなければならない。

b. 一例では、 $4 \times N$ 及び／又は $N \times 4$ ($N \geq 8$) に関し、RSTが適用される場合、最初の2つの 4×4 CGは少なくとも1つの非ゼロ係数を含まなければならない。

c. 一例では、 4×8 及び／又は 8×4 に関し、RSTが適用される場合、左上 4×4 は少なくとも1つの非ゼロ係数を含まなければならない。

d. 一例では、 $W \times H$ ($W \geq 8$ 及び $H \geq 8$) に関し、RSTが適用される場合、左上 4×4 は少なくとも1つの非ゼロ係数を含まなければならない。

e. 適合ビットストリームは、上記の1つ又は複数の条件を満たさなければならない。

12. RST関連のシンタックス要素は、残差（例えば、変換係数／直接的に量子化されるもの）をコーディングする前にシグナリングされてもよい。

a. 一例では、ゼロ・アウト領域における非ゼロ係数の数（例えば、numZeroOutSigCoeff）及びブロック全体における非ゼロ係数の数（例えば、numSigCoeff）をカウントすることは、係数の解析プロセスで除外される。

b. 一例では、RST関連のシンタックス要素（例えば、st_idx）は、residual_codingの前にコーディングされてもよい。

c. RST関連シンタックス要素は、条件付きで（例えば、コーディングされたブロック・フラグ、TSモードの使用法に従って）シグナリングされてもよい。

vi. 一例では、RST関連シンタックス要素（例えば、st_idx）は、コーディングされたブロック・フラグのシグナリングの後、又はTS/MTS関連シンタックス要素のシグナリングの後にコーディングされてもよい。

vii. 一例では、TSモードがイネーブルにされている場合（例えば、復

10

20

30

40

50

号化された`transform_skip_flag`が1に等しい場合)、RST関連シンタックス要素のシグナリングはスキップされる。

d. 残差関連シンタックスは、ゼロ・アウトCGに対してシグナリングされない可能性がある。

e. 残差をどのようにコーディングするか(例えば、スキヤニング順序、二進化、復号化されるシンタックス、コンテキスト・モデリング)は、RSTに依存してもよい。

i. 一例では、対角線右上向きスキヤニング順序ではなく、ラスタ・スキヤニング順序が適用されてもよい。

1) ラスタ・スキヤニング順序は左から右へ、上から下へ、又はその逆の順序である。

2) 代替的に、対角線右上向きスキヤニング順序の代わりに、垂直スキヤニング順序(上から下へ、左から右へ、又はその逆の順序)が適用されてもよい。

3) 代替的に、更に、コンテキスト・モデリングが修正されてもよい。

a. 一例では、コンテキスト・モデリングは、右、下、右下の近隣を使用するのではなく、テンプレートの中で、スキャン順序で最近のN個の近隣である以前にコーディングされた情報に依存してもよい。

b. 一例では、コンテキスト・モデリングは、スキャンされたインデックスに従って、テンプレートの中で、以前にコーディングされた情報(例えば、現在のインデックスが0に等しいと仮定して、-1, -2, ...)に依存してもよい。

ii. 一例では、異なる二値化方法(例えば、ライス・パラメータ導出)を適用して、RSTコーディングされたブロック及びRSTコーディングされていないブロックに関連する残差をコーディングしてもよい。

iii. 一例では、特定のシンタックス要素のシグナリングは、RSTコーディングされたブロックに対してスキップされてもよい。

1) RSTが適用されるCGに対するCGコード化ブロック・フラグ(`coded_sub_block_flag`)のシグナリングはスキップされてもよい。

a. 一例では、RST 8×8 が対角線スキャン順序で最初の3つのCGに適用された場合、CGコード化ブロック・フラグのシグナリングは、第2及び第3CG、例えば、ブロックの左上 8×8 領域の右上 4×4 CG及び左下 4×4 CGについてスキップされる。

i. 代替的に、更に、対応するCGコード化ブロック・フラグは0であると推定される、即ち、全ての係数はゼロである。

b. 一例では、RSTが1ブロックに適用される場合に、CGコード化ブロック・フラグのシグナリングは、スキヤニング順序で最初のCG(又は逆のスキヤニング順序で最後のCG)に対してスキップされる。

ii. 代替的に、更に、ブロック内の左上CGに対するCGコード化ブロック・フラグは、1であると推定される、即ち、それは少なくとも1つの非ゼロ係数を含む。

c. 8×8 ブロックの一例が図21に示されている。RST 8×8 又はRST 4×4 が 8×8 ブロックに適用される場合、CG0の`coded_sub_block_flag`は1であると推定され、CG1及びCG2の`coded_sub_block_flag`は0であると推定される。

2) 特定の座標に対する係数の大きさ及び／又は符号フラグのシグナリングはスキップされてもよい。

a. 一例では、スキャン順序における1つのCGに対するインデックスが、非ゼロ係数が存在し得る最大許容インデックスより小さくない場合(例えば、セクション0における`nonZeroSize`)、係数のシグナリングはスキップされてもよい。

b. 一例では、`sig_coeff_flag`, `abs_level_gtX_flag`, `par_level_flag`, `abs_remainder`, `coeff_sign_flag`, `dec_abs_level`のようなシンタックス要素のシグナリングは、スキップされてもよい。

10

20

30

40

50

3) 代替的に、残差のシグナリング（例えば、CGコード化ブロック・フラグ、係数の大きさ、及び／又は特定の座標の符号フラグ）は維持されるかもしれないが、コンテキスト・モデリングは、他のCGと異なるように修正される可能性がある。

iv. 一例では、RSTが適用されるCG及び他のCGにおける残差のコーディングは異なってもよい。

1) 上記のサブ項目に関し、それらはRSTが適用されるCGにのみ適用されてもよい。

10. RST関連シンタックス要素は、変換スキップ及び／又はMTSインデックスのような他の変換指示の前にシグナリングされてもよい。

a. 一例では、変換スキップのシグナリングは、RST情報に依存してもよい。

i. 一例では、RSTがブロック内で適用される場合、変換スキップ指示はシグナリングされず、ブロックについて0であると推定される。

b. 一例では、MTSインデックスのシグナリングは、RST情報に依存してもよい。

i. 一例では、RSTがブロック内で適用される場合、1つ以上のMTS変換指示はシグナリングされず、ブロックについて使用されないと推定される。

11. 1つのブロック内の異なる部分に対する算術符号化において、異なるコンテキスト・モデリング法を使用することが提案される。

a. 一例では、ブロックは、スキニング順序における最初のM個のCGと残りのCGという2つの部分であるように扱われる。

i. 一例において、Mは1に設定される。

ii. 一例において、Mは $4 \times N$ 及び $N \times 4$ ($N \geq 8$) ブロックに対して2に設定され、他のすべての場合に対して1に設定される。

b. 一例では、ブロックは、RSTが適用されるサブ領域とRSTが適用されないサブ領域、という2つの部分であるように扱われる。

i. RST 4×4 が適用される場合、RSTが適用されるサブ領域は、現在のブロックの最初の1つ又は2つのCGである。

ii. RST 4×4 が適用される場合、RSTが適用されるサブ領域は、現在のブロックの最初の3つのCGである。

c. 一例では、1つのブロック内の最初の部分に対するコンテキスト・モデリング・プロセスにおいて、以前にコーディングされた情報の使用をディセーブルにするが、2番目の部分に対してはそれをイネーブルにすることが提案される。

d. 一例では、第1CGを復号化する場合に、残りの1つ以上のCGの情報は、使用されることを許容されない場合がある。

i. 一例では、第1CGに対してCGコード化ブロック・フラグをコーディングする場合に、第2CGの値（例えば右又は下）は考慮されない。

ii. 一例では、第1CGに対するCGコード化ブロック・フラグをコーディングする場合に、第2及び第3CGの値（例えば、 $W \times H$ ($W \geq 8$ 及び $H \geq 8$)) に対する右及び下のCG) は考慮されない。

iii. 一例では、現在の係数をコーディングする場合に、コンテキスト・テンプレートにおける近隣が異なるCG内にある場合、この近隣からの情報は、使用されることを禁止される。

e. 一例では、RSTが適用される領域内の係数を復号化する場合に、RSTが適用されない残りの領域の情報は、使用されることを禁止される場合がある。

f. 代替的に、更に、上記方法は特定の条件下で適用されてもよい。

i. 条件は、RSTがイネーブルであるかどうかを含んでもよい。

ii. 条件は、ブロック・ディメンジョンを含んでもよい。

RSTサイド情報の算術コーディングにおけるコンテキスト・モデリング

15. RSTインデックスをコーディングする場合に、コンテキスト・モデリングは、明示的又は暗黙的な多重変換選択 (multiple transform selection, MTS) がイネーブ

ルにされているかどうか依存してもよい。

a. 一例では、暗黙のMTSがイネーブルである場合に、同じイントラ予測モードでコーディングされたブロックに対して、異なるコンテキストが選択されてもよい。

i. 一例では、形状（正方形又は非正方形）のようなブロック・ディメンジョンが、コンテキストを選択するために使用される。

b. 一例では、明示的なMTSに対してコーディングされる変換インデックス（例えば、tu_mts_idx）をチェックする代わりに、変換行列の基底が使用されてもよい。

i. 一例では、水平及び垂直1-D変換の両方についてDCT-IIを用いた変換行列の基底に対して、対応するコンテキストは、他の種類の変換行列と相違していてもよい。

16. RSTインデックスをコーディングする場合に、コンテキスト・モデリングはCCLMがイネーブルにされているかどうか依存してもよい（例えば、sps_cclm_enabled_flag）。

a. 代替的に、RSTインデックス・コーディングのためのコンテキストを選択することをイネーブルにするかどうか、又はどのように選択するかは、CCLMが1つのブロックに適用されるかどうか依存してもよい。

b. 一例では、コンテキスト・モデリングは、現在のブロックに対してCCLMがイネーブルにされているかどうか依存してもよい。

i. 以下は一例である。

```
intraModeCtx = sps_cclm_enabled_flag ? ( intra_chroma_pred_mode[ x0 ][ y0 ] is CCLM : intra_chroma_pred_mode[ x0 ][ y0 ] is DM ) ? 1 : 0.
```

c. 代替的に、RSTインデックス・コーディングのためのコンテキストを選択することをイネーブルにするかどうか、又はどのように選択するかは、現在のクロマ・ブロックがDMモードでコーディングされるかどうか依存してもよい。

i. 以下は一例である。

```
intraModeCtx = ( intra_chroma_pred_mode[ x0 ][ y0 ] == (sps_cclm_enabled_flag ? 7:4) ) ? 1 : 0.
```

17. RSTインデックスをコーディングする場合に、コンテキスト・モデリングはブロック・ディメンジョン／分割深度（例えば、四分木深度及び／又はBT/TT深度）に依存してもよい。

18. RSTインデックスをコーディングする場合に、コンテキスト・モデリングは、カラー・フォーマット及び／又は色成分に依存してもよい。

19. RSTインデックスをコーディングする場合に、コンテキスト・モデリングは、イントラ予測モード及び／又はMTSインデックスから独立していてもよい。

20. RSTインデックスをコーディングする場合に、第1及び／又は第2ピンは、1つのコンテキストのみでコンテキスト・コーディングされるか、又はバイパス・コーディングされる可能性がある。

条件下でRSTプロセスを起動すること

21. インバースRSTプロセスを起動するかどうかは、CGコード化ブロック・フラグに依存してもよい。

a. 一例では、左上のCGコード化ブロック・フラグがゼロである場合、プロセスを起動する必要はない。

i. 一例では、左上CGコード化ブロック・フラグがゼロであり、ブロック・サイズが $4 \times N / N \times 4$ ($N \geq 8$) と等しくない場合、プロセスを起動する必要はない。

b. 一例では、スキャニング順序で最初の2つのCGコード化ブロック・フラグが両方ともゼロに等しい場合、プロセスを起動する必要はない。

i. 一例では、スキャニング順序で最初の2つのCGコード化ブロック・フラグが両方ともゼロに等しく、ブロック・サイズが $4 \times N / N \times 4$ ($N \geq 8$) に等しい場合、プロセスを起動する必要はない。

22. インバースRSTプロセスを起動するかどうかは、ブロック・ディメンジョンに依

10

20

30

40

50

存してもよい。

a. 一例では、 $4 \times 8 / 8 \times 4$ のような特定のブロック・ディメンジョンに対して、RSTはディセーブルにされてもよい。代替的に、更に、RST関連シンタックス要素のシグナリングはスキップされてもよい。

デュアル・ツリー及びシングル・ツリー・コーディングの統一

23. RSTの使用法及び／又はRST関連情報のシグナリングは、デュアル・ツリー及びシングル・ツリー・コーディングにおいて同じ方法で決定されてもよい。

a. 例えば、カウントされる非ゼロ係数の数（例えば、JVET-N0193で規定されているnumSigCoeff）が、デュアル・ツリー・コーディングのケースでT1より大きくはないか、又はシングル・ツリー・コーディングのケースでT2より大きくはない場合、RSTは適用されるべきではなく、関連する情報はシグナリングされず、ここでT1はT2に等しい。

b. 一例では、T1及びT2は両方ともN、に設定され、例えば、 $N = 1$ 又は2である。

CU内の複数のTUを考察する。

24. RSTを適用するかどうか、及び／又はどのように適用するかは、ブロック・ディメンジョン $W \times H$ に依存する可能性がある。

a. 一例では、 $W > T1$ 又は $H > T2$ ならば、RSTは適用されない場合がある。

b. 一例では、 $W > T1$ 及び $H > T2$ ならば、RSTは適用されない場合がある。

c. 一例では、 $W * H > T$ ならば、RSTは適用されない場合がある。

d. 上記の例に関し、以下が適用される：

i. 一例では、ブロックはCUである。

ii. 一例では、 $T1 = T2 = 64$ である。

iii. 一例では、T1及び／又はT2は許容最大変換サイズに依存する可能性があり、例えば、 $T1 = T2 =$ 許容最大変換サイズである。

iv. 一例では、Tは4096に設定される。

e. 代替的に、更に、RSTは適用されないことに決定された場合、関連する情報はシグナリングされなくてもよい。

22. CU内にN個の TUが存在する場合($N > 1$)、RSTの使用法及び／又はRST関連情報のシグナリングを決定するために、N個のTUのうちの唯1つのコーディングされた情報が使用される。

a. 一例では、復号化順序でのCUの最初のTUが、判定を行うために使用されてもよい。

b. 一例では、復号化順序におけるCUの左上TUが、判定を行うために使用されてもよい。

c. 一例では、特定のTUを用いる判定は、CUに1つのTUしかないケースと同じ方法で行われてもよい。

23. RSTの使用法及び／又はRST関連情報のシグナリングは、CUレベルの代わりにTUレベル又はPUレベルで実行されてもよい。

a. 代替的に、更に、CU内の異なるTU/PUは、異なる二次変換行列を選択したり、制御フラグを有効/無効にしたりしてもよい。

b. 代替的に、更に、デュアル・ツリーのケースに関し、クロマ・ブロックがコーディングされ、異なる色成分は、異なる二次変換行列を選択したり、制御フラグを有効/無効にしたりしてもよい。

c. 代替的に、どのビデオ・ユニット・レベルでRST関連の情報をシグナリングするかどうかは、パーティション・ツリー・タイプ（デュアル又はシングル）に依存してもよい。

d. 代替的に、どのビデオ・ユニット・レベルでRST関連の情報をシグナリングするかどうかは、CU/PU/TUと最大許容変換ブロック・サイズの間関係、例えばより大きい又はより小さいことに依存してもよい。

【0181】

5 開示される技術の実装例

以下の例示的な実施形態では、最新のJVET-N0193に対する変更が太字で且つイタリック体で強調されている。削除されるテキストは、二重括弧でマーキングされる（例えば、[a]）は、文字“a”の削除を示す）。

【0182】

【数29】

5.1 実施形態#1

RSTインデックスのシグナリングは、ブロック全体ではなく、ブロックのサブ領域内の非ゼロ係数の数に依存する。

7.3.6.11 残差コーディング・シンタックス

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if((tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbWidth > 4)	
log2TbWidth = 4	
else	
log2TbWidth = Min(log2TbWidth, 5)	
if(tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbHeight > 4)	
log2TbHeight = 4	
else	
log2TbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth < 2 && cIdx == 0) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2 && cIdx == 0) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][1]	

10

20

30

40

50

inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
remBinsPass1 = ((log2SbW + log2SbH) < 4 ? 8 : 32)	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = -1	
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	10
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)	
&&	
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
if(i == 0 (i == 1 && (log2TbWidth + log2TbHeight == 5)))	20
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3	
&& log2TbHeight == 3) && n >= 8 && i == 0) ((log2TbWidth >=	
3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
}	
abs_level_gt1_flag [n]	ae(v)
remBinsPass1--	
if(abs_level_gt1_flag[n]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gt3_flag [n]	ae(v)
remBinsPass1--	30
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
...	
}	
}	

代替的に、条件は以下により置換されてもよい：

if(i == 0 (i == 1 && (log2TbWidth + log2TbHeight == 5)))	
---	--

40

【 0 1 8 3 】

【 数 3 0 】

5.2 実施形態#2

RST は、特定の CG のコーディングされたブロック・フラグに従って起動されない場合がある。

8.7.4 スケーリングされた変換係数に対する変換プロセス

8.7.4.1 概要

このプロセスに対する入力は次のとおりである：

- 現在のピクチャの左上ルマ・サンプルに対する、現在のルマ変換ブロックの左上サンプルを指定するルマ・ロケーション($xTbY, yTbY$)。
- 現在の変換ブロックの幅を指定する変数 $nTbW$ 。
- 現在の変換ブロックの高さを指定する変数 $nTbH$ 。
- 現在のブロックの色成分を指定する変数 $cIdx$ 。
- スケーリングされた変換係数の $(nTbW) \times (nTbH)$ アレイ $d[x][y]$, $x = 0..nTbW - 1$, $y = 0..nTbH - 1$

10

このプロセスの出力は、残差サンプルの $(nTbW) \times (nTbH)$ アレイ $r[x][y]$ であり、 $x = 0..nTbW - 1, y = 0..nTbH - 1$ である。

変数 $bInvokeST$ は 0 に設定され、更に、以下の条件のうちの 1 つが真である場合には 1 に修正される：

- $coded_sub_block_flag[0][0]$ が 1 に等しく、 $nTbW \times nTbH \neq 32$ である場合
- $coded_sub_block_flag[0][0]$ 及び $coded_sub_block_flag[0][1]$ が 1 に等しく、 $nTbW$ が 4 に等しく、 $nTbH$ が 8 より大きい場合
- $coded_sub_block_flag[0][0]$ 及び $coded_sub_block_flag[1][0]$ が 1 に等しく、 $nTbW$ が 8 より大きく、 $nTbH$ が 4 に等しい場合

20

$bInvokeST$ が 1 に等しく、且つ $st_idx[xTbY][yTbY]$ が 0 に等しくない場合は、以下が適用される：

1. 変数 $nStSize$, $\log2StSize$, $numStX$, $numStY$ 及び $nonZeroSize$ は、以下のように導出される：
 - $nTbW$ 及び $nTbH$ 双方が 8 以上である場合、 $\log2StSize$ は 3 に設定され、 $nStOutSize$ は 48 に設定される。
 - それ以外の場合、 $\log2StSize$ は 2 に設定され、 $nStOutSize$ は 16 に設定される。
 - $nStSize$ は $(1 \ll \log2StSize)$ に設定される。
 - $nTbH$ は 4 に等しく、 $nTbW$ は 8 より大きく、 $numStX$ は 2 に等しく設定される。
 - それ以外の場合、 $numStX$ は 1 に等しく設定される。
 - $nTbW$ は 4 に等しく、 $nTbH$ は 8 より大きい場合、 $numStY$ は 2 に等しく設定される。
 - それ以外の場合、 $numStY$ は 1 に等しく設定される。
 - $nTbW$ 及び $nTbH$ 双方が 4 に等しいか、又は $nTbW$ 及び $nTbH$ 双方が 8 に等しい場合、 $nonZeroSize$ は 8 に等しく設定される。
 - それ以外の場合、 $nonZeroSize$ は 16 に等しく設定される。
2. $xSbIdx = 0..numStX - 1$ 及び $ySbIdx = 0..numStY - 1$ に関し、以下が適用される：
 - 変数アレイ $u[x]$ ($x = 0..nonZeroSize - 1$) は、以下のようにして導出される：

$$xC = (xSbIdx \ll \log2StSize) + DiagScanOrder[\log2StSize][\log2StSize][x][0]$$

$$yC = (ySbIdx \ll \log2StSize) + DiagScanOrder[\log2StSize][\log2StSize][x][1]$$

$$u[x] = d[xC][yC]$$
 - $[x]$ ($x = 0..nonZeroSize - 1$) は、条項 8.7.4.4 で指定されるような 1 次元変換プロセスを起動することによって、変数アレイ $v[x]$ ($x = 0..nStOutSize - 1$) に変換され、そのプロセスは、スケーリングされた変換係数の変換入力長 $nonZeroSize$ と、変換出力長

30

40

- nStOutSizeと、リストu[x] (x = 0..nonZeroSize - 1)と、変換セット選択のためのインデックスstPredModeIntraと、変換セットにおける変換選択のためのインデックスst_idx[xTbY][yTbY]とを入力とし、出力はリストv[x] (x = 0..nStOutSize - 1)である。変数stPredModeIntraは、条項8.4.4.2.1.で指定されるpredModeIntraに設定される。
- アレイd[(xSbIdx << log2StSize) + x][(ySbIdx << log2StSize) + y], x = 0..nStSize - 1, y = 0..nStSize - 1 は、次のように導出される :
 - stPredModeIntraが34以下であるか、又はINTRA_LT_CCLM, INTRA_T_CCLM, 又はINTRA_L_CCLMに等しい場合、以下が適用される :

$$d[(xSbIdx << \log 2StSize) + x][(ySbIdx << \log 2StSize) + y] =$$

$$(y < 4) ? v[x + (y << \log 2StSize)] : ((x < 4) ? v[32 + x + ((y - 4) << 2)] :$$

$$d[(xSbIdx << \log 2StSize) + x][(ySbIdx << \log 2StSize) + y])$$
 - それ以外の場合、以下が適用される :

$$d[(xSbIdx << \log 2StSize) + x][(ySbIdx << \log 2StSize) + y] =$$

$$(y < 4) ? v[y + (x << \log 2StSize)] : ((x < 4) ? v[32 + (y - 4) + (x << 2)] :$$

$$d[(xSbIdx << \log 2StSize) + x][(ySbIdx << \log 2StSize) + y])$$

10

変数 implicitMtsEnabled は、以下のように導出される :

- sps_mts_enabled_flag が 1 に等しく、以下の条件のうちの 1 つが真である場合、implicitMtsEnabled は 1 に等しく設定される :
 - IntraSubPartitionsSplitType はISP_NO_SPLITに等しくない。
 - cu_sbt_flag は 1 に等しく、Max(nTbW, nTbH)が 32 以下である。
 - sps_explicit_mts_intra_enabled_flag 及び sps_explicit_mts_inter_enabled_flag が両方とも 0 に等しく、CuPredMode[xTbY][yTbY]が MODE_INTRA に等しい。
- それ以外の場合、implicitMtsEnabledは0に等しく設定される。

20

水平変換カーネルを指定する変数trTypeHorと垂直変換カーネルを指定する変数trTypeVerは、以下のように導出される :

- cIdxが0より大きい場合、trTypeHorとtrTypeVerは0に等しく設定される。
- それ以外の場合、implicitMtsEnabledが1に等しいならば、以下が適用される :
 - IntraSubPartitionsSplitTypeがISP_NO_SPLITに等しくない場合、trTypeHorとtrTypeVerは、intraPredModeに依存してテーブル8-15で指定される。
 - それ以外の場合、cu_sbt_flagが1に等しいならば、trTypeHorとtrTypeVerは、cu_sbt_horizontal_flagとcu_sbt_pos_flagに依存してテーブル8-14で指定される。
 - それ以外の場合 (sps_explicit_mts_intra_enabled_flag と sps_explicit_mts_inter_enabled_flag が 0 に等しい) 、trTypeHor と trTypeVerは以下のように導出される :

$$trTypeHor = (nTbW \geq 4 \ \&\& \ nTbW \leq 16 \ \&\& \ nTbW \leq nTbH) ? 1 : 0(8-1029)$$

$$trTypeVer = (nTbH \geq 4 \ \&\& \ nTbH \leq 16 \ \&\& \ nTbH \leq nTbW) ? 1 : 0(8-1030)$$
- それ以外の場合、trTypeHor と trTypeVerは、tu_mts_idx[xTbY][yTbY]に依存してテーブル8-13で指定される。

30

変数nonZeroW とnonZeroHは、以下のように導出される :

$$nonZeroW = \text{Min}(nTbW, (trTypeHor > 0) ? 16 : 32) \quad (8-1031)$$

$$nonZeroH = \text{Min}(nTbH, (trTypeVer > 0) ? 16 : 32) \quad (8-1032)$$

40

残差サンプルの $(nTbW) \times (nTbH)$ アレイ r は、以下のように導出される：

1. $nTbH$ が 1 より大きい場合、スケーリングされた変換係数 $d[x][y]$ ($x = 0..nonZeroW - 1, y = 0..nonZeroH - 1$) の各々の（垂直）列は、条項 8.7.4.2 で規定されるような 1 次元変換プロセスを、各々の列 $x = 0..nonZeroW - 1$ について起動することによって、 $e[x][y]$ ($x = 0..nonZeroW - 1, y = 0..nTbH - 1$) に変換され、そのプロセスは、変換ブロックの高さ $nTbH$ と、スケーリングされた変換係数の非ゼロ高さ $nonZeroH$ と、リスト $d[x][y]$ ($y = 0..nonZeroH - 1$) と、 $trTypeVer$ に等しく設定される変換タイプ変数 $trType$ とを入力とし、出力はリスト $e[x][y]$ ($y = 0..nTbH - 1$) である。
2. $nTbH$ と $nTbW$ が両方とも 1 より大きい場合、中間サンプル値 $g[x][y]$ ($x = 0..nonZeroW - 1, y = 0..nTbH - 1$) は、以下のように導出される：

$$g[x][y] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, (e[x][y] + 64) \gg 7) \quad (8-1033)$$

10

$nTbW$ が 1 より大きい場合、結果のアレイ $g[x][y]$ ($x = 0..nonZeroW - 1, y = 0..nTbH - 1$) の各々の（水平）行は、条項 8.7.4.2 で規定されるような 1 次元変換プロセスを、各々の行 $y = 0..nTbH - 1$ について起動することによって、 $r[x][y]$ ($x = 0..nTbW - 1, y = 0..nTbH - 1$) に変換され、そのプロセスは、変換ブロックの幅 $nTbW$ と、結果のアレイ $g[x][y]$ の非ゼロ幅 $nonZeroW$ と、リスト $g[x][y]$ ($x = 0..nonZeroW - 1$) と、 $trTypeHor$ に等しく設定される変換タイプ変数 $trType$ とを入力とし、出力はリスト $r[x][y]$ ($x = 0..nTbW - 1$) である。

【 0 1 8 4 】

20

【 数 3 1 】

30

40

50

5.3 実施形態#3

RST インデックスのコンテキスト・モデリングを改訂する。

5.3.1 代替例#1

9.5.4.2.8 シンタックス要素 **st_idx** に対する **ctxInc** の導出プロセス

このプロセスに対する入力、色成分インデックス **cIdx** と、**cIdx** に依存する現在のピクチャの左上サンプルに対する現在のルマ又はクロマ・コーディング・ブロックの左上サンプルを指定するルマ又はクロマ位置 (**x0**, **y0**) と、ツリー・タイプ **treeType** と、条項 8.4.2 で指定されるようなルマ・イントラ予測モード **IntraPredModeY[x0][y0]** と、条項 7.4.7.5 で指定されるようなクロマ・サンプルのイントラ予測モードを指定するシンタックス要素 **intra_chroma_pred_mode[x0][y0]** と、ブロック幅 **nTbW** 及び高さ **nTbH** と、多重変換選択インデックス **tu_mts_idx[x0][y0]** である。

このプロセスの出力は変数 **ctxInc** である。

変数 **intraModeCtx** は、以下のように導出される：

cIdx が 0 に等しい場合、**intraModeCtx** は以下のように導出される：

intraModeCtx = (**IntraPredModeY[x0][y0]** <= 1) ? 1 : 0

それ以外の場合 (**cIdx** は 0 より大きい) 、**intraModeCtx** は以下のように導出される：

intraModeCtx = (**intra_chroma_pred_mode[x0][y0]** >= 4) ? 1 : 0

変数 **mtsCtx** は、以下のように導出される：

mtsCtx = ((*sps_explicit_mts_intra_enabled_flag* ? **tu_mts_idx[x0][y0]** == 0 : **nTbW** == **nTbH**)

&& **treeType** != SINGLE_TREE) ? 1 : 0

変数 **ctxInc** は、以下のように導出される：

ctxInc = (**binIdx** << 1) + **intraModeCtx** + (**mtsCtx** << 2)

5.3.2 代替例#2

シンタックス要素構造	シンタックス要素	二進化	
		プロセス	入力パラメータ
.....			
	st_idx[][]	TR	cMax = 2, cRiceParam = 0

テーブル 9-15 — コンテキスト・コード化ビンによる **ctxInc** のシンタックス要素に対する割り当て

シンタックス要素	binIdx					
	0	1	2	3	4	>= 5
.....						
st_idx[][]	0[[,1,4,5]] (clause 9.5.4.2.8)	2[[,3,6,7]] (clause 9.5.4.2.8)	na	na	na	na
.....						

[9.5.4.2.8 シンタックス要素 **st_idx** に対する **ctxInc** の導出プロセス

このプロセスに対する入力、色成分インデックス **cIdx** と、**cIdx** に依存する現在のピクチャの左上サンプルに対する現在のルマ又はクロマ・コーディング・ブロックの左上サンプルを指定するルマ又はクロマ位置 (**x0**, **y0**) と、ツリー・タイプ **treeType** と、条項 8.4.2 で指定されるようなルマ・イントラ予測モード **IntraPredModeY[x0][y0]** と、条項 7.4.7.5 で指定されるようなクロマ・サンプルのイントラ予測モードを指定するシンタックス要素 **intra_chroma_pred_mode[x0][y0]** と、多重変換選択インデックス **tu_mts_idx[x0][y0]** とである。

このプロセスの出力は変数 **ctxInc** である。

10

20

30

40

50

変数 `intraModeCtx` は、以下のように導出される：

`cIdx` が 0 に等しい場合、`intraModeCtx` は以下のように導出される：

```
intraModeCtx = ( IntraPredModeY[ x0 ][ y0 ] <= 1 ) ? 1 : 0
```

それ以外の場合（`cIdx` は 0 より大きい）、`intraModeCtx` は以下のように導出される：

```
intraModeCtx = ( intra_chroma_pred_mode[ x0 ][ y0 ] >= 4 ) ? 1 : 0
```

変数 `mtsCtx` は、以下のように導出される：

```
mtsCtx = ( tu_mts_idx[ x0 ][ y0 ] == 0 && treeType != SINGLE_TREE ) ? 1 : 0
```

変数 `ctxInc` は、以下のように導出される：

```
ctxInc = ( binIdx << 1 ) + intraModeCtx + ( mtsCtx << 2 ) ]]
```

5.4 実施形態 #4

条項 7.c 及び 7.d に対応している。

7.3.7.11 残差コーディング・シンタックス

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDeSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1 --	
if(sig_coeff_flag[xC][yC])	
inferSbDeSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
if(xC < 4 && yC < 4)	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3)) && n >= 8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
}	
abs_level_gt1_flag[n]	ae(v)
...	

代替例において、以下が適用されてもよい：

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDeSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1 --	
if(sig_coeff_flag[xC][yC])	
inferSbDeSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
if(xC < SigRangeX && yC < SigRangeY)	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3)) && n >= 8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
}	
abs_level_gt1_flag[n]	ae(v)
...	

10

20

30

40

50

一例では、以下が適用されてもよい：

$\log_2 \text{TbWidth} > 3$ && $\log_2 \text{TbHeight} == 2$ である場合、 SigRangeX は 8 に等しく、それ以外の場合には 4 に等しい。

$\log_2 \text{TbHeight} > 3$ && $\log_2 \text{TbWidth} == 2$ である場合、 SigRangeY は 8 に等しく、それ以外の場合には 4 に等しい。

5.5 実施形態 #5

条項 19 に対応している。

7.3.6.5 コーディング・ユニット・シンタックス

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
if(Min(cbWidth, cbHeight) >= 4 && sps_st_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT) {	
if((numSigCoeff > [((treeType == SINGLE_TREE) ? 2 : 1) [[]]]) && numZeroOutSigCoeff == 0) {	
st_idx[x0][y0]	ae(v)
}	
}	
}	
}	
}	

5.6 実施形態 #6

条項 20 に対応している。

7.3.6.5 コーディング・ユニット・シンタックス

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	

10

20

30

40

50

if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
if(Min(cbWidth, cbHeight) >= 4 && sps_st_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) {	
st_idx[x0][y0]	ae(v)
}	
}	
}	
}	
}	

10

20

5.7 実施形態 #7

条項 21 に対応している。

7.3.7.11 残差コーディング・シンタックス

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	Descriptor
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC] && x0 == CbX[x0][y0] && y0 == CbU[x0][y0]) {	
if(!transform_skip_flag[x0][y0]) {	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3)) && n >= 8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
abs_level_gt1_flag[n]	ae(v)
...	

30

(CbX[x0][y0], CbY[x0][y0]) は、ポジション(x0, y0)をカバーするコーディング・ユニットの左上ポジションを指定している。

40

5.8 実施形態#8

RST 変換セット・インデックスは、非ノーマル・イントラ予測モードに割り当てられたデフォルト・モードから導出される。新たに追加される部分は太字のイタリック体で強調表示されており、削除される部分は二重括弧でマークされている（**[[a]]**は文字「a」の削除を示す）。

8.7.4.1 一般的な事項

このプロセスに対する入力は以下のとおりである：

- 現在のピクチャの左上ルマ・サンプルに対する、現在のルマ変換ブロックの左上サンプルを指定するルマ位置(x_{TbY}, y_{TbY})、
- 現在の変換ブロックの幅を指定する変数 n_{TbW} 、
- 現在の変換ブロックの高さを指定する変数 n_{TbH} 、
- 現在のブロックの色成分を指定する変数 $cIdx$ 、
- スケーリングされた変換係数の $(n_{TbW}) \times (n_{TbH})$ array $d[x][y]$, $x = 0..n_{TbW}-1$, $y = 0..n_{TbH}-1$.

このプロセスの出力は、残差サンプルの $(n_{TbW}) \times (n_{TbH})$ アレイ $r[x][y]$, $x = 0..n_{TbW}-1$, $y = 0..n_{TbH}-1$ である。

$lfst_idx[x_{TbY}][y_{TbY}]$ が 0 に等しくなく、 n_{TbW} と n_{TbH} の両方が 4 以上である場合、以下が適用される：

- 変数 $predModeIntra$, $nLfstOutSize$, $log2LfstSize$, $nLfstSize$, 及び $nonZeroSize$ は次のように導出される：

```

predModeIntra =
  (  $cIdx == 0$  ) ?  $IntraPredModeY[x_{TbY}][y_{TbY}]$  :  $IntraPredModeC[x_{TbY}][y_{TbY}]$  (8-965)
 $nLfstOutSize = (n_{TbW} \geq 8 \ \&\& \ n_{TbH} \geq 8) ? 48 : 16$  (8-966)
 $log2LfstSize = (n_{TbW} \geq 8 \ \&\& \ n_{TbH} \geq 8) ? 3 : 2$  (8-967)
 $nLfstSize = 1 \ll log2LfstSize$  (8-968)
 $nonZeroSize = ((n_{TbW} == 4 \ \&\& \ n_{TbH} == 4) || (n_{TbW} == 8 \ \&\& \ n_{TbH} == 8)) ? 8 : 16$  (8-969)

```

- $intra_mip_flag[x_{TbComp}][y_{TbComp}]$ が 1 に等しく、 $cIdx$ が 0 に等しい場合、 $predModeIntra$ は $INTRA_PLANAR$ に等しく設定される。
- $predModeIntra$ が、 $INTRA_LT_CCLM$, $INTRA_L_CCLM$, 又は $INTRA_T_CCLM$ の何れかに等しい場合、 $predModeIntra$ は、次のように導出される：

[[$IntraPredModeY[x_{TbY} + n_{TbW}/2][y_{TbY} + n_{TbH}/2]$ に等しく設定される]]

- **$intra_mip_flag[x_{TbY}][y_{TbY}]$ が 1 に等しい場合、 $predModeIntra$ は $INTRA_PLANAR$ に等しく設定される。**
- **それ以外の場合、 $CuPredMode[0][x_{TbY}][y_{TbY}]$ が $MODE_IBC$ 又は $MODE_PLT$ に等しいならば、 $predModeIntra$ は $INTRA_DC$ に等しく設定される。**
- **それ以外の場合、 $predModeIntra$ は $IntraPredModeY[x_{TbY} + n_{TbW}/2][y_{TbY} + n_{TbH}/2]$ に等しく設定される。**

- 条項 8.4.5.2.6 で規定されるような広角予測モード・マッピング・プロセスは、 $predModeIntra$, n_{TbW} , n_{TbH} 及び $cIdx$ を入力とし、修正された $predModeIntra$ を出力として呼び出される。

代替的に、以下が適用されてもよい：

- $predModeIntra$ が $INTRA_LT_CCLM$, $INTRA_L_CCLM$, or $INTRA_T_CCLM$ の何れかに等しい場合、 $predModeIntra$ は、次のように導出される：

[[$IntraPredModeY[x_{TbY} + n_{TbW}/2][y_{TbY} + n_{TbH}/2]$ に等しく設定される]]

- **$intra_mip_flag[x_{TbY} + n_{TbW}/2][y_{TbY} + n_{TbH}/2]$ が 1 に等しい場合、 $predModeIntra$ は $INTRA_PLANAR$ に等しく設定される。**

10

20

30

40

50

- それ以外の場合、 $CuPredMode[0][xTbY + nTbW/2][yTbY + nTbH/2]$ が $MODE_IBC$ 又は $MODE_PLT$ に等しいならば、 $predModeIntra$ は $INTRA_DC$ に等しく設定される。
- それ以外の場合、 $predModeIntra$ は $IntraPredModeY[xTbY + nTbW/2][yTbY + nTbH/2]$ に等しく設定される。

5.9 実施形態#9

RST 変換セット・インデックスは、非ノーマル・イントラ予測モードに割り当てられたデフォルト・モードから、及びカラー・フォーマットに依存して導出される。新たに追加される部分は太字のイタリック体で強調表示されており、削除される部分は二重括弧でマークされている（[[a]]は文字「a」の削除を示す）。

10

8.7.4.2 一般的な事項

このプロセスに対する入力には以下のとおりである：

- 現在のピクチャの左上ルマ・サンプルに対する、現在のルマ変換ブロックの左上サンプルを指定するルマ位置 ($xTbY, yTbY$)、
- 現在の変換ブロックの幅を指定する変数 $nTbW$ 、
- 現在の変換ブロックの高さを指定する変数 $nTbH$ 、
- 現在のブロックの色成分を指定する変数 $cIdx$ 、
- スケーリングされた変換係数の $(nTbW) \times (nTbH)$ array $d[x][y]$, $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

このプロセスの出力は、残差サンプルの $(nTbW) \times (nTbH)$ アレイ $r[x][y]$, $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ である。

20

$lfst_idx[xTbY][yTbY]$ が 0 に等しくなく、 $nTbW$ と $nTbH$ の両方が 4 以上である場合、以下が適用される：

- 変数 $predModeIntra$, $nLfstOutSize$, $\log2LfstSize$, $nLfstSize$, 及び $nonZeroSize$ は次のように導出される：

```

predModeIntra =
  ( cIdx == 0 ) ? IntraPredModeY[ xTbY ][ yTbY ] : IntraPredModeC[ xTbY ][ yTbY ] (8-965)
nLfstOutSize = ( nTbW >= 8 && nTbH >= 8 ) ? 48 : 16 (8-966)
log2LfstSize = ( nTbW >= 8 && nTbH >= 8 ) ? 3 : 2 (8-967)
nLfstSize = 1 << log2LfstSize (8-968)
nonZeroSize = ( ( nTbW == 4 && nTbH == 4 ) || ( nTbW == 8 && nTbH == 8 ) ) ? 8 : 16 (8-969)

```

- $intra_mip_flag[xTbComp][yTbComp]$ が 1 に等しく、 $cIdx$ が 0 に等しい場合、 $predModeIntra$ は $INTRA_PLANAR$ に等しく設定される。
- $predModeIntra$ が、 $INTRA_LT_CCLM$, $INTRA_L_CCLM$, 又は $INTRA_T_CCLM$ の何れかに等しい場合、 $predModeIntra$ は、次のように導出される：

[[$IntraPredModeY[xTbY + nTbW/2][yTbY + nTbH/2]$ に等しく設定される]]

- $intra_mip_flag[xTbY][yTbY]$ が 1 に等しい場合、 $predModeIntra$ は $INTRA_PLANAR$ に等しく設定される。
- それ以外の場合、 $CuPredMode[0][xTbY][yTbY]$ が $MODE_IBC$ 又は $MODE_PLT$ に等しいならば、 $predModeIntra$ は $INTRA_DC$ に等しく設定される。
- それ以外の場合、 $predModeIntra$ は $IntraPredModeY[xTbY + nTbW/2][yTbY + nTbH/2]$ に等しく設定される。
- $chroma_format_idc$ が 2 に等しい場合、 $predModeIntra$ はテーブル 8-3 に従って更に修正される。クロマ・イントラ予測モード X は $predModeIntra$ に等しく設定され、クロマ・イントラ予測モード Y を導出する。その後、 $predModeIntra$ は Y に等しく設定される。

30

40

- 条項8.4.5.2.6で規定されるような広角予測モード・マッピング・プロセスは、predModeIntra, nTbW, nTbH 及び cIdx を入力とし、修正されたpredModeIntraを出力として呼び出される。

テーブル8-3 chroma_format_idcが2に等しい場合のクロマ・イントラ予測モードXからモードYへの4:2:2マッピング・プロセスの仕様

mode X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mode Y	0	1	61	62	63	64	65	66	2	3	4	6	8	10	12	13	14	16
mode X	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
mode Y	18	20	22	23	24	26	28	30	32	33	34	35	36	37	38	39	40	41
mode X	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
mode Y	42	43	44	44	44	45	46	46	46	47	48	48	48	49	50	51	52	52
mode X	54	55	56	57	58	59	60	61	62	63	64	65	66					
mode Y	52	53	54	54	54	55	56	56	56	57	58	59	60					

10

代替的に、以下が適用されてもよい：

- predModeIntra がINTRA_LT_CCLM, INTRA_L_CCLM, 又は INTRA_T_CCLMの何れかに等しい場合、predModeIntraは、次のように導出される：
[[IntraPredModeY[xTbY + nTbW / 2][yTbY + nTbH / 2]に等しく設定される]]
- intra_mip_flag[xTbY + nTbW / 2][yTbY + nTbH / 2]が1に等しい場合、predModeIntra は INTRA_PLANARに等しく設定される。
- それ以外の場合、CuPredMode[0][xTbY + nTbW / 2][yTbY + nTbH / 2] がMODE_IBC 又は MODE_PLTに等しいならば、predModeIntra はINTRA_DCに等しく設定される。
- それ以外の場合、predModeIntra はIntraPredModeY[xTbY + nTbW / 2][yTbY + nTbH / 2]に等しく設定される。
- chroma_format_idcが2に等しい場合、predModeIntraはテーブル8-3に従って更に修正される。クロマ・イントラ予測モードXはpreModeIntraに等しく設定され、クロマ・イントラ予測モードYを導出する。その後、predModeIntraはYに等しく設定される。

20

5.10 実施形態#10

本実施形態では、現在のクロマ・ブロックの対応するルマ領域の中央ルマ・サンプルが、MIP 又は IBC 又はパレット・モードでコーディングされるかどうかチェックされ；DMの設定は中央ルマ・サンプルにも基づいている。

新たに追加される部分は太字のイタリック体で強調表示されており、削除される部分は二重括弧でマークされている（[[a]]は文字「a」の削除を示す）。

30

8.4.3 クロマ・イントラ予測モードの導出プロセス

このプロセスに対する入力は以下のとおりである：

- 現在のピクチャの左上ルマ・サンプルに対する、現在のクロマ・コーディング・ブロックの左上サンプルを指定するルマ位置(xCb, yCb)、
- ルマ・サンプルにおける現在のコーディング・ブロックの幅を指定する変数cbWidth、
- ルマ・サンプルにおける現在のコーディング・ブロックの高さを指定する変数cbHeight、

このプロセスでは、クロマ・イントラ予測モードIntraPredModeC[xCb][yCb]が導出される。

対応するルマ・イントラ予測モードlumaIntraPredModeは次のように導出される：

- intra_mip_flag[xCb + cbWidth / 2][yCb + cbHeight / 2] が1に等しい場合、 lumaIntraPredMode はINTRA_PLANARに等しく設定される。
- それ以外の場合、CuPredMode[0][xCb + cbWidth / 2][yCb + cbHeight / 2] がMODE_IBC 又は MODE_PLTに等しい場合、lumaIntraPredMode はINTRA_DCに等しく設定される。

40

- － それ以外の場合、lumaIntraPredModeはIntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2] に等しく設定される。

クロマ・イントラ予測モード IntraPredModeC[xCb][yCb] は、テーブル 8-2 で規定されるような cclm_mode_flag, cclm_mode_idx, intra_chroma_pred_mode 及び lumaIntraPredMode を用いて導出される。

テーブル 8-2 — cclm_mode_flag, cclm_mode_idx, intra_chroma_pred_mode 及び lumaIntraPredMode に依存する IntraPredModeC[xCb][yCb] の仕様

cclm_mode_flag	cclm_mode_idx	intra_chroma_pred_mode	lumaIntraPredMode				
			0	5 0	1 8	1	X (0 <= X <= 66)
0	-	0	66	0	0	0	0
0	-	1	50	6 6	5 0	5 0	50
0	-	2	18	1 8	6 6	1 8	18
0	-	3	1	1	1	6 6	1
0	-	4	0	5 0	1 8	1	X
1	0	-	81	8 1	8 1	8 1	81
1	1	-	82	8 2	8 2	8 2	82
1	2	-	83	8 3	8 3	8 3	83

chroma_format_idc が 2 に等しい場合、クロマ・イントラ予測モード Y は、テーブル 8-3 に規定されるようなテーブル 8-2 におけるクロマ・イントラ予測モード X を用いて導出され、その後、クロマ・イントラ予測モード X はクロマ・イントラ予測モード Y に等しく設定される。

5.11 実施形態 #11

本実施形態では、現在のクロマ・ブロックの対応するルマ領域の左上ルマ・サンプルが、MIP 又は IBC 又はパレット・モードでコーディングされるかどうかチェックされ；DM の設定は中央ルマ・サンプルにも基づいている。

新たに追加される部分は太字のイタリック体で強調表示されており、削除される部分は二重括弧でマークされている（[[a]]は文字「a」の削除を示す）。

8.4.3 クロマ・イントラ予測モードの導出プロセス

このプロセスに対する入力は以下のとおりである：

- － 現在のピクチャの左上ルマ・サンプルに対する、現在のクロマ・コーディング・ブロックの左上サンプルを指定するルマ位置(xCb, yCb)、
- － ルマ・サンプルにおける現在のコーディング・ブロックの幅を指定する変数cbWidth、
- － ルマ・サンプルにおける現在のコーディング・ブロックの高さを指定する変数cbHeight、

このプロセスでは、クロマ・イントラ予測モードIntraPredModeC[xCb][yCb]が導出される。

対応するルマ・イントラ予測モードlumaIntraPredModeは次のように導出される：

10

20

30

40

50

- $\text{intra_mip_flag}[\text{xCb}][\text{yCb}]$ が 1 に等しい場合、 lumaIntraPredMode は INTRA_PLANAR に等しく設定される。
- それ以外の場合、 $\text{CuPredMode}[0][\text{xCb}][\text{yCb}]$ が MODE_IBC 又は MODE_PLT に等しい場合、 lumaIntraPredMode は INTRA_DC に等しく設定される。
- それ以外の場合、 lumaIntraPredMode は $\text{IntraPredModeY}[\text{xCb}[(+\text{cbWidth}/2)][\text{yCb}[(+\text{cbHeight}/2)]]$ に等しく設定される。

クロマ・イントラ予測モード $\text{IntraPredModeC}[\text{xCb}][\text{yCb}]$ は、テーブル 8-2 で規定されるような $\text{cclm_mode_mode_flag}$, cclm_mode_idx , $\text{intra_chroma_pred_mode}$ 及び lumaIntraPredMode を用いて導出される。

10

テーブル 8-2 — cclm_mode_flag , cclm_mode_idx , $\text{intra_chroma_pred_mode}$ and lumaIntraPredMode に依存する $\text{IntraPredModeC}[\text{xCb}][\text{yCb}]$ の仕様

cclm_mode_flag	cclm_mode_idx	$\text{intra_chroma_pred_mode}$	lumaIntraPredMode				
			0	5 0	1 8	1	X ($0 \leq X \leq 66$)
0	-	0	66	0	0	0	0
0	-	1	50	6 6	5 0	5 0	50
0	-	2	18	1 8	6 6	1 8	18
0	-	3	1	1	1	6 6	1
0	-	4	0	5 0	1 8	1	X
1	0	-	81	8 1	8 1	8 1	81
1	1	-	82	8 2	8 2	8 2	82
1	2	-	83	8 3	8 3	8 3	83

20

30

chroma_format_idc が 2 に等しい場合、クロマ・イントラ予測モード Y は、テーブル 8-3 に規定されるようなテーブル 8-2 におけるクロマ・イントラ予測モード X を用いて導出され、その後、クロマ・イントラ予測モード X はクロマ・イントラ予測モード Y に等しく設定される。

【0185】

上述の例は、以下に説明される方法、例えば方法 2200, 2210, 2220, 2230, 2240 及び 2250 の状況で組み込まれることが可能であり、これらはビデオ・デコーダ又はビデオ・エンコーダにおいて実装されてもよい。

40

【0186】

図 22A は、ビデオ処理のための例示的な方法のフローチャートを示す。方法 2200 は、ステップ 2202 において、現在のビデオ・ブロックに縮小二次変換を適用するための変換セット又は変換マトリクスを、現在のビデオ・ブロックの特徴に基づいて選択することを含む。

【0187】

本方法 2200 は、ステップ 2204 において、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換の一部として、選択された変換セット又は変換マトリクスを現在のビデオ・ブロックの一部に適用することを含む。

【0188】

50

幾つかの実施形態において、現在のビデオ・ブロックの一部は、現在のビデオ・ブロックの右上サブ領域、右下サブ領域、左下サブ領域、又は中央サブ領域である。

【0189】

幾つかの実施形態において、現在のビデオ・ブロックの特徴は、現在のビデオ・ブロックのイントラ予測モード又はプライマリ変換行列である。

【0190】

幾つかの実施形態において、特徴は現在のビデオブロックの色成分である。一例において、第1変換セットは、現在のビデオ・ブロックのルマ成分に対して選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックの1つ以上のクロマ成分に対して選択される。

【0191】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックのイントラ予測モード又はイントラ・コーディング方法である。一例では、イントラ予測方法は、多重参照ライン（MRL）ベースの予測法又は行列ベースのイントラ予測法を含む。別の例では、第1変換セットは、現在のビデオ・ブロックがクロス・コンポーネント線型モデル（CCLM）コード化ブロックである場合に選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックが非CCLMコード化ブロックである場合に選択される。更に別の例では、第1変換セットは、現在のビデオ・ブロックがジョイント・クロマ残差コーディング方法でコーディングされる場合に選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックがジョイント・クロマ残差コーディング方法でコーディングされない場合に選択される。

【0192】

幾つかの実施形態において、特徴は現在のビデオ・ブロックの一次変換である。

【0193】

図22Bは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2210は、ステップ2212において、現在のビデオ・ブロックのビットストリーム表現における縮小二次変換（RST）の適用のためのサイド情報のシグナリングの選択的な包含に関する決定を、現在のビデオ・ブロックに関連する1つ以上の係数に基づいて行うことを含む。

【0194】

方法2210は、ステップ2214において、決定に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックのビットストリーム表現を含むビデオとの間の変換を実行することを含む。

【0195】

幾つかの実施形態において、1つ以上の係数は、現在のビデオ・ブロックのスキャン順序における最後の非ゼロ係数を含む。

【0196】

幾つかの実施形態において、1つ以上の係数は、現在のビデオ・ブロックの部分的な領域内の複数の係数を含む。一例では、部分的な領域は、RSTが適用され得る1つ以上のコーディング・グループを含む。別の例では、部分的な領域は、現在のビデオ・ブロックのスキャン順序で先頭からM個のコーディング・グループ又は末尾からM個のコーディング・グループを含む。更に別の例では、部分的な領域は、現在のビデオ・ブロックの逆スキャン順序で先頭からM個のコーディング・グループ又は末尾からM個のコーディング・グループを含む。更に別の例では、決定を行うステップは、複数の係数の1つ以上の非ゼロ係数のエネルギーに更に基づいている。

【0197】

図22Cは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2220は、ステップ2222において、現在のビデオ・ブロックに縮小二次変換を適用するために、現在のビデオ・ブロックのビットストリーム表現を構成することを含み、ここで、RSTに関連するシンタックス要素は、残差情報をコーディングする前に、ビットストリーム表現においてシグナリングされる。

【0198】

方法2220は、ステップ2224において、上記の構成に基づいて、現在のビデオ・ブロックと現在のビデオ・ブロックのビットストリーム表現との間の変換を実行することを含む。

【0199】

幾つかの実施形態において、RSTに関連するシンタックス要素をシグナリングすることは、少なくとも1つのコーディングされたブロック・フラグ又は変換選択モードの使用に基づいている。

【0200】

幾つかの実施形態において、ビットストリーム表現は、全てゼロ係数を有するコーディング・グループに対応するコーディング残差情報を除外する。

【0201】

幾つかの実施態様において、コーディング残差情報は、RSTの適用に基づいている。

【0202】

図22Dは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2230は、ステップ2232において、現在のビデオ・ブロックに縮小二次変換を適用するために、現在のビデオ・ブロックのビットストリーム表現を構成することを含み、ここで、RSTに関連するシンタックス要素は、変換スキップ指示又は多重変換セット（MTS）インデックスの何れかの前に、ビットストリーム表現においてシグナリングされる。

【0203】

方法2230は、ステップ2234において、上記の構成に基づいて、現在のビデオ・ブロックと現在のビデオ・ブロックのビットストリーム表現との間の変換を実行することを含む。

【0204】

幾つかの実施形態において、変換スキップ指示又はMTSインデックスは、RSTに関連するシンタックス要素に基づいている。

【0205】

図22Eは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2240は、ステップ2242において、現在のビデオ・ブロックの特徴に基づいて、縮小二次変換（RST）のインデックスをコーディングするためのコンテキスト・モデルを構成することを含む。

【0206】

方法2240は、ステップ2244において、上記の構成に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換を実行することを含む。

【0207】

幾つかの実施形態では、特徴は、多重変換選択（MTS）プロセスの明示的又は暗示的な実施可能性である。

【0208】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックにおけるクロス・コンポーネント線型モデル（CCLM）コーディング・モードの実施可能性である。

【0209】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックのサイズである。

【0210】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックに適用されるパーティショニング・プロセスの分割深度である。一例では、パーティショニング・プロセスは、四分木（QT）パーティショニング・プロセス、二分木（BT）パーティショニング・プロセス又は三分木（TT）パーティショニング・プロセスである。

【0211】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックのカラー・フォーマット又は色成分である。

【0212】

幾つかの実施態様において、特徴は、現在のビデオ・ブロックのイントラ予測モードと

10

20

30

40

50

多重変換選択（MTS）プロセスのインデックスとを除外する。

【0213】

図22Fは、ビデオ処理のための例示的な方法のフローチャートを示す。この方法2250は、ステップ2252において、現在のビデオ・ブロックに対するインバース二次変換（RST）の選択的適用に関する決定を、現在のビデオ・ブロックの特徴に基づいて行うことを含む。

【0214】

方法2250は、ステップ2254において、決定に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換を実行することを含む。

【0215】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックのコーディング・グループのコーディングされたブロック・フラグである。一例では、インバースRSTプロセスは適用されず、ここで、左上コーディング・グループのコーディングされたブロック・フラグはゼロである。別の例では、インバースRSTプロセスは適用されず、ここで、現在のビデオ・ブロックのスキャン順序における第1及び第2コーディング・グループに対するコーディングされたブロック・フラグはゼロである。

【0216】

幾つかの実施形態において、特徴は、現在のビデオ・ブロックの高さ（M）又は幅（N）である。一例では、インバースRSTプロセスは適用されず、ここで、(i) $M = 8$ 及び $N = 4$ 、又は (ii) $M = 4$ 及び $N = 8$ である。

【0217】

図26Aは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2610は、ステップ2612において、ビデオのビデオ・ユニットの現在のビデオ・ブロックとビデオのコーディングされた表現との間の変換について、特定のイントラ予測モードを使用してコーディングされたビデオ・ユニットのデフォルト・イントラ予測モードを決定し、その結果、現在のビデオ・ブロックの予測ブロックが、ある方向に沿って現在のビデオ・ブロックの隣接するピクセルを外挿することなく生成されるようにすることを含む。方法2610は、更に、ステップ2614において、決定に基づいて変換を実行することを含む。

【0218】

図26Bは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2620は、ステップ2622において、ルールを使用して、ビデオのクロマ・ブロックの所定の位置をカバーするビデオのルマ・ブロックの決定を行うことを含む。方法2620は、ステップ2624において、決定に基づいて、ビデオとビデオのコーディングされた表現との間の変換を実行することを更に含む。幾つかの実装形態において、クロマ・ブロックは、イントラ予測モードを用いてコーディングされた表現で表現される。

【0219】

図26Cは、ビデオ処理のための例示的な方法のフローチャートを示す。方法2630は、ステップ2632において、ルールを使用して、クロマ・ブロックに対応するルマ・ブロックのコーディング・モードに基づいて、ビデオのクロマ・ブロックのイントラ予測モードを導出することを含む。方法2630は、ステップ2634において、導出されたイントラ予測モードに基づいて、そのクロマ・ブロックとビデオのコーディングされた表現との間の変換を実行することを更に含む。幾つかの実装形態において、ルマ・ブロックの符号化モードが、特定のイントラ予測モードであって、ルマ・ブロックの予測ブロックが、ルマ・ブロックの隣接するピクセルを或る方向に沿って外挿することなく生成されるものである場合に、ルールは、デフォルトのイントラ予測モードを使用することを規定している。

【0220】

図27Aはビデオ処理のための例示的な方法のフローチャートを示す。方法2710は、ステップ2712において、ビデオのクロマ・ブロックに関し、非ノーマル・クロマ・イントラ予測モードがビデオのクロマ・ブロックに適用されるかどうかの第1判定を行うことを含む。方法2710は、ステップ2714において、クロマ・ブロックに対応するルマ・ブロッ

10

20

30

40

50

クに関し、ルマ・イントラ予測モードがルマ・ブロックに適用される第2判定を行うことを更に含む。方法2710は、ステップ2716において、ルマ・イントラ予測モードに基づいてクロマ・ブロックに変換セット又は変換行列が適用される第3判定を行うことを更に含む。方法2710は、ステップ2718において、第3判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行することを更に含む。幾つかの実装において、非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む。

【0221】

図27Bはビデオ処理のための例示的な方法のフローチャートを示す。方法2720は、ステップ2722において、ビデオのクロマ・ブロックに関し、クロマ・ブロックに対応するルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用してコーディングされる第1判定を行うことを含む。方法2720は、ステップ2724において、第1判定に基づいて、ルールに従って、クロマ・ブロックに対する変換セット又は変換行列の第2判定を行うことを更に含む。方法2720は、ステップ2726において、第2判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行することを更に含む。幾つかの実装において、ルールは、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用してコーディングされることに起因して、クロマ・ブロックに関連する1つ以上のデフォルト・モード又はデフォルト変換セットは、クロマ・ブロックが非ノーマル・クロマ・イントラ予測モードを使用してコーディングされる場合に、変換セット又は変換行列を決定することを規定しており、ここで、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックをコーディングすることを含み、またここで、非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む。

【0222】

図27Cはビデオ処理のための例示的な方法のフローチャートを示す。方法2730は、ステップ2732において、ビデオの現在のビデオ・ブロックとビデオのコーディングされた表現との間の変換に関し、1つの色成分の現在のビデオ・ブロックに適用される第2変換ツールの適用可能性を、1) 別の色成分の対応するブロックのコーディング・モード、又は2) 現在のビデオ・ブロックのコーディング・モードのうちの少なくとも1つに基づいて決定することを含む。方法2730は、ステップ2734において、決定に基づいて変換を実行することを更に含む。幾つかの実装において、二次変換ツールを使用して：符号化の間に、量子化の前に、現在のビデオ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は復号化の間に、インバース一次変換を適用する前に、現在のビデオ・ブロックの逆量子化の出力にインバース二次変換が適用される。

【0223】

図27Dはビデオ処理のための例示的な方法のフローチャートを示す。方法2740は、ステップ2742において、ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定された位置をカバーしているルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うことを含む。方法2740は、ステップ2744において、第1判定に基づいて、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用して符号化されることに起因して、予め決定されたイントラ予測モードをクロマ・ブロックに適用する第2判定を行うことを更に含む。方法2740は、ステップ2746において、第2判定に従って、ビデオとビデオのコーディングされた表現との変換を実行することを更に含む。幾つかの実装において、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックを符号化することを含む。

【0224】

図27Eはビデオ処理のための例示的な方法のフローチャートを示す。方法2750は、ステップ2752において、ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定

された位置をカバーしているルマ・ブロックが、ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うことを含む。方法2750は、ステップ2754において、第1判定に基づいて、ルマ・ブロックのノーマル・ルマ・イントラ予測モードに基づいて、クロマ・イントラ予測モードを導出する第2判定を行うことを更に含む。方法2750は、ステップ2756において、第2判定に従って、ビデオとビデオのコーディングされた表現との変換を実行することを更に含み、ここで、ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用して、ルマ・ブロックを符号化することを含む。

【0225】

図23Aは、ビデオ処理装置2300のブロック図である。装置2300は、本願で説明される1つ以上の方法を実装するために使用されてもよい。装置2300は、スマートフォン、タブレット、コンピュータ、モノのインターネット（IoT）受信機などで具体化されてもよい。装置2300は、1つ以上のプロセッサ2302、1つ以上のメモリ2304、及びビデオ処理ハードウェア2306を含んでもよい。プロセッサ2302は、本文書で説明される1つ以上の方法（方法2200ないし2750を含むが、これらに限定されない）を実装するように構成されてもよい。メモリ（memories）2304は、本願で説明される方法及び技術を実装するために使用されるデータ及びコードを記憶するために使用されてもよい。ビデオ処理ハードウェア2306は、ハードウェア回路において、本文書で説明される幾つかの技術を実装するために使用されてもよい。幾つかの実施形態において、ハードウェア2306は、例えばグラフィックス・コプロセッサのように少なくとも部分的にプロセッサ2302内にあってもよい。

【0226】

図23Bは、開示される技術が実装され得るビデオ処理システムの別の例のブロック図である。図23Bは、本願で開示される種々の技術が実装され得る例示的なビデオ処理システム2400を示すブロック図である。種々の実装は、システム4100の構成要素の一部又は全部を含んでもよい。システム2400は、ビデオ・コンテンツを受信するための入力2402を含んでもよい。ビデオ・コンテンツは、生の又は非圧縮のフォーマット、例えば、8又は10ビットの多重成分ピクセル値で受信されてもよいし、又は圧縮された又は符号化されたフォーマットにおけるものであってもよい。入力2402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、又はストレージ・インターフェースを表現している可能性がある。ネットワーク・インターフェースの例は、イーサネット、光受動ネットワーク（PON）などの有線インターフェースや、Wi-Fi又はセルラー・インターフェースのような無線インターフェースを含む。

【0227】

システム2400は、本文書で説明される種々のコーディング又は符号化方法を実装することが可能なコーディング構成要素2404を含んでもよい。コーディング構成要素2404は、コーディング構成要素2404の入力2402から出力までのビデオの平均ビットレートを低減して、ビデオのコーディングされた表現を生成することができる。従って、コーディング技術は、ビデオ圧縮又はビデオ・トランスコーディング技術と呼ばれることが間々ある。コーディング構成要素2404の出力は、記憶されてもよいし、あるいは構成要素2406によって表現されているように接続された通信を介して伝送されてもよい。入力2402で受信されたビデオの記憶又は通信されるビットストリーム（又はコーディングされた）表現は、ディスプレイ・インターフェース2410に送信されるピクセル値又は表示可能なビデオを生成するために、構成要素2408によって使用されてもよい。ビットストリーム表現から、ユーザーが視聴可能なビデオを生成するプロセスは、ビデオ解凍と呼ばれることが間々ある。更に、特定のビデオ処理演算は、“コーディングする”演算又はツールと称されるが、コーディング・ツール又は演算はエンコーダで使用され、コーディングの結果を逆向きに処理する対応する復号化ツール又は演算はデコーダによって実行される、ということとは理解されるであろう。

【0228】

周辺バス・インターフェース又はディスプレイ・インターフェースの例は、ユニバーサル・シリアル・バス（USB）又は高解像度マルチメディア・インターフェース（HDMI（登録商標））、ディスプレイポート（Displayport）などを含む可能性がある。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（serial advanced technology attachment, SATA）、PCI、IDEインターフェースなどを含む。本文書で説明される技術は、携帯電話、ラップトップ、スマートフォン、又はその他のデバイスであって、デジタル・データ処理及び／又はビデオ表示を実行することが可能なデバイス、のような種々の電子デバイスで具現化されることが可能である。

【0229】

幾つかの実施形態において、本文書で説明されるビデオ処理方法は、図23A又は23Bに関して説明したようにハードウェア・プラットフォーム上に実装された装置を使用して実装されてもよい。

10

【0230】

開示される技術の幾つかの実施形態は、ビデオ処理ツール又はモードを動作可能にする判断又は決定を行うことを含む。一例では、ビデオ処理ツール又はモードがイネーブルにされている場合に、エンコーダは、ビデオのブロックの処理においてツール又はモードを使用又は実装するであろうが、結果として生じるビットストリームをツール又はモードの使用に基づいて必ずしも修正しない可能性がある。即ち、ビデオのブロックからビデオのビットストリーム表現への変換は、判断又は決定に基づいてイネーブルにされた場合に、ビデオ処理ツール又はモードを使用するであろう。別の例では、ビデオ処理ツール又はモードがイネーブルにされている場合に、デコーダは、ビットストリームがビデオ処理ツール又はモードに基づいて修正されている旨の情報とともに、ビットストリームを処理するであろう。即ち、ビデオのビットストリーム表現からビデオのブロックへの変換は、判断又は決定に基づいてイネーブルにされたビデオ処理ツール又はモードを使用して実行されるであろう。

20

【0231】

開示される技術の幾つかの実施形態は、ビデオ処理ツール又はモードをディセーブルにする判断又は決定を行うことを含む。一例では、ビデオ処理ツール又はモードがディセーブルにされる場合、エンコーダは、ビデオのブロックをビデオのビットストリーム表現に変換する際に、ツール又はモードを使用しないであろう。別の例では、ビデオ処理ツール又はモードがディセーブルにされる場合、デコーダは、ビットストリームが、その判断又は決定に基づいてディセーブルにされたビデオ処理ツール又はモードを使用して修正されていない旨の情報とともに、ビットストリームを処理するであろう。

30

【0232】

本文書において、“ビデオ処理”という用語は、ビデオ符号化、ビデオ復号化、ビデオ圧縮又はビデオ解凍を指す可能性がある。例えば、ビデオ圧縮アルゴリズムは、ビデオのピクセル表現から、対応するビットストリーム表現へ、又はその逆への変換の間に適用される可能性がある。現在のビデオ・ブロックのビットストリーム表現は、例えばシンタックスによって定義されるように、ビットストリーム内で同等位置にあるか又は異なる場所に分散されるビットに対応してもよい。例えば、マクロブロックは、変換されたコーディングされたエラー残差値の観点から、また、ビットストリーム内のヘッダ及びその他のフィールドのビットを使用して符号化されてもよい。

40

【0233】

様々な技術及び実施形態は以下の条項形式を用いて記述することができる。第1セットの条項は、上記のセクションにおいて開示される技術の特定の特徴及び態様を記述している。

【0234】

1. ビデオ処理方法であって、

現在のビデオ・ブロックの特徴に基づいて、縮小二次変換を現在のビデオ・ブロックに適用するための変換セット又は変換行列を選択するステップ；及び

50

現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換の一部として、選択された変換セット又は変換行列を、現在のビデオ・ブロックの一部分に適用するステップ；

を含む方法。

【0235】

2. 現在のビデオ・ブロックの一部分は、現在のビデオ・ブロックの右上サブ領域、右下サブ領域、左下サブ領域又は中央サブ領域である、条項1に記載の方法。

【0236】

3. 現在のビデオ・ブロックの特徴は、現在のビデオ・ブロックのイントラ予測モード又は一次変換行列である、条項1又は2に記載の方法。

【0237】

4. 上記の特徴は、現在のビデオ・ブロックの色成分である、条項1に記載の方法。

【0238】

5. 第1変換セットは、現在のビデオ・ブロックのルマ成分に対して選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックの1つ以上のクロマ成分に対して選択される、条項4に記載の方法。

【0239】

6. 上記の特徴は、現在のビデオ・ブロックのイントラ予測モード又はイントラ・コーディング方法である、条項1に記載の方法。

【0240】

7. イントラ予測方法は、多重参照ライン（MRL）ベースの予測方法又は行列ベースのイントラ予測方法を含む、条項6に記載の方法。

【0241】

8. 第1変換セットは、現在のビデオ・ブロックがクロス・コンポーネント線型モデル（CCLM）コード化ブロックである場合に選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックが非CCLMコード化ブロックである場合に選択される、条項6に記載の方法。

【0242】

9. 第1変換セットは、現在のビデオ・ブロックがジョイント・クロマ残差コーディング方法でコーディングされる場合に選択され、第1変換セットとは異なる第2変換セットは、現在のビデオ・ブロックがジョイント・クロマ残差コーディング方法でコーディングされない場合に選択される、条項6に記載の方法。

【0243】

10. 上記の特徴は、現在のビデオ・ブロックの一次変換である、条項1に記載の方法。

【0244】

11. ビデオ処理方法であって：

現在のビデオ・ブロックに関連する1つ以上の係数に基づいて、現在のビデオ・ブロックのビットストリーム表現における縮小二次変換（RST）の適用のためのサイド情報のシグナリングの選択的な包含に関する決定を行うステップ；及び

決定に基づいて、現在のビデオ・ブロックと現在のビデオ・ブロックのビットストリーム表現を含むビデオとの間の変換を実行するステップ；

を含む方法。

【0245】

12. 1つ以上の係数は、現在のビデオ・ブロックのスキヤニング順序における最後の非ゼロ係数を含む、条項11に記載の方法。

【0246】

13. 1つ以上の係数は、現在のビデオ・ブロックの部分的な領域内の複数の係数を含む、条項11に記載の方法。

【0247】

14. 部分的な領域は、RSTが適用され得る1つ以上のコーディング・グループを含む、

10

20

30

40

50

条項13に記載の方法。

【0248】

15. 部分的な領域は、現在のビデオ・ブロックのスキニング順序で先頭からM個のコーディング・グループ又は最後からM個のコーディング・グループを含む、条項13に記載の方法。

【0249】

16. 部分的な領域は、現在のビデオ・ブロックの逆スキニング順序で先頭からM個のコーディング・グループ又は最後からM個のコーディング・グループを含む、条項13に記載の方法。

【0250】

17. 決定を行うことは、複数の係数の1つ以上の非ゼロ係数のエネルギーに更に基づいている、条項13に記載の方法。

【0251】

18. ビデオ処理方法であって：

縮小二次変換（RST）を現在のビデオ・ブロックに適用することに関し、現在のビデオ・ブロックのビットストリーム表現を構築するステップであって、RSTに関連するシンタックス要素は、残差情報をコーディングする前に、ビットストリーム表現においてシグナリングされる、ステップ；及び

その構築に基づいて、現在のビデオ・ブロックと現在のビデオ・ブロックのビットストリーム表現との間の変換を実行するステップ；

を含む方法。

【0252】

19. RSTに関連するシンタックス要素をシグナリングすることは、少なくとも1つのコーディングされたブロック・フラグ又は変換選択モードの使用法に基づいている、条項18に記載の方法。

【0253】

20. ビットストリーム表現は、全てゼロ係数であるコーディング・グループに対応するコーディング残差情報を除外している、条項18に記載の方法。

【0254】

21. コーディング残差情報は、RSTの適用に基づいている、条項18に記載の方法。

【0255】

22. ビデオ処理方法であって：

縮小二次変換（RST）を現在のビデオ・ブロックに適用するために、現在のビデオ・ブロックのビットストリーム表現を構築するステップであって、RSTに関連するシンタックス要素は、変換スキップ指示又は多重変換セット（MTS）インデックスの何れかの前にビットストリーム表現でシグナリングされる、ステップ；及び

その構築に基づいて、現在のビデオ・ブロックと現在のビデオ・ブロックのビットストリーム表現との間の変換を実行するステップ；

を含む方法。

【0256】

23. 変換スキップ指示又はMTSインデックスは、RSTに関連するシンタックス要素に基づいている、条項22に記載の方法。

【0257】

24. ビデオ処理方法であって：

現在のビデオ・ブロックの特徴に基づいて、縮小二次変換（RST）のインデックスをコーディングするためのコンテキスト・モデルを構築するステップ；及び

その構築に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換を実行するステップ；

を含む方法。

【0258】

10

20

30

40

50

25. 上記の特徴は、多重変換選択（MTS）プロセスの明示的又は暗示的な実施可能性である、条項24に記載の方法。

【0259】

26. 上記の特徴は、現在のビデオ・ブロックにおけるクロス・コンポーネント線型モデル（CCLM）コーディング・モードの実施可能性である、条項24に記載の方法。

【0260】

27. 上記の特徴は、現在のビデオ・ブロックのサイズである、条項24に記載の方法。

【0261】

28. 上記の特徴は、現在のビデオ・ブロックに適用されるパーティショニング・プロセスの分割深度である、条項24に記載の方法。

【0262】

29. パーティショニング・プロセスは、四分木（QT）パーティショニング・プロセス、二分木（BT）パーティショニング・プロセス、又は三分木（TT）パーティショニング・プロセスである、条項28に記載の方法。

【0263】

30. 上記の特徴は、現在のビデオ・ブロックのカラー・フォーマット又は色成分である、条項24に記載の方法。

【0264】

31. 上記の特徴は、現在のビデオ・ブロックのイントラ予測モードと、多重変換選択（MTS）プロセスのインデックスとを除外している、条項24に記載の方法。

【0265】

32. ビデオ処理方法であって：

現在のビデオ・ブロックの特徴に基づいて、現在のビデオ・ブロックに対するインバース縮小二次変換（RST）プロセスの選択的適用に関する決定を行うステップ；及び
決定に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換を実行するステップ；
を含む方法。

【0266】

33. 上記の特徴は、現在のビデオ・ブロックのコーディング・グループのコーディングされたブロック・フラグである、条項32に記載の方法。

【0267】

34. インバースRSTプロセスは適用されず、左上コーディング・グループのコーディングされたブロック・フラグはゼロである、条項33に記載の方法。

【0268】

35. インバースRSTプロセスは適用されず、現在のビデオ・ブロックのスキニング順序における第1及び第2コーディング・グループに対するコーディングされたブロック・フラグはゼロである、条項33に記載の方法。

【0269】

36. 上記の特徴は、現在のビデオ・ブロックの高さ（M）又は幅（N）である、条項32に記載の方法。

【0270】

37. インバースRSTプロセスは適用されず、（i）M=8及びN=4、又は（ii）M=4及びN=8である、条項36に記載の方法。

【0271】

38. ビデオ処理方法であって：

現在のビデオ・ブロックの特徴に基づいて、現在のビデオ・ブロックに対するインバース縮小二次変換（RST）プロセスの選択的適用に関する決定を行うステップ；及び
決定に基づいて、現在のビデオ・ブロックと、現在のビデオ・ブロックを含むビデオのビットストリーム表現との間の変換を実行するステップ；
を含み、ビットストリーム表現はRSTに関するサイド情報を含み、サイド情報は、現在

10

20

30

40

50

のビデオ・ブロックのルマ成分又はシングル・カラーの係数に基づいて包含される、方法。

【0272】

39. サイド情報は、現在のビデオ・ブロックのディメンジョンに更に基づいて包含される、条項38に記載の方法。

【0273】

40. サイド情報は、現在のビデオ・ブロックに対するブロック情報を考慮することなく包含される、条項38又は39に記載の方法。

【0274】

41. 変換は、ビットストリーム表現を現在のビデオ・ブロックから生成することを含む、条項1-40のうちの何れか1項に記載の方法。

【0275】

42. 変換は、現在のビデオ・ブロックをビットストリーム表現から生成することを含む、条項1-40のうちの何れか1項に記載の方法。

【0276】

43. プロセッサと命令を伴う非一時的なメモリとを含むビデオ・システムにおける装置であって、命令は、プロセッサにより実行されると、条項1-42のうちの何れか1つにおける方法をプロセッサに実行させる、装置。

【0277】

44. 非一時的なコンピュータ読み取り可能な媒体に記憶されるコンピュータ・プログラム製品であって、条項1-42のうちの何れか1つにおける方法を実行するためのプログラム・コードを含む、コンピュータ・プログラム製品。

【0278】

第2セットの条項は、上記のセクション、例えば例示的なアイテム4及び5において開示される技術の特定の特徴及び態様を記述している。

【0279】

1. ビデオ処理方法であって、

ビデオのビデオ・ユニットの現在のビデオ・ブロックとビデオのコーディングされた表現との間の変換について、特定のイントラ予測モードを使用してコーディングされたビデオ・ユニットのデフォルト・イントラ予測モードを決定し、その結果、現在のビデオ・ブロックの予測ブロックが、ある方向に沿って現在のビデオ・ブロックの隣接するピクセルを外挿することなく生成されるようにするステップと、

決定に基づいて変換を実行するステップとを含む方法。

【0280】

2. ビデオユニットは、コーディング・ユニット、予測ユニット、コーディング・ブロック、又は予測ブロックに対応する、条項1に記載の方法。

【0281】

3. ビデオ・ユニットは、行列ベクトル乗算を使用して予測ブロックを生成する行列ベースのイントラ予測（MIP）を使用してコーディングされる、条項1に記載の方法。

【0282】

4. ビデオ・ユニットは、現在のビデオ・ブロックを含むビデオ・フレームを指すブロック・ベクトルを少なくとも使用して予測ブロックを生成するイントラ・ブロック・コピー（IBC）モードを使用してコーディングされる、条項1に記載の方法。

【0283】

5. ビデオ・ユニットは、代表的なサンプル値のパレットを使用して現在のビデオ・ブロックを表現又は再構成することを可能にするパレット・モードを使用してコーディングされる、条項1に記載の方法。

【0284】

6. デフォルトのイントラ予測モードは、現在のビデオ・ブロックのコーディング・モードに基づいて決定される、条項1に記載の方法。

【0285】

7. デフォルトのイントラ予測モードに関する情報は、コーディングされた表現に含まれる、条項1に記載の方法。

【0286】

8. デフォルトのイントラ予測モードに関する情報は、シグナリングされることなく導出される、条項1に記載の方法。

【0287】

9. デフォルトのイントラ予測モードの決定は、クロマ導出モード（DM）の導出で更に使用される、第1項に記載の方法。

【0288】

10. デフォルトのイントラ予測モードの決定は、ビデオの他のブロックのイントラ予測モードを予測するために更に使用される、第1項に記載の方法。

【0289】

11. ある色成分の現在のビデオ・ブロックに割り当てられたデフォルトのイントラ予測モードの決定が、別の色成分の変換セット又は変換インデックスの導出に使用される、条項1に記載の方法。

【0290】

12. デフォルトのイントラ予測モードは、ビデオのブロックの予測モードと共に記憶される、第1項に記載の方法。

【0291】

13. デフォルトのイントラ予測モードは、インター・コーディングされたブロックには割り当てられない、条項1に記載の方法。

【0292】

14. デフォルトのイントラ予測モードが平面イントラ予測モードである、条項1に記載の方法。

【0293】

15. ビデオ処理方法であって、
ルールを使用して、ビデオのクロマ・ブロックの所定の位置をカバーするビデオのルマ・ブロックの決定を行うステップと、
決定に基づいて、ビデオとビデオのコーディングされた表現との間の変換を実行するステップと
を含み、クロマ・ブロックは、イントラ予測モードを用いてコーディングされた表現で表現される、方法。

【0294】

16. 変換を実行するステップは、ルマ・ブロックの予測モード又はコーディング・モードをチェックするステップと、ルマ・ブロックのイントラ予測モードをフェッチするステップとを含む、第15項に記載の方法。

【0295】

17. ルマ・ブロックは、クロマ・ブロックの中央にあるクロマ・サンプルに対応するように位置するルマ・サンプルを含む、条項15に記載の方法。

【0296】

18. ルマ・ブロックは、クロマ・ブロックの左上にあるクロマ・サンプルに対応するように位置するルマ・サンプルを含む、条項15に記載の方法。

【0297】

19. ビデオ処理方法であって、
ルールを使用して、クロマ・ブロックに対応するルマ・ブロックのコーディング・モードに基づいて、ビデオのクロマ・ブロックのイントラ予測モードを導出するステップと、
導出されたイントラ予測モードに基づいて、そのクロマ・ブロックとビデオのコーディングされた表現との間の変換を実行するステップと
を含み、ルマ・ブロックのコーディング・モードが、特定のイントラ予測モードであっ

て、ルマ・ブロックの予測ブロックが、ルマ・ブロックの隣接するピクセルを或る方向に沿って外挿することなく生成される特定のイントラ予測モードである場合に、ルールは、デフォルトのイントラ予測モードを使用することを規定している、方法。

【0298】

20. ルールは、ルマ・ブロックのコーディング・モードが特定のイントラ予測モードでない場合に、復号化されたイントラ予測モードを使用することを規定している、条項19に記載の方法。

【0299】

21. ルマ・ブロックのコーディング・モードは、行列ベクトル乗算を使用して予測ブロックを生成する行列ベースのイントラ予測（MIP）に対応する、条項19に記載の方法。

【0300】

22. 復号化されたイントラ予測モードは、平面イントラ予測モードである、条項21に記載の方法。

【0301】

23. ルマ・ブロックのコーディング・モードは、現在のビデオ・ブロックを含むビデオ・フレームを指すブロック・ベクトルを少なくとも使用して予測ブロックを生成するイントラ・ブロック・コピー（IBC）モードに対応する、条項19に記載の方法。

【0302】

24. ルマ・ブロックのコーディング・モードは、代表的なサンプル値のパレットを使用して、現在のビデオ・ブロックを表現又は再構成することを可能にするパレット・モードに対応する、条項19に記載の方法。

【0303】

25. 復号化されたイントラ予測モードが、DCイントラ予測モードである、条項23又は24に記載の方法。

【0304】

返還を実行するステップは、二次変換ツールを現在のビデオ・ブロックに適用するステップを含み、前記二次変換ツールを使用して、符号化の間に、量子化の前に、現在のビデオ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は復号化の間に、インバース一次変換を適用する前に、前記現在のビデオ・ブロックの逆量子化の出力にインバース二次変換が適用される、条項1ないし25のうちの何れか1項に記載の方法。

【0305】

27. 変換の実行は、コーディングされた表現をビデオから生成することを含む、条項1ないし26のうちの何れか1項に記載の方法。

【0306】

28. 変換の実行は、コーディングされた表現からビデオを生成することを含む、条項1ないし26のうちの何れか1項に記載の方法。

【0307】

29. 条項1ないし28のうちの何れか1項に記載の方法を実行するように構成されたプロセッサを含むビデオ処理装置。

【0308】

30. 実行された場合に、条項1ないし28のうちの任意の1つ以上に記載の方法をプロセッサに実行させるプログラム・コードを記憶したコンピュータ読み取り可能な媒体。

【0309】

第3セットの条項は、上記のセクション、例えば例示的なアイテム5及び6において開示される技術の特定の特徴及び態様を記述している。

【0310】

1. ビデオ処理方法であって、

ビデオのクロマ・ブロックに関し、非ノーマル・クロマ・イントラ予測モードがビデオの前記クロマ・ブロックに適用されるかどうかの第1判定を行うステップと、

10

20

30

40

50

クロマ・ブロックに対応するルマ・ブロックに関し、ルマ・イントラ予測モードがルマ・ブロックに適用される第2判定を行うステップと、

ルマ・イントラ予測モードに基づいてクロマ・ブロックに変換セット又は変換行列が適用される第3判定を行うステップと、

第3判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行するステップと

を含み、非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む、方法。

【0311】

2. 変換セット又は変換行列は、クロマ・ブロックに適用される二次変換ツールで
使用される、条項1に記載の方法。

【0312】

3. 二次変換ツールは、低周波ノン・セパラブル変換（LFNST）ツールに対応しており、二次変換ツールを使用して、

符号化の間に、量子化の前に、クロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の間に、インバース一次変換を適用する前に、クロマ・ブロックの逆量子化の出力にインバース二次変換が適用される、条項2に記載の方法。

【0313】

4. 非ノーマル・クロマ・イントラ予測モードは、線型モデルを使用してクロマ・コンポーネントの予測値を他のコンポーネントから導出するクロス・コンポーネント線型モデル（CCLM）予測モードに対応している、条項1に記載の方法。

【0314】

5. ルマ・ブロックは、クロマ・ブロックの予め決定された位置にあるクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項1に記載の方法。

【0315】

6. ルマ・ブロックは、クロマ・ブロックの中央のクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項1に記載の方法。

【0316】

7. ルマ・ブロックは、現在のビデオ・ブロックの左上のクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項1に記載の方法。

【0317】

8. ルマ・イントラ予測は非ノーマル・ルマ・イントラ予測モードであり、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックをコーディングすることを含む、条項1に記載の方法。

【0318】

9. 非ノーマル・ルマ・イントラ予測モードは、行列ベクトル乗算を使用して予測値を生成するマトリクス・ベースのイントラ予測（MIP）に対応している、条項8に記載の方法。

【0319】

10. 非ノーマル・ルマ・イントラ予測モードは、ルマ・ブロックを含むビデオ・フレームを指すブロック・ベクトルを少なくとも使用して予測値を生成するイントラ・ブロック・コピー（IBC）モードに対応している、条項8に記載の方法。

【0320】

11. 非ノーマル・ルマ・イントラ予測モードは、代表サンプル値のパレットを使用してルマ・ブロックを表現又は再構成することを可能にするパレット・モードに対応している、条項8に記載の方法。

【0321】

12. ルマ・イントラ予測モードが非ノーマル・ルマ・イントラ予測モードであること

10

20

30

40

50

に応答して、予め決定されたイントラ予測モードがクロマ・ブロックのために指定され、非ノーマル・クロマ・イントラ予測モードの代わりに、予め決定されたイントラ予測モードを更に使用して、クロマ・ブロックに適用される変換セット又は変換行列を導出する、条項8ないし11のうちの何れか1項に記載の方法。

【0322】

13. 予め決定されたイントラ予測モードは、ルマ・ブロックの非ノーマル・ルマ・イントラ予測モードに基づいて決定される、条項12に記載の方法。

【0323】

14. 非ノーマル・ルマ・イントラ予測モードがIBCモード又はパレット・モードであることに応答して、予め決定されたイントラ予測モードはDCモードとして決定される、条項13に記載の方法。

10

【0324】

15. 非ノーマル・ルマ・イントラ予測モードがMIPモードであることに応答して、予め決定されたイントラ予測モードは平面モードとして決定される、条項13に記載の方法。

【0325】

16. ルマ・イントラ予測はノーマル・ルマ・イントラ予測モードであり、ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用して、ルマ・ブロックをコーディングすることを含む、条項1に記載の方法。

【0326】

17. ルマ・イントラ予測モードが非ノーマル・ルマ・イントラ予測モードであることに応答して、ルマ・イントラ予測がクロマ・ブロックのために指定され、非ノーマル・クロマ・イントラ予測モードの代わりに、ルマ・イントラ予測モードを更に使用して、クロマ・ブロックに適用される変換セット又は変換行列を導出する、条項16に記載の方法。

20

【0327】

18. ビデオ処理方法であって、
ビデオのクロマ・ブロックに関し、クロマ・ブロックに対応するルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用してコーディングされる第1判定を行うステップと、

第1判定に基づいて、ルールに従って前記クロマ・ブロックに対する変換セット又は変換行列の第2判定を行うステップと、

30

第2判定に従って、ビデオとビデオのコーディングされた表現との間の変換を実行するステップと

を含み、ルールは、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用してコーディングされることに起因して、クロマ・ブロックに関連する1つ以上のデフォルト・モード又はデフォルト変換セットは、クロマ・ブロックが非ノーマル・クロマ・イントラ予測モードを使用してコーディングされる場合に変換セット又は変換行列を決定することを規定しており、

非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックをコーディングすることを含み、

非ノーマル・クロマ・イントラ予測モードは、クロマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、クロマ・ブロックをコーディングすることを含む、方法。

40

【0328】

19. 非ノーマル・クロマ・イントラ予測モードは、線型モデルを使用してクロマ・コンポーネントの予測値を他のコンポーネントから導出するクロス・コンポーネント線型モデル(CCLM)予測モードに対応している、条項18に記載の方法。

【0329】

20. 非ノーマル・ルマ・イントラ予測モードは、行列ベクトル乗算を使用して予測値を生成するマトリクス・ベースのイントラ予測(MIP)に対応している、条項18に記載の方法。

【0330】

50

21. 非ノーマル・ルマ・イントラ予測モードは、ルマ・ブロックを含むビデオ・フレームを指すブロック・ベクトルを少なくとも使用して予測値を生成するイントラ・ブロック・コピー（IBC）モードに対応している、条項18に記載の方法。

【0331】

22. 非ノーマル・ルマ・イントラ予測モードは、代表サンプル値のパレットを使用してルマ・ブロックを表現又は再構成することを可能にするパレット・モードに対応している、条項18に記載の方法。

【0332】

23. ルマ・ブロックは、クロマ・ブロックの予め決定された位置にあるクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項18に記載の方法。

【0333】

24. ルマ・ブロックは、クロマ・ブロックの中央のクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項18に記載の方法。

【0334】

25. ルマ・ブロックは、現在のビデオ・ブロックの左上のクロマ・サンプルに対応するルマ・サンプルをカバーしている、条項18に記載の方法。

【0335】

26. ルールは、デフォルト変換セット又はデフォルト変換行列のうちの少なくとも1つを使用するか、或いはルマ・ブロックのイントラ・ルマ予測モードから前記変換セット又は変換行列を導出するかを、ルマ・ブロックのコーディング・モードに基づいて規定している、条項18に記載の方法。

【0336】

27. ルールは、ルマ・ブロックの隣接するピクセルを外挿することによってルマ・ブロックの予測ブロックを生成するノーマル・イントラ予測モードを使用してルマ・ブロックが符号化されることに起因して、ルマ・ブロックのイントラ・ルマ予測モードに従って、変換セット又は変換行列を導出することを規定している、条項18に記載の方法。

【0337】

28. ルールは、ブロック・ベースのデルタ・パルス・コード変調（BDPCM）モードを使用してルマ・ブロックが符号化されることに起因して、ルマ・ブロックのイントラ・ルマ予測モードに従って、変換セット又は変換行列を導出することを規定している、条項18に記載の方法。

【0338】

29. ルールは、非ノーマル・ルマ・イントラ予測モードを使用してルマ・ブロックが符号化されることに起因して、デフォルト変換セット又はデフォルト変換行列のうちの少なくとも1つを使用することを規定している、条項26に記載の方法。

【0339】

30. ビデオ処理方法であって、
ビデオの現在のビデオ・ブロックとビデオのコーディングされた表現との間の変換に関し、1つの色成分の現在のビデオ・ブロックに適用される第2変換ツールの適用可能性を、
1) 別の色成分の対応するブロックのコーディング・モード、又は2) 現在のビデオ・ブロックのコーディング・モードのうちの少なくとも1つに基づいて決定するステップと、
決定に基づいて前記変換を実行するステップと、
を含み、二次変換ツールを使用して、
符号化の間に、量子化の前に、現在のビデオ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は
復号化の間に、インバース一次変換を適用する前に、現在のビデオ・ブロックの逆量子化の出力にインバース二次変換が適用される、方法。

【0340】

31. 二次変換ツールは、低周波ノン・セパラブル変換（LFNST）ツールに対応している、条項30に記載の方法。

10

20

30

40

50

【0341】

32. 現在のビデオ・ブロックはクロマ・ブロックに対応し、対応するブロックはクロマ・ブロックに対応するルマ・ブロックに対応している、条項30に記載の方法。

【0342】

33. 対応するブロックのコーディング・モードが、i) 予測方向に沿って外挿された隣接するピクセルの値を使用して、対応するブロックを符号化することを含むノーマル・イントラ予測モード、及び／又はii) ブロック・ベースのデルタ・パルス・コード変調 (BDPCM) モードであることに起因して、決定するステップは、第2変換ツールを前記現在のビデオ・ブロックに適用することを決定する、条項31に記載の方法。

【0343】

34. 対応するブロックのコーディング・モードが、予測方向に沿って外挿された隣接するピクセルの値を使用せずに、対応するブロックを符号化することを含む非ノーマル・イントラ予測モードであることに起因して、決定するステップは、現在のビデオ・ブロックに対して第2変換ツールをディセーブルにすることを決定する、条項31に記載の方法。

【0344】

35. 現在のビデオ・ブロックのコーディング・モードは、線型モデルを使用してクロマ・コンポーネントの予測値を他のコンポーネントから導出するクロス・コンポーネント線型モデル (CCLM) 予測モードに対応している、条項30に記載の方法。

【0345】

36. ビデオ処理方法であって、
ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定された位置をカバーしているルマ・ブロックが、非ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うステップと、

第1判定に基づいて、ルマ・ブロックが非ノーマル・ルマ・イントラ予測モードを使用して符号化されることに起因して、予め決定されたイントラ予測モードをクロマ・ブロックに適用する第2判定を行うステップと、

第2判定に従って、ビデオとビデオのコーディングされた表現との変換を実行するステップと、

を含み、非ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用せずに、ルマ・ブロックを符号化することを含む、方法。

【0346】

37. クロマ・ブロックに適用される二次変換ツールで使用される変換セット又は変換行列は、予め決定されたイントラ予測モードに基づいて選択される、条項36に記載の方法。

【0347】

38. 二次変換ツールは、低周波ノン・セパラブル変換 (LFNST) ツールに対応しており、前記二次変換ツールを使用して、

符号化の間に、量子化の前に、クロマ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の間に、インバース一次変換を適用する前に、クロマ・ブロックの逆量子化の出力にインバース二次変換が適用される、条項36に記載の方法。

【0348】

39. クロマ・ブロックに適用される予め決定されたイントラ予測モードは、ルマ・ブロックの非ノーマル・ルマ・イントラ予測モードに依存している、条項36ないし38のうちの何れか1項に記載の方法。

【0349】

40. 予め決定されたイントラ予測モードは、DCモード、平面モード、垂直モード、水平モード、45度モード、又は135度モードのうちの少なくとも1つに対応している、条項36ないし38のうちの何れか1項に記載の方法。

【0350】

41. 非ノーマル・ルマ・イントラ予測モードは、イントラ・ブロック・コピー（IBC）モード、パレット・モード、マトリクス・ベースのイントラ予測（MIP）、又はブロック・ベースのデルタ・パルス・コード変調（BDPCM）モードのうちの少なくとも1つに対応している、条項36ないし38のうちの何れか1項に記載の方法。

【0351】

42. 非ノーマル・ルマ・イントラ予測モードが、マトリクス・ベースのイントラ予測（MIP）に対応している場合に、MIPモードは、ルマ・ブロックのディメンジョンと前記MIPに基づいて特定のイントラ予測モードにマッピングされる、条項36ないし38のうちの何れか1項に記載の方法。

【0352】

43. 予め決定されたイントラ予測モードは複数の候補の中から選択される、条項36ないし38のうちの何れか1項に記載の方法。

【0353】

44. 予め決定されたイントラ予測モードはシグナリングされる、条項36ないし38のうちの何れか1項に記載の方法。

【0354】

45. 予め決定されたイントラ予測モードは導出される、条項36ないし38のうちの何れか1項に記載の方法。

【0355】

46. ビデオ処理方法であって、
ビデオのクロマ・ブロックに関し、クロマ・ブロックの予め決定された位置をカバーしているルマ・ブロックが、ノーマル・ルマ・イントラ予測モードを使用して符号化される第1判定を行うステップと、

第1判定に基づいて、ルマ・ブロックのノーマル・ルマ・イントラ予測モードに基づいて、クロマ・イントラ予測モードを導出する第2判定を行うステップと、

第2判定に従って、ビデオとビデオのコーディングされた表現との変換を実行するステップと、

を含み、ノーマル・ルマ・イントラ予測モードは、ルマ予測方向に沿って外挿された隣接するピクセルの値を使用して、ルマ・ブロックを符号化することを含む、方法。

【0356】

47. クロマ・ブロックに適用される二次変換ツールで使用される変換セット又は変換行列は、クロマ・イントラ予測モードに基づいて選択される、条項46に記載の方法。

【0357】

48. 二次変換ツールは、低周波ノン・セパラブル変換（LFNST）ツールに対応しており、二次変換ツールを使用して、

符号化の間に、量子化の前に、現在のビデオ・ブロックの残差に適用されたフォワード一次変換の出力にフォワード二次変換が適用され、又は

復号化の間に、インバース一次変換を適用する前に、現在のビデオ・ブロックの逆量子化の出力にインバース二次変換が適用される、条項46に記載の方法。

【0358】

49. ルマ・ブロックは、クロマ・ブロックの中央にあるクロマ・サンプルに対応する位置にあるルマ・サンプルを含む、条項46ないし48のうちの何れか1項に記載の方法。

【0359】

50. ルマ・ブロックは、クロマ・ブロックの左上にあるクロマ・サンプルに対応する位置にあるルマ・サンプルを含む、条項46ないし48のうちの何れか1項に記載の方法。

【0360】

51. 変換を実行するステップは、コーディングされた表現をビデオから生成するステップ、或いはビデオをコーディングされた表現から生成するステップを含む、条項1ないし50のうちの何れか1項に記載の方法。

【0361】

10

20

30

40

50

52. 条項1ないし51のうちの何れか1項に記載された方法を実行するように構成されたプロセッサを含むビデオ処理装置。

【0362】

53. 実行された場合に、条項1ないし51のうちの何れか1項に記載された方法をプロセッサに実行させるプログラム・コードを記憶したコンピュータ読み取り可能な記憶媒体。

【0363】

以上から、本件で開示される技術の特定の実施形態が、説明の目的で本明細書で説明されてきたが、本発明の範囲を逸脱することなく、種々の修正が行われてもよいことは理解されるであろう。従って、ここに開示される技術は、添付のクレームによる場合を除いて限定されない。

【0364】

本特許文献で説明される対象事項及び機能的動作の実装は、本明細書で開示される構造及びそれらの構造的均等物を含む、種々のシステム、デジタル電子回路、又はコンピュータ・ソフトウェア、ファームウェア、又はハードウェア、又はそれらの1つ以上の組み合わせにおいて実現することができる。本明細書で説明される対象事項の実装は、1つ以上のコンピュータ・プログラム製品として、即ち、データ処理装置による実行のための、又はその動作を制御するための、有形且つ非一時的なコンピュータ読み取り可能な媒体上で符号化されているコンピュータ・プログラム命令の1つ以上のモジュールとして、実装することができる。コンピュータ読み取り可能な媒体は、機械読み取り可能なストレージ・デバイス、機械読み取り可能なストレージ基板、メモリ・デバイス、機械読み取り可能な伝搬信号に影響を与える物質の組成、又はそれらの1つ以上の組み合わせであることが可能である。用語“データ処理ユニット”又は“データ処理装置”は、例えば、プログラマブル・プロセッサ、コンピュータ、又は複数のプロセッサ又はコンピュータを含む、データを処理するための全ての装置、デバイス、及び機械を包含する。装置は、ハードウェアに加えて、問題としているコンピュータ・プログラムの実行環境を生成するコード、例えば、プロセッサ・ファームウェア、プロトコル・スタック、データベース管理システム、オペレーティング・システム、又はそれらの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。

【0365】

コンピュータ・プログラム（プログラム、ソフトウェア、ソフトウェア・アプリケーション、スクリプト、又はコードとしても知られているもの）は、コンパイル又は解釈された言語を含む、任意の形式のプログラミング言語で書くことが可能であり、それは、スタンド・アロン・プログラムとして、又はモジュール、コンポーネント、サブルーチン、又はコンピューティング環境での使用に適したその他のユニットとしてのものを含む、任意の形式で配備することが可能である。コンピュータ・プログラムは、必ずしもファイル・システム内のファイルに対応するとは限らない。プログラムは、他のプログラム又はデータを保持するファイルの一部分（例えば、マークアップ言語文書に保存される1つ以上のスクリプト）内に、問題としているプログラムに専用の単一ファイル内に、又は複数の調整されたファイル（例えば、1つ以上のモジュール、サブ・プログラム、又はコードの一部分を記憶するファイル）内に、保存されることが可能である。コンピュータ・プログラムは、1つのコンピュータ上で又は複数のコンピュータ上で実行されるように配備することが可能であり、複数のコンピュータは、1つのサイトに配置されるか、又は複数のサイトにわたって分散されて通信ネットワークによって相互接続されている。

【0366】

本明細書で説明されるプロセス及びロジックの流れは、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・プロセッサによって実行され、入力データに作用して出力を生成することによって機能を実行することができる。プロセス及びロジックのフローはまた、例えばFPGA（フィールド・プログラマブル・ゲート・アレイ）又はASIC（特定用途向け集積回路）のような特殊目的論理回路によって実行されることが可能であり、また、それらとして装置を実装することも可能である。

【0367】

コンピュータ・プログラムの実行に適したプロセッサは、例えば、汎用及び専用双方のマイクロプロセッサ、及び任意の種類のデジタル・コンピュータの任意の1つ以上のプロセッサを含む。一般に、プロセッサは、リード・オンリ・メモリ又はランダム・アクセス・メモリ又は双方から命令及びデータを受信するであろう。コンピュータの本質的な要素は、命令を実行するためのプロセッサと、命令及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。一般に、コンピュータはまた、データを記憶するための1つ以上の大容量ストレージ・デバイス、例えば磁気的なもの、磁気光ディスク、又は光ディスクを含み、あるいはそれらからデータを受信したり、それらへデータを転送したり、或いは双方のために動作可能に結合される。しかしながら、コンピュータは、そのようなデバイスを有することを必須とはしない。コンピュータ・プログラム命令及びデータを記憶するのに適したコンピュータ読み取り可能な媒体は、例えば、EPROM、EEPROM、及びフラッシュ・メモリ・デバイスのような半導体メモリ・デバイスを含む、あらゆる形態の不揮発性メモリ、媒体及びメモリ・デバイスを含む。プロセッサ及びメモリは、特殊目的論理回路によって補足されるか、又はそこに内蔵されることが可能である。

10

【0368】

明細書は図面と共に模範としてのみ考慮されることが意図されており、ここで模範は例を意味する。本願で使用されるように、“又は”の使用は、文脈が明らかに別意を示していない限り、“及び／又は”を含むように意図されている。

【0369】

本特許文献は多くの詳細を含んでいるが、これらは、何れかの発明やクレームされ得るものの範囲に関する限定として解釈されるべきではなく、むしろ特定の発明の特定の実施形態に特有である可能性のある特徴の説明として解釈されるべきである。別々の実施形態の文脈で本特許文献で説明されている特定の特徵は、組み合わせて単一の実施形態で実施することも可能である。逆に、単一の実施形態の文脈で説明されている種々の特徴は、複数の実施形態において別々に、又は任意の適切なサブコンビネーションで実施することも可能である。更に、特徴が、特定の組み合わせにおいて作用するものとして上述されていたり、当初にはそのようにクレームされていたりさえするかもしれないが、クレームされた組み合わせ中の1つ以上の特徴は、場合によっては、組み合わせから切り出されることが可能であり、クレームされた組み合わせは、サブコンビネーション又はサブコンビネーションの変形例に仕向けられる可能性がある。

20

30

【0370】

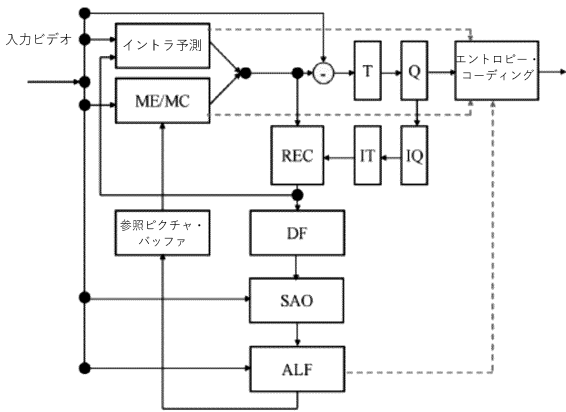
同様に、図中、動作は特定の順序で描かれているが、これは、所望の結果を達成するために、このような動作が図示の特定の順序で又は順番通りに実行されること、又は、例示された全ての動作が実行されること、を要求するものとして理解されるべきではない。更に、この特許文献で説明される実施形態における種々のシステム構成要素の分け方は、全ての実施形態でこのような分け方を要求とするものとして理解されるべきではない。

【0371】

僅かな実装例及び実施例のみが記述されているに過ぎず、本特許文献で説明され図示されているものに基づいて他の実装、拡張及び変更を行うことが可能である。

40

【図面】
【図 1】



【図 2】

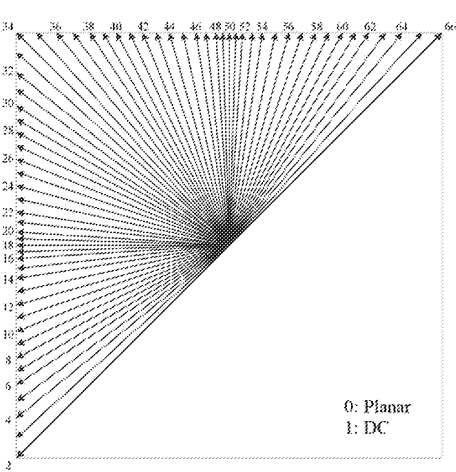
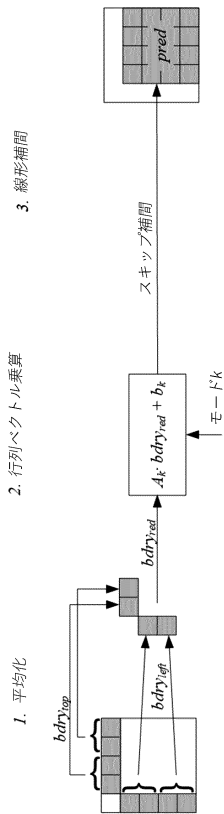
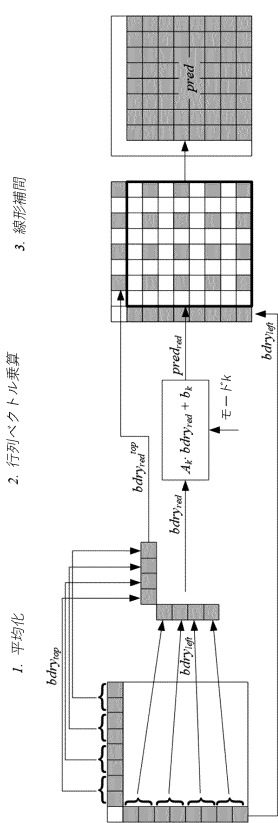


FIG. 2

【図 3】



【図 4】



10

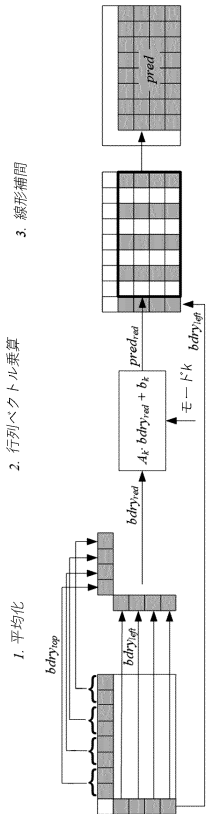
20

30

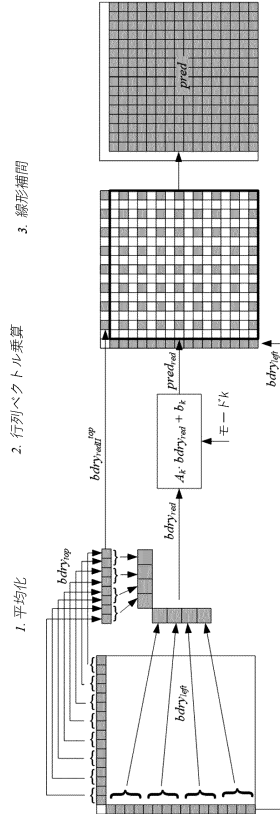
40

50

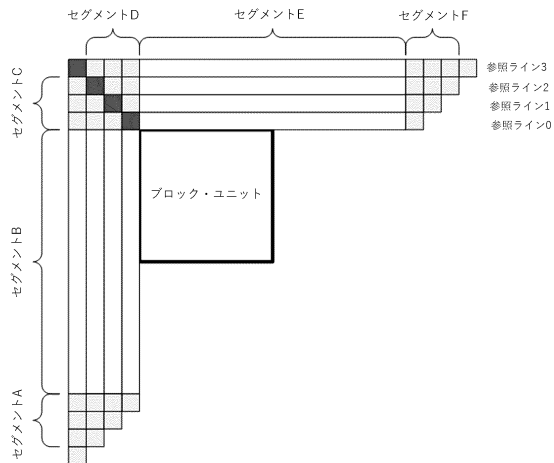
【図 5】



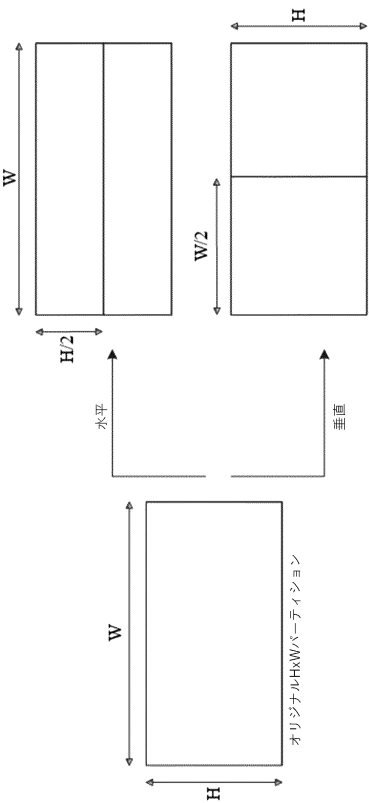
【図 6】



【図 7】



【図 8】



10

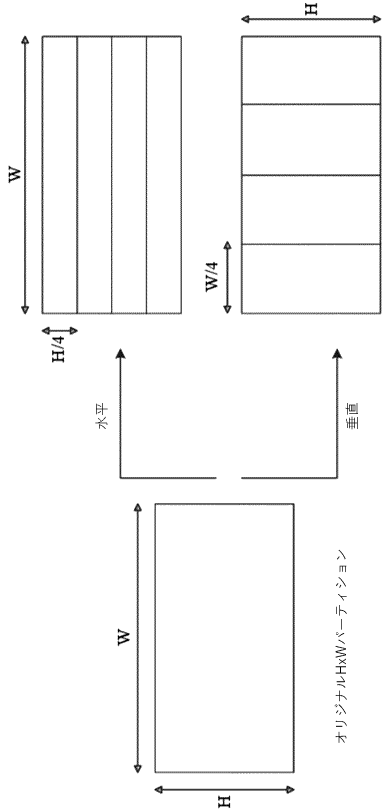
20

30

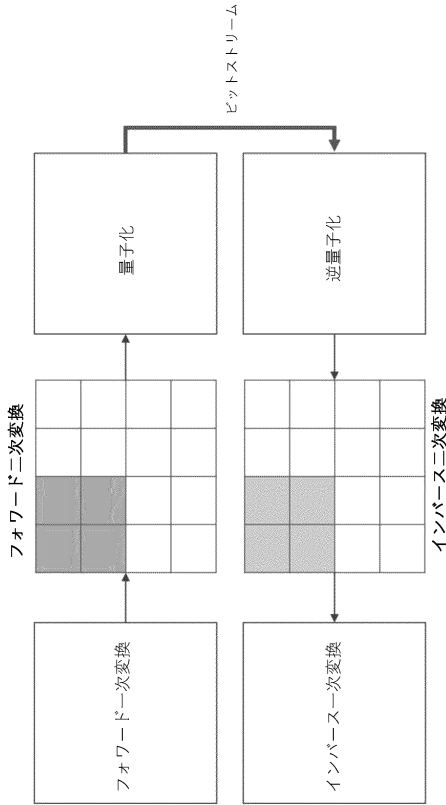
40

50

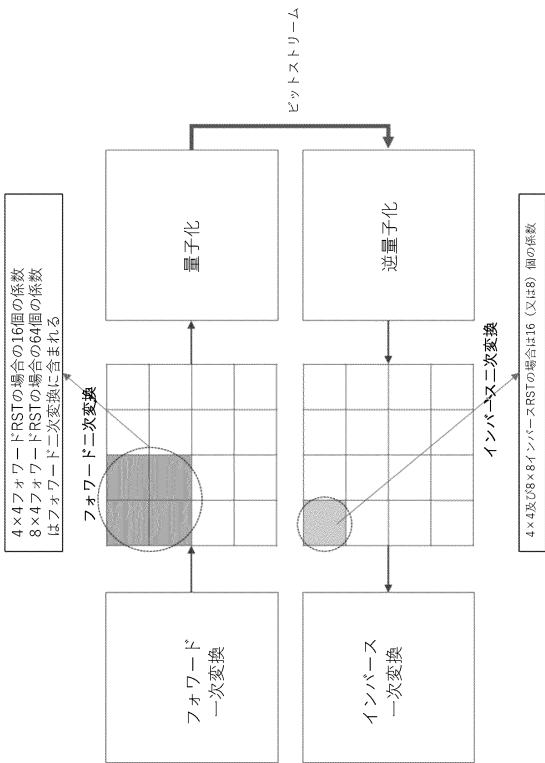
【図 9】



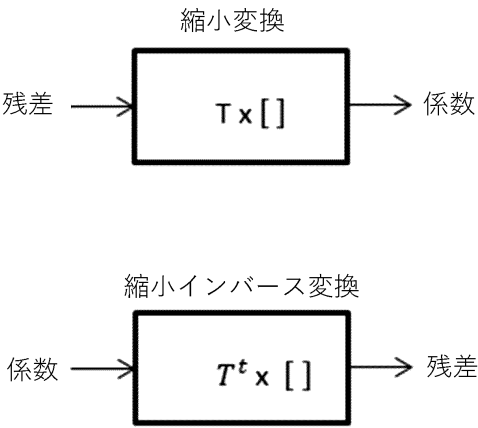
【図 10】



【図 11】



【図 12】



10

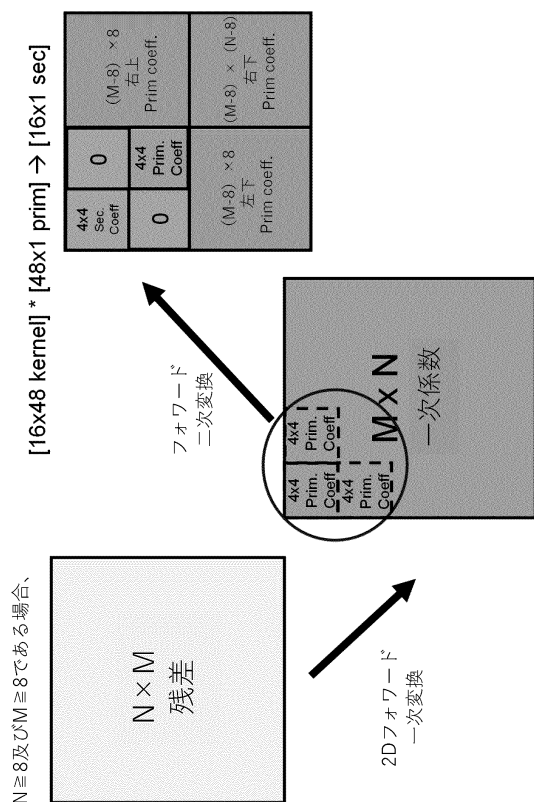
20

30

40

50

【図 13】

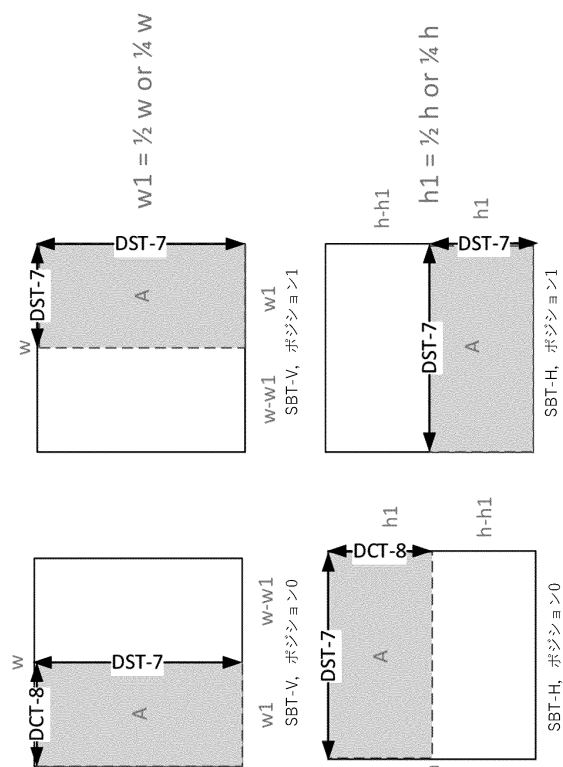


【図 14】

[illegible]

FIG. 14

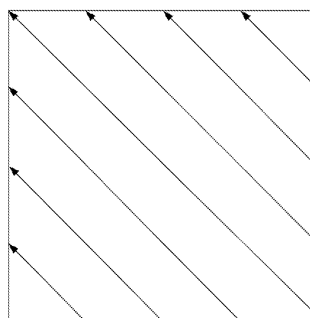
【図 15】



【図 16】

	Y = 0	1	2	3
X = 0	0	1	3	6
1	2	4	7	10
2	5	8	11	13
3	9	12	14	15

FIG. 16



【図 1 7】

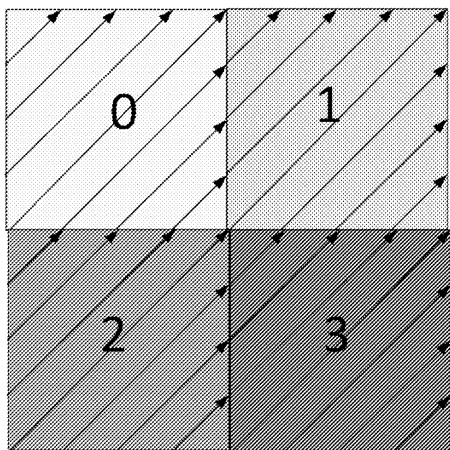


FIG. 17

【図 1 8】

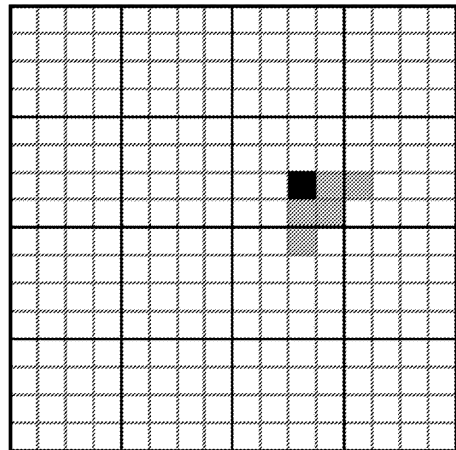


FIG. 18

【図 1 9】

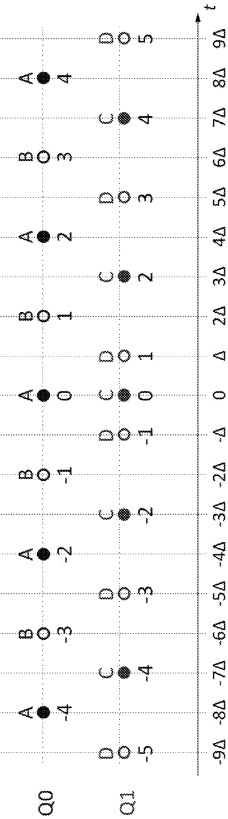
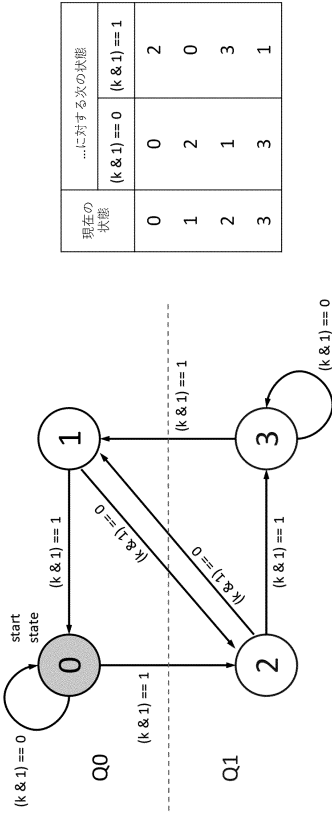


FIG. 19

【図 2 0】



10

20

30

40

50

【図 2 1】

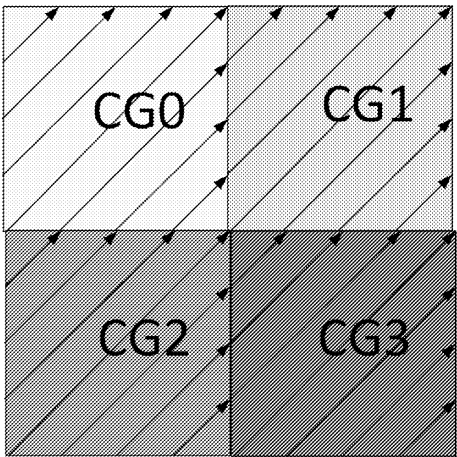
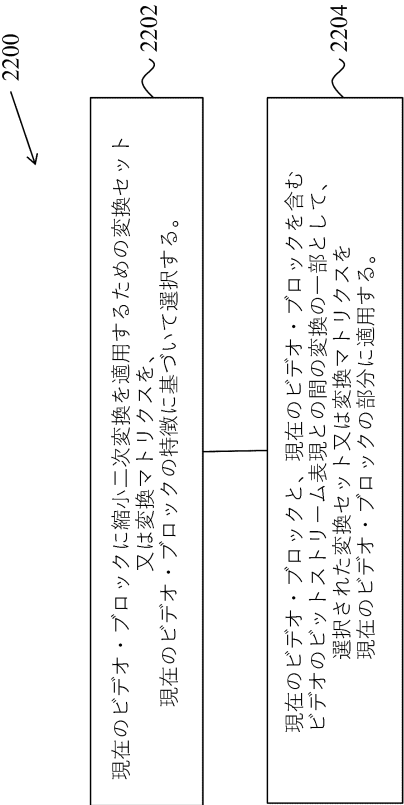


FIG. 21

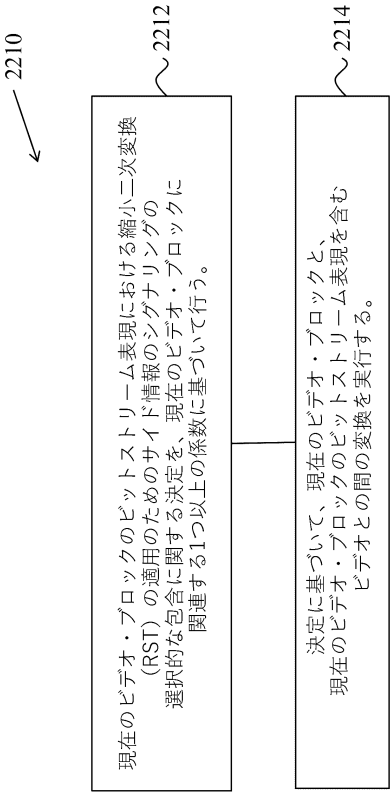
【図 2 2 A】



10

20

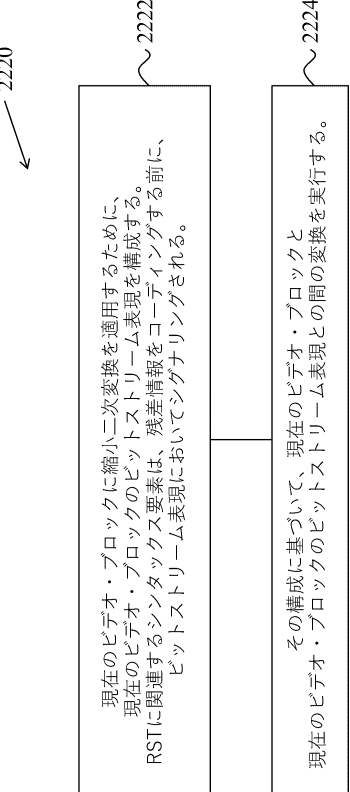
【図 2 2 B】



30

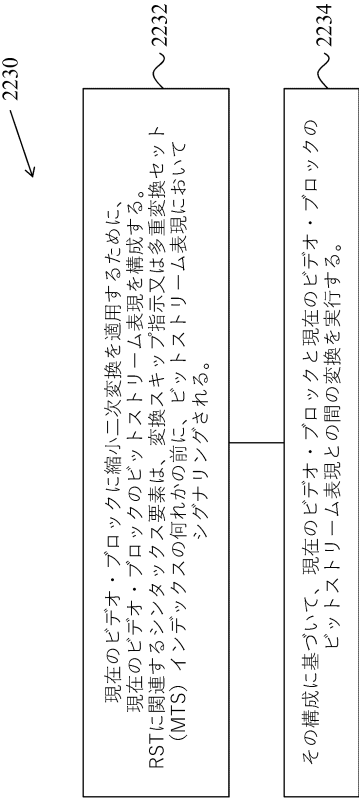
40

【図 2 2 C】

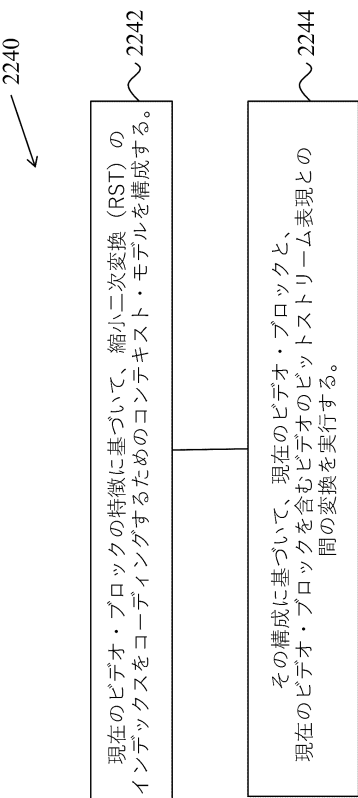


50

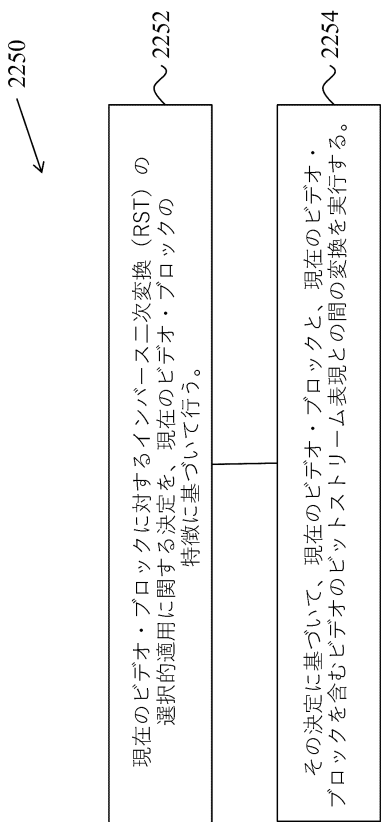
【図 2 2 D】



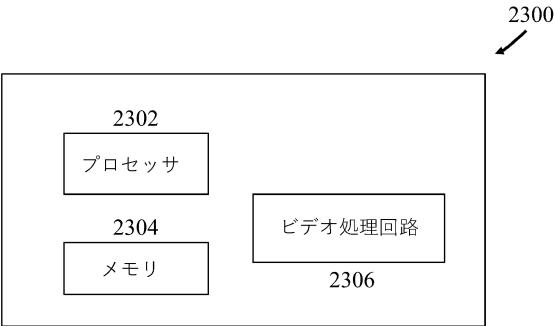
【図 2 2 E】



【図 2 2 F】



【図 2 3 A】



10

20

30

40

50

【図 2 3 B】

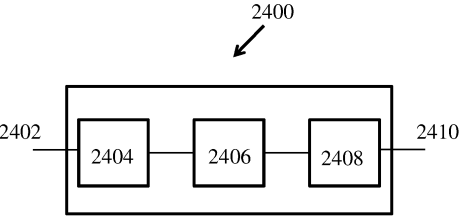


FIG. 23B

【図 2 4 A】

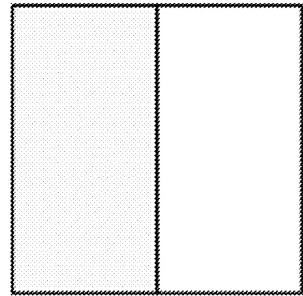


FIG. 24A

【図 2 4 B】

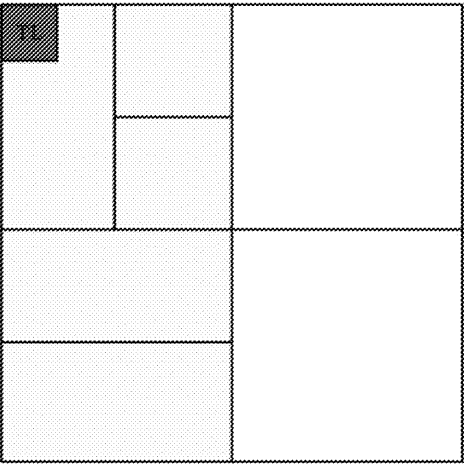


FIG. 24B

【図 2 5】

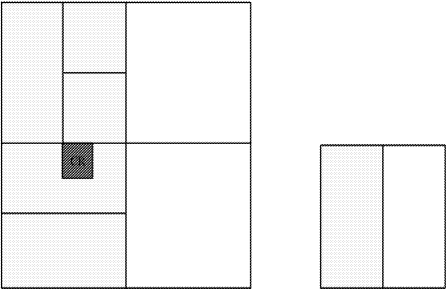


FIG. 25

10

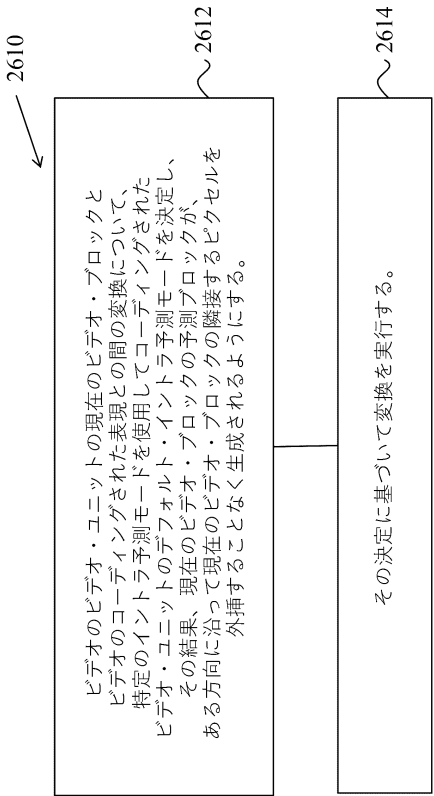
20

30

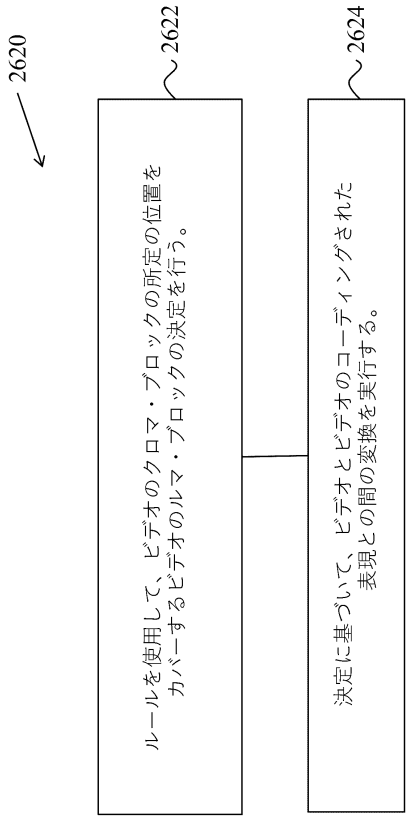
40

50

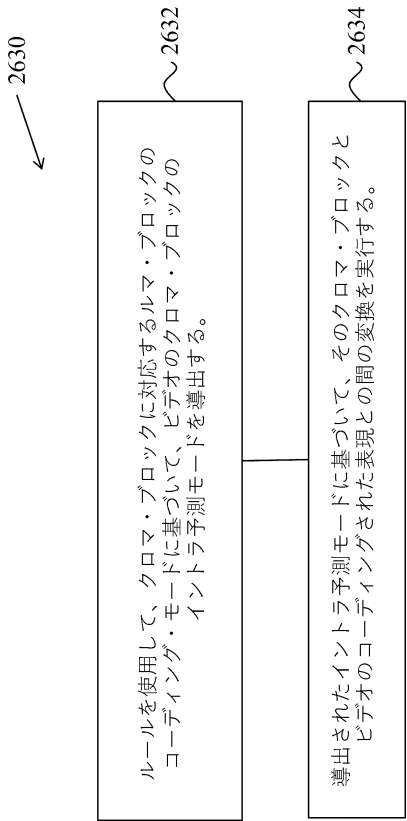
【図 2 6 A】



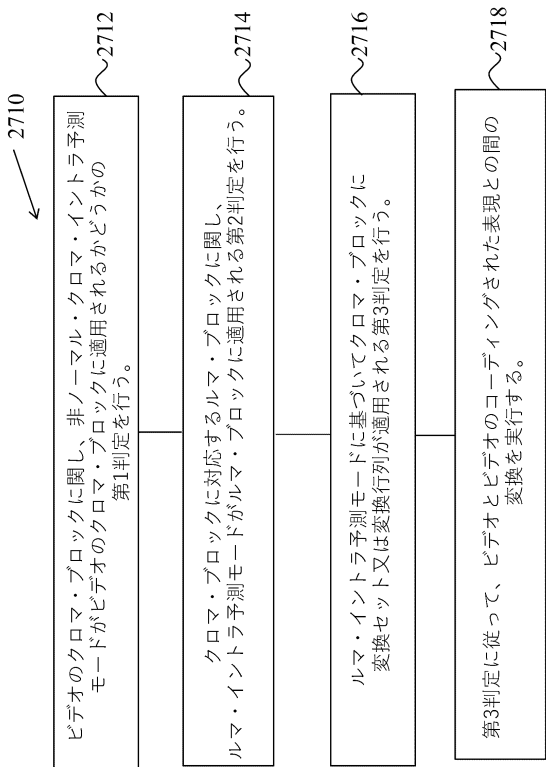
【図 2 6 B】



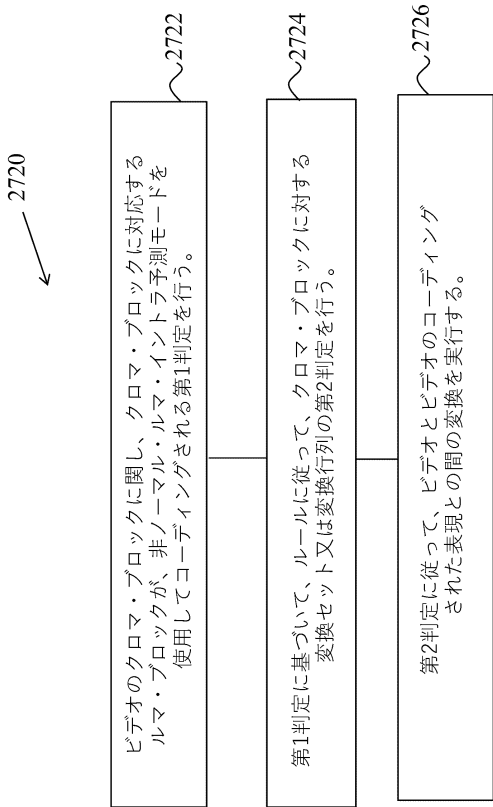
【図 2 6 C】



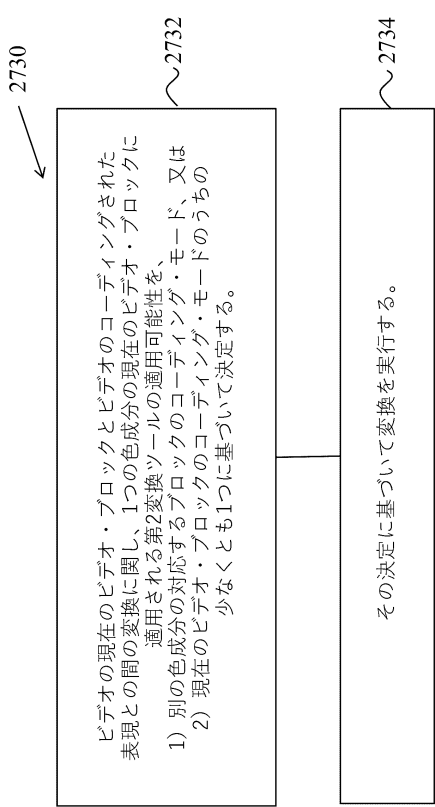
【図 2 7 A】



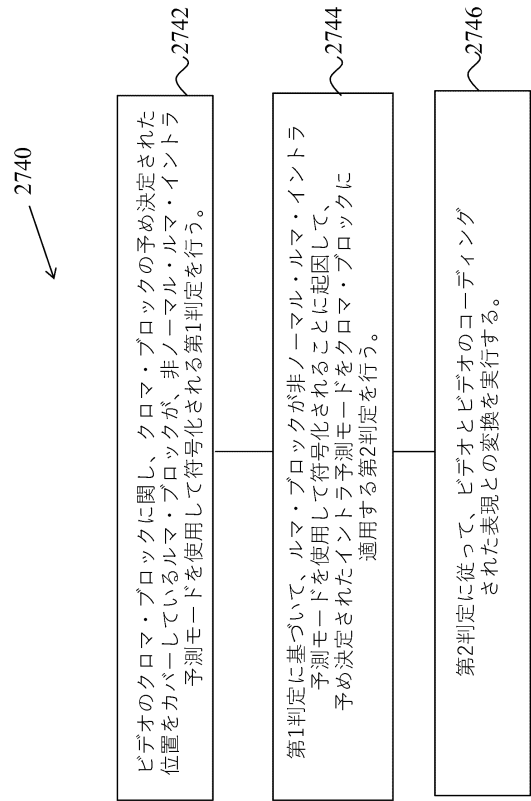
【図 2 7 B】



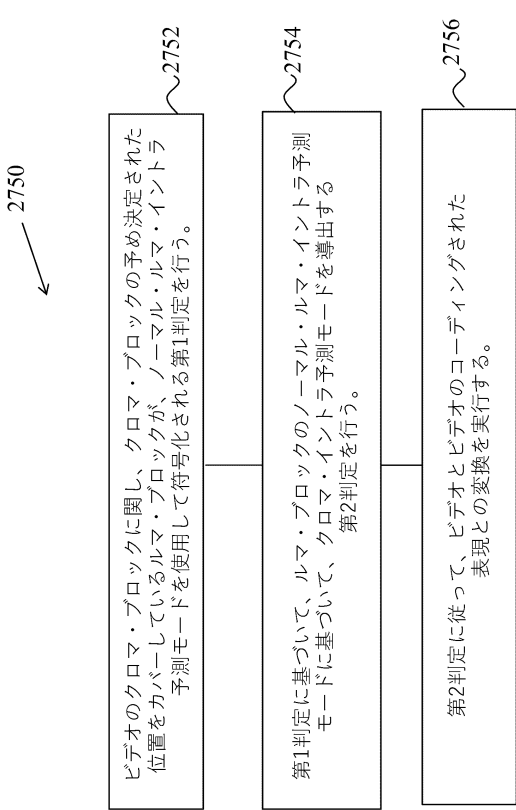
【図 2 7 C】



【図 2 7 D】



【図 2 7 E】



10

20

30

40

50

フロントページの続き

BYTEDANCE INC.

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137 Los Angeles, California 90066 United States of America

(74)代理人 100107766

弁理士 伊東 忠重

(74)代理人 100070150

弁理士 伊東 忠彦

(74)代理人 100135079

弁理士 宮崎 修

(72)発明者 ザン, リー

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

(72)発明者 ザン, カイ

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

(72)発明者 リュウ, ホンビン

中国 100080 ベイジン, ハイディエン ディストリクト, ズィチュン ロード, ナンバー・63, チャイナ サテライト コミュニケーションズ タワー, ジンリトウシャオ ポスト オフィス

(72)発明者 ワン, ユエ

中国 100080 ベイジン, ハイディエン ディストリクト, ズィチュン ロード, ナンバー・63, チャイナ サテライト コミュニケーションズ タワー, ジンリトウシャオ ポスト オフィス

審査官 大桃 由紀雄

(56)参考文献 特表2022-542851 (JP, A)

特表2022-541700 (JP, A)

BROSS, Benjamin et al., Versatile Video Coding (Draft 6), JVET-O2001 (version 14), 2019年07月31日, pp.19,137-138,164-165,285-298

Kai Zhang, et al., Non-CE3: Fixed MPMs for MIP, JVET-00255 (Version 1), 2019年06月26日

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00-19/98