

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7451504号
(P7451504)

(45)発行日 令和6年3月18日(2024.3.18)

(24)登録日 令和6年3月8日(2024.3.8)

(51)国際特許分類

H 0 4 N 19/70 (2014.01)

F I

H 0 4 N 19/70

請求項の数 12 (全97頁)

(21)出願番号	特願2021-515143(P2021-515143)
(86)(22)出願日	令和1年9月20日(2019.9.20)
(65)公表番号	特表2022-501896(P2022-501896)
A)	
(43)公表日	令和4年1月6日(2022.1.6)
(86)国際出願番号	PCT/KR2019/012291
(87)国際公開番号	WO2020/060328
(87)国際公開日	令和2年3月26日(2020.3.26)
審査請求日	令和4年8月26日(2022.8.26)
(31)優先権主張番号	10-2018-0114350
(32)優先日	平成30年9月21日(2018.9.21)
(33)優先権主張国・地域又は機関	韓国(KR)
(31)優先権主張番号	10-2018-0114345
(32)優先日	平成30年9月21日(2018.9.21)

最終頁に続く

(73)特許権者	516227559 オッポ広東移動通信有限公司 GUANGDONG OPPO MOBI LE TELECOMMUNICATI ONS CORP., LTD. 中華人民共和国広東省東莞市長安鎮烏沙 海浜路18号 No. 18 Haibin Road, Wusha, Chang'an, Don guan, Guangdong 52 3860 China
(74)代理人	100091487 弁理士 中村 行孝
(74)代理人	100120031 弁理士 宮嶋 学

最終頁に続く

(54)【発明の名称】 ビデオ信号符号化 / 復号化方法およびそのための機器

(57)【特許請求の範囲】**【請求項 1】**

ビデオ復号化方法であって、

残差係数が非ゼロであるか否かを示す非ゼロフラグをピットストリームから解析すること、

前記非ゼロフラグが、前記残差係数が非ゼロであることを示す場合、前記ピットストリームから絶対値情報を解析することであって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、ことと、

前記絶対値情報に基づいて、前記残差係数の絶対値を決定することと、を含み、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記ピットストリームからパリティフラグをさらに解析し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに解析し、前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに解析し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ復号化方法。

【請求項 2】

前記調整残余係数が第2値以下である場合、前記パリティフラグの値に従って、前記残差係数が $2N$ または $2N+1$ (N は前記第2値である)であると決定する、

請求項1に記載のビデオ復号化方法。

【請求項 3】

前記調整残余係数が前記第2値を超える場合、残差値情報をさらに解析し、

前記残差値情報は、前記調整残余係数から前記第2値を差し引いた値である、

請求項1に記載のビデオ復号化方法。

【請求項 4】

ビデオ符号化方法であって、

残差係数が非ゼロであるか否かを示す非ゼロフラグを符号化することと、

前記残差係数が非ゼロである場合、絶対値情報を符号化することであって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、ことと、を含み、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記残差係数のパリティフラグをさらに符号化し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに符号化し、

前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに符号化し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ符号化方法。

【請求項 5】

前記調整残余係数が前記第2値以下である場合、前記パリティフラグの値に従って、前記残差係数が $2N$ または $2N+1$ であると決定する、

請求項4に記載のビデオ符号化方法。

【請求項 6】

前記調整残余係数が前記第2値を超える場合、残差値情報をさらに符号化し、

前記残差値情報は、前記調整残余係数から前記第2値を差し引いた値である、

請求項4に記載のビデオ符号化方法。

【請求項 7】

ビデオ復号化デバイスであって、

残差係数が非ゼロであるか否かを示す非ゼロフラグをビットストリームから解析するための手段と、

前記非ゼロフラグが、前記残差係数が非ゼロであることを示す場合、前記ビットストリームから絶対値情報を解析するための手段であって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、手段と、

前記絶対値情報に基づいて、前記残差係数の絶対値を決定するための手段と、を含み、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記ビットストリームからパリティフラグをさらに解析し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに解析し、

前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに解析し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ復号化方法。

10

20

30

40

50

す、前記ビデオ復号化デバイス。

【請求項 8】

ビデオ符号化デバイスであって、

残差係数が非ゼロであるか否かを示す非ゼロフラグを符号化するための手段と、

前記残差係数が非ゼロである場合、絶対値情報を符号化するための手段であって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、手段と、を含み、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記残差係数のパリティフラグをさらに符号化し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに符号化し、前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに符号化し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ符号化デバイス。

【請求項 9】

ビデオ復号化デバイスであって、

命令を記憶したメモリと、プロセッサと、を含み、前記プロセッサは前記命令を実行して、

残差係数が非ゼロであるか否かを示す非ゼロフラグをピットストリームから解析することと、

前記非ゼロフラグが、前記残差係数が非ゼロであることを示す場合、前記ピットストリームから絶対値情報を解析することであって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、ことと、

前記絶対値情報に基づいて、前記残差係数の絶対値を決定することと、を実行するよう構成され、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記ピットストリームからパリティフラグをさらに解析し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに解析し、前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに解析し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ復号化デバイス。

【請求項 10】

ビデオ符号化デバイスであって、

命令を記憶したメモリと、プロセッサと、を含み、前記プロセッサは前記命令を実行して、

残差係数が非ゼロであるか否かを示す非ゼロフラグを符号化することと、

前記残差係数が非ゼロである場合、絶対値情報を符号化することであって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、ことと、を実行するよう構成され、

前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み、

前記残差係数が前記第1値を超える場合にのみ、前記残差係数のパリティフラグをさら

10

20

30

40

50

に符号化し、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し、

前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに符号化し、前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示し、

前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに符号化し、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す、前記ビデオ符号化デバイス。

【請求項11】

コンピュータ可読記憶媒体であって、

コンピュータに請求項1から3のいずれか一項に記載のビデオ復号化方法を実行させるためのプログラムを記憶した、前記コンピュータ可読記憶媒体。

【請求項12】

コンピュータ可読記憶媒体であって、

コンピュータに請求項4から6のいずれか一項に記載のビデオ符号化方法を実行させるためのプログラムを記憶した、前記コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオ信号符号化／復号化方法およびそのための機器に関するものである。

20

【背景技術】

【0002】

ディスプレイパネルがますます大きくなる傾向に伴い、ますます高い品質のビデオサービスが求められている。高解像度ビデオサービスの最大の問題は、データ量の大幅な増加である。この問題を解決するために、ビデオ圧縮率を向上させるための研究が活発に進められている。代表的な例として、2009年に、MPEG (Motion Picture Experts Group)とITU-T (International Telecommunication Union - Telecommunication)傘下のVCEG (Video Coding Experts Group)は、JCT-VC (Joint Collaborative Team on Video Coding)を設立した。JCT-VCは、H.264 / AVCに比べて約2倍の圧縮性能を有するビデオ圧縮規格であるHEVC (High Efficiency Video Coding)を提案し2013年1月25日に承認された。高解像度ビデオサービスの急速な発展に伴い、HEVCの性能は徐々にその限界を露呈した。

30

【発明の概要】

【発明が解決しようとする課題】

【0003】

本発明は、ビデオ信号を符号化／復号化するときに、残差係数を符号化／復号化する方法およびその方法のための機器を提供することを目的とする。

40

【0004】

本発明は、ビデオ信号を符号化／復号化するときに、残差係数の大きさを閾値と比較するフラグを使用して、残差係数を符号化／復号化する方法およびその方法のための機器を提供することを目的とする。

【0005】

本発明は、ビデオ信号を符号化／復号化するときに、残差係数が偶数であるか奇数あるかを示すフラグを使用して、残差係数を符号化／復号化する方法およびその方法のための機器を提供することを目的とする。

【0006】

本発明で解決しようとする技術的課題は、上記の技術的課題に限定されず、本発明が属する技術分野の当業者は、以下の説明を介して、本発明で言及されていない他の技術的課

50

題を明確に理解することができる。

【課題を解決するための手段】

【0007】

本発明のビデオ信号復号化方法は、残差係数が非ゼロであるか否かを示す非ゼロフラグをビットストリームから解析することと、前記非ゼロフラグが、前記残差係数が非ゼロではないことを示す場合、前記ビットストリームから絶対値情報解析することであって、前記絶対値情報は前記残差係数の絶対値を決定するために使用される、ことと、前記絶対値情報に基づいて、前記残差係数の絶対値を決定することと、を含む。ここで、前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み得、前記残差係数が第1値を超える場合にのみ、前記ビットストリームからパリティフラグをさらに解析する。

【0008】

本発明のビデオ信号符号化方法は、残差係数が非ゼロであるか否かを示す非ゼロフラグを符号化することと、前記残差係数が非ゼロではない場合、絶対値情報を符号化することとを含み得、前記絶対値情報は前記残差係数の絶対値を決定するために使用される。前記絶対値情報は、前記残差係数が第1値を超えるか否かを示す残差係数比較フラグを含み得、前記残差係数が第1値を超える場合にのみ、前記残差係数のパリティフラグをさらに符号化する。

【0009】

本発明のビデオ信号復号化／符号化方法では、前記パリティフラグは、前記残差係数の値が偶数であるか奇数であるかを示し得る。

【0010】

本発明のビデオ信号復号化／符号化方法では、前記残差係数が前記第1値を超える場合、第1調整残余係数比較フラグをさらに解析することができ、前記第1調整残余係数比較フラグは、前記残差係数を1ビット右にシフトして導出された調整残余係数が第2値を超えるか否かを示す。

【0011】

本発明のビデオ信号復号化／符号化方法では、前記調整残余係数が第2値以下である場合、前記パリティフラグの値に従って、前記残差係数が $2N$ または $2N+1$ であると決定することができる。

【0012】

本発明のビデオ信号復号化／符号化方法では、前記調整残余係数が前記第2値を超える場合、第2調整残余係数比較フラグをさらに解析することができ、前記第2調整残余係数比較フラグは、前記調整残余係数が第3値を超えるか否かを示す。

【0013】

本発明のビデオ信号復号化／符号化方法では、前記調整残余係数が前記第2値を超える場合、残差値情報をさらに解析することができ、前記残差値情報は、前記調整残余係数から前記第2値を差し引いた値である。

【0014】

本発明について上記の簡単に要約された特徴は、後述する本発明の詳細な説明の例示的な実施形態に過ぎず、本発明の範囲を限定するものではない。

【発明の効果】

【0015】

本発明によれば、残差係数を効率的に符号化／復号化することができる。

【0016】

本発明によれば、残差係数の大きさを閾値と比較するフラグを使用することにより、残差係数を効率的に符号化／復号化することができる。

【0017】

本発明によれば、残差係数が偶数であるか奇数であるかを示すフラグを使用することにより、残差係数を効率的に符号化／復号化することができる。

10

20

30

40

50

【0018】

本発明で得られる効果は、上記の効果に限定されず、本発明が属する技術分野の当業者は、以下の説明を介して、本発明で言及されていない他の効果を明確に理解することができる。

【図面の簡単な説明】

【0019】

【図1】本発明の実施例に係るビデオ符号化器（encoder）のブロック図である。

【図2】本発明の実施例に係るビデオ復号化器（decoder）のブロック図である。

【図3】本発明の実施例に係る基本符号化ツリーユニットを示す図である。

【図4】符号化ブロックの様々な分割形態を示す図である。 10

【図5】符号化ツリーユニットの分割例を示す図である。

【図6】プリセットされたサイズの符号化ツリーユニットより小さいサイズのブロックが画像境界に現れる例を示す図である。

【図7】非定型境界の境界ブロックに対して四分木分割を実行する例を示す図である。

【図8】画像境界に隣接するブロックに対して四分木分割を実行する例を示す図である。

【図9】画像境界に隣接するブロックの分割パターンを示す図である。

【図10】画像境界に隣接するブロックの符号化パターンを示す図である。

【図11】本発明の実施例によるインター予測方法のフローチャートである。

【図12】マージモードで現在のブロックの動き情報を導出するプロセスのフローチャートである。 20

【図13】マージ候補を導出するための候補ブロックを示す図である。

【図14】参照サンプルの位置を示す図である。

【図15】マージ候補を導出するための候補ブロックを示す図である。

【図16】参照サンプルの位置を変更する例を示す図である。

【図17】参照サンプルの位置を変更する例を示す図である。

【図18】インター領域動き情報テーブルの更新例を説明するための図である。

【図19】インター領域マージ候補テーブルの更新例を示す図である。

【図20】記憶されたインター領域マージ候補のインデックスが更新される例を示す図である。

【図21】代表サブブロックの位置を示す図である。 30

【図22】インター予測モードごとにインター領域動き情報テーブルを生成する例を示す図である。

【図23】長期動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加する例を示す図である。

【図24】マージ候補の一部のみに対して冗長検査を実行する例を示す図である。

【図25】特定のマージ候補について冗長検査を省略する例を示す図である。

【図26】現在のブロックと同じ並列マージ領域に含まれる候補ブロックがマージ候補として使用できないように設定される例を示す図である。

【図27】一時動き情報テーブルを示す図である。

【図28】インター領域動き情報テーブルと一時動き情報テーブルをマージする例を示す図である。 40

【図29】本発明の実施例によるイントラ予測方法のフローチャートである。

【図30】各参照サンプルラインに含まれる参照サンプルを示す図である。

【図31】イントラ予測モードを示す図である。

【図32】参照サンプルを一列に配列する一次元配列の例を示す図である。

【図33】参照サンプルを一列に配列する一次元配列の例を示す図である。

【図34】方向性イントラ予測モードとx軸に平行な直線との間に形成される角度を示す図である。

【図35】現在のブロックが非正方形である場合に予測サンプルを取得する例を示す図である。 50

- 【図36】広角イントラ予測モードを示す図である。
- 【図37】PDPの適用を示す図である。
- 【図38】二次変換が実行されるサブブロックを示す図である。
- 【図39】二次変換が実行されるサブブロックを示す図である。
- 【図40】現在のブロックの変換タイプを決定する例を説明するための図である。
- 【図41】残差係数を符号化する方法のフローチャートである。
- 【図42】異なる走査順序で残差係数の配列順序を示す図である。
- 【図43】異なる走査順序で残差係数の配列順序を示す図である。
- 【図44】最後の非ゼロ係数の位置を符号化する例を示す図である。
- 【図45】残差係数の絶対値を符号化するプロセスのフローチャートである。
- 【図46】残差係数の絶対値を符号化するプロセスのフローチャートである。
- 【図47】残差係数の絶対値を符号化するプロセスのフローチャートである。
- 【図48】ブロック強度を決定するプロセスのフローチャートである。
- 【図49】事前定義されたフィルタ候補を示す図である。

【発明を実施するための形態】

【0020】

以下、図面を参照して、本発明の実施例を詳細に説明する。

【0021】

ビデオの符号化及び復号化は、ブロック単位で行われる。例えば、符号化ブロック、変換ブロック、または予測ブロックについて、変換、量子化、予測、インループフィルタリング、または復元などの符号化／復号化処理を実行することができる。

【0022】

以下、符号化／復号化されるブロックを「現在のブロック」と呼ぶ。例えば、現在の符号化／復号化処理プロセスに応じて、現在のブロックは、符号化ブロック、変換ブロック、または予測ブロックを示すことができる。

【0023】

さらに、本明細書で使用される用語「ユニット」は、特定の符号化／復号化プロセスを実行するための基本ユニットを表し、「ブロック」は、所定の大きさのサンプルアレイを示すものと理解することができる。別の説明がない限り、「ブロック」と「ユニット」は、同等の意味で使用することができる。例えば、後述される実施例では、符号化ブロックと符号化ユニットは、相互同等の意味を持つものと理解することができる。

【0024】

図1は、本発明の実施例に係るビデオ符号化器(encoder)のブロック図である。

【0025】

図1を参照すると、ビデオ符号化機器100は、画像分割部110、予測部120、125、変換部130、量子化部135、再並べ替え部160、エントロピー符号化部165、逆量子化部140、逆変換部145、フィルタ部150、およびメモリ155を含み得る。

【0026】

図1に示される各構成要素は、ビデオ符号化機器における異なる特徴的な機能を示すために、独立して図示したものであり、各構成要素が別個のハードウェアまたは単一のソフトウェア構成要素から構成されることを意味するものではない。つまり、各構成要素について、説明の便宜のために、少なくとも2つの構成要素が1つの構成要素に組み合わされるか、または1つの構成要素が複数の構成要素に分割されて機能を実行することができ、本発明の本質から逸脱しない限り、これらの各構成要素を統合する実施例と各構成要素を分離する実施例も、本発明の保護範囲に含まれる。

【0027】

また、構成要素の一部は、本発明の本質的な機能を実行するために必要な構成要素ではなく、単に性能を向上させるためのオプションの構造要素であり得る。本発明は、単に性能を向上させるための構成要素以外の、本発明の本質を実現するために必要な構成要素の

10

20

30

40

50

みを含み得、単に性能を向上させるための構成要素以外の必要な構造要素のみを含構造も、本発明の保護範囲に含まれる。

【0028】

画像分割部110は、入力された画像を少なくとも1つの処理ユニットに分割することができる。この場合、処理ユニットは、予測ユニット(PU: Prediction Unit)であってもよいし、変換ユニット(TU: Transform Unit)であってもよいし、符号化ユニット(CU: Coding Unit)であってもよい。画像分割部110は、1つの画像を複数の符号化ユニット、予測ユニット、および変換ユニットの組み合わせに分割し、所定の標準(コスト関数など)に基づいて、1つの符号化ユニット、予測ユニット、および変換ユニットの組み合わせを選択して、画像を符号化することができる。10

【0029】

例えば、1つの画像を複数の符号化ユニットに分割することができる。画像を符号化ユニットに分割するためには、四分木構造(Quad Tree Structure)のような再帰ツリー構造を使用でき、1つのビデオまたは最大符号化ユニット(largest coding unit)をルート(root)として使用して、他の符号化ユニットに分割される符号化ユニットは、分割された符号化ユニットの数のサブノードを持って分割されることができる。特定の制限に応じてこれ以上分割されない符号化ユニットは、リーフノードとなる。つまり、1つの符号化ユニットに対して正方形分割のみが可能であると仮定した場合、1つの符号化ユニットは、最大4つの他の符号化ユニットに分割されることができる。20

【0030】

以下、本発明の実施例では、符号化ユニットは、符号化を実行するユニットを意味してもよいし、復号化を実行するユニットを意味してもよい。

【0031】

1つの符号化ユニット内の予測ユニットを同じサイズの正方形および長方形のうちの少なくとも1つに分割することができ、または、1つの符号化ユニット内で分割された予測ユニットのいずれかの予測ユニットが別の1つの予測ユニットと異なる形状および/またはサイズを有するように分割することができる。

【0032】

符号化ユニットに基づいてイントラ予測を実行する予測ユニットが最小の符号化ユニットでない場合、複数の予測ユニット $N \times N$ に分割することなくイントラ予測を実行することができる。30

【0033】

予測部120、125は、インター予測を実行するインター予測部120およびイントラ予測を実行するイントラ予測部125を含むことができる。予測ユニットに対して、インター予測を実行するかイントラ予測を実行するかを決定し、各予測方法に基づいて具体的な情報(例えば、イントラ予測モード、動きベクトル、参照画像など)を決定できる。この場合、予測を実行する処理ユニットは、予測方法および具体的な内容を決定する処理ユニットとは異なる場合がある。例えば、予測ユニットによって予測方法と予測モードなどを決定でき、変換ユニットによって予測を実行することができる。生成された予測ブロックと元のブロックとの間の残差値(残差ブロック)を変換部130に入力することができる。また、予測に使用される予測モード情報、動きベクトル情報などは、残差値と一緒にエントロピー符号化部165で符号化されて復号化器に伝送されることができる。特定の符号化モードを使用する場合、予測ユニット120および125によって予測されたブロックを生成せずに、元のブロックを直接符号化して復号化器に伝送することもできる。40

【0034】

インター予測部120は、現在の画像の前の画像またはそれ以降の画像のうちの少なくとも1つの画像の情報に基づいて、予測ユニットを予測することができ、場合によっては、現在の画像内の符号化された領域の一部の情報に基づいて予測ユニットを予測すること

10

20

30

40

50

もできる。インター予測部 120 は、参照画像補間部、動き予測部、および動き補償部を含み得る。

【0035】

参照画像補間部は、メモリ 155 から参照画像情報を受け取り、参照画像から整数画素以下の画素情報を生成することができる。輝度画素の場合、1 / 4 画素単位で整数画素以下の画素情報を生成するために、異なるフィルタ係数を持つ DCT ベースの 8 タップ補間フィルタ (DCT-based Interpolation Filter) を使用できる。彩度信号の場合、1 / 8 画素単位で整数画素以下の画素情報を生成するために、異なるフィルタ係数を持つ DCT ベースの 4 タップ補間フィルタ (DCT-based Interpolation Filter) を使用できる。

10

【0036】

動き予測部は、参照画像補間部によって補間された参照画像に基づいて動き予測を実行することができる。動きベクトルを計算するための方法については、完全検索ベースのブロックマッチングアルゴリズム (FBMA: Full search-based Block Matching Algorithm)、3ステップ検索法 (TSS: Three Step Search)、新しい3ステップ検索法 (NTS: New Three-Step Search Algorithm) など、様々な方法可能である。動きベクトルは、補間された画素に基づき 1 / 2 画素または 1 / 4 画素を単位とする動きベクトル値を持つことができる。動き予測部で異なる動き予測方法を採用することによって現在の予測ユニットを予測することができる。動き予測方法については、スキップ (Skip) 方法、マージ (Merge) 方法、アドバンスド動きベクトル予測 (AMVP: Advanced Motion Vector Prediction) 方法、イントラブロックコピー (Intra Block Copy) 方法など、様々な方法が使用可能である。

20

【0037】

イントラ予測部 125 は、現在の画像内の画素情報である現在のブロック周辺の参照画像情報に基づいて、予測ユニットを生成することができる。現在の予測ユニットの隣接ブロックが、インター予測が行われたブロックであり、参照画素がインター予測が行われた画素である場合、インター予測が行われたブロックに含まれる参照画素を、周辺のイントラ予測が行われたブロックの参照画素情報として使用することができる。つまり、参照画素が使用可能ではない場合、使用可能な参照画素のうちの少なくとも 1 つの参照画素を使用して、使用できない参照画素情報を置き換えることができる。

30

【0038】

イントラ予測では、予測モードは、予測方向に応じて参照画素情報を使用する方向性予測モードと、予測を実行するときに方向情報を使用しない非方向性モードとを有し得る。輝度情報を予測するためのモードと、彩度情報を予測するためのモードが異なることができ、彩度情報を予測するために、輝度情報を予測するために使用されるイントラ予測モード情報または予測された輝度信号情報を活用することができる。

【0039】

イントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと同じである場合、予測ユニットの左側に存在する画素、左上に存在する画素および上側に存在する画素に基づいて、予測ユニットに対してイントラ予測を実行できる。ただし、イントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと異なる場合、変換ユニットに基づく参照画素を使用してイントラ予測を実行することができる。また、最小符号化ユニットにのみ、N × N 分割のイントラ予測を使用することができる。

40

【0040】

イントラ予測方法は、予測モードに応じて、適応イントラスムージング (AIS: Adaptive Intra Smoothing) フィルタを参照画素に適用した後、予測ブロックを生成することができる。参照画素に適用される AIS フィルタのタイプは異なる場合がある。イントラ予測方法を実行するために、現在の予測ユニットの周辺に存在する予測ユニットのイントラ予測モードから、現在の予測ユニットのイントラ予測モードを

50

予測することができる。周辺予測ユニットから予測されたモード情報を使用して、現在の予測ユニットの予測モードを予測する場合、現在の予測ユニットと周辺予測ユニットのイントラ予測モードが同じである場合、所定のフラグ情報を使用して、現在の予測ユニットと周辺予測ユニットの予測モードが同じであることを示す情報を伝送することができ、現在の予測ユニットと周辺予測ユニットの予測モードが異なる場合、エントロピー符号化を実行して、現在のブロックの予測モード情報を符号化することができる。

【0041】

また、予測部120、125で生成された予測ユニットに基づいて予測を行った予測単位と予測ユニットの元のブロックとの間の差である残差値情報を含む残差ブロックを生成することができる。生成された残差ブロックを変換部130に入力することができる。 10

【0042】

変換部130は、離散コサイン変換(DCT)、離散正弦変換(DST)、またはK-L変換(KLT)などの変換方法を使用して、元のブロック、および予測部120、125で生成された予測ユニットの残差係数情報を含む残差ブロックを変換することができる。残差ブロックを変換するためにDCTを適用するか、DSTを適用するか、またはKLTを適用するかは、残差ブロックを生成するために使用される予測ユニットのイントラ予測モード情報に従って決定することができる。

【0043】

量子化部135は、変換部130で周波数領域に変換された値を量子化することができる。量子化係数は、ブロックまたはビデオの重要度に応じて変換できる。量子化部135で算出された値を、逆量子化部140と再並べ替え部160に提供することができる。 20

【0044】

再並べ替え部160は、量子化された残差値に対して係数値の並べ替えを実行することができる。

【0045】

再並べ替え部160、係数走査(Coefficient Scanning)方法により、2次元のブロック形状係数を1次元のベクトル形に変更することができる。例えば、再並べ替え部160は、ジグザグ走査(Zig-Zag Scan)方法を使用して、DC係数から高周波数領域の係数まで走査して、それを1次元ベクトル形に変更することができる。変換ユニットのサイズおよびイントラ予測モードに従って、ジグザグ走査の代わりに、2次元ブロック形状係数を列方向に走査する垂直走査と2次元ブロック形状係数を行方向に走査する水平走査を使用することもできる。つまり、変換ユニットのサイズおよびイントラ予測モードに従って、ジグザグ走査、垂直方向走査、および水平方向走査のうちのどの走査方法を使用するかを決定することができる。 30

【0046】

エントロピー符号化部165は、再並べ替え部160によって算出された値に基づいて、エントロピー符号化を実行することができる。例えば、エントロピー符号化については、指数ゴロンブ(Exponential Golomb)符号化、コンテキスト適応型可変長符号化(CAVLC: Context-Adaptive Variable Length Coding)、コンテキスト適応型バイナリ算術符号化(CABAC: Context-Adaptive Binary Arithmetic Coding)などの様々な符号化方法が使用可能である。 40

【0047】

エントロピー符号化部165は、再並べ替え部160と予測部120、125からの符号化ユニットの残差値係数情報およびブロックタイプ情報、予測モード情報、分割ユニット情報、予測ユニット情報および伝送ユニット情報、動きベクトル情報、参照フレーム情報、ブロックの補間情報、フィルタリング情報などの様々な情報を符号化することができる。

【0048】

エントロピー符号化部165で、再並べ替え部160から入力された符号化ユニットの

係数値に対してエントロピー符号化を実行することができる。

【0049】

逆量子化部140および逆変換部145は、量子化部135で量子化された複数の値を逆量子化し、変換部130で変換された値を逆変換する。逆量子化部140および逆変換部145で生成された残差値は、予測部120、125に含まれる動き予測部、動き補償部、およびイントラ予測部によって予測された予測ユニットとマージして、復元ブロック(Reconstructed Block)を生成することができる。

【0050】

フィルタ部150は、デブロッキングフィルタ、オフセット補正部、適応ループフィルタ(ALF: Adaptive Loop Filter)のうちの少なくとも1つを含み得る。

10

【0051】

デブロッキングフィルタは、復元された画像でブロック間の境界に生じたブロック歪みを除去することができる。デブロッキングを行うか否かを判断するために、ブロックに含まれるいくつかの列または行に含まれる画素に基づいて、現在のブロックにデブロッキングフィルタを適用するか否かを決定できる。ブロックにデブロッキングフィルタを適用する場合、必要なデブロッキングフィルタリング強度に応じて、強いフィルタ(Strong Filter)または弱いフィルタ(Weak Filter)を適用できる。また、デブロッキングフィルタを使用する過程で、垂直フィルタリングと水平フィルタリングを実行する場合、水平方向フィルタリングと垂直方向フィルタリングを並行処理することができる。

20

【0052】

オフセット補正部は、デブロッキングされたビデオに対して、画素単位で元のビデオとのオフセットを補正することができる。特定の画像に対するオフセット補正を実行するために、ビデオに含まれる画素を一定数の領域に分割した後、オフセットを実行する領域を決定し、対応する領域にオフセットを適用する方法、または各画素のエッジ情報を考慮してオフセットを適用する方法を使用することができる。

【0053】

適応ループフィルタ(ALF: Adaptive Loop Filtering)は、フィルタリングされた復元画像と元のビデオを比較した値に基づいて実行されることができる。ビデオに含まれる画素を所定のグループに分割した後、対応するグループに使用される1つのフィルタを決定することにより、グループごとに差別的にフィルタリングを行うことができる。ALFを適用するか否かに関する情報および輝度信号は、符号化ユニット(CU: Coding Unit)で伝送でき、それぞれのブロックに応じて適用されるALFフィルタの形状とフィルタ係数は異なる可能性がある。また、適用対象のブロックの特性に関係なく、同じタイプ(固定タイプ)のALFフィルタを適用することもできる。

30

【0054】

メモリ155は、フィルタ部150によって算出された復元ブロックまたは画像を記憶することができ、インター予測を実行するときに、記憶された復元ブロックまたは画像を予測部120、125に提供することができる。

40

【0055】

図2は、本発明の実施例に係るビデオ復号化器(decoder)のブロック図である。

【0056】

図2を参照すると、ビデオ復号化器200は、エントロピー復号化部210、再並べ替え部215、逆量子化部220、逆変換部225、予測部230、予測部235、フィルタ部240、およびメモリ245を含み得る。

【0057】

ビデオビットストリームをビデオ符号化器に入力する場合、入力されたビットストリームは、ビデオ符号化器と逆のステップで復号化することができる。

50

【0058】

エントロピー復号化部210は、ビデオ符号化器のエントロピー符号化部でエントロピー符号化を実行するステップとは逆のステップでエントロピー復号化を実行することができる。例えば、ビデオ符号化器で実行される方法に対応して、指数ゴロンブ(Exponential Golomb)符号化、コンテキスト適応型可変長符号化(CAVLC: Context-Adaptive Variable Length Coding)、コンテキスト適応型バイナリ算術符号化(CABAC: Context-Adaptive Binary Arithmetic Coding)などの様々な方法が適用可能である。

【0059】

エントロピー復号化部210は、符号化器によって実行されるイントラ予測およびインター予測に関する情報を復号化することができる。

【0060】

再並べ替え部215は、エントロピー復号化部210でエントロピー復号化されたビットストリームを符号化部で並べ替える方法に基づいて、並べ替えを実行することができる。1次元ベクトルの形で表される複数の係数を、再度2次元のブロック形状の係数に復元して並べ替えることができる。再並べ替え部215は、符号化部で実行される係数走査に関する情報を受け取り、対応する符号化部によって実行される走査順序に基づいて、逆走査の方法によって並べ替えを実行することができる。

【0061】

逆量子化部220は、符号化器によって提供される量子化パラメータと並べ替えられたブロックの係数値に基づいて逆量子化を実行することができる。

【0062】

ビデオ符号化器によって実行された量子化結果について、逆変換部225は、変換部によって実行される変換、すなわち、DCT、DST、およびKLTに対して逆変換(すなわち、逆DCT、逆DST、および逆KLT)を実行することができる。逆変換は、ビデオ符号化器で決定された伝送ユニットに基づいて実行することができる。ビデオ復号化器の逆変換部225において、予測方法、現在のブロックのサイズおよび予測方向などの複数の情報に基づいて変換方法(例えば、DCT、DST、KLT)を選択的に実行することができる。

【0063】

予測部230、235は、エントロピー復号化部210によって提供される予測ブロックの生成に関する情報、およびメモリ245によって提供される以前に復号化されたブロックまたは画像に関する情報に基づいて予測ブロックを生成することができる。

【0064】

前述したように、ビデオ符号化器での動作と同様な方式でイントラ予測実行する場合、予測ユニットのサイズと変換ユニットのサイズが同じである場合、予測ユニットの左側に存在する画素、左上に存在する画素、上側に存在する画素に基づいて予測ユニットに対してイントラ予測を実行し、イントラ予測を実行する際の予測ユニットのサイズと変換ユニットのサイズが異なる場合、変換ユニットに基づく参照画素を使用してイントラ予測を実行することができる。また、最小符号化ユニットにのみ、N×N分割のイントラ予測を使用することができる。

【0065】

予測部230、235は、予測ユニット判別部、インター予測部、およびイントラ予測部を含むことができる。予測ユニット判別部は、エントロピー復号化部210から入力された予測ユニット情報、イントラ予測方法の予測モード情報、インター予測方法の動き予測関連情報などの様々な情報を受け取り、現在の符号化ユニットに従って予測ユニットを分類することにより、予測ユニットがインター予測を実行するか、イントラ予測を実行するかを判別することができる。インター予測部230は、ビデオ符号化器によって提供される現在の予測ユニットのインター予測に必要な情報を使用して、現在の予測ユニットが

10

20

30

40

50

属する現在の画像の前の画像または以降の画像のうちの少なくとも1つの画像に含まれる情報に基づいて、現在の予測ユニットに対してインター予測を実行することができる。あるいは、現在の予測ユニットが属する現在の画像内で復元された領域の一部の情報に基づいて、インター予測を実行することもできる。

【0066】

インター予測を実行するために、符号化ユニットに基づいて対応する符号化ユニットに含まれる予測ユニットの動き予測方法がスキップモード (S k i p M o d e)、マージモード (M e r g e M o d e)、アドバンスド動きベクトル予測モード (A M V P M o d e)、イントラブロックコピーモードのうちのどのモードであるかを判断することができる。

10

【0067】

イントラ予測部 235 は、現在の画像内の画素情報に基づいて予測ブロックを生成することができる。予測ユニットがイントラ予測を行った予測ユニットである場合、ビデオ符号化器によって提供される予測ユニットのイントラ予測モード情報に基づいてイントラ予測を実行することができる。イントラ予測部 235 は、適応イントラスムージング (A I S : A d a p t i v e I n t r a S m o o t h i n g) フィルタ、参照画素補間部、および D C フィルタを含み得る。A I S フィルタは、現在のブロックの参照画素に対してフィルタリングを実行する要素であり、現在の予測ユニットの予測モードに従ってフィルタを適用するか否かを決定することができる。ビデオ符号化器によって提供される予測ユニットの予測モードおよび A I S フィルタ情報を使用して、現在のブロックの参照画素に対して A I S フィルタリングを実行することができる。現在のブロックの予測モードが A I S フィルタリングを実行しないモードである場合、A I S フィルタを適用しないことができる。

20

【0068】

予測ユニットの予測モードが、補間された参照画素の画素値に基づいてイントラ予測を実行する予測ユニットである場合、参照画素補間部は、参照画素を補間することによって整数値または分数値の画素単位の参照画素を生成することができる。現在の予測ユニットの予測モードが、参照画素を補間することなく予測ブロックを生成する予測モードである場合、参照画素を補間しなくてもよい。現在のブロックの予測モードが D C モードである場合、D C フィルタは、フィルタリングにより予測ブロックを生成することができる。

30

【0069】

復元されたブロックまたは画像をフィルタ部 240 に提供することができる。フィルタ部 240 は、デブロッキングフィルタ、オフセット補正部、および A L F を含み得る。

【0070】

ビデオ符号化器から、ブロックまたは画像にデブロッキングフィルタを適用するか否かに関する情報、およびデブロッキングフィルタを適用するときに強いフィルタリングを適用するか弱いフィルタリングを適用するかに関する情報を受信することができる。ビデオ復号化器のデブロッキングフィルタから、ビデオ符号化器によって提供されるデブロッキングフィルタの関連情報を受信でき、ビデオ復号化器で、対応するブロックに対してデブロッキングフィルタリングを実行することができる。

40

【0071】

オフセット補正部は、符号化ときにビデオに使用されるオフセット補正のタイプおよびオフセット量情報などに基づいて、復元されたビデオに対してオフセット補正を実行することができる。

【0072】

符号化器によって提供される、A L F を適用するか否かに関する情報、A L F 係数情報などに基づいて、A L F を符号化ユニットに適用することができる。このような A L F 情報は、特定のパラメータセットに含まれることによって提供できる。

【0073】

メモリ 245 は、復元された画像またはブロックを記憶し、前記画像またはブロックを

50

参照画像または参照ブロックとして使用でき、復元された画像を出力部に提供することができる。

【0074】

図3は、本発明の実施例に係る基本符号化ツリーユニットを示す図である。

【0075】

最大サイズの符号化ブロックを符号化ツリーブロックとして定義することができる。1つの画像は、複数の符号化ツリーユニット(CTU: Coding Tree Unit)に分割されることがある。符号化ツリーユニットは、最大サイズの符号化ユニットであり、最大符号化ユニット(LCU:Largest Coding Unit)と呼ばれることもできる。図3は、1つの画像が複数の符号化ツリーユニットに分割された例を示す。

10

【0076】

符号化ツリーユニットのサイズは、画像レベルまたはシーケンスレベルで定義できる。このため、符号化ツリーユニットのサイズを示す情報は、画像パラメータセットまたはシーケンスパラメータセットを介してシグナリングできる。

【0077】

例えば、シーケンス内の画像全体の符号化ツリーユニットのサイズを 128×128 に設定できる。あるいは、画像レベルで 128×128 または 256×256 のいずれかを符号化ツリーユニットのサイズとして決定できる。例えば、第1画像では符号化ツリーユニットのサイズが 128×128 に設定され、第2画像では符号化ツリーユニットのサイズが 256×256 に設定されることができる。

20

【0078】

符号化ツリーユニットを分割することによって符号化ブロックを生成することができる。符号化ブロックは、符号化/復号化処理を実行するための基本ユニットを表す。例えば、異なる符号化ブロックに従って予測または変換を実行でき、または異なる符号化ブロックに従って予測符号化モードを決定することができる。ここで、予測符号化モードは予測画像を生成する方法を表す。例えば、予測符号化モードは、画面内予測(Intra Prediction、イントラ予測)、画面間予測(Inter Prediction、インター予測)、現在の画像参照(Current Picture Referencing、またはイントラブロックコピー(IBC: Intra Block Copy))または複合予測(Combined Prediction)を含み得る。符号化ブロックについては、イントラ予測、インター予測、現在の画像参照または複合予測のうちの少なくとも1つの予測符号化モードを使用して、符号化ブロックに関する予測ブロックを生成することができる。

30

【0079】

現在のブロックの予測符号化モードを示す情報は、ビットストリームを介してシグナリングできる。例えば、前記情報は、予測符号化モードがイントラモードであるか、またはインター予測モードであるかを示す1ビットのフラグであり得る。現在のブロックの予測符号化モードがインター予測モードであると決定された場合にのみ、現在の画像参照または複合予測を使用できる。

40

【0080】

現在の画像参照は、現在の画像を参照画像に設定し、現在の画像内の符号化/復号化された領域から、現在のブロックの予測ブロックを取得するために使用される。ここで、現在の画像は、現在のブロックを含む画像を意味する。現在の画像参照を現在のブロックに適用するか否かを示す情報は、ビットストリームを介してシグナリングできる。例えば、前記情報は1ビットのフラグであり得る。前記フラグがtrueである場合、現在のブロックの予測符号化モードを現在の画像参照として決定でき、前記フラグが偽である場合、現在のブロックの予測モードをインター予測として決定できる。

【0081】

あるいは、参照画像インデックスに基づいて現在のブロックの予測符号化モードを決定できる。例えば、参照画像インデックスが現在の画像を指す場合、現在のブロックの予測

50

符号化モードを現在の画像参照として決定できる。参照画像インデックスが、現在の画像ではなく他の画像を指す場合、現在のブロックの予測符号化モードをインター予測として決定できる。つまり、現在の画像参照は、現在の画像内の符号化 / 復号化された領域の情報を使用する予測方法であり、インター予測は、符号化 / 復号化された他の画像の情報を使用する予測方法である。

【0082】

複合予測は、イントラ予測、インター予測、および現在の画像参照のうちの 2 つ以上を組み合わせた符号化モードを示す。例えば、複合予測を適用する場合、イントラ予測、インター予測、または現在の画像参照のいずれかに基づいて、第 1 予測ブロックを生成し、他の 1 つに基づいて、第 2 予測ブロックを生成することができる。第 1 予測ブロックおよび第 2 予測ブロックを生成する場合、第 1 予測ブロックおよび第 2 予測ブロックの平均演算または加重合演算を介して、最終予測ブロックを生成できる。複合予測を適用するか否かを示す情報は、ビットストリームを介してシグナリングできる。前記情報は 1 ビットのフラグであり得る。

10

【0083】

図 4 は、符号化ブロックの様々な分割形態を示す図である。

【0084】

四分木分割、二分木分割、または三分木分割に基づいて、符号化ブロックを複数の符号化ブロックに分割することができる。四分木分割、二分木分割、または三分木分割に基づいて、分割された符号化ブロックを再び複数の符号化ブロックに分割することもできる。

20

【0085】

四分木分割は、現在のブロックを 4 つのブロックに分割する分割手法を指す。四分木分割の結果として、現在のブロックを 4 つの正方形のパーティションに分割できる（図 4 (a) の「SPLIT_QT」を参照）。

【0086】

二分木分割は、現在のブロックを 2 つのブロックに分割する分割手法を指す。垂直方向に沿って（すなわち、現在のブロックを横切る垂直線を使用する）現在のブロックを 2 つのブロックに分割することを垂直方向二分木分割と呼ぶことができ、水平方向に沿って（すなわち、現在のブロック横切る水平線を使用する）現在のブロックを 2 つのブロックに分割することを水平方向二分木分割と呼ぶことができる。二分木分割により、現在のブロックを 2 つの非正方形のパーティションに分割することができる。図 4 (b) の「SPLIT_BT_VER」は、垂直方向二分木分割結果を示し、図 4 (c) の「SPLIT_BT_HOR」は、水平方向二分木分割結果を示す。

30

【0087】

三分木分割は、現在のブロックを 3 つのブロックに分割する分割手法を指す。垂直方向に沿って（すなわち、現在のブロックを横切る 2 つの垂直線を使用する）現在のブロックを 3 つのブロックに分割することを垂直方向三分木分割と呼ぶことができ、水平方向に沿って（すなわち、現在のブロック横切る 2 つの水平線を使用する）現在のブロックを 3 つのブロックに分割することを水平方向三分木分割と呼ぶことができる。三分木分割により、現在のブロックを 3 つの非正方形のパーティションに分割できる。この場合、現在のブロックの中央に位置するパーティションの幅 / 高さは、他のパーティションの幅 / 高さの 2 倍であり得る。図 4 (d) の「SPLIT_TT_VER」は、垂直方向三分木分割結果を示し、図 4 (e) の「SPLIT_TT_HOR」は、水平方向三分木分割結果を示す。

40

【0088】

符号化ツリーユニットの分割回数を分割深さ（Partitioning Depth）として定義できる。シーケンスまたは画像レベルにより符号化ツリーユニットの最大分割深さを決定できる。これにより、符号化ツリーユニットの最大分割深さは、異なるシーケンスまたは画像によって異なることができる。

【0089】

50

あるいは、複数の分割手法のそれぞれの最大分割深さを個別に決定することができる。例えば、四分木分割が許可される最大分割深さは、二分木分割および／または三分木分割が許可される最大分割深さとは異なることができる。

【0090】

符号化器は、現在のブロックの分割形状または分割深さのうちの少なくとも1つを示す情報を、ビットストリームを介してシグナリングすることができる。復号化器は、ビットストリームから解析された情報に基づいて、符号化ツリーユニットの分割形状および分割深さを決定することができる。

【0091】

図5は、符号化ツリーユニットの分割例を示す図である。

10

【0092】

四分木分割、二分木分割、および／または三分木分割などの分割手法を使用して、符号化ブロックを分割することを多分木分割(Multi Tree Partitioning)と呼ぶことができる。

【0093】

多分木分割を符号化ブロックに適用することによって生成される複数の符号化ブロックを下位符号化ブロックと呼ぶことができる。符号化ブロックの分割深さがkである場合、複数の下位符号化ブロックの分割深さをk+1に設定する。

【0094】

一方、分割深さがk+1である複数の符号化ブロックについて、分割深さがkである符号化ブロックを上位符号化ブロックと呼ぶことができる。

20

【0095】

上位符号化ブロックの分割形状または隣接符号化ブロックの分割タイプのうちの少なくとも1つに基づいて、現在の符号化ブロックの分割タイプを決定できる。ここで、隣接符号化ブロックは、現在の符号化ブロックに隣接し、当該隣接符号化ブロックは、現在の符号化ブロックの上部隣接ブロック、左側隣接ブロックまたは左上隅に隣接する隣接ブロックのうちの少なくとも1つを含み得る。ここで、分割タイプは、四分木分割するか否か、二分木分割するか否か、二分木の分割方向、三分木分割するか否か、または三分木の分割方向のうちの少なくとも1つを含み得る。

【0096】

符号化ブロックの分割形状を決定するために、符号化ブロックが分割されるか否かを示す情報が、ビットストリームを介してシグナリングできる。前記情報が1ビットのフラグ「split_cu_flag」であり、現在の記フラグがtrueである場合、多分木分割手法によって符号化ブロックを分割することができる。

30

【0097】

「split_cu_flag」がtrueである場合、符号化ブロックが四分木分割されるか否かを示す情報が、ビットストリームを介してシグナリングできる。前記情報が1ビットのフラグ「split_qt_flag」であり、前記フラグがtrueである場合、符号化ブロックは、4つのブロックに分割されることがある。

【0098】

例えば、図5に示される例では、符号化ツリーユニットが四分木分割により分割深さが1である4つの符号化ブロックが生成されることが示される。また、四分木分割の結果として生成された4つの符号化ブロックのうちの最初の符号化ブロックおよび4番目の符号化ブロックに再び四分木分割が適用されることが示される。その結果、分割深さが2である4つの符号化ブロックを生成できる。

40

【0099】

また、分割深さが2である符号化ブロックに再び四分木分割を適用することによって、分割深さが3である符号化ブロックを生成できる。

【0100】

符号化ブロックに四分木分割が適用されない場合、符号化ブロックのサイズ、符号化ブ

50

ロックが画像境界に位置するか否か、最大分割深さまたは隣接ブロックの分割形状のうちの少なくとも1つを考慮して、前記符号化ブロックに対して二分木分割を実行するか三分木分割を実行するかを決定できる。前記符号化ブロックに対して二分木分割または三分木分割を実行すると決定した場合、分割方向を示す情報が、ビットストリームを介してシグナリングできる。前記情報は1ビットのフラグ「`m t t_s p l i t_c u_v e r t i c a l_f l a g`」であり得る。前記フラグに基づいて、分割方向が垂直方向であるか水平方向であるかを決定できる。さらに二分木分割または三分木分割のどちらが前記符号化ブロックに適用されるかを示す情報が、ビットストリームを介してシグナリングできる。前記情報は1ビットのフラグ「`m t t_s p l i t_c u_b i n a r y_f l a g`」であり得る。前記フラグに基づいて、前記符号化ブロックに二分木分割を適用するか三分木分割を適用するかを決定できる。

10

【0101】

例えば、図5に示される例では、分割深さが1である符号化ブロックに垂直方向二分木分割が適用され、前記分割結果として生成された符号化ブロックのうち、左側の符号化ブロックに垂直方向三分木分割が適用され、右側の符号化ブロックに垂直方向二分木分割が適用されることが示される。

【0102】

画像を符号化ツリーユニットに分割する場合、画像の右境界または下境界に隣接する領域に、事前定義されたサイズより小さいサイズのブロックが存在することができる。前記ブロックが符号化ツリーユニットである場合、画像の右境界または下境界に、事前定義されたサイズより小さいサイズの符号化ツリーユニットが生成されることがある。この場合、シーケンスパラメータセットまたは画像パラメータセットを介してシグナリングされる情報に基づいて、符号化ツリーユニットのサイズを決定できる。

20

【0103】

図6は、プリセットされたサイズの符号化ツリーユニットより小さいサイズのブロックが画像境界に現れる例を示す図である。

【0104】

図6に示される例のように、 1292×1080 サイズの画像を 128×128 サイズの符号化ツリーユニットに分割する場合、画像の右境界および下境界に 128×128 未満のサイズのブロックが存在することになる。後述される実施例では、画像境界で生成される事前定義されたサイズ未満の符号化ツリーユニットのブロックは非定型境界の境界ブロックと呼ばれる。

30

【0105】

非定型境界の境界ブロックについては、事前定義された分割方法のみを許可することができる。ここで、事前定義された分割方法は、四分木分割、三分木分割、または二分木分割のうちの少なくとも1つを含み得る。

【0106】

例えば、非定型境界の境界ブロックについては、四分木分割のみを許可することができる。この場合、画像境界のブロックが最小四分木分割サイズになるまで、四分木分割を繰り返すことができる。ここで、符号化器および復号化器で最小四分木分割サイズを事前定義することができる。あるいは、最小四分木分割サイズを示す情報が、ビットストリームを介してシグナリングできる。

40

【0107】

図7は、非定型境界の境界ブロックに対して四分木分割を実行する例を示す図である。説明の便宜のために、最小四分木分割のサイズが 4×4 であると仮定する。

【0108】

非定型境界の境界ブロックに対する分割は、正方形のブロックに基づいて実行できる。前記正方形のブロックは、非定型境界の境界ブロックの幅または高さの大きい方にに基づいて導出できる。例えば、前記参照値より大きく、前記参照値に最も近い2のべき級数は、正方形のブロックの1辺の長さと見なすことができる。例えば、図7に示される 12×2

50

0 ブロックは、 32×32 ブロックに属するものと見なされ、 32×32 の分割結果を 12×20 ブロックに適用することができる。

【0109】

12×20 のブロックに対して四分木分割を実行する場合、前記ブロックは、 12×16 サイズのブロックおよび 12×4 サイズのブロックに分割されることがある。それらのそれぞれに対して再び四分木分割を実行する場合、 12×16 ブロックは 2 つの 8×8 ブロックと、2 つの 4×8 ブロックに分割され、 12×4 サイズのブロックは、 8×4 サイズのブロックと 4×4 サイズのブロックに分割される。

【0110】

画像境界に位置する 4×8 ブロックに対して再び四分木分割を実行し、その結果、2 つの 4×4 サイズのブロックを生成できる。同様に、画像境界に位置する 8×4 ブロックに対して再び四分木分割を実行し、その結果、2 つの 4×4 サイズのブロックを生成できる。10

【0111】

あるいは、ブロックの幅または高さのうちの少なくとも 1 つが、最小四分木分割サイズと同じかそれよりも小さい場合、二分木分割を実行することもできる。ここで、最小四分木分割サイズは、最小四分木分割幅または最小四分木分割高さを示すことができる。例えば、最小四分木分割サイズが 4×4 である場合、最小四分木分割幅と最小四分木分割高さは 4 であり得る。

【0112】

この場合、ブロックの幅が最小四分木分割サイズと同じかそれよりも小さい場合、垂直方向二分木分割を実行でき、ブロックの高さが最小四分木分割高さと同じかそれよりも小さい場合、水平方向二分木分割を実行できる。20

【0113】

一方、ブロックの幅または高さが最小四分木分割サイズよりも大きい場合、四分木分割を実行できる。例えば、ブロックの右上の位置と左下の位置が画像を脱し、ブロックの幅または高さが最小四分木分割サイズよりも大きい場合、対応するブロックに四分木分割を適用できる。

【0114】

図 8 は、画像の右境界と下境界に同時に隣接するブロックに対して四分木分割を実行する例を示す図である。説明の便宜のために、最小四分木分割のサイズが 4×4 であると仮定する。30

【0115】

図 8 に示される例では、 12×20 サイズのブロックを含む 32×32 サイズのブロックに対して四分木分割を実行する場合、4 つの 16×16 サイズのブロックが生成される。これらのテクスチャデータを含む 2 つの 16×16 サイズのブロックに対して、再び四分木分割を実行できる。その結果、x 軸座標および y 軸座標が、画像境界を抜け出し、 4×4 サイズのテクスチャデータを含む 8×8 サイズのブロックを生成することができる。前記 8×8 サイズのブロックの幅および高さが最小四分木分割サイズより大きいため、前記ブロックに対して四分木分割を実行することができる。

【0116】

12×20 のブロックに対して四分木分割を実行する場合、前記ブロックは、 12×16 サイズのブロックおよび 12×4 サイズのブロックに分割されることがある。それらのそれぞれに対して再び四分木分割を実行する場合、 12×16 ブロックは 2 つの 8×8 ブロックと、2 つの 4×8 ブロックに分割され、 12×4 サイズのブロックは、 8×4 サイズのブロックと 4×4 サイズのブロックに分割される。40

【0117】

画像境界に位置する 4×8 ブロックの幅は、最小四分木分割サイズと同じであるため、 4×8 ブロックに対して二分木分割を実行できる。具体的には、 4×8 ブロックを含む正方形ブロック（すなわち、 8×8 ）に基づいて、垂直方向二分木分割を実行できる。

【0118】

また、画像境界に位置する 8×4 ブロックの幅は、最小四分木分割サイズと同じであるため、 8×4 ブロックに対して二分木分割を実行できる。具体的には、 4×8 ブロックを含む正方形ブロック（すなわち、 8×8 ）に基づいて、水平方向二分木分割を実行できる。

【0119】

前記分割結果は、 8×4 サイズのブロック、 4×4 サイズのブロック、および 4×8 サイズのブロックが画像境界に隣接配置することができる。

【0120】

あるいは、ブロックの幅または高さのうちの少なくとも 1 つが閾値より小さいか等しい場合二分木分割を実行し、そうでない場合、四分木分割を実行できる。ここで、閾値は、最小四分木分割サイズに基づいて導出できる。例えば、最小四分木分割サイズが $\min Q_{\text{size}}$ である場合、閾値を「 $\min Q_{\text{size}} << 1$ 」に設定できる。あるいは、閾値を決定するための情報が、ビットストリームを介してシグナリングできる。

10

【0121】

図 9 は、画像境界に隣接するブロックの分割パターンを示す図である。説明の便宜のために、最小四分木分割のサイズが 4×4 であると仮定する。閾値は 8 に設定できる。

【0122】

先ず、 12×20 ブロックに対して四分木分割を実行する。その結果、前記ブロックは 12×16 サイズのブロックおよび 12×4 サイズのブロックに分割できる。 12×16 サイズのブロックの幅および高さの両方が閾値を超えるため、前記ブロックに対して四分木分割を適用することができる。これにより、前記ブロックは 2 つの 8×8 ブロックと 2 つの 4×8 ブロックに分割できる。

20

【0123】

12×4 サイズのブロックの幅は閾値を超える。これにより、前記 12×4 サイズのブロックに対して四分木分割を適用することができる。その結果、前記ブロックは 8×4 サイズのブロックと 4×4 サイズのブロックに分割できる。

【0124】

その後、画像境界に位置する 4×8 サイズのブロックと 8×4 サイズのブロックの幅と高さの両方が、閾値より小さいか等しいため、これらのブロックに対して二分木分割を適用することができる。

【0125】

30

上述した例とは逆に、ブロックの幅または高さのうちの少なくとも 1 つが閾値を超える場合、二分木分割を実行でき、そうでない場合、四分木分割を実行できる。

【0126】

あるいは、非定型境界の境界ブロックに対して、四分木分割のみを適用するか、または二分木分割のみを適用することができる。例えば、画像境界に位置するブロックが最小サイズを持つまで四分木分割を繰り返して実行するか、または、画像境界に位置するブロックが最小サイズを持つまで二分木分割を繰り返して実行することができる。

【0127】

非定型境界の境界ブロックを符号化ユニットとして指定し、非定型境界の境界ブロックについては、スキップモードを固定的に適用するか、または変換係数すべて 0 に設定することができる。これにより、非定型境界の境界ブロックに 0 以外の変換係数が存在する否かを示す符号化ブロックフラグ（CBF : C o d e d B l o c k F l a g）の値が 0 に設定できる。スキップモードで符号化された符号化ユニットまたは変換係数が 0 に設定された符号化ユニットは、境界ゼロ符号化ユニットと呼ぶことができる。

40

【0128】

あるいは、非定型境界の境界ブロックを分割して生成された符号化ブロックの幅または高さのうちの少なくとも 1 つを閾値と比較して、前記符号化ユニットを境界ゼロ符号化ユニットに設定するか否かを決定することができる。例えば、符号化ブロックの幅または高さのうちの少なくとも 1 つが閾値未満の符号化ブロックをスキップモードで符号化するか、または前記符号化ブロックの変換係数を 0 に設定することができる。

50

【 0 1 2 9 】

図 10 は、画像境界に隣接するプロックの符号化パターンを示す図である。閾値が 8 であると仮定する。

【 0 1 3 0 】

非定型境界の境界プロックを分割して生成された符号化プロックのうち、幅または高さのうちの少なくとも 1 つが閾値未満である符号化プロックを境界ゼロ符号化ユニットとして設定できる。

【 0 1 3 1 】

例えば、図 10 に示される例では、 4×16 サイズの符号化プロック、 8×4 サイズの符号化プロック、および 4×4 サイズの符号化プロックを境界ゼロ符号化ユニットとして設定できる。これにより、スキップモードで前記プロックを符号化するか、または前記プロックの変換係数を 0 に設定できる。

10

【 0 1 3 2 】

幅と高さが閾値より大きいか等しい符号化プロックについては、スキップモードを適用するか変換係数を 0 に設定するかを選択的に決定することができる。このため、前記符号化プロックにスキップモードを適用するか否かを示すフラグ、または変換係数を 0 に設定するか否かを示すフラグを符号化してシグナリングすることができる。

【 0 1 3 3 】

あるいは、二分木分割で生成された符号化ユニットのみが境界ゼロ符号化ユニットとして設定できることを許可することができる。あるいは、四分木分割で生成された符号化ユニットのみが境界ゼロ符号化ユニットとして設定できることを許可することができる。

20

【 0 1 3 4 】

インター予測は、前の画像の情報を使用して、現在のプロックを予測する予測符号化モードである。例えば、前の画像内の現在のプロックと同じ位置のプロック（以下、同等位置プロック（C o l l o c a t e d b l o c k ）と呼ぶ）を現在のプロックの予測プロックとして設定できる。以下、現在のプロックと同じ位置のプロックに基づいて生成された予測プロックを同等位置予測プロック（C o l l o c a t e d P r e d i c t i o n B l o c k ）と呼ぶ。

【 0 1 3 5 】

一方、前の画像に存在したオブジェクトが現在の画像の別の位置に移動した場合、オブジェクトの動きを使用して、現在のプロックを効果的に予測できる。例えば、前の画像と現在の画像を比較することにより、オブジェクトの移動方向とサイズを知ることができる場合、オブジェクトの動き情報を考慮して、現在のプロックの予測プロック（または予測画像）を生成することができる。以下、動き情報を使用して生成された予測プロックを動き予測プロックと呼ぶことができる。

30

【 0 1 3 6 】

現在のプロックから予測プロックを差し引くことによって、残差プロック（r e s i d u a l b l o c k ）を生成することができる。この場合、オブジェクトの動きが存在する場合、同等位置予測プロックの代わりに動き予測プロックを使用することにより、残差プロックのエネルギーを削減し、これにより、残差プロックの圧縮性能を向上させることができる。

40

【 0 1 3 7 】

前述したように、動き情報を使用して予測プロックを生成することを、動き補償予測を呼ぶことができる。ほとんどのインター予測では、動き補償予測に基づいて予測プロックを生成することができる。

【 0 1 3 8 】

動き情報は、動きベクトル、参照画像インデックス、予測方向または双方向重み値インデックスのうちの少なくとも 1 つを含み得る。動きベクトルは、オブジェクトの移動方向とサイズを表す。参照画像インデックスは、参照画像リストに含まれる複数の参照画像のうちの現在のプロックの参照画像を指定する。予測方向は、一方向 L 0 予測、一方向 L 1

50

予測、または双方向予測（L0予測およびL1予測）のいずれかを指す。現在のブロックの予測方向に従って、L0方向の動き情報またはL1方向の動き情報のうちの少なくとも1つを使用することができる。双方向重み値インデックスは、L0予測ブロックに適用される重み値およびL1予測ブロックに適用される重み値を指定する。

【0139】

図11は、本発明の実施例によるインター予測方法のフローチャートである。

【0140】

図11を参照すると、インター予測方法は、現在のブロックのインター予測モードを決定することと（S1101）、決定されたインター予測モードに従って、現在のブロックの動き情報取得することと（S1102）、取得された動き情報に基づいて、現在のブロックの動き補償予測を実行することと（S1103）を含む。10

【0141】

ここで、インター予測モードは、現在のブロックの動き情報を決定するための様々な手法を表し、並進（Translation）動き情報を使用するインター予測モードと、アフィン（Affine）動き情報を使用するインター予測モードを含み得る。例えば、並進動き情報を使用するインター予測モードは、マージモードおよび高レベル動きベクトル予測モードを含み得、アフィン動き情報を使用するインター予測モードは、アフィンマージモードおよびアフィン動きベクトル予測モードを含み得る。インター予測モードに従って、現在のブロックに隣接する隣接ブロックまたはビットストリームから解析された情報に基づいて、現在のブロックの動き情報を決定できる。20

【0142】

以下、並進動き情報を用いたインター予測方法について詳細に説明する。

【0143】

現在のブロックの他のブロックの動き情報から、現在のブロックの動き情報を導出することができる。ここで、他のブロックは、現在のブロックよりも先にインター予測で符号化／復号化されたブロックであり得る。現在のブロックの動き情報を他のブロックの動き情報と同じに設定したことをマージモードと定義することができる。また、他のブロックの動きベクトルを現在のブロックの動きベクトルの予測値に設定したことを動きベクトル予測モードと定義することができる。30

【0144】

図12は、マージモードで現在のブロックの動き情報を導出するプロセスのフローチャートである。

【0145】

現在のブロックのマージ候補を導出することができる（S1201）。現在のブロックのマージ候補は、現在のブロックよりも先にインナー予測で符号化／復号化されたブロックから導出されることがある。

【0146】

図13は、マージ候補を導出するための候補ブロックを示す図である。

【0147】

候補ブロックは、現在のブロックに隣接するサンプルを含む隣接ブロック、または現在のブロックに隣接しないサンプルを含む非隣接ブロックのうちの少なくとも1つを含み得る。以下、候補ブロックを決定するために使用されるサンプルを参照サンプルとして指定する。また、現在のブロックに隣接する参照サンプルを隣接参照サンプルと呼び、現在のブロックに隣接しない参照サンプルを非隣接参照サンプルと呼ぶ。40

【0148】

隣接参照サンプルは、現在のブロックの最左側列の隣接列に含まれるか、または現在のブロックの最上行の隣接行に含まれることができる。例えば、現在のブロックの左上のサンプルの座標が（0、0）である場合、（-1, H-1）の位置の参照サンプルをブロック、（W-1, -1）位置の参照サンプルを含むブロック、（W, -1）の位置の参照サンプルを含むブロック、（-1, H）の位置の参照サンプルを含むブロック、または（-

1、 - 1) の位置の参照サンプルを含むブロックのうちの少なくとも 1 つを候補ブロックとして使用することができる。図面を参照すると、インデックス 0 ~ 4 の隣接ブロックを候補ブロックとして使用できる。

【 0 1 4 9 】

非隣接参照サンプルは、現在のブロックに隣接する参照サンプルとの x 軸距離または y 軸距離のうちの少なくとも 1 が事前定義された値を持つサンプルを示す。例えば、左側参照サンプルとの x 軸距離が事前定義された値である参照サンプルを含むブロック、上部参照サンプルとの y 軸距離が事前定義された値である非隣接サンプルのブロック、または左上参照サンプルとの x 軸距離と y 軸距離が事前定義された値である非隣接サンプルであるブロックのうちの少なくとも 1 を候補ブロックとして使用できる。事前定義された値は 4 10 、 8 、 12 、 16 などの整数であり得る。図面を参照すると、インデックス 5 ~ 26 のブロックのうちの少なくとも 1 つを候補ブロックとして使用できる。

【 0 1 5 0 】

隣接参照サンプルと同一垂直線、水平線、または対角線上に位置しないサンプルを非隣接参照サンプルとして設定できる。

【 0 1 5 1 】

図 14 は、参照サンプルの位置を示す図である。

【 0 1 5 2 】

図 14 に示される例のように、上部非隣接参照サンプルの x 座標は、上部隣接参照サンプルの x 座標とは異なるように設定できる。例えば、上部隣接参照サンプルの位置が (W 20 - 1 、 - 1) である場合、 y 軸に沿って上部隣接参照サンプルから N 離れた上部非隣接参照サンプルの位置を ((W / 2) - 1 、 - 1 - N) に設定でき、 y 軸に沿って上部隣接参照サンプルから 2 N 離れた上部非隣接参照サンプルの位置を (0 、 - 1 - 2 N) に設定できる。つまり、非隣接参照サンプルの位置は、隣接参照サンプルの位置と隣接参照サンプルとの距離に基づいて決定ができる。

【 0 1 5 3 】

以下、候補ブロックのうちの隣接参照サンプルを含む候補ブロックを隣接ブロックと呼び、非隣接参照サンプルを含むブロックを非隣接ブロックと呼ぶ。

【 0 1 5 4 】

現在のブロックと候補ブロックとの間の距離が閾値より大きいか等しい場合、前記候補ブロックをマージ候補として使用できないものに設定できる。前記閾値は、符号化ツリーユニットのサイズに基づいて決定できる。例えば、前記閾値を符号化ツリーユニットの高さ (c t u _ h e i g h t) に設定でき、または符号化ツリーユニットの高さからオフセット値を加算または減算することによって得られた値 (例えば、 c t u _ h e i g h t ± N) に設定できる。オフセット値 N は符号化器と復号化器で事前定義された値であり、 4 30 、 8 、 16 、 32 または c t u _ h e i g h t に設定できる。

【 0 1 5 5 】

現在のブロックの y 軸座標と候補ブロックに含まれるサンプルの y 軸座標との間の差が閾値を超える場合、候補ブロックをマージ候補として使用できないと決定できる。

【 0 1 5 6 】

あるいは、現在のブロックと同じ符号化ツリーユニットに属しない候補ブロックを、マージ候補として使用できないものに設定できる。例えば、参照サンプルが現在のブロックに属する符号化ツリーユニットの上部境界を超えない場合、参照サンプルを含む候補ブロックをマージ候補として使用できないものに設定できる。

【 0 1 5 7 】

現在のブロックの上部境界が符号化ツリーユニットの上部境界と隣接する場合、複数の候補ブロックがマージ候補として使用できないものに設定され、これにより、現在のブロックの符号化 / 復号化効率が低下する。上記の問題を解決するために、候補ブロックを設定することにより、現在のブロックの上方に位置する候補ブロックの数が現在のブロックの左側の候補ブロックの数よりも多くなるように設定できる。

【 0 1 5 8 】

図 15 は、マージ候補を導出するための候補ブロックを示す図である。

【 0 1 5 9 】

図 15 に示される例のように、現在のブロックの上部 N 個のブロック列に属する上部ブロックと、現在のブロックの左側 M 個のブロック列に属する左側ブロックを候補ブロックに設定できる。この場合、M を N より大きく設定することにより、左側候補ブロックの数を上部候補ブロックの数より大きく設定することができる。

【 0 1 6 0 】

例えば、現在のブロック内の参照サンプルの y 軸座標と、候補ブロックとして使用できる上部ブロックの y 軸座標との間の差を、現在のブロックの高さの N 倍を超えないように設定できる。また、現在のブロック内の参照サンプルの x 軸座標と、候補ブロックとして使用できる左側ブロックの x 軸座標との間の差を、現在のブロックの幅の M 倍を超えないように設定できる。

10

【 0 1 6 1 】

例えば、図 15 に示される例では、現在のブロックの上部の 2 つのブロック列に属するブロックと、現在のブロックの左側の 5 つのブロック列に属するブロックが候補ブロックに設定されることを示す。

【 0 1 6 2 】

別の例として、候補ブロックが現在のブロックと同じ符号化ツリーユニットに属しない場合、前記候補ブロックの代わりに、現在のブロックと同じ符号化ツリーユニットに属するブロック、または前記符号化ツリーユニットの境界に隣接する参照サンプルを含むブロックを使用してマージ候補を導出することができる。

20

【 0 1 6 3 】

図 16 は、参照サンプルの位置を変更する例を示す図である。

【 0 1 6 4 】

参照サンプルが現在のブロックとは異なる符号化ツリーユニットに含まれ、前記参照サンプルが前記符号化ツリーユニット境界に隣接しない場合、前記参照サンプルの代わりに、前記符号化ツリーユニットの境界に隣接する参照サンプルを使用して候補ブロックを決定することができる。

30

【 0 1 6 5 】

例えば、図 16 (a) と図 16 (b) に示される例では、現在のブロックの上部境界と符号化ツリーユニットの上部境界が接する場合、現在のブロックの上部の参照サンプルは、現在のブロックとは異なる符号化ツリーユニットに属することになる。符号化ツリーユニットの上部境界に隣接するサンプルを使用して、現在のブロックとは異なる符号化ツリーユニットに属する参照サンプルのうち、符号化ツリーユニットの上部境界に隣接しない参照サンプルを置き換えることができる。

【 0 1 6 6 】

例えば、図 16 (a) に示される例のように、位置 6 の参照サンプルが、符号化ツリーユニットの上部境界の位置 6' のサンプルに置き換えられ、図 16 (b) に示される例のように、位置 15 の参照サンプルが、符号化ツリーユニットの上部境界の位置 15' のサンプルに置き換えられる。この場合、置換サンプルの y 座標は、符号化ツリーユニットの隣接位置に変更でき、置換サンプルの x 座標を参照サンプルと同じに設定できる。例えば、位置 6' のサンプルは、位置 6 のサンプルと同じ x 座標を有し得、位置 15' のサンプルは、位置 15 のサンプルと同じ x 座標を有し得る。

40

【 0 1 6 7 】

あるいは、参照サンプルの x 座標からオフセット値を加算または減算して得られた値を、置換サンプルの x 座標として設定できる。例えば、現在のブロックの上部の隣接参照サンプルと、非隣接参照サンプルの x 座標が同じである場合、参照サンプルの x 座標からオフセット値を加算または減算して得られた値を、置換サンプルの x 座標として設定できる。これは、非隣接参照サンプルを置換する置換サンプルが他の非隣接参照サンプルまたは

50

隣接参照サンプルと同じ位置になることを防止するためである。

【 0 1 6 8 】

図 1 7 は、参照サンプルの位置を変更する例を示す図である。

【 0 1 6 9 】

符号化ツリーユニットの境界に位置するサンプルを使用して、現在のブロックとは異なる符号化ツリーユニットに含まれかつ符号化ツリーユニットの境界に隣接しない参照サンプルを置き換える場合、参照サンプルの x 座標からオフセット値を加算または減算して得られた値を、置換サンプルの x 座標として設定できる。

【 0 1 7 0 】

例えば、図 1 7 に示される例では、位置 6 の参照サンプルと位置 1 5 の参照サンプルは、それぞれ、符号化ツリーユニットの上部境界に隣接する行と同じ y 座標を有する、位置 6' のサンプルおよび位置 1 5' のサンプルに置き換えることができる。この場合、位置 6' のサンプルの x 座標は、位置 6 の参照サンプルの x 座標から $W / 2$ を差し引いた値に設定でき、位置 1 5' のサンプルの x 座標は、位置 1 5 の参照サンプルの x 座標から $W - 1$ を差し引いた値に設定できる。

【 0 1 7 1 】

図 1 6 および図 1 7 に示される例とは異なり、現在のブロックの最上部の行の上に位置する行の y 座標、または符号化ツリーユニットの上部境界の y 座標を、置換サンプルの y 座標に設定することもできる。

【 0 1 7 2 】

図示されていないが、符号化ツリーユニットの左境界に基づいて、参照サンプルを置換するサンプルを決定することができる。例えば、参照サンプルが現在のブロックと同じ符号化ツリーユニットに含まれず、符号化ツリーユニットの左境界に隣接しない場合、前記参照サンプルを符号化ツリーユニットの左境界に隣接するサンプルに置き換えることができる。この場合、置換サンプルは、参照サンプルと同じ y 座標を有し得るか、または参照サンプルの y 座標からオフセット値を加算または減算して得られた y 座標を有し得る。

【 0 1 7 3 】

その後、置換サンプルを含むブロックを候補ブロックに設定し、前記候補ブロックに基づいて現在のブロックのマージ候補を導出することができる。

【 0 1 7 4 】

現在のブロックとは異なる画像に含まれる時間的隣接するブロックからマージ候補を導出することもできる。例えば、同等位置画像に含まれる同等位置ブロックからマージ候補を導出することができる。

【 0 1 7 5 】

マージ候補の動き情報を候補ブロックの動き情報と同じように設定できる。例えば、候補ブロックの動きベクトル、参照画像インデックス、予測方向または双方向重み値インデックスのうちの少なくとも 1 つをマージ候補の動き情報に設定できる。

【 0 1 7 6 】

マージ候補を含むマージ候補リストを生成できる (S 1 2 0 2)。前記マージ候補は、現在のブロックに隣接する隣接ブロックから導出された隣接マージ候補と、非隣接ブロックから導出された非隣接マージ候補に分類することができる。

【 0 1 7 7 】

所定の順序応じて、マージ候補リスト内の複数のマージ候補のインデックスを割り当てることができる。例えば、隣接マージ候補に割り当たられるインデックスは、非隣接マージ候補に割り当たられるインデックスより小さい値を持つことができる。あるいは、図 1 3 または図 1 5 に示される各ブロックのインデックスに基づいて、各マージ候補にインデックスを割り当てることができる。

【 0 1 7 8 】

マージ候補リストに複数のマージ候補が含まれる場合、前記複数のマージ候補のうちの少なくとも 1 つを選択することができる (S 1 2 0 3)。この場合、現在のブロックの動

10

20

30

40

50

き情報が隣接マージ候補から導出されるか否かを示す情報が、ビットストリームを介してシグナリングできる。前記情報は1ビットのフラグであり得る。例えば、現在のブロックの動き情報が隣接マージ候補から導出されるか否かを示す構文要素 `isAdjacentMergeFlag` が、ビットストリームを介してシグナリングできる。構文要素 `isAdjacentMergeFlag` の値が1である場合、現在のブロックの動き情報は、隣接マージ候補に基づいて導出できる。一方、構文要素 `isAdjacentMergeFlag` の値が0である場合、現在のブロックの動き情報は、非隣接マージ候補に基づいて導出できる。

【0179】

表1は、構文要素 `isAdjacentMergeFlag` を含む構文テーブルを示す。

10

20

30

40

50

【表 1】

coding_unit (x0, y0, cbWidth, cbHeight, treeType) {	記述子
if (slice_type != I) {	
pred_mode_flag	ae(v)
}	
if (CuPredMode[x0][y0] == MODE_INTRA) {	
if (treeType == SINGLE_TREE treeType ==	
DUAL_TREE_LUMA) {	
intra_luma_mpm_flag [x0][y0]	
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)
}	
if (cu_skip_falg[x0][y0]) {	
if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx [x0][y0]	ae(v)
}	
NA_merge_idx [x0][y0]	ae(v)
}	
}	

10

20

30

40

50

<code>}</code>		
<code>else /* MODE_INTER */</code>		
<code>merge_flag[x0][y0]</code>	<code>ae(v)</code>	
<code>if (merge_flag[x0][y0]) {</code>		
<code> if (MaxNumMergeCand > 1) {</code>		
<code> isAdjacentMergeflag</code>	<code>ae(v)</code>	10
<code> if (isAdjcanetMergeflag) {</code>		
<code> merge_idx[x0][y0]</code>	<code>ae(v)</code>	
<code> } else {</code>		
<code> NA_merge_idx[x0][y0]</code>	<code>ae(v)</code>	
<code> }</code>		
<code> }</code>		20
<code>}</code>		
<code>if (CuPredMode[x0][y0] != MODE_INTRA)</code>		
<code> cu_cbf</code>	<code>ae(v)</code>	
<code> if (cu_cbf) {</code>		
<code> transform_tree (x0, y0, cbWidth, cbHeight, treeType)</code>		
<code>}</code>		30

【0180】

複数のマージ候補のいずれかを指定するための情報が、ビットストリームを介してシグナリングできる。例えば、マージ候補リストに含まれるマージ候補のいずれかのインデックスを示す情報が、ビットストリームを介してシグナリングできる。

【0181】

`isAdjacentMergeflag`が1である場合、隣接マージ候補のいずれかを決定するための構文要素`merge_idx`がシグナリングできる。構文要素`merge_idx`の最大値を、隣接マージ候補の数から1を差し引いた値に設定できる。

【0182】

`isAdjacentMergeflag`が0である場合、非隣接マージ候補のいずれかを決定するための構文要素`NA_merge_idx`がシグナリングできる。構文要素`NA_merge_idx`は、非隣接マージ候補のインデックスから隣接マージ候補の数を減算して得られた値を示す。復号化器は、`NA_merge_idx`によって決定されたインデックスに隣接マージ候補の数を追加することにより、非隣接マージ候補を選択することができる。

【0183】

マージ候補リストに含まれるマージ候補の数が最大値未満である場合、インター領域動き情報テーブルに含まれるマージ候補をマージ候補リストに追加することができる。イン

ター領域動き情報テーブルは、現在のブロックの前に符号化／復号化されたブロックに基づいて導出されたマージ候補を含み得る。

【0184】

インター領域動き情報テーブルは、現在の画像内のインター予測に基づいて符号化／復号化されたブロックから導出されたマージ候補を含む。例えば、インター領域動き情報テーブルに含まれるマージ候補の動き情報は、インター予測に基づいて符号化／復号化されたブロックの動き情報と同じであるように設定できる。ここで、動き情報は、動きベクトル、参照画像インデックス、予測方向または双方向重み値インデックスのうちの少なくとも1つを含み得る。

【0185】

説明の便宜のために、インター領域動き情報テーブルに含まれるマージ候補をインター領域マージ候補と呼ぶ。

【0186】

符号化器および復号化器で、インター領域動き情報テーブルに含めることができるマージ候補の最大数を事前定義することができる。例えば、インター領域動き情報テーブルに含めることができるマージ候補の最大数は、1、2、3、4、5、6、7、8またはそれ以上（例えば、16など）であり得る。

【0187】

あるいは、インター領域動き情報テーブルのマージ候補の最大数を示す情報が、ビットストリームを介してシグナリングできる。前記情報は、シーケンスレベル、画像レベルまたはスライスレベルでシグナリングができる。

10

【0188】

あるいは、画像のサイズ、スライスのサイズまたは符号化ツリーユニットのサイズに従って、インター領域動き情報テーブルの最大マージ候補数を決定できる。

【0189】

インター領域動き情報テーブルは、画像、スライス、タイル、ブリック、符号化ツリーユニットまたは符号化ツリーユニットライン（行または列）の単位で初期化できる。例えば、スライスが初期化される場合、インター領域動き情報テーブルも初期化され、インター領域動き情報テーブルは、何のマージ候補も含まない可能性がある。

【0190】

あるいは、インター領域動き情報テーブルを初期化するか否かを示す情報が、ビットストリームを介してシグナリングできる。前記情報は、スライスレベル、タイルレベル、ブリックレベルまたはブロックレベルでシグナリングができる。前記情報がインター領域動き情報テーブルを初期化することを指示するまで、構成されたインター領域動き情報テーブルを使用することができる。

20

【0191】

あるいは、画像パラメータセットまたはスライスヘッダを介して与インター領域マージ候補に関する情報がシグナリングされることができる。スライスが初期化されても、インター領域動き情報テーブルは、初期インター領域マージ候補を含み得る。これにより、スライス内の最初の符号化／復号化されるブロックにも、インター領域マージ候補を使用することができる。

30

【0192】

符号化／復号化順序に従って、ブロックを符号化／復号化し、符号化／復号化順序に従って、インター予測に基づいて符号化／復号化された複数のブロックをインター領域マージ候補として順次設定することができる。

40

【0193】

図18は、インター領域動き情報テーブルの更新例を説明するための図である。

【0194】

現在のブロックに対してインター予測を実行する場合（S1801）、現在のブロックに基づいてインター領域マージ候補を導出することができる（S1802）。インター領

50

域マージ候補の動き情報は、現在のブロックの動き情報と同じように設定することができる。

【0195】

インター領域動き情報テーブルが空の状態である場合（S1803）、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加することができる（S1804）。

【0196】

インター領域動き情報テーブルが既に前記インター領域マージ候補を含んでいる場合（S1803）、現在のブロックの動き情報（または現在のブロックに基づいて導出されたインター領域マージ候補）に対して冗長検査を実行することができる（S1805）。冗長検査は、インター領域動き情報テーブルに記憶されたインター領域マージ候補の動き情報と、現在のブロックの動き情報が同じであるか否かを決定するために使用される。インター領域動き情報テーブルに記憶されたすべてのインター領域マージ候補をオブジェクトとして、冗長検査を実行することが。あるいは、インター領域動き情報テーブルに記憶されたインター領域マージ候補のうち、インデックスが閾値以上または閾値以下であるインター領域マージ候補をオブジェクトとして、冗長検査を実行することができる。

10

【0197】

現在のブロックの動き情報と同じ動き情報を持つインター領域マージ候補を含まない場合、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加することができる（S1808）。インター領域マージ候補の動き情報（例えば、動きベクトルおよび／または参照画像インデックスなど）が同じであるか否かに基づいて、インター領域マージ候補が同じであるか否かを決定できる。

20

【0198】

この場合、インター領域動き情報テーブルに既に最大数のインター領域マージ候補が記憶されている場合（S1806）、最も古いインター領域マージ候補を削除し（S1807）、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加することができる（S1808）。

【0199】

各インデックスに従って、複数のインター領域マージ候補を識別することができる。現在のブロックから導出されたインター領域マージ候補をインター領域動き情報テーブルに追加する場合、前記インター領域マージ候補に最も低いインデックス（0など）を割り当て、記憶されたインター領域マージ候補のインデックスを1ずつ増加させることができる。この場合、インター領域動き情報テーブルに既に最大数のインター領域マージ候補が記憶されている場合、インデックスが最大のインター領域マージ候補を除去する。

30

【0200】

あるいは、現在のブロックから導出されたインター領域マージ候補をインター領域動き情報テーブルに追加する場合、前記インター領域マージ候補に最大のインデックスを割り当てることができる。例えば、インター領域動き情報テーブルに記憶されたインター領域マージ候補の数が最大値未満である場合、記憶されたインター領域マージ候補の数と同じ値のインデックスを前記インター領域マージ候補に割り当てることができる。あるいは、インター領域動き情報テーブルに記憶されたインター領域マージ候補の数が最大値に等しい場合、最大値から1を差し引いたインデックスを前記インター領域マージ候補に割り当てることができる。また、インデックスが最も小さいインター領域マージ候補を除去し、残りの記憶された複数のインター領域マージ候補のインデックスを1ずつ減少させる。

40

【0201】

図19は、インター領域マージ候補テーブルの更新例を示す図である。

【0202】

現在のブロックから導出されたインター領域マージ候補をインター領域マージ候補テーブルに追加し、前記インター領域マージ候補に最大のインデックスを割り当てる場合を仮定する。さらに、インター領域マージ候補テーブルには、既に最大数のインター領域マージ候

50

補が記憶されていると仮定する。

【0203】

現在のブロックから導出されたインター領域マージ候補 $Hmvpcand[n+1]$ をインター領域マージ候補テーブル $HmvpcandList$ に追加する場合、記憶されたインター領域マージ候補から、インデックスが最も小さいインター領域マージ候補 $Hmvpcand[0]$ を削除し、複数の剩余インター領域マージ候補のインデックスを 1 ずつ減少させることができる。また、現在のブロックから導出されたインター領域マージ候補 $Hmvpcand[n+1]$ のインデックスを最大値（図 19 に示される例では n である）に設定できる。

【0204】

現在のブロックに基づいて導出されたインター領域マージ候補と同じであるインター領域マージ候補が記憶されている場合（S1805）、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加しないことができる（S1809）。

【0205】

あるいは、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加しつつ、前記インター領域マージ候補と同じ記憶されたインター領域マージ候補を削除することもできる。この場合、記憶されたインター領域マージ候補のインデックスが新たに更新されることと同じ効果が生じる。

【0206】

図 20 は、記憶されたインター領域マージ候補のインデックスが更新される例を示す図である。

【0207】

現在のブロックに基づいて導出されたインター領域マージ候補 $mvcand$ と同じである記憶されたインター領域マージ候補のインデックスが $hIdx$ である場合、前記記憶されたインター領域マージ候補を削除し、インデックスが $hIdx$ を超えるインター領域マージ候補のインデックスを 1 ずつ減少させることができる。例えば、図 20 に示される例では、 $mvcand$ と同じ $Hmvpcand[2]$ がインター領域動き情報テーブル $HvmpcandList$ から削除され、 $Hmvpcand[3]$ から $Hmvpcand[n]$ までのインデックスが 1 ずつ減少することを示す。

【0208】

さらに、現在のブロックに基づいて導出されたインター領域マージ候補 $mvcand$ をインター領域動き情報テーブルの最後に追加することができる。

【0209】

あるいは、現在のブロックに基づいて導出されたインター領域マージ候補と同じである記憶されたインター領域マージ候補のインデックスを更新することができる。例えば、記憶されたインター領域マージ候補のインデックスを最小値または最大値に変更することができる。

【0210】

所定の領域に含まれるブロックの動き情報を、インター領域動き情報テーブルに追加できないものに設定できる。例えば、並列マージ領域に含まれるブロックの動き情報に基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加しないことができる。並列マージ領域に含まれるブロックの符号化 / 復号化順序が指定されていないため、他のブロックに対してインター予測を実行する場合、前記ブロックのいずれかの動き情報を使用することは不適切である。これにより、並列マージ領域に含まれるブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブルに追加しないことができる。

【0211】

サブブロックユニットを用いて動き補償予測を実行する場合、現在のブロックに含まれる複数のサブブロックのうちの代表的なサブブロックの動き情報に基づいてインター領域

10

20

30

40

50

マージ候補を導出することができる。例えば、現在のブロックについて、サブブロックマージ候補が使用される場合、サブブロックのうちの代表的なサブブロックの動き情報に基づいてインター領域マージ候補を導出することができる。

【0212】

サブブロックの動きベクトルは、次の順序で導出されることがある。先ず、現在のブロックのマージ候補リストに含まれるマージ候補のいずれかを選択し、選択されたマージ候補の動きベクトルに基づいて初期シフトベクトル (*shVector*) を導出することができる。さらに、符号化ブロック内の各サブブロックの参照サンプル（例えば、左上のサンプルまたは中間位置のサンプル）の位置 (*xSb*、*ySb*) に初期シフトベクトルを加算して、参照サンプルの位置が (*xColSb*、*yColSb*) であるシフトサブブロックを導出することができる。下記式 1 は、シフトサブブロックを導出するための式を示す。

【数1】

[式1]

$$(xColSb, yColSb) = (xSb + shVector[0] >> 4, ySb + shVector[1] >> 4)$$

10

20

【0213】

次に、(*xColSb*、*yColSb*) を含むサブブロックの中心位置に対応する同等位置ブロックの動きベクトルを、(*xSb*、*ySb*) を含むサブブロックの動きベクトルに設定する。

【0214】

代表的なサブブロックは、現在のブロックを含む左上のサンプルまたは中央サンプルを含むサブブロックを意味することができる。

【0215】

図 21 は、代表サブブロックの位置を示す図である。

【0216】

図 21 (a) は、現在のブロックの左上に位置するサブブロックが代表サブブロックに設定された例を示し、図 21 (b) は、現在のブロックの中央に位置するサブブロックが代表サブブロックに設定された例を示す。サブブロック単位で動き補償予測が実行された場合、現在のブロックの左上のサンプルを含むサブブロックまたは現在のブロックの中央サンプルを含むサブブロックの動きベクトルに基づいて、現在のブロックのインター領域マージ候補を導出することができる。

【0217】

現在のブロックのインター予測モードに基づいて、現在のブロックをインター領域マージ候補として使用するか否かを決定することもできる。例えば、アフィン動きモデルに基づいて符号化 / 復号化されたブロックをインター領域マージ候補として使用できないものに設定できる。これにより、現在のブロックがインター予測で符号化 / 復号化されていても、現在のブロックのインター予測モードがアフィン予測モードである場合、現在のブロックに基づいてインター領域動き情報テーブルを更新しない。

【0218】

あるいは、アフィン動きモデルに基づいて符号化 / 復号化されたブロックに含まれるサブブロックのうちの少なくとも 1 つのサブブロックベクトルに基づいて、インター領域マージ候補を導出することができる。例えば、現在のブロックの左上に位置するサブブロック、中央に位置するサブブロックまたは右上に位置するサブブロックを使用して、インター領域マージ候補を導出することができる。あるいは、複数のサブブロックのサブブロックベクトルの平均値をインター領域マージ候補の動きベクトルに設定できる。

30

40

50

【 0 2 1 9 】

あるいは、アフィン動きモデルに基づいて符号化／復号化されたブロックのアフィンシードベクトルの平均値に基づいて、インター領域マージ候補を導出することができる。例えば、現在のブロックの第1アフィンシードベクトル、第2アフィンシードベクトル、または第3アフィンシードベクトルのうちの少なくとも1つの平均値をインター領域マージ候補の動きベクトルに設定できる。

【 0 2 2 0 】

あるいは、インター予測モードごとにインター領域動き情報テーブルを構成することができる。例えば、イントラブロックコピーによって符号化／復号化されたブロックのためのインター領域動き情報テーブル、並進動きモデルに基づいて符号化／復号化されたブロックのためのインター領域動き情報テーブル、またはアフィン動きモデルに基づいて符号化／復号化されたブロックのためのインター領域動き情報テーブルのうちの少なくとも1つを定義することができる。現在のブロックのインター予測モードに従って、複数のインター領域動き情報テーブルのいずれかを選択することができる。

10

【 0 2 2 1 】

図22は、インター予測モードごとにインター領域動き情報テーブルを生成する例を示す図である。

【 0 2 2 2 】

非アフィン動きモデルに基づいてブロックを符号化／復号化する場合、前記ブロックに基づいて導出されたインター領域マージ候補 $m \vee C \text{ and } L \text{ is } t$ をインター領域非アフィン動き情報テーブル $H \vee p \text{ C and L is t}$ に追加することができる。一方、アフィン動きモデルに基づいてブロックを符号化／復号化する場合、前記ブロックに基づいて導出されたインター領域マージ候補 $m \vee A \text{ f } C \text{ and } L \text{ is } t$ をインター領域アフィン動き情報テーブル $H \vee p \text{ A f } C \text{ and L is t}$ に追加することができる。

20

【 0 2 2 3 】

ブロックのアフィンシードベクトルは、アフィン動きモデルに基づいて符号化／復号化されたブロックから導出されるインター領域マージ候補に記憶されることができる。これにより、前記インター領域マージ候補を、現在のブロックのアフィンシードベクトルを導出するためのマージ候補として使用することができる。

【 0 2 2 4 】

30

説明したインター領域動き情報テーブルに加えて、別のインター領域動き情報テーブルを定義することもできる。上記説明したインター領域動き情報テーブル（以下、第1インター領域動き情報テーブルと呼ぶ）加えて、長期動き情報テーブル（以下、第2インター領域動き情報テーブルと呼ぶ）を定義することもできる。ここで、長期動き情報テーブルは長期マージ候補を含む。

【 0 2 2 5 】

第1インター領域動き情報テーブルおよび第2インター領域動き情報テーブルがすべて空である場合、まず、インター領域マージ候補を第2インター領域動き情報テーブルに追加することができる。第2インター領域動き情報テーブルに使用可能なインター領域マージ候補の数が最大数に達した後にのみ、第1インター領域動き情報テーブルにインター領域マージ候補を追加することができる。

40

【 0 2 2 6 】

あるいは、1つのインター領域マージ候補を、第2インター領域動き情報テーブルと第1インター領域動き情報テーブルの両方に追加することもできる。

【 0 2 2 7 】

この場合、構成済みの第2インター領域動き情報テーブルをもはや更新しないことができる。あるいは、復号化領域がスライスの所定の比率以上である場合、第2インター領域動き情報テーブルを更新することができる。あるいは、N個の符号化ツリーユニットラインごとに、第2インター領域動き情報テーブルを更新することができる。

【 0 2 2 8 】

50

一方、インター予測で符号化 / 復号化されたブロックが生成されるたびに、第 1 インターフィールド動き情報テーブルを更新することができる。ただし、第 2 インターフィールド動き情報テーブルに追加されるインター領域マージ候補は、第 1 インターフィールド動き情報テーブルを更新するために使用されないように設定することもできる。

【 0 2 2 9 】

第 1 インターフィールド動き情報テーブルまたは第 2 インターフィールド動き情報テーブルのいずれかを選択するための情報が、ビットストリームを介してシグナリングできる。マージ候補リストに含まれるマージ候補の数が最大値未満である場合、前記情報で指示される、インター領域動き情報テーブルに含まれるマージ候補をマージ候補リストに追加することができる。

10

【 0 2 3 0 】

あるいは、現在のブロックのサイズ、形状、インター予測モード、双方向予測するか否か、動きベクトルをリファインするか否か、または三角分割するか否かに基づいて、インター領域動き情報テーブルを選択することができる。

【 0 2 3 1 】

あるいは、第 1 インターフィールド動き情報テーブルに含まれるインター領域マージ候補を追加しても、マージ候補リストに含まれるマージ候補の数が最大マージ数を超えない場合、第 2 インターフィールド動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加することができる。

【 0 2 3 2 】

20

図 23 は、長期動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加する例を示す図である。

【 0 2 3 3 】

マージ候補リストに含まれるマージ候補の数が最大数未満である場合、第 1 インターフィールド動き情報テーブル HmvpCanLis t に含まれるインター領域マージ候補をマージ候補リストに追加することができる。第 1 インターフィールド動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加しても、マージ候補リストに含まれるマージ候補の数が最大数未満である場合、長期動き情報テーブル HmvpLTCandLis t に含まれるインター領域マージ候補をマージ候補リストに追加することができる。

【 0 2 3 4 】

30

表 2 は、長期動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加するプロセスを示す。

40

50

【表2】

HMVPCandList内のインデックスHMVPLTIdx = 1..numHMVPLTCandである各候補について、combStopがtrueになるまで以下の順序付けられたステップを繰り返す

- sameMotionをfalseに設定する
- hmpStopがfalseであり、numCurrMergeCandidateが (MaxNumMergeCandidate - 1) より小さい場合、hmvpLTをtrueに設定する
- HMVPLTCandList [NumLTHmvp - HMVPLTIdx] が同じ動きベクトルと同じ参照インデックスを持ち、Iが0である..numOriginalMergeCandidate - 1の任意のmergeCandidateList [i] とHasBeenPruned [i] がfalseである場合、sameMotionをtrueに設定する
- sameMotionがfalseである場合、mergeCandidateList [numCurrMergeCandidate++] をHMVPLTCandList [NumLTHmvp - HMVPLTIdx] に設定する
- numCurrMergeCandidateが (MaxNumMergeCandidate - 1) と等しい場合、hmvpLTSStopをtrueに設定する

10

20

30

【0235】

インター領域マージ候補が動き情報に加えて追加情報を含むように設定することができる。例えば、インター領域マージ候補のブロックのサイズ、形状、またはブロックの分割情報を追加保存することができる。現在のブロックのマージ候補リストを構成する場合、インター領域マージ候補のうち、サイズ、形状または分割情報が現在のブロックと同じまたは類似するインター領域マージ候補のみを使用するか、またはサイズ、形状または分割情報が現在のブロックと同じまたは類似するインター領域マージ候補を、マージ候補リストに優先的に追加することができる。

40

【0236】

あるいは、ブロックサイズ、形状、または分割情報ごとに、インター領域動き情報テーブルを生成することができる。複数のインター領域動き情報テーブルのうち、現在のブロックの形状、サイズまたは分割情報に対応するインター領域動き情報テーブルを使用して、現在のブロックのマージ候補リストを生成することができる。

【0237】

あるいは、動きベクトルの解像度ごとに、インター領域動き情報テーブルを生成することができる。例えば、現在のブロックの動きベクトルが1/4画素の解像度を持つ場合、現在のブロックから導出されたインター領域マージ候補をインター予測領域4分の1画素

50

動き情報リストに追加することができる。現在のブロックの動きベクトルが1つの整数画素の解像度を持つ場合、現在のブロックから導出されたインター領域マージ候補をインター予測領域整数画素動き情報リストに追加することができる。現在のブロックの動きベクトルが4つの整数画素の解像度を持つ場合、現在のブロックから導出されたインター領域マージ候補をインター予測領域4の整数画素動き情報リストに追加することができる。符号化／復号化オブジェクトブロックの動きベクトル解像度に応じて、複数のインター動き情報リストのいずれかを選択することができる。

【0238】

マージオフセットベクトル符号化方法を現在のブロックに適用する場合、現在のブロックに基づいて導出されたインター領域マージ候補をインター領域動き情報テーブル $HmvpCandList$ に追加する代わりに、インター予測領域マージオフセット動き情報テーブル $HmvpHMVD CandList$ に追加することができる。この場合、前記インター領域マージ候補は、現在のブロックの動きベクトルオフセット情報を含み得る。 $HmvpHMVD CandList$ は、マージオフセットベクトル符号化方法が適用されたブロックのオフセットを導出するために使用されることができる。

10

【0239】

現在のブロックのマージ候補リストに含まれるマージ候補の数が最大値でない場合、インター領域動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加することができる。上記の追加プロセスは、インデックスの昇順または降順で実行される。例えば、インデックスが最大のインター領域マージ候補からマージ候補リストに追加することができる。

20

【0240】

インター領域動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加する場合、インター領域マージ候補とマージ候補リストに記憶された複数のマージ候補との間の冗長検査を実行することができる。

【0241】

例えば、表3は、インター領域マージ候補をマージ候補リストに追加するプロセスを示す。

30

40

50

【表 3】

<p>HMVPC and List内の、インデックスがHMVPIdx = 1..numCheckedHMVPC and である各候補について、combsStopがtrueになるまで以下の順序付けられたステップを繰り返す</p> <ul style="list-style-type: none"> — sameMotionをfalseに設定する — HMVPC and List [NumHmp - HMVPIdx] が同じ動きベクトルと同じ参照インデックスを持ち、Iが0である.. numOrigMergeC and - 1の任意のmergeC and List [i] と Has Been Pruned [i] が falseである場合、sameMotionをtrueに設定する — sameMotionがfalseである場合、mergeC and List [numCurrMergeC and ++] をHMVPC and List [NumHmp - HMVPIdx] に設定する — numCurrMergeC and が (MaxNumMergeC and - 1) と等しい場合、hmpStopをtrueに設定する 	10 20 20
--	----------------

【0242】

30 インター領域動き情報テーブルに含まれるインター領域マージ候補の一部に対してのみ、冗長検査を実行することもできる。例えば、インデックスが閾値以上または閾値以下であるインター領域マージ候補に対してのみ、冗長検査を実行することができる。

【0243】

あるいは、マージ候補リストに記憶されたマージ候補の一部に対してのみ、冗長検査を実行することができる。例えば、インデックスが閾値以上または閾値以下であるマージ候補または特定の位置のブロックから導出されたマージ候補に対してのみ、冗長検査を実行することができる。ここで、特定の位置は、現在のブロックの左側隣接ブロック、上部隣接ブロック、右上隣接ブロックまたは左下隣接ブロックのうちの少なくとも1つを含み得る。

【0244】

40 図24は、マージ候補の一部のみに対して冗長検査を実行する例を示す図である。

【0245】

インター領域マージ候補 HmpC and [j] をマージ候補リストに追加する場合、インター領域マージ候補について、インデックスが最大の2つのマージ候補 mergeC and List [NumMerge - 2] と mergeC and List [NumMerge - 1]との冗長検査を実行することができる。ここで、NumMergeは、使用可能な空間マージ候補と時間マージ候補の数を示す。

【0246】

第1インター領域マージ候補と同じマージ候補を見つかった場合、第2インター領域マージ候補に対して冗長検査を実行する場合、前記第1インター領域マージ候補と同じマ

30

40

50

ジ候補との冗長検査を省略することができる。

【0247】

図25は、特定のマージ候補について冗長検査を省略する例を示す図である。

【0248】

インデックスが i であるインター領域マージ候補 $HmvpCand[i]$ をマージ候補リストに追加する場合、前記インター領域マージ候補とマージ候補リストに記憶されたマージ候補との間の冗長検査を実行することができる。この場合、インター領域マージ候補 $HmvpCand[i]$ と同じマージ候補 $mergeCandList[j]$ を見つかった場合、インター領域マージ候補 $HmvpCand[i]$ をマージ候補リストに追加せずに、インデックスが $i - 1$ であるインター領域マージ候補 $HmvpCand[i - 1]$ とマージ候補との間の冗長検査を実行することができる。この場合、インター領域マージ候補 $HmvpCand[i - 1]$ とマージ候補 $mergeCandList[j]$ との間の冗長検査を省略することができる。

10

【0249】

例えば、図25に示される例では、 $HmvpCand[i]$ と $mergeCandList[2]$ が同じであると判断された。これにより、 $HmvpCand[i]$ は、マージ候補リストに追加されず、 $HmvpCand[i - 1]$ に対する冗長検査を実行することができる。この場合、 $HvmpCand[i - 1]$ と $mergeCandList[2]$ との間の冗長検査を省略することができる。

20

【0250】

現在のブロックマージ候補リストに含まれるマージ候補の数が最大値未満である場合、インター領域マージ候補に加えて、ペアワイズ(*pairwise*)マージ候補またはゼロマージ候補のうちの少なくとも1つを含むこともできる。ペアワイズマージ候補とは、2つ以上のマージ候補の動きベクトルの平均値を動きベクトルとしたマージ候補を指し、ゼロマージ候補とは、動きベクトルが0であるマージ候補を指す。

【0251】

現在のブロックのマージ候補リストは、以下の順序でマージ候補を追加することができる。

【0252】

空間マージ候補 - 時間マージ候補 - インター領域マージ候補 - (インター予測アフィンマージ候補) - ペアワイズマージ候補 - 零マージ候補

30

空間マージ候補とは、隣接ブロックまたは非隣接ブロックのうちの少なくとも1つから導出されたマージ候補を指し、時間マージ候補とは、前の参照画像から導出されたマージ候補を指す。インター予測アフィンマージ候補とは、アフィン動きモデルで符号化 / 復号化されたブロックから導出されたインター領域マージ候補を指す。

【0253】

高レベル動きベクトル予測モードでもインター領域動き情報テーブルを使用することができる。例えば、現在のブロックの動きベクトル予測候補リストに含まれる動きベクトル予測候補の数が最大値未満である場合、インター領域動き情報テーブルに含まれるインター領域マージ候補を、現在のブロックの動きベクトル予測候補として設定することができる。具体的には、インター領域マージ候補の動きベクトルを動きベクトル予測候補として設定することができる。

40

【0254】

現在のブロックの動きベクトル予測候補リストに含まれる動きベクトル予測候補のいずれかが選択される場合、選択された候補を、現在のブロックの動きベクトル予測者として設定することができる。その後、現在のブロックの動きベクトル残差値を復号化した後、動きベクトル予測値と動きベクトル残差値を加算することによって、現在のブロックの動きベクトルを取得することができる。

【0255】

現在のブロックの動きベクトル予測候補リストは、以下の順序で構成できる。

50

【 0 2 5 6 】

空間動きベクトル予測候補 - 時間動きベクトル予測候補 - インター復号化領域マージ候補 - (インター復号化領域アフィンマージ候補) - ゼロ動きベクトル予測候補

空間動きベクトル予測候補とは、隣接ブロックまたは非隣接ブロックのうちの少なくとも1つから導出された動きベクトル予測候補を指し、時間動きベクトル予測候補とは、前の参照画像から導出された動きベクトル予測候補を指す。インター予測アフィンマージ候補とは、アフィン動きモデルで符号化 / 復号化されたブロックから導出されたインター予測動きベクトル予測候補を指す。ゼロ動きベクトル予測候補とは、動きベクトルの値が0である候補を指す。

【 0 2 5 7 】

10

符号化ブロックよりも大きいサイズのマージ処理領域を指定することができる。マージ処理領域に含まれる符号化ブロックは、順次符号化 / 復号化せずに、並列処理することができる。ここで、順次符号化 / 復号化されないのは、符号化 / 復号化順序を指定しないことを意味する。これにより、マージ処理領域に含まれるブロックの符号化 / 復号化プロセスを独立して処理することができる。あるいは、マージ処理領域に含まれるブロックは、マージ候補を共有することができる。ここで、マージ候補は、マージ処理領域に基づいて導出されることができる。

【 0 2 5 8 】

上述した特徴に応じて、マージ処理領域を並列処理領域、マージ共有領域 (S M R : S h a r e d M e r g e R e g i o n)、またはマージ推定領域 (M E R : M e r g e E s t i m a t i o n R e g i o n) と呼ぶこともできる。

20

【 0 2 5 9 】

現在のブロックのマージ候補は、符号化ブロックに基づいて導出されることができる。ただし、現在のブロックが現在のブロックよりも大きいサイズの並列マージ領域に含まれている場合、現在のブロックと同じ並列マージ領域に含まれる候補ブロックを、マージ候補として使用できないものに設定することができる。

【 0 2 6 0 】

図26は、現在のブロックと同じ並列マージ領域に含まれる候補ブロックがマージ候補として使用できないように設定される例を示す図である。

【 0 2 6 1 】

30

図26(a)に示される例では、CU5を符号化 / 復号化する場合、CU5に隣接する参照サンプルを含むブロックを候補ブロックとして設定することができる。この場合、CU5と同じ並列マージ領域に含まれる候補ブロックX3とX4を、CU5のマージ候補として使用できないものに設定することができる。一方、CU5と同じ並列マージ領域に含まれない候補ブロックX0、X1、およびX2を、マージ候補として使用できるものに設定することができる。

【 0 2 6 2 】

図26(b)に示される例では、CU8を符号化 / 復号化する場合、CU8に隣接する参照サンプルを含むブロックを候補ブロックとして設定することができる。この場合、CU8と同じ並列マージ領域に含まれる候補ブロックX6、X7、およびX8を、マージ候補として使用できないものに設定することができる。一方、CU5と同じ並列マージ領域に含まれない候補ブロックX5とX9を、マージ候補として使用できるものに設定することができる。

40

【 0 2 6 3 】

並列マージ領域は、正方形または非正方形であり得る。並列マージ領域を決定するための情報が、ビットストリームを介してシグナリングできる。前記情報は、並列マージ領域の形状を示す情報および並列マージ領域のサイズを示す情報のうちの少なくとも1つを含み得る。並列マージ領域が非正方形である場合、並列マージ領域のサイズを示す情報、並列マージ領域の幅および / または高さを示す情報、または並列マージ領域の幅と高さの比率を示す情報のうちの少なくとも1つが、ビットストリームを介してシグナリングできる。

50

【 0 2 6 4 】

並列マージ領域のサイズは、ビットストリームを介してシグナリングされる情報、画像解像度、スライスのサイズまたはタイルのサイズのうちの少なくとも1つに基づいて決定することができる。

【 0 2 6 5 】

並列マージ領域に含まれるブロックに対して動き補償予測を実行する場合、動き補償予測が行われたブロックの動き情報に基づいて導出されたインター領域マージ候補をインターポーチメント動き情報テーブルに追加することができる。

【 0 2 6 6 】

ただし、並列マージ領域に含まれるブロックから導出されたインター領域マージ候補をインター領域動き情報テーブルに追加する場合、上記ブロックより実在の符号化／復号化が遅れて並列マージ領域内の他のブロックを符号化／復号化する場合、上記のブロックから導出されたインター領域マージ候補を使用する場合がある。つまり、並列マージ領域に含まれるブロックを符号化／復号化する際のブロック間の依存関係を排除する必要があるが、並列マージ領域に含まれる別のブロックの動き情報を使用して動き予測補償を実行する場合がある。このような問題を解決するために、並列マージ領域に含まれるブロックに対する符号化／復号化が完了しても、符号化／復号化が完了したブロックの動き情報をインター領域動き情報テーブルに追加しないことができる。

10

【 0 2 6 7 】

あるいは、並列マージ領域に含まれるブロックに対して動き補償予測を実行する場合、前記ブロックから導出されたインター領域マージ候補を事前定義された順序でインター領域動き情報テーブルに追加することができる。ここで、事前定義された順序は、並列マージ領域または符号化ツリーユニット内の符号化ブロックの走査順序に基づいて決定できる。前記走査順序は、ラスター走査、水平走査、垂直走査、またはジグザグ走査のうちの少なくとも1つであり得る。あるいは、事前定義された順序は、各ブロックの動き情報を有するブロックの数に基づいて決定することができる。

20

【 0 2 6 8 】

あるいは、一方向の動き情報を含むインター領域マージ候補を、双方向動き情報を含むインター領域マージ候補よりも先にインター予測領域マージリストに追加することができる。これとは逆に、双方向動き情報を含むインター領域マージ候補を、一方向動き情報を含むインター領域マージ候補よりも先にインター領域マージ候補テーブルに追加することができる。

30

【 0 2 6 9 】

あるいは、並列マージ領域または符号化ツリーユニット内の使用頻度が高い順または使用頻度の低い順に従って、インター領域マージ候補をインター領域動き情報テーブルに追加することができる。

【 0 2 7 0 】

現在のブロックが並列マージ領域に含まれており、現在のブロックのマージ候補リストに含まれるマージ候補の数が最大数未満である場合、インター領域動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加することができる。この場合、現在のブロックと同じ並列マージ領域に含まれるブロックから導出されたインター領域マージ候補を、現在のブロックのマージ候補リストに追加しないように設定することができる。

40

【 0 2 7 1 】

あるいは、現在のブロックが並列マージ領域に含まれる場合、インター領域動き情報テーブルに含まれるインター領域マージ候補を使用しないように設定することができる。つまり、現在のブロックのマージ候補リストに含まれるマージ候補の数が最大数未満であっても、インター領域動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加しないことができる。

【 0 2 7 2 】

50

並列マージ領域または符号化ツリーユニットのインター領域動き情報テーブルを構成することができる。前記インター領域動き情報テーブルは、並列マージ領域に含まれるブロックの動き情報を一時的に記憶する役割を果たす。一般的なインター領域動き情報テーブルと、並列マージ領域または符号化ツリーユニットのためのインター領域動き情報テーブルを区別するために、並列マージ領域または符号化ツリーユニットのためのインター領域動き情報テーブルを一時動き情報テーブルと呼ぶ。さらに、一時動き情報テーブルに記憶されたインター領域マージ候補を一時マージ候補と呼ぶ。

【0273】

図27は、一時動き情報テーブルを示す図である。

【0274】

符号化ツリーユニットまたは並列マージ領域のための一時動き情報テーブルを構成することができる。符号化ツリーユニットまたは並列マージ領域に含まれる現在のブロックの動き補償予測が行われた場合、前記ブロックの動き情報をインター領域動き情報テーブル $HmvpCandList$ に追加しないことができる。代わりに、前記ブロックから導出された一時マージ候補を一時動き情報テーブル $HmvpMERCandList$ に追加することができる。つまり、一時動き情報テーブルに追加された一時マージ候補をインター領域動き情報テーブルに追加しないことができる。これにより、インター領域動き情報テーブルは、現在のブロックを含む符号化ツリーユニットまたは並列マージ領域に含まれるブロックの動き情報に基づいて導出されたインター領域マージ候補を含まない可能性がある。

10

【0275】

一時動き情報テーブルが包含できるマージ候補の最大数を、インター領域動き情報テーブルが包含できるマージ候補の最大数と同じように設定することができる。あるいは、一時動き情報テーブルが包含できるマージ候補の最大数は、符号化ツリーユニットまたは並列マージ領域のサイズに従って決定できる。

20

【0276】

符号化ツリーユニットまたは並列マージ領域に含まれる現在のブロックが、前記符号化ツリーユニットまたは前記並列マージ領域の一時動き情報テーブルを使用しないように設定することができる。つまり、現在のブロックのマージ候補リストに含まれるマージ候補の数が最大値未満である場合、インター領域動き情報テーブルに含まれるインター領域マージ候補をマージ候補リストに追加し、一時動き情報テーブルに含まれる一時マージ候補をマージ候補リストに追加しないことができる。これにより、現在のブロックと同じ符号化ツリーユニットまたは並列マージ領域に含まれる他のブロックの動き情報を現在のブロックの動き補償予測に使用しないことができる。

30

【0277】

符号化ツリーユニットまたは並列マージ領域に含まれるすべてのブロックの符号化 / 復号化が完了すると、インター領域動き情報テーブルと一時動き情報テーブルをマージすることができる。

【0278】

図28は、インター予測動き情報テーブルと一時動き情報テーブルをマージする例を示す図である。

40

【0279】

符号化ツリーユニットまたは並列マージ領域に含まれるすべてのブロックの符号化 / 復号化が完了すると、図28に示されるように、一時動き情報テーブルに含まれる一時マージ候補をインター領域動き情報テーブルに更新できる。

【0280】

この場合、一時動き情報テーブルに含まれる一時マージ候補は、一時動き情報テーブルに挿入された順序（すなわち、インデックス値の昇順または降順）に従ってレーム間動き情報テーブルに追加されることができる。

【0281】

50

別の例として、事前定義された順序に従って、一時動き情報テーブルに含まれる一時マージ候補をインター領域動き情報テーブルに追加することができる。

【 0 2 8 2 】

ここで、事前定義された順序は、並列マージ領域または符号化ツリーユニット内の符号化ブロックの走査順序に基づいて決定できる。前記走査順序は、ラスター走査、水平走査、垂直走査、またはジグザグ走査のうちの少なくとも1つであり得る。あるいは、事前定義された順序は、各ブロックの動き情報を有するブロックの数に基づいて決定されることがある。

【 0 2 8 3 】

あるいは、一方向の動き情報を含む一時マージ候補を、双方向の動き情報の一時マージ候補よりも先にインター予測マージリストに追加することができる。一方、双方向の動き情報を含む一時マージ候補を、一方向の動き情報を含む一時マージ候補よりも先にインター領域マージ候補リストに追加することができる。

10

【 0 2 8 4 】

あるいは、並列マージ領域または符号化ツリーユニット内の使用頻度が高い順または使用頻度の低い順に従って、一時マージ候補をインター領域動き情報テーブルに追加することができる。

【 0 2 8 5 】

一時動き情報テーブルに含まれる一時マージ候補をインター領域動き情報テーブルに追加する場合、一時マージ候補に対して冗長検査を実行することができる。例えば、一時動き情報テーブルに含まれる一時マージ候補と同じインター領域マージ候補がインター領域動き情報テーブルに記憶されている場合、一時マージ候補をインター領域動き情報テーブルに追加しないことができる。この場合、インター領域動き情報テーブルに含まれるインター領域マージ候補の一部に対して冗長検査を実行することができる。例えば、インデックスが閾値より大きいか等しいインター領域マージ候補に対して冗長検査を実行することができる。例えば、一時マージ候補が、インデックスが事前定義された値より大きいか等しいインター領域マージ候補と同一である場合、一時マージ候補をインター領域動き情報テーブルに追加しないことができる。

20

【 0 2 8 6 】

イントラ予測は、現在のブロック周辺の符号化 / 復号化が完了した復元サンプルを使用して、現在のブロックを予測することである。この場合、現在のブロックのイントラ予測は、インループフィルタが適用される前の復元サンプルを使用することができる。

30

【 0 2 8 7 】

イントラ予測技術は、行列 (Matrix) に基づくイントラ予測および周辺復元サンプルとの方向性を考慮した一般的なイントラ予測を含む。現在のブロックのイントラ予測技術を示す情報は、ビットストリームを介してシグナルリングできる。前記情報は1ビットのフラグであり得る。あるいは、現在のブロックの位置、サイズ、形状、または隣接ブロックのイントラ予測技術のうちの少なくとも1つに基づいて、現在のブロックのイントラ予測技術を決定できる。例えば、現在のブロックが画像境界をまたがって存在する場合、現在のブロックを、現在のブロックを行列に基づくイントラ予測が適用されないように設定することができる。

40

【 0 2 8 8 】

行列に基づくイントラ予測は、符号化器と復号化器に記憶された行列と、現在のブロック周辺の復元サンプルとの間の行列積に基づいて、現在のブロックの予測ブロックを取得する方法である。記憶された複数の行列のいずれかを指定するための情報が、ビットストリームを介してシグナリングできる。復号化器は、前記情報および現在のブロックのサイズに基づいて、現在のブロックのイントラ予測に使用される行列を決定することができる。

【 0 2 8 9 】

一般的なイントラ予測は、非方向性イントラ予測モードまたは方向性イントラ予測モードに基づいて、現在のブロックに関する予測ブロックを取得する方法である。以下、図面

50

を参照して、一般的なイントラ予測に基づいてイントラ予測を実行するプロセスについて詳細に説明する。

【0290】

図29は、本発明の実施例によるイントラ予測方法のフローチャートである。

【0291】

現在のブロックの参照サンプルラインを決定できる(S2901)。参照サンプルラインとは、現在のブロックの上部および／または左側からK番目離れたラインに含まれる参照サンプルの集合を指す。現在のブロックの周辺の符号化／復号化が完了した復元サンプルから参照サンプルを導出することができる。

【0292】

複数の参照サンプルライン内の現在のブロックの参照サンプルラインを識別するためのインデックス情報が、ビットストリームを介してシグナルリングできる。複数の参照サンプルラインは、現在のブロックの上部および／または左側の1行／列目、2行／列目、3行／列目または4行／列目のうちの少なくとも1つを含み得る。表4は、各参照サンプルラインのそれぞれに割り当てるインデックスを示す。表4では、1行／列目、2行／列目、および4行／列目が参照サンプルラインの候補として使用されると想定する。

【表4】

インデックス	参照サンプルライン
0	第1参照サンプルライン
1	第2参照サンプルライン
2	第4参照サンプルライン

【0293】

現在のブロックの位置、サイズ、形状、または隣接ブロックの予測符号化モードのうちの少なくとも1つに基づいて、現在のブロックの参照サンプルラインを決定できる。例えば、現在のブロックが、画像、タイル、スライスまたは符号化ツリーユニットの境界に接する場合、第1参照サンプルラインを現在のブロックの参照サンプルラインとして決定できる。

【0294】

参照サンプルラインは、現在のブロックの上部に位置する上部参照サンプル、および現在のブロック左側に位置する左側参照サンプルを含み得る。現在のブロックの周辺の復元サンプルから、上部参照サンプルおよび左側参照サンプルを導出することができる。前記復元サンプルは、インループフィルタが適用される前の状態であることができる。

【0295】

図30は、各参照サンプルラインに含まれる参照サンプルを示す図である。

【0296】

現在のブロックのイントラ予測モードに従って、参照サンプルラインに属する参照サンプルのうちの少なくとも1つを使用して、予測サンプルを取得することができる。

【0297】

次に、現在のブロックのイントラ予測モードを決定することができる(S2902)。現在のブロックのイントラ予測モードの場合、非方向性イントラ予測モードまたは方向性イントラ予測モードのうちの少なくとも1つを、現在のブロックのイントラ予測モードとして決定することができる。非方向性イントラ予測モードは、Planar(plane r)およびDCを含み、方向性イントラ予測モードは、左下下対角方向から右上対角方向に33個または65個のモードを含む。

【0298】

10

20

30

40

50

図31は、イントラ予測モードを示す図である。

【0299】

図31(a)は、35個のイントラ予測モードを示し、図31(b)は、67個のイントラ予測モードを示す。

【0300】

図31に示されるものよりも多いまたは少ない数のイントラ予測モードを定義することができる。

【0301】

現在のブロックに隣接する隣接ブロックのイントラ予測モードに基づいて、最も可能なモード(MPM: Most Probable Mode)を設定することができる。ここで、隣接ブロックは、現在のブロックの左側に隣接する左側隣接ブロック、および現在のブロックの上部に隣接する上部隣接ブロックを含み得る。現在のブロックの左上のサンプルの座標が(0, 0)である場合、左側隣接ブロックは、(-1, 0)、(-1, H-1)または(-1, (H-1)/2)位置のサンプルを含み得る。ここで、Hは現在のブロックの高さを表す。上部隣接ブロックは、(0, -1)、(W-1, -1)または((W-1)/2, -1)位置のサンプルを含み得る。ここで、Wは現在のブロックの幅を表す。

10

【0302】

一般的なイントラ予測で隣接ブロックを符号化する場合、隣接ブロックのイントラ予測モードに基づいてMPMを導出することができる。具体的には、左側隣接ブロックのイントラ予測モードを変数c and IntraPredMode Aに設定し、上部隣接ブロックのイントラ予測モードを変数c and IntraPredMode Bに設定することができる。

20

【0303】

この場合、隣接ブロックが使用不可能な場合(例えば、隣接ブロックがまだ符号化/復号化されていない場合、または隣接ブロックの位置が画像境界から離れている場合)、隣接ブロックが行列に基づいたイントラ予測で符号化され、かつ隣接ブロックがインター予測で符号化された場合または隣接ブロックが現在のブロックとは異なる符号化ツリーユニットに含まれる場合、隣接ブロックのイントラ予測モードに基づいて導出された変数c and IntraPredMode X(ここで、XはAまたはBである)をデフォルトモードとして設定することができる。ここで、デフォルトモードは、Planarモード、DCモード、垂直方向モードまたは水平方向モードのうちの少なくとも1を含み得る。

30

【0304】

あるいは、行列に基づくイントラ予測で隣接ブロックを符号化する場合、行列のいずれかを指定するためのインデックス値に対応するイントラ予測モードをc and IntraPredMode Xに設定することができる。そのために、行列を指定するためのインデックス値とイントラ予測モードのマッピング関係を示すルックアップテーブルを、事前に符号化器および復号化器に記憶することができる。

【0305】

変数c and IntraPredMode Aおよび変数c and IntraPredMode Bに基づいてMPMを導出することができる。符号化器および復号化器でMPMリストに含まれるMPMの数を事前定義することができる。例えば、MPMの数は、3、4、5または6であり得る。あるいは、MPMの数を示す情報が、ビットストリームを介してシグナリングできる。あるいは、隣接ブロックの予測符号化モード、現在のブロックのサイズまたは形状のうちの少なくとも1つに基づいて、MPMの数を決定することができる。

40

【0306】

後述される実施例では、MPMの数が3であると仮定し、この3つのMPMをMPM[0]、MPM[1]およびMPM[2]と呼ぶ。MPMの数が3を超える場合、MPMは、後述される実施例で説明する3つのMPMを含み得る。

【0307】

50

`c and IntraPredA` と `c and IntraPredB` が等しく、`c and IntraPredA` が `Planar` モードまたは `DC` モードである場合、`MPM[0]` と `MPM[1]` をそれぞれ `Planar` モードと `DC` モードに設定することができる。`MPM[2]` を、垂直イントラ予測モード、水平イントラ予測モードまたは対角方向イントラ予測モードに設置することができる。対角方向イントラ予測モードは、左下対角方向のイントラ予測モード、左上方向イントラ予測モードまたは右上方向イントラ予測モードであり得る。

【0308】

`c and IntraPredA` と `c and IntraPredB` が等しく、`c and IntraPredA` が方向性イントラ予測モードである場合、`MPM[0]` は `c and IntraPredA` と同じように設定することができる。`MPM[1]` および `MPM[2]` を、`c and IntraPredA` と同様のイントラ予測モードとして設定することができる。`c and IntraPredA` と同様のイントラ予測モードは、`c and IntraPredA` のインデックス差値が ± 1 または ± 2 であるイントラ予測モードであり得る。モジュロ演算 (%) とオフセット量を使用して、`c and IntraPredA` と同様のイントラ予測モードを導出することができる。

10

【0309】

`c and IntraPredA` と `c and IntraPredB` が異なる場合、`MPM[0]` を `c and IntraPredA` と同じように設定し、`MPM[1]` を `c and IntraPredA` と `c and IntraPredB` の両方が非方向性イントラ予測モードである場合、`MPM[2]` を垂直イントラ予測モード、水平イントラ予測モードまたは対角方向イントラ予測モードに設定することができる。あるいは、`c and IntraPredA` および `c and IntraPredB` のうちの少なくとも 1 つが方向性イントラ予測モードである場合、`MPM[2]` を、`Planar`、`DC`、または `c and IntraPredA` または `c and IntraPredB` のうちのより大きな値を持つものにオフセット値を加算または減算することによって導出されたイントラ予測モードに設定することができる。ここで、オフセット値は 1 または 2 であり得る。

20

【0310】

複数の `MPM` を含む `MPM` リストを生成でき、現在のブロックのイントラ予測モードと同じ `MPM` が `MPM` リストに含まれているか否かを示す情報がビットストリームを介してシグナルリングすることができる。前記情報は 1 ビットフラグであり、`MPM` フラグを呼ぶことができる。`MPM` フラグが、現在のブロックと同じ `MPM` が `MPM` リストに含まれていることを示す場合、`MPM` のいずれかを識別するためのインデックス情報がビットストリームを介してシグナルリングすることができる。前記インデックス情報によって指定された `MPM` を、現在のブロックのイントラ予測モードに設定することができる。`MPM` フラグが、現在のブロックと同じ `MPM` が `MPM` リストに含まれていないことを示す場合、`MPM` を除く残りのイントラ予測モードのいずれかを指示する残差モード情報が、ビットストリームを介してシグナルリングすることができる。残差モード情報は、`MPM` を除く残りのイントラ予測モードにインデックスを再び割り当てる場合、現在のブロックのイントラ予測モードに対応するインデックス値を示す。復号化器は、`MPM` を昇順にソートしつつ残差モード情報を `MPM` と比較して、現在のブロックのイントラ予測モードを決定することができる。例えば、残差モード情報が `MPM` より小さいか等しい場合、残差モード情報に 1 を加算して、現在のブロックのイントラ予測モードを導出することができる。

30

【0311】

デフォルトモードを `MPM` に設定する代わりに、現在のブロックのイントラ予測モードがデフォルトモードであるか否かを示す情報を、ビットストリームを介してシグナルリングすることができる。前記情報は 1 ビットフラグであり、前記フラグをデフォルトモードフラグと呼ぶことができる。`MPM` フラグが現在のブロックと同じ `MPM` が `MPM` リストに含まれていることを示す場合にのみ、デフォルトモードフラグをシグナルリングするこ

40

50

とができる。前述したように、デフォルトモードは、Planar、DC、垂直方向モードまたは水平方向モードのうちの少なくとも1つを含み得る。例えば、Planarがデフォルトモードに設定された場合、デフォルトモードフラグは、現在のブロックのイントラ予測モードがPlanarであるか否かを指示することができる。デフォルトモードフラグが、現在のブロックのイントラ予測モードがデフォルトモードでないことを指示する場合、インデックス情報によって指示されるMPMのいずれかを、現在のブロックのイントラ予測モードとして設定することができる。

【0312】

複数のイントラ予測モードをデフォルトモードに設定する場合、デフォルトモードのいずれかを指示するインデックス情報がシグナルリングすることができる。現在のブロックのイントラ予測モードを、前記インデックス情報によって示されるデフォルトモードとして設定することができる。10

【0313】

現在のブロックの参照サンプルラインのインデックスがでない場合、デフォルトモードを使用できないように設定することができる。これにより、参照サンプルラインのインデックスが0でない場合、デフォルトモードフラグをシグナルリングせずに、前記デフォルトモードフラグの値を事前定義された値（すなわち、false）に設定することができる。

【0314】

現在のブロックのイントラ予測モードが決定された場合、決定されたイントラ予測モードに基づいて、現在のブロックの予測サンプルを取得することができる（S2903）。20

【0315】

DCモードが選択された場合、参照サンプルの平均値に基づいて、現在のブロックに関する予測サンプルを生成することができる。具体的には、参照サンプルの平均値に基づいて、予測ブロック内全体のサンプルの値を生成することができる。現在のブロックの上部に位置する上部参照サンプルおよび現在のブロックの左側に位置する左側参照サンプルのうちの少なくとも1つを使用して、平均値を導出することができる。

【0316】

現在のブロックの形状に従って、平均値を導出するために使用される参照サンプルの数または範囲が異なる場合がある。例えば、現在のブロックが幅が高さより大きい非正方形ブロックである場合、上部参照サンプルのみを使用して、平均値を計算することができる。一方、現在のブロックが幅が高さより小さい非正方形ブロックである場合、左側参照サンプルのみを使用して、平均値を計算することができる。つまり、現在のブロックの幅と高さが異なる場合、長さがより長い方に隣接する参照サンプルのみを使用して、平均値を計算することができる。あるいは、現在のブロックの幅と高さの比率に基づいて、上部参照サンプルのみを使用して平均値を計算するか否か、または左側参照サンプルのみを使用して平均値を計算するか否かを決定することができる。30

【0317】

Planarモードを選択した場合、水平方向予測サンプルと垂直方向予測サンプルを使用して、予測サンプルを取得することができる。ここで、予測サンプルと同じ水平線上に位置する左側参照サンプルと右側参照サンプルに基づいて、水平方向予測サンプルを取得し、予測サンプルと同じ垂直線上に位置する上部参照サンプルと下部参照サンプルを使用して、垂直方向予測サンプルを取得する。ここで、現在のブロックの右上隅に隣接する参照サンプルをコピーして、右側参照サンプルを生成でき、現在のブロックの左下隅に隣接する参照サンプルをコピーして、下部参照サンプルを生成できる。左側参照サンプルと右側参照サンプルの加重合演算に基づいて、水平方向予測サンプルを取得でき、上部参照サンプルと下部参照サンプルの加重合演算に基づいて、垂直方向予測サンプルを取得できる。この場合、予測サンプルの位置に基づいて、各参照サンプルに付与される重み値を決定できる。水平方向予測サンプルと垂直方向予測サンプルの平均演算または加重合演算に基づいて、予測サンプルを取得することができる。加重合演算を実行する場合、予測サン40

プルの位置に基づいて、水平方向予測サンプルと垂直方向予測サンプルに付与される重み値を決定することができる。

【0318】

方向性予測モードが選択された場合、選択された方向性予測モードの予測方向（または予測角度）を表すパラメータを決定することができる。下記表5は、各イントラ予測モードのイントラ予測パラメータ `intрапредAng` を示す。

【表5】

PredModeIntra IntraPredAng	1-	232	326	421	517	613	79
PredModeIntra IntraPredAng	85	92	100	11-2	12-5	13-9	14-1 3
PredModeIntra IntraPredAng	15-17	16-21	17-26	18-32	19-26	20-21	21-1 7
PredModeIntra IntraPredAng	22-13	23-9	24-5	25-2	260	272	285
PredModeIntra IntraPredAng	299	3013	3117	3221	3326	3432	

10

20

30

40

【0319】

表5は、35個のイントラ予測モードが定義されたときに、インデックスが2～34のいずれかである各イントラ予測モードのイントラ方向パラメータを示す。33個よりも多くの方向性イントラ予測モードが定義された場合、表5をより細分化して、各方向性イントラ予測モードのイントラ方向パラメータを設定することができる。

【0320】

現在のブロックの上部参照サンプルおよび左側参照サンプルを一列に配列した後、イントラ方向パラメータの値に基づいて、予測サンプルを取得することができる。この場合、イントラ方向パラメータの値が負の数である場合、左側参照サンプルと上部参照サンプルを一列に配列することができる。

【0321】

図32および図33は、参照サンプルを一列に配列する一次元配列の例を示す図である。

【0322】

図32は、参照サンプルを垂直方向に配列する垂直方向一次元配列の例を示し、図33は、参照サンプルを水平方向に配列する水平方向一次元配列の例を示す。35個のイントラ予測モードが定義された場合を想定して、図32および図33の実施例を説明する。

【0323】

イントラ予測モードインデックスが11～18のいずれかである場合、上部参照サンプルを反時計回りに回転した水平方向一次元配列を適用し、レーム内予測モードインデックスが19～25のいずれかである場合、左側参照サンプルを時計回りに回転した垂直方向一次元配列を適用することができる。参照サンプルを一列に配列する場合、イントラ予測モードの角度を考慮することができる。

【0324】

50

イントラ方向パラメータに基づいて、参照サンプル決定パラメータを決定することができる。参照サンプル決定パラメータは、参照サンプルを指定するための参照サンプルインデックスおよび参照サンプルに適用される重み値を決定するための重み値パラメータを含み得る。

【0 3 2 5】

参照サンプルインデックス i_{Idx} と重みパラメータ i_{fact} は、それぞれ下記の式 2 および式 3 を介して取得することができる。

【数 2】

[式 2]

$$i_{Idx} = (y+1) * P_{ang} / 32$$

10

【数 3】

[式 3]

$$i_{fact} = [(y+1) * P_{ang}] \& 31$$

20

【0 3 2 6】

式 2 および式 3において、`Pang` はイントラ方向パラメータを表す。参照サンプルインデックス i_{Idx} によって指定された参照サンプルは、整数画素 (`Integer` `pel`) に該当する。

【0 3 2 7】

予測サンプルを導出する宇ために、少なくとも 1 つ以上の参照サンプルを指定することができる。具体的には、予測モードの傾きを考慮して、予測サンプルを導出するために使用される参照サンプルの位置を指定することができる。例えば、参照サンプルインデックス i_{Idx} を使用して、予測サンプルを導出するために使用される参照サンプルを指定することができる。

30

【0 3 2 8】

この場合、イントラ予測モードの傾きが 1 つの参照サンプルによって表されない場合、複数の参照サンプルを補間して予測サンプルを生成することができる。例えば、イントラ予測モードの傾きが、予測サンプルと第 1 参照サンプルとの間の傾きと、予測サンプルと第 2 参照サンプルとの間の傾きの間の値である場合、第 1 参照サンプルと第 2 参照サンプルを補間して予測サンプルを取得することができる。つまり、イントラ予測角度に従う角線 (`Angular Line`) が整数画素に位置する参照サンプルを通過しない場合、前記角線が通過する位置の左右または上下に隣接する参照サンプルを補間して、予測サンプルを取得することができる。

40

【0 3 2 9】

下記式 4 は、参照サンプルに基づいて予測サンプルを取得する例を示す。

【数 4】

[式 4]

$$P(x,y) = ((32 - i_{fact}) / 32) * Ref_1D(x + i_{Idx} + 1) + (i_{fact} / 32) * Ref_1D(x + i_{Idx} + 2)$$

【0 3 3 0】

50

式 4において、Pは予測サンプルを表し、Ref_1Dは、一次元配列された参照サンプルのいずれかを表す。この場合、予測サンプルの位置(x, y)と参照サンプルインデックスidxに基づいて参照サンプルの位置を決定することができる。

【0331】

イントラ予測モードの傾きが1つの参照サンプルで表現可能な場合、重み値パラメータifactを0に設定することができる。したがって、式4は、下記式5のように簡略化できる。

【数5】

[式5]

$$P(x,y)=Ref_1D(x+idx+1)$$

10

【0332】

複数のイントラ予測モードに基づいて、現在のブロックに対してイントラ予測を実行することができる。例えば、予測サンプルごとにイントラ予測モードを導出し、各予測サンプルに割り当てられたイントラ予測モードに基づいて、予測サンプルを導出することができる。

【0333】

あるいは、領域ごとにイントラ予測モードを導出し、各領域に割り当てられたイントラ予測モードに基づいて、各領域に対してイントラ予測を実行することができる。ここで、前記領域は、少なくとも1つのサンプルを含み得る。現在のブロックのサイズ、形状、またはイントラ予測モードのうちの少なくとも1つに基づいて、前記領域のサイズまたは形状のうちの少なくとも1つを適応的に決定することができる。あるいは、符号化器と復号化器では、現在のブロックのサイズや形状に関係なく、領域のサイズまたは形状のうちの少なくとも1つを事前定義することができる。

20

【0334】

あるいは、複数のイントラ予測のそれぞれに基づいてイントラ予測を実行し、複数回のイントラ予測を介して取得された複数の予測サンプルの平均演算または加重合演算に基づいて、最終的な予測サンプルを導出することができる。例えば、第1イントラ予測モードに基づいてイントラ予測を実行して、第1予測サンプルを取得し、第2イントラ予測モードに基づいてイントラ予測を実行して、第2予測サンプルを取得することができる。その後、第1予測サンプルと第2予測サンプルとの間の平均演算または加重合演算に基づいて、最終的な予測サンプルを取得することができる。この場合、第1イントラ予測モードが非方向性/方向性予測モードであるか否か、第2イントラ予測モードが非方向性/方向性予測モードであるか否か、または隣接ブロックのイントラ予測モードのうちの少なくとも1つを考慮して、第1予測サンプルと第2予測サンプルのそれぞれに割り当てられる重み値を決定することができる。

30

【0335】

複数のイントラ予測モードは、非方向性イントラ予測モードと方向性予測モードの組み合わせ、方向性予測モードの組み合わせ、または非方向性予測モードの組み合わせであり得る。

40

【0336】

図34は、方向性イントラ予測モードとx軸に平行な直線との間に形成される角度を示す図である。

【0337】

図34に示されるように、方向性予測モードは、左下対角方向から右上対角方向の間に存在することができる。x軸と方向性予測モードが形成する角度で説明すれば、方向性予測モードは、45度(左下対角方向)から、-135度(右上対角方向)の間に存在する

50

ことができる。

【0338】

現在のブロックが非正方形の形である場合、現在のブロックのイントラ予測モードに基づいて、イントラ予測角度に従う角線上に位置する参照サンプルのうち、予測サンプルに近い参照サンプルではなく、参照サンプルからより離れた位置にある参照サンプルを使用して、予測サンプルを導出する。

【0339】

図35は、現在のブロックが非正方形である場合に予測サンプルを取得する例を示す図である。

【0340】

例えば、図35(a)に示されるように、現在のブロックが幅が高さより大きい非正方形であり、現在のブロックのイントラ予測モードが、0度から45度の間の角度を有する方向性イントラ予測モードであると仮定する。この場合、現在のブロックの右側の列付近の予測サンプルAを導出する場合、前記角度に従う角度モード上に位置する参照サンプルのうち、前記予測サンプルに近い上部参照サンプルTの代わりに、前記予測サンプルから遠く離れた左側参照サンプルLを使用する場合が発生する可能性がある。

10

【0341】

別の例として、図35(b)に示されるように、現在のブロックが、高さが幅より大きい非正方形であり、現在のブロックのイントラ予測モードが、-90度から-135度の間の角度を有する方向性イントラ予測モードであると仮定する。この場合、現在のブロックの下段付近の予測サンプルAを導出する場合、前記角度に従う角度モード上に位置する参照サンプルのうち、前記予測サンプルに近い左側参照サンプルLの代わりに、前記予測サンプルから遠く離れた上部参照サンプルTを使用する場合が発生する可能性がある。

20

【0342】

上記の問題を解決するためには、現在のブロックが非正方形である場合、現在のブロックのイントラ予測モードを逆方向のイントラ予測モードに置き換えることができる。これにより、非正方形ブロックについては、図31に示される方向性予測モードよりも大きいまたは小さい角度を有する方向性予測モードを使用することができる。このような方向性イントラ予測モードを広角イントラ予測モードと定義することができる。広角イントラ予測モードは、45度から-135度範囲に属さない方向性イントラ予測モードを示す。

30

【0343】

図36は、広角イントラ予測モードを示す図である。

【0344】

図36に示される例では、インデックスが-1~-14であるイントラ予測モードおよびインデックスが67~80であるイントラ予測モードが広角イントラ予測モードを示す。

【0345】

図36では、45度より大きい角度を有する14個の広角イントラ予測モード(-1から-14)および-135度より小さい角度を有する14個の広角イントラ予測モード(67から80)が図示されたが、これよりも大きいまたは少ない数の広角イントラ予測モードを定義することができる。

40

【0346】

広角イントラ予測モードを使用する場合、上部参照サンプルの長さを $2W+1$ に設定し、左側参照サンプルの長さを $2H+1$ に設定することができる。

【0347】

広角イントラ予測モードを使用する場合、参照サンプルTを使用して、図35(a)に示されるサンプルAを予測し、参照サンプルLを使用して、図35(b)に示されるサンプルAを予測することができる。

【0348】

既存のイントラ予測モードと、N個の広角イントラ予測モードを加え、合計 $67+N$ 個のイントラ予測モードを使用することができる。例えば、表6は、20個の広角イントラ

50

予測モードを定義した場合、イントラ予測モードのイントラ方向パラメータを示す。

【表 6】

PredModeIntra	-10	-9	-8	-7	-6	-5	-4	-3	-2
intraPredAngle	114	93	79	68	60	54	49	45	39
PredModeIntra	-1	2	3	4	5	6	7	8	9
intraPredAngle	35	32	29	26	23	21	19	17	15
PredModeIntra	10	11	12	13	14	15	16	17	18
intraPredAngle	13	11	9	7	5	3	2	1	0
PredModeIntra	19	20	21	22	23	24	25	26	27
intraPredAngle	-1	-2	-3	-5	-7	-9	-11	-13	-15
PredModeIntra	28	29	30	31	32	33	34	35	36
intraPredAngle	-17	-19	-21	-23	-26	-29	-32	-29	-26
PredModeIntra	37	38	39	40	41	42	43	44	45
intraPredAngle	-23	-21	-19	-17	-15	-13	-11	-9	-7
PredModeIntra	46	47	48	49	50	51	52	53	54
intraPredAngle	-5	-3	-2	-1	0	1	2	3	5
PredModeIntra	55	56	57	58	59	60	61	62	63
intraPredAngle	7	9	11	13	15	17	19	21	23
PredModeIntra	64	65	66	67	68	69	70	71	72
intraPredAngle	26	29	32	35	39	45	49	54	60
PredModeIntra	73	74	75	76					
intraPredAngle	68	79	93	114					

【0349】

イントラ方向パラメータは、現在のブロックのサイズ、形状、または参照サンプルラインのうちの少なくとも1つに基づいて異なるように設定することができる。例えば、現在のブロックが正方形である場合と、現在のブロックが非正方形である場合において、特定のイントラ予測モードのイントラ方向パラメータが異なることができる。例えば、イントラ予測モード15のイントラ方向パラメータ `intraPredAngle` は、現在のブロックが非正方形である場合よりも、現在のブロックが正方形である場合の方が、より大きな値を有することができる。

【0350】

あるいは、イントラ予測モード75のイントラ方向パラメータ `intraPredAngle` は、現在のブロックの参照サンプルラインのインデックスが0である場合よりも、現在のブロックの参照サンプルラインのインデックスが1以上である場合の方が、より大

10

20

30

40

50

きな値を有することができる。

【0351】

現在のブロックが非正方形であり、ステップ S 2 9 0 2 で取得された現在のブロックのイントラ予測モードが変換範囲に属する場合、現在のブロックのイントラ予測モードを広角イントラ予測モードに変換することができる。現在のブロックのサイズ、形状、または比率のうちの少なくとも 1 つに基づいて、前記変換範囲を決定することができる。ここで、前記比率は、現在のブロックの幅と高さの間の比率を示すことができる。

【0352】

現在のブロックが幅が高さより大きい非正方形である場合、変換範囲を、右上対角方向のイントラ予測モードインデックス（例えば、66）から（右上対角方向であるイントラ予測モードのインデックス - N）に設定することができる。ここで、N は現在のブロックの比率に基づいて決定できる。現在のブロックのイントラ予測モードが変換範囲に属する場合、前記イントラ予測モードを広角イントラ予測モードに変換することができる。前記イントラ予測モードから事前定義された値を差し引くことによって前記変換を実行でき、前記事前定義された値は、広角イントラ予測モードを除くイントラ予測モードの総数（例えば、67）であり得る。

10

【0353】

前記実施例に基づき、66番から53番の間のイントラ予測モードを、それぞれ - 1番から - 14番の間の広角イントラ予測モードに変換することができる。

【0354】

20

現在のブロックが、高さが幅より大きい非正方形である場合、変換範囲を、左下対角方向のイントラ予測モードインデックス（例えば、2）から（左下対角方向のイントラ予測モードのインデックス + M）に設定することができる。ここで、M は、現在のブロックの比率に基づいて決定できる。現在のブロックのイントラ予測モードが変換範囲に属する場合、前記イントラ予測モードを広角イントラ予測モードに変換することができる。前記イントラ予測モードに事前定義された値を追加することによって前記変換を実行でき、事前定義された値は、広角イントラ予測モードを除く方向性イントラ予測モードの総数（例えば、65）であり得る。

【0355】

前記実施例に基づき、2番から15番の間のイントラ予測モードを、それぞれ67番から80番の間の広角イントラ予測モードに変換することができる。

30

【0356】

以下、変換範囲に属するイントラ予測モードを広角イントラ代替予測モードと呼ぶ。

【0357】

変換範囲は、現在のブロックの比率に基づいて決定できる。例えば、表7および表8はそれぞれ、広角イントラ予測モードを除く35個のイントラ予測モードが定義された場合と67個のイントラ予測モードが定義された場合の変換範囲を示す。

【表7】

条件	置き換えられたイントラ予測モード
$W/H = 2$	モード 2、3、4
$W/H > 2$	モード 2、3、4、5、6
$W/H = 1$	なし
$H/W = 1/2$	モード 32、33、34
$H/W < 1/2$	モード 30、31、32、33、34

40

50

【表 8】

条件	置き換えられたイントラ予測モード
$W/H = 2$	モード 2、3、4、5、6、7
$W/H > 2$	モード 2、3、4、5、6、7、8、9、10、11
$W/H = 1$	なし
$H/W = 1/2$	モード 61、62、63、64、65、66
$H/W < 1/2$	モード 57、58、59、60、61、62、63、64、65、66

10

【0358】

表7および表8に示されるように、現在のブロックの比率に応じて、変換範囲に含まれる広角イントラ代替予測モードの数が異なることができる。

20

【0359】

既存のイントラ予測モードに加えて、広角イントラ予測モードを使用することにより、広角イントラ予測モードを符号化するのに必要なリソースが増加するため、符号化効率が低下する可能性がある。これにより、広角イントラ予測モードをそのまま符号化する代わりに、広角イントラ予測モードに関する代替イントラ予測モードを符号化することにより、符号化効率を向上させることができる。

【0360】

例えば、67番の広角イントラ予測モードを使用して現在のブロックを符号化する場合、67番の広角代替イントラ予測モードである2番を、現在のブロックのイントラ予測モードで符号化することができる。さらに、-1番の広角イントラ予測モードを使用して、現在のブロックを符号化する場合、-1番の広角代替イントラ予測モードである66番を、現在のブロックのイントラ予測モードで符号化することができる。

30

【0361】

復号化器は、現在のブロックのイントラ予測モードを復号化し、復号化されたイントラ予測モードが変換範囲に含まれるか否かを決定することができる。復号化されたイントラ予測モードが広角代替イントラ予測モードである場合、イントラ予測モードを広角イントラ予測モードに変換することができる。

【0362】

あるいは、広角イントラ予測モードで現在のブロックを符号化する場合、広角イントラ予測モードを直接符号化することもできる。

40

【0363】

イントラ予測モードの符号化は、前記MPMリストに基づいて実現されることがある。具体的には、以広角イントラ予測モードで隣接ブロックを符号化する場合、広角イントラ予測モードに対応する広角代替イントラ予測モードに基づいてMPMを設定することができる。例えば、広角イントラ予測モードで隣接ブロックを符号化する場合、変数 `cand_IntraPredX` (`X`はAまたはBである) を広角代替イントラ予測モードとして設定することができる。

【0364】

イントラ予測を実行した結果として予測ブロックが生成されると、予測ブロックに含まれる各予測サンプルの位置に基づき予測サンプルを更新することができる。上記のような

50

更新方法をサンプル位置に基づくイントラ加重予測方法（または位置依存の予測の組み合わせ（P D P C：Position Dependent Prediction Combination））と呼ぶことができる。

【0365】

現在のブロックのイントラ予測モード、現在のブロックの参照サンプルライン、現在のブロックのサイズ、またはカラー成分を考慮して、P D P Cを使用するか否かを決定することができる。例えば、現在のブロックのイントラ予測モードがP l a n a rモード、D Cモード、垂直方向モード、水平方向モード、垂直方向よりもインデックス値が小さいモード、または水平方向よりもインデックス値が大きいモードのうちの少なくとも1つである場合、P D P Cを使用することができる。あるいは、現在のブロックの幅または高さのうちの少なくとも1つが4を超える場合にのみ、P D P Cを使用することができる。あるいは、現在のブロックの参照画像ラインのインデックスが0である場合にのみ、P D P Cを使用することができる。あるいは、現在のブロックの参照画像ラインのインデックスが事前定義された値より大きいか等しい場合にのみ、P D P Cを使用することができる。あるいは、輝度成分に対してのみ、P D P Cを使用することができる。あるいは、前記列挙された条件のうちの2つ以上を満たすか否かに応じて、P D P Cを使用するか否かを決定することができる。

10

【0366】

別の例として、P D P Cが適用されるか否かを示す情報が、ビットストリームを介してシグナリングできる。

20

【0367】

イントラ予測サンプルを介して予測サンプルを取得する場合、取得された予測サンプルの位置に基づいて、前記予測サンプルを補正するために使用される参照サンプルを決定できる。説明の便宜のために、後述される実施例では、予測サンプルを補正するために使用される参照サンプルをP D P C参照サンプルと呼ぶ。さらに、イントラ予測を介して取得された予測サンプルを第1予測サンプルと呼び、第1予測サンプルを補正して取得された予測サンプルを第2予測サンプルと呼ぶ。

【0368】

図37は、P D P Cの適用を示す図である。

【0369】

少なくとも1つのP D P C参照サンプルを使用して、第1予測サンプルを補正することができる。P D P C参照サンプルは、現在のブロックの左上隅に隣接する参照サンプル、現在のブロックの上部に位置する上部参照サンプル、または現在のブロック左側に位置する左側参照サンプルのうちの少なくとも1つを含み得る。

30

【0370】

現在のブロックの参照サンプルラインに属する参照サンプルのうちの少なくとも1つをP D P C参照サンプルとして設定することができる。あるいは、現在のブロックの参照サンプルラインとは関係なく、インデックス0の参照サンプルラインに属する参照サンプルのうちの少なくとも1つをP D P C参照サンプルとして設定することができる。例えば、インデックス1またはインデックス2の参照サンプルラインに含まれる参照サンプルを使用して第1予測サンプルを取得した場合でも、インデックス0の参照サンプルラインに含まれる参照サンプルを使用して第2予測サンプルを取得することができる。

40

【0371】

現在のブロックのイントラ予測モード、現在のブロックのサイズ、現在のブロックの形状、または第1予測サンプルの位置のうちの少なくとも1つを考慮して、第1予測サンプルを補正するために使用されるP D P C参照サンプルの数または位置を決定することができる。

【0372】

例えば、現在のブロックのイントラ予測モードがP l a n a rモードまたはD Cモードである場合、上部参照サンプルおよび左側参照サンプルを使用して、第2予測サンプルを

50

取得することができる。この場合、上部参照サンプルは、第1予測サンプルに垂直な参照サンプル（例えば、 x 座標が同じ参照サンプル）であり得、左側参照サンプルは、第1予測サンプルに水平な参照サンプル（例えば、 y 座標が同じ参照サンプル）であり得る。

【0373】

現在のブロックのイントラ予測モードが水平イントラ予測モードである場合、上部参照サンプルを使用して、第2予測サンプルを取得することができる。この場合、上部参照サンプルは、第1予測サンプルに垂直な参照サンプルであり得る。

【0374】

現在のブロックのイントラ予測モードが垂直イントラ予測モードである場合、左側参照サンプルを使用して、第2予測サンプルを取得することができる。この場合、左側参照サンプルは、第1予測サンプルに水平な参照サンプルであり得る。

10

【0375】

現在のブロックのイントラ予測モードが左下対角方向イントラ予測モードまたは右上対角方向のイントラ予測モードである場合、左上参照サンプル、上部参照サンプル、および左側参照サンプルに基づいて、第2予測サンプルを取得することができる。左上参照サンプルは、現在のブロックの左上隅に隣接する参照サンプル（例えば、 $(-1, -1)$ 位置の参照サンプル）であり得る。上部参照サンプルは、第1予測サンプルの右上対角方向に位置する参照サンプルであり得、左側参照サンプルは、第1予測サンプルの左下対角方向に位置する参照サンプルであり得る。

【0376】

要約すると、第1予測サンプルの位置が (x, y) である場合、 $R(-1, -1)$ を左上参照サンプルとして設定し、 $R(x + y + 1, -1)$ または $R(x, -1)$ を上部参照サンプルとして設定することができる。また、 $R(-1, x + y + 1)$ または $R(-1, y)$ を左側参照サンプルとして設定することができる。

20

【0377】

別の例として、現在のブロックの形状、または広角イントラモードが適用されたか否かのうちの少なくとも1を考慮して、左側参照サンプルまたは上部参照サンプルの位置を決定することができる。

【0378】

具体的には、現在のブロックのイントラ予測モードが広角イントラ予測モードである場合、第1予測サンプルの対角方向に位置する参照サンプルから、オフセット値程度離れた参照サンプルをPDC参照サンプルとして設定することができる。例えば、上部参照サンプル $R(x + y + k + 1, -1)$ と左側参照サンプル $R(-1, x + y - k + 1)$ をPDC参照サンプルとして設定することができる。

30

【0379】

この場合、広角イントラ予測モードに基づいてオフセット値 k を決定することができる。式6と式7は、広角イントラ予測モードに基づいてオフセット値を導出する例を示す。

【数6】

[式6]

40

$$\begin{aligned} k &= \text{CurrIntraMode} - 66 \\ \text{if } (\text{CurrIntraMode} > 66) \end{aligned}$$

【数7】

50

[式 7]

```

k=-CurrIntraMode
if(CurrIntraMode < 0)

```

【0 3 8 0】

第 1 予測サンプルと P D P C 参照サンプルとの間の加重合演算に基づいて、第 2 予測サンプルを決定することができる。例えば、下記式 8 に基づいて第 2 予測サンプルを取得することができる。

【数 8】

[式 8]

$$pred(x,y) = (xL * R_L + wT * R_T - wTL * R_{TL} + (64 - wL - wT + wTL) * pred(x,y) + 32) >> 6$$

【0 3 8 1】

式 8において、R L は左側参照サンプルを表し、R T は上部参照サンプルを表し、R TL は左上参照サンプルを表す。pred(x, y) は (x, y) 位置の予測サンプルを表す。w L は左側参照サンプルに割り当てられる重み値を表し、w T は上部参照サンプルに割り当てられる重み値を表し、w TL は左上参照サンプルに割り当てられる重み値を表す。第 1 予測サンプルに割り当てられる重み値は、最大値から参照サンプルに割り当てられる重み値を差し引くことによって導出することができる。説明の便宜のために、P D P C 参照サンプルに割り当てられる重み値を P D P C 重み値と呼ぶ。

【0 3 8 2】

現在のブロックのイントラ予測モードまたは第 1 予測サンプルの位置のうちの少なくとも 1 つに基づいて、各参照サンプルに割り当てられる重み値を決定することができる。

【0 3 8 3】

例えば、w L、w T または w TL のうちの少なくとも 1 つと、予測サンプルの x 軸座標値または y 軸座標値のうちの少なくとも 1 つは、比例関係または反比例関係にあり得る。あるいは、w L、w T または w TL のうちの少なくとも 1 つと、現在のブロックの幅または高さのうちの少なくとも 1 つは、比例関係または反比例関係にあり得る。

【0 3 8 4】

現在のブロックのイントラ予測モードが DC モードである場合、P D P C 重み値は、下記式 9 に示されるように決定できる。

【数 9】

[式 9]

```

wT=32>> ((y<<1) >> shift)
wL=32 >> ((x<<1) >> shift)
wTL=(wL>>4) + (wT>>4)

```

【0 3 8 5】

式 9において、x および y は、第 1 予測サンプルの位置を表す。

【0 3 8 6】

10

20

30

40

50

式 9において、現在のブロックの幅または高さに基づいて、ビットシフト演算に使用される変数 `s h i f t` を決定できる。例えば、下記式 10 または式 11 に基づいて変数 `s h i f t` を導出することができる。

【数 1 0】

[式 1 0]

$$\text{shift} = (\log_2(\text{width}) - 2 + \log_2(\text{height}) - 2 + 2) \gg 2$$

10

【数 1 1】

[式 1 1]

$$\text{shift} = ((\text{Log}_2(nTbW) + \text{Log}_2(nTbH) - 2) \gg 2)$$

【0 3 8 7】

あるいは、現在のブロックのイントラ方向パラメータを考慮して、変数 `s h i f t` を導出することができる。

20

【0 3 8 8】

変数 `s h i f t` を導出するために使用されるパラメータの数や種類は、現在のブロックのイントラ予測モードに応じて異なるように決定することができる。例えば、現在のブロックのイントラ予測モードが `Planear` モード、`DC` モード、垂直方向モード、または水平方向モードである場合、式 10 または式 11 に示されるように、現在のブロックの幅と高さを使用して変数 `s h i f t` を導出することができる。現在のブロックのイントラ予測モードが、垂直方向のイントラ予測モードよりも大きいインデックスを有するイントラ予測モードである場合、現在のブロックの高さとイントラ方向パラメータを使用して、変数 `s h i f t` を導出することができる。現在のブロックのイントラ予測モードが、水平方向のイントラ予測モードよりも小さいインデックスを有するイントラ予測モードである場合、現在のブロックの幅とイントラ方向パラメータを使用して、変数 `s h i f t` を導出することができる。

30

【0 3 8 9】

現在のブロックのイントラ予測モードが `Planear` モードである場合、`wT L` の値を 0 に設定することができる。下記式 12 に基づいて `wL` と `wT` を導出することができる。

【数 1 2】

[式 1 2]

$$\begin{aligned} wT[y] &= 32 \gg ((y \ll 1) \gg nScale) \\ wL[x] &= 32 \gg ((x \ll 1) \gg nScale) \end{aligned}$$

40

【0 3 9 0】

現在のブロックのイントラ予測モードが水平イントラ予測モードである場合、`wT` を 0 に設定し、`wT L` と `wL` を同じに設定することができる。一方、現在のブロックのイントラ予測モードが垂直イントラ予測モードである場合、`wL` を 0 に設定し、`wT L` と `wT` を同じに設定することができる。

【0 3 9 1】

現在のブロックのイントラ予測モードが、垂直方向のイントラ予測モードより大きいイ

50

ンデックス値を有する、右上方向に向かうイントラ予測モードである場合、下記式 1 3 のように、P D P C 重み値を導出することができる。

【数 1 3】

[式 1 3]

$$\begin{aligned} wT &= 16 >> ((y << 1) >> shift) \\ wL &= 16 >> ((x << 1) >> shift) \\ wTL &= 0 \end{aligned}$$

10

【0 3 9 2】

一方、現在のブロックのイントラ予測モードが、水平方向のイントラ予測モードより小さいインデックス値を有する、左下方向に向かうイントラ予測モードである場合、下記式 1 4 のように、P D P C 重み値を導出することができる。

【数 1 4】

[式 1 4]

$$\begin{aligned} wT16 &>> ((y << 1) >> shift) \\ wL &= 16 >> ((x << 1) >> shift) \\ wTL &= 0 \end{aligned}$$

20

【0 3 9 3】

上述した実施例のように、予測サンプルの位置 x と y に基づいて、P D P C 重み値を決定することができる。

【0 3 9 4】

別の例として、サブブロック単位で各 P D P C 参照サンプルに割り当てる重み値を決定することもできる。サブブロックに含まれる予測サンプルは、同じ P D P C 重み値を共有することができる。

30

【0 3 9 5】

符号化器と復号化器で、重み値を決定するために使用される基本単位であるサブブロックのサイズを事前定義することができる。例えば、 2×2 サイズまたは 4×4 サイズの各サブブロックについて、重み値を決定することができる。

【0 3 9 6】

あるいは、現在のブロックのサイズまたは形状に応じて、サブブロックのサイズ、形状、または数を決定することができる。例えば、符号化ブロックのサイズとは関係なく、符号化ブロックを 4 つのサブブロックに分割することができる。あるいは、符号化ブロックのサイズに基づいて、符号化ブロックを 4 つまたは 16 つのサブブロックに分割することができます。

40

【0 3 9 7】

あるいは、現在のブロックのイントラ予測モードに基づいて、サブブロックのサイズ、形状、または数を決定することもできる。例えば、現在のブロックのイントラ予測モードが水平方向モードである場合、N 個の列（または N 個の行）を 1 つのサブブロックとして設定し、逆に、現在のブロックのイントラ予測モードが垂直方向モードである場合、N 個の行（または N 行の列）を 1 つのサブブロックとして設定することができます。

【0 3 9 8】

式 1 5 ~ 1 7 は、 2×2 サイズのサブブロックの P D P C 重み値を決定する例を示す。

50

式 1 5 は、現在のブロックのイントラ予測モードが DC モードである場合の例を示す。

【数 1 5】

[式 1 5]

$$\begin{aligned} wT &= 32 >> (((y << \log_2 K) >> \log_2 K) << 1) >> shift \\ wL &= 32 >> (((x << \log_2 K) >> \log_2 K) << 1) >> shift \\ wTL &= (wL >> 4) + (wT >> 4) \end{aligned}$$

10

【0 3 9 9】

式 1 5 において、K は、サブブロックのサイズまたはイントラ予測モードに基づいて決定される値であり得る。

【0 4 0 0】

式 1 6 は、現在のブロックのイントラ予測モードが、垂直方向のイントラ予測モードより大きいインデックス値を有する、右上方向に向かうイントラ予測モードである場合の例を示す。

【数 1 6】

[式 1 6]

20

$$\begin{aligned} wT &= 16 >> (((y << \log_2 K) >> \log_2 K) << 1) >> shift \\ wL &= 16 >> (((x << \log_2 K) >> \log_2 K) << 1) >> shift \\ wTL &= 0 \end{aligned}$$

【0 4 0 1】

式 1 7 は、現在のブロックのイントラ予測モードが、水平方向のイントラ予測モードより小さいインデックス値を有する左下方向に向かうイントラ予測モードである場合の例を示す。

30

【数 1 7】

[式 1 7]

$$\begin{aligned} wT &= 16 >> (((y << \log_2 K) >> \log_2 K) << 1) >> shift \\ wL &= 16 >> (((x << \log_2 K) >> \log_2 K) << 1) >> shift \\ wTL &= 0 \end{aligned}$$

【0 4 0 2】

式 1 5 ~ 1 7 において、x 及び y は、サブブロック内の参照サンプルの位置を示す。参照サンプルは、サブブロックの左上に位置するサンプル、サブブロックの中央に位置するサンプル、またはサブブロックの右下に位置するサンプルのいずれかであり得る。

40

【0 4 0 3】

式 1 8 ~ 2 0 は、4 × 4 サイズのサブブロックの P D P C 重み値を決定する例を示す。式 1 8 は、現在のブロックのイントラ予測モードが DC モードである場合の例を示す。

【数 1 8】

50

[式 1 8]

$$\begin{aligned} wT &= 32 >> (((y << 2) >> 2) << 1) >> shift \\ wL &= 32 >> (((x << 2) >> 2) << 1) >> shift \\ wTL &= (wL >> 4) + (wT >> 4) \end{aligned}$$

【0 4 0 4】

式 1 9 は、現在のブロックのイントラ予測モードが、垂直方向のイントラ予測モードより大きいインデックス値を有する、右上方向に向かうイントラ予測モードである場合の例を示す。

【数 1 9】

[式 1 9]

$$\begin{aligned} wT &= 16 >> (((y << 2) >> 2) << 1) >> shift \\ wL &= 16 >> (((x << 2) >> 2) << 1) >> shift \\ wTL &= 0 \end{aligned}$$

10

20

【0 4 0 5】

式 2 0 は、現在のブロックのイントラ予測モードが、水平方向のイントラ予測モードより小さいインデックス値を有する左下方向に向かうイントラ予測モードである場合の例を示す。

【数 2 0】

[式 2 0]

$$\begin{aligned} wT &= 16 >> (((y << 2) >> 2) << 1) >> shift \\ wL &= 16 >> (((x << 2) >> 2) << 1) >> shift \\ wTL &= 0 \end{aligned}$$

30

【0 4 0 6】

上述した実施例では、第 1 予測サンプルまたはサブブロックに含まれる予測サンプルの位置を考慮して、P D P C 重み値を決定することについて説明した。現在のブロックの形状をさらに考慮して、P D P C 重み値を決定することもできる。

【0 4 0 7】

例えば、D C モードの場合、現在のブロックが幅が高さより大きい非正方形であるか、または高さが幅により大きい非正方形であるかに応じて、P D P C 重み値の導出方法が異なることができる。

【0 4 0 8】

式 2 1 は、現在のブロックが幅が高さより大きい非正方形である場合、P D P C 重み値を導出する例を示し、式 2 2 は、現在のブロックが、高さが幅より大きい非正方形の場合、P D P C 重み値を導出する例を示す。

【数 2 1】

40

50

[式 2 1]

$$\begin{aligned} wT &= 32 >> ((y << 1) >> shift) \\ wL &= 32 >> (x >> shift) \\ wTL &= (wL >> 4) + (wT >> 4) \end{aligned}$$

【数 2 2】

$$\begin{aligned} wT &>> (y >> shift) \\ wL &= 32 >> ((x << 1) >> shift) \\ wTL &= (xL >> 4) + (wT >> 4) \end{aligned}$$

【0 4 0 9】

現在のブロックが非正方形である場合、広角イントラ予測モードを使用して、現在のブロ
ックを予測することができる。このように、広角イントラ予測モードが適用される場合
にも、P D P Cを適用して、第1予測サンプルを更新することができる。

【0 4 1 0】

現在のブロックに広角イントラ予測が適用される場合、符号化ブロックの形状を考慮し
て、P D P C重み値を決定することができる。

【0 4 1 1】

例えば、現在のブロックが幅が高さより大きい非正方形である場合、第1予測サンプル
の位置に基づいて、第1予測サンプルの左下に位置する左側参照サンプルよりも、第1予
測サンプルの右上に位置する上部参照サンプルの方が、第1予測サンプルにより近い場
合が発生する。これにより、第1予測サンプルを補正する場合、上部参照サンプルに適用さ
れる重み値を、左側参照サンプルに適用される重み値よりも大きな値を有するように設
定することができる。

【0 4 1 2】

一方、現在のブロックが、高さが幅より大きい非正方形である場合、第1予測サンプル
の位置に基づいて、第1予測サンプルの右上に位置する上部参照サンプルによりも、第1予
測サンプルの左下に位置する左側参照サンプルの方が、第1予測サンプルにより近い場
合が発生する。これにより、第1予測サンプルを補正する場合、左側参照サンプルに適用さ
れる重み値を、上部参照サンプルに適用される重み値よりも大きな値を有するように設
定することができる。

【0 4 1 3】

式2 3は、現在のブロックのイントラ予測モードが、インデックスが6 6より大きい広
角イントラ予測モードである場合、P D P C重み値を導出する例を示す。

【数 2 3】

$$\begin{aligned} wT &= 16 >> (y >> shift) \\ wL &= 16 >> ((x << 1) >> shift) \\ wTL &= 0 \end{aligned}$$

10

20

30

40

50

【 0 4 1 4 】

式 2 4 は、現在のブロックのイントラ予測モードが、インデックスが 0 より小さい広角イントラ予測モードである場合、P D P C 重み値を導出する例を示す。

【 数 2 4 】

[式 2 4]

$$wT = 16 >> ((y << 1) >> shift)$$

$$wL = 16 >> (x >> shift)$$

$$wTL = 0$$

10

【 0 4 1 5 】

現在のブロックの比率に基づいて P D P C 重み値を決定することもできる。現在のブロックの比率は、現在のブロックの幅と高さの比率を示し、下記式 2 5 のように定義することができる。

【 数 2 5 】

[式 2 5]

$$whRatio = CUwidth / CUheight$$

20

【 0 4 1 6 】

現在のブロックのイントラ予測モードに基づいて、P D P C 重み値を導出する方法を可変的に決定することができる。

【 0 4 1 7 】

例えば、式 2 6 および式 2 7 は、現在のブロックのイントラ予測モードが D C モードである場合、P D P C 重み値を導出する例を示す。具体的には、式 2 6 は、現在のブロックが幅が高さより大きい非正方形である場合の例であり、式 2 7 は、現在のブロックが、高さが幅よりも大きい非正方形である場合の例である。

30

【 数 2 6 】

[式 2 6]

$$wT = 32 >> ((y << 1) >> shift)$$

$$wL = 32 >> (((x << 1) >> whRatio) >> shift)$$

$$wTL = (wL >> 4) + (wT >> 4)$$

40

【 数 2 7 】

[式 2 7]

$$wT = 32 >> (((y << 1) >> 1 / whRatio) >> shift)$$

$$wL = 32 >> ((x << 1) >> shift)$$

$$wTL = (wL >> 4) + (wT >> 4)$$

【 0 4 1 8 】

50

式28は、現在のブロックのイントラ予測モードが、インデックスが66より大きい広角イントラ予測モードである場合、PDP C重み値を導出する例を示す。

【数28】

[式28]

```
wT = 16 >> (((y<<1) >> 1/whRatio) >> shift)
wL = 16 >> ((x<<1) >> shift)
wTL=0
```

10

【0419】

式29は、現在のブロックのイントラ予測モードが、インデックスが0より小さい広角イントラ予測モードである場合、PDP C重み値を導出する例を示す。

【数29】

[式29]

```
wT = 16 >> ((y<<1) >> shift)
wL = 16 >> (((x<<1) >> whRatio) >> shift)
wTL=0
```

20

【0420】

元画像から予測画像を差分して、導出された残差画像を導出することができる。この場合、残差画像を周波数領域にする場合、周波数成分のうちの高周波数成分を除去しても、ビデオの品質を大幅に低下させることはない。これにより、高周波数成分の値小さく変更し、高周波数成分の値を0に設定すると、明らかな視覚的歪みを発生させることなく、圧縮効率を向上させることができる効果がある。上記の特性を反映して、現在のブロックを変換して、残差画像を2次元周波数成分に分解することができる。前記変換は、離散余弦変換(DCT: Discrete Cosine Transform)または離散正弦変換(DST: Discrete Sine Transform)などの変換方法を使用して行うことができる。

30

【0421】

DCTは余弦変換を使用して、残差画像を2次元周波数成分に分解(または変換)するものであり、DSTは使用正弦変換を使用して、残差画像を2次元周波数成分に分解(または変換)するものである。残差画像の変換結果、周波数成分は基本画像で表現することができる。例えば、N×Nサイズのブロックに対してDCT変換を実行する場合、N2個の基本パターン成分を取得することができる。変換により、N×Nサイズのブロックに含まれる各基本パターン成分のサイズを取得することができる。使用された変換技術に基づいて、基本パターン成分のサイズをDCT係数またはDST係数と呼ぶことができる。

40

【0422】

変換技術DCTは、主に、0以外の低周波数成分が多く分布された画像を変換するためを使用される。変換技術DSTは、主に、高周波数成分が多く分布された画像に使用される。

【0423】

DCTまたはDST以外の変換技術を使用して、残差画像を変換することもできる。

【0424】

以下、残差画像を2次元周波数成分に変換することを2次元画像変換と呼ぶ。さらに、変換結果によって得られた基本パターン成分のサイズを変換係数と呼ぶ。例えば、変換係

50

数は、DCT係数またはDST係数を指し得る。後述される一次変換と二次変換を同時に適用する場合、変換係数は、二次変換の結果によって生成される基本パターン成分のサイズを表すことができる。

【0425】

ブロック単位で変換技術を決定することができる。現在のブロックの予測符号化モード、現在のブロックのサイズ、または現在のブロックの形状のうちの少なくとも1つに基づいて、変換技術を決定することができる。例えば、イントラ予測モードで現在のブロックを符号化し、現在のブロックのサイズが $N \times N$ よりも小さい場合、変換技術DSTを使用して変換を実行できる。一方、上記の条件を満たしない場合、変換技術DCTを使用して変換を実行できる。

10

【0426】

残差画像において、一部のブロックに対して2次元画像変換を実行しないこともできる。2次元画像変換を実行しないことを変換スキップ(Transform Skip)と呼ぶことができる。変換スキップが適用される場合、変換が実行されていない残差値に量子化を適用することができる。

【0427】

DCTまたはDSTを使用して、現在のブロックを変換した後、変換された現在のブロックを再変換することができる。この場合、DCTまたはDSTに基づく変換を一次変換として定義でき、一次変換が適用されたブロックを再変換することを二次変換として定義できる。

20

【0428】

一次変換は、複数の変換コア候補のいずれかを使用して実行することができる。例えば、DCT2、DCT8、またはDCT7のいずれかを使用して、一次変換を実行することができる。

【0429】

水平方向と垂直方向について、異なる変換コアを使用することもできる。水平方向の変換コアと垂直方向の変換コアの組み合わせを示す情報が、ビットストリームを介してシグナリングされることもできる。

【0430】

一次変換および二次変換の実行単位が異なる場合がある。例えば、 8×8 ブロックに対して一次変換を実行し、変換された 8×8 ブロックのうち 4×4 サイズのサブブロックに対して二次変換を実行できる。この場合、二次変換が実行されない限りの領域の変換係数を0に設定することもできる。

30

【0431】

あるいは、 4×4 ブロックに対して一次変換を実行し、変換された 4×4 ブロックを含む 8×8 サイズの領域に対して二次変換を実行することもできる。

【0432】

二次変換を実行するか否かを示す情報が、ビットストリームを介してシグナリングできる。

【0433】

あるいは、水平方向変換コアと垂直方向変換コアが同じであるか否かに基づいて、二次変換を実行する否かを決定することができる。例えば、水平方向変換コアと垂直方向変換コアが同一である場合にのみ、二次変換を実行することができる。あるいは、水平方向変換コアと垂直方向変換コアが異なる場合にのみ、二次変換を実行することができる。

40

【0434】

あるいは、水平方向の変換と垂直方向の変換が事前定義された変換コアを使用する場合にのみ、二次変換が可能になることができる。例えば、水平方向の変換と垂直方向の変換にDCT2変換コアが使用される場合に、二次変換が可能になることができる。

【0435】

あるいは、現在のブロックの非ゼロ変換係数の数に基づいて、二次変換を実行するか否

50

かを決定できる。例えば、現在のブロックの非ゼロ変換係数が閾値より小さいか等しい場合、二次変換を使用しないように設定し、現在のブロックの非ゼロ変換係数が閾値を超える場合、二次変換を使用するように設定することができる。現在のブロックがイントラ予測で符号化される場合にのみ、二次変換を使用するように設定することもできる。

【0436】

現在のブロックの形状に基づいて、二次変換を実行するサブブロックのサイズや形状を決定することができる。

【0437】

図38および図39は、二次変換が実行されるサブブロックを示す図である。

【0438】

現在のブロックが正方形である場合、一次変換を実行した後、現在のブロックの左上の $N \times N$ サイズのサブブロックに対して二次変換を実行することができる。例えば、現在のブロックが 8×8 サイズの符号化ブロックである場合、現在のブロックに対して一次変換を実行した後、現在のブロックの左上の 4×4 サイズのサブブロックに対して二次変換を実行することができる（図38参照）。

【0439】

現在のブロックが幅が高さよりも4倍以上大きい非正方形である場合、一次変換を実行した後、現在のブロックの左上の $(kN) \times (4kN)$ サイズのサブブロックに対して二次変換を実行することができる。例えば、現在のブロックが 16×4 サイズの非正方形である場合、現在のブロックに対して一次変換を実行した後、現在のブロックの左上の 2×8 サイズのサブブロックに対して二次変換を実行することができる（図39(a)参照）。

【0440】

現在のブロックが、高さが幅よりも4倍以上大きい非正方形である場合、一次変換を実行した後、現在のブロックの左上の $(4kN) \times (kN)$ サイズのサブブロックに対して二次変換を実行することができる。例えば、現在のブロックが 16×4 サイズの非正方形である場合、現在のブロックに対して一次変換を実行した後、現在のブロックの左上の 2×8 サイズのサブブロックに対して二次変換を実行することができる（図39(b)参照）。

【0441】

復号化器では、二次変換の逆変換（第2逆変換）を実行し、その実行結果に対して、一次変換の逆変換（第1逆変換）を実行することができる。第2逆変換と第1逆変換の実行結果、現在のブロックの残差信号を取得することができる。

【0442】

量子化は、ブロックのエネルギーを低減するために使用され、量子化プロセスは、変換係数を特定の定数で除算するプロセスを含む。前記定数は、量子化パラメータによって導出でき、量子化パラメータは、1～63の値として定義できる。

【0443】

符号化器で変換と量子化を実行する場合、復号化器は、逆量子化と逆変換を介して残差ブロックを取得することができる。復号化器では、予測ブロックと残差ブロックを加算して、現在のブロックの復元ブロックを取得することができる。

【0444】

現在のブロックの変換タイプを示す情報が、ビットストリームを介してシグナルリングできる。前記情報は、水平方向の変換タイプと垂直方向の変換タイプの組み合わせのうちの1つを示すインデックス情報 $t u_m t s_i d_x$ であり得る。

【0445】

インデックス情報 $t u_m t s_i d_x$ によって指定された変換タイプの候補に基づいて、垂直方向の変換コアと水平方向の変換コアを決定することができる。表9および表10は、 $t u_m t s_i d_x$ による変換タイプの組み合わせを示す。

10

20

30

40

50

【表 9】

t u_m t s_i d x	変換タイプ	
	水平	垂直
0	スキップ	スキップ
1	D C T - I I	D C T - I I
2	D S T - V I I	D S T - V I I
3	D C T - V I I I	D S T - V I I
4	D S T - V I I I	D C T - V I I I
5	D C T - V I I I	D C T - V I I I

10

【表 10】

t u_m t s_i d x	変換タイプ	
	水平	垂直
0	D C T - I I	D C T - I I
1	スキップ	スキップ
2	D S T - V I I	D S T - V I I
3	D C T - V I I I	D S T - V I I
4	D S T - V I I I	D C T - V I I I
5	D C T - V I I I	D C T - V I I I

20

30

【0446】

変換タイプは、D C T 2、D S T 7、D C T 8、または変換スキップのいずれかとして決定できる。あるいは、変換スキップを除き、変換コアのみを使用して、変換タイプの組み合わせの候補を構成することもできる。

【0447】

表9を使用する場合、t u_m t s_i d xが0である場合、水平方向と垂直方向に変換スキップを適用することができる。t u_m t s_i d xが1である場合、水平方向と垂直方向にD C T 2を適用することができる。t u_m t s_i d xが3である場合、水平方向にD C T 8を適用し、垂直方向にD C T 7を適用することができる。

40

【0448】

表10を使用する場合、t u_m t s_i d xが0である場合、水平方向と垂直方向にD C T 2を適用することができる。t u_m t s_i d xが1である場合、水平方向と垂直方向に変換スキップを適用することができる。若t u_m t s_i d xが3である場合、水平方向にD C T 8を適用し、垂直方向にD C T 7を適用することができる。

【0449】

現在のブロックのサイズ、形状、または非ゼロ係数の数のうちの少なくとも1つに基づ

50

いて、インデックス情報を符号化するか否かを決定することができる。例えば、非ゼロ係数の数が閾値より小さいか等しい場合、インデックス情報をシグナーリングせずに、デフォルトの変換タイプを現在のブロックに適用することができる。ここで、デフォルトの変換タイプは D S T 7 であり得る。あるいは、現在のブロックのサイズ、形状、またはイントラ予測モードに応じて、デフォルトモードが異なることができる。

【 0 4 5 0 】

現在のブロックのサイズまたは形状に基づいて閾値を決定することができる。例えば、現在のブロックのサイズが 32×32 より小さいか等しい場合、閾値を 2 に設定し、現在のブロックのサイズが 32×32 を超える場合（例えば、現在のブロックが、 32×64 サイズまたは 64×32 サイズの符号化ブロックである場合）、閾値を 4 に設定することができる。

10

【 0 4 5 1 】

複数のルックアップテーブルを事前に符号化器 / 復号化器に記憶できる。複数のルックアップテーブルにおいて、変換タイプの組み合わせ候補に割り当てられるインデックス値、変換タイプの組み合わせ候補の種類、または変換タイプの組み合わせ候補の数のうちの少なくとも 1 つが異なってもよい。

【 0 4 5 2 】

現在のブロックのサイズ、形状、予測符号化モード、イントラ予測モード、二次変換を適用するか否か、または変換スキップを隣接ブロックに適用するか否かのうちの少なくとも 1 に基づいて、現在のブロックのルックアップテーブルを選択することができる。

20

【 0 4 5 3 】

例えば、現在のブロックのサイズが 4×4 より小さいか等しい場合、またはインター予測で現在のブロックを符号化する場合、表 9 のルックアップテーブルを使用し、現在のブロックのサイズが 4×4 を超える場合、またはブロック内予測で現在のブロックを符号化する場合、表 10 のルックアップテーブルを使用することができる。

【 0 4 5 4 】

あるいは、複数のルックアップテーブルのいずれかを示す情報が、ビットストリームを介してシグナーリングできる。復号化器は、前記情報に基づいて、現在のブロックのルックアップテーブルを選択することができる。

【 0 4 5 5 】

別の例として、現在のブロックのサイズ、形状、予測符号化モード、イントラ予測モード、二次変換を適用するか否か、または変換スキップを隣接ブロックに適用するか否かのうちの少なくとも 1 つに基づいて、変換タイプの組み合わせ候補に割り当てられるインデックスを適応的に決定することができる。例えば、現在のブロックのサイズが 4×4 である場合、変換スキップに割り当てられるインデックス値が、現在のブロックのサイズが 4×4 を超える場合に変換スキップに割り当てられるインデックス値より小さいことができる。具体的には、現在のブロックのサイズが 4×4 である場合、変換スキップにインデックス 0 を割り当てることが可能である。現在のブロックが 4×4 より大きいかつ 16×16 以下である場合、変換スキップに 0 よりも大きいインデックス（例えば、インデックス 1）を割り当てることができる。現在のブロックが 16×16 を超える場合、変換スキップのインデックスに最大値（例えば、5）を割り当てることができる。

30

【 0 4 5 6 】

あるいは、現在のブロックがインター予測で符号化された場合、変換スキップにインデックス 0 を割り当てることができる。現在のブロックがイントラ予測で符号化された場合、変換スキップに 0 よりも大きいインデックス（例えば、インデックス 1）を割り当てることができる。

40

【 0 4 5 7 】

あるいは、現在のブロックがインター予測で符号化された 4×4 サイズのブロックである場合、変換スキップにインデックス 0 を割り当てることができる。一方、現在のブロックがインター予測で符号化されていないか、または現在のブロックが 4×4 を超える場合

50

、変換スキップに 0 より大きいインデックス（例えば、インデックス 1）を割り当てることができる。

【 0 4 5 8 】

表 9 および表 10 に列挙された変換タイプの組み合わせ候補とは異なる変換タイプの組み合わせ候補を定義して使用することもできる。例えば、水平方向変換または垂直方向変換の一方に変換スキップを適用し、D C T 7、D C T 8 またはD S T 2 などの変換コアが適用される変換タイプの組み合わせ候補を他方に使用することができる。この場合、現在のブロックのサイズ（例えば、幅および / または高さ）、形状、予測符号化モード、またはイントラ予測モードのうちの少なくとも 1 つに基づいて、水平方向または垂直方向の変換タイプ候補として変換スキップを使用するか否かを決定することができる。

10

【 0 4 5 9 】

あるいは、特定の変換タイプ候補が使用できるか否かを示す情報が、ビットストリームを介してシグナリングできる。例えば、水平方向と垂直方向の変換スキップを、変換タイプの候補として使用できるか否かを示すフラグがシグナリングされることがある。前記フラグに基づいて、複数の変換タイプの組み合わせ候補に特定の変換タイプの組み合わせ候補が含まれるか否かを決定することができる。

【 0 4 6 0 】

あるいは、特定の変換タイプの候補が現在のブロックに適用されるか否かを示す情報が、ビットストリームを介してシグナリングできる。例えば、水平方向と垂直方向に D C T 2 を適用するか否かを示すフラグ c u _ m t s _ f l a g がシグナリングされることがある。c u _ m t s _ f l a g の値が 1 である場合、D C T 2 を垂直方向と水平方向の変換コアとして設定することができる。c u _ m t s _ f l a g の値が 0 である場合、D C T 8 またはD S T 7 を垂直方向と水平方向の変換コアとして設定することができる。あるいは、c u _ m t s _ f l a g の値が 0 である場合、複数の変換タイプの組み合わせ候補のいずれかを指定する情報 t u _ m t s _ i d x をシグナリングすることができる。

20

【 0 4 6 1 】

現在のブロックが幅が高さより大きい非正方形または高さが幅より大きい非正方形である場合、c u _ m t s _ f l a g の符号化を省略し、c u _ m t s _ f l a g の値が 0 であると見なすことができる。

30

【 0 4 6 2 】

現在のブロックのサイズ、形状、またはイントラ予測モードに応じて、使用可能な変換タイプの組み合わせ候補の数を異なるように設定することができる。例えば、現在のブロックが正方形である場合、3 つ以上の変換タイプの組み合わせ候補を使用し、現在のブロックが非正方形である場合、2 つの変換タイプの組み合わせ候補を使用することができる。あるいは、現在のブロックが正方形である場合、変換タイプの組み合わせ候補のうち、水平方向の変換タイプと垂直方向の変換タイプが異なる変換タイプの組み合わせ候補のみを使用することができる。

【 0 4 6 3 】

現在のブロックが使用できる変換タイプの組み合わせ候補が 3 つ以上である場合、変換タイプの組み合わせ候補のうちの 1 つを指示するインデックス情報 t u _ m t s _ i d x をシグナリングすることができる。一方、現在のブロックが使用できる変換タイプの組み合わせ候補が 2 つである場合、変換タイプの組み合わせ候補のいずれかを指示するフラグ m t s _ f l a g をシグナリングすることができる。次の表 11 は、現在のブロックの形状による変換タイプの組み合わせ候補を指定するための情報を符号化するプロセスを示す。

40

50

【表 1 1】

residual_coding (x0, y0, log2TbWidth, log2TbHeight, cIdx) {	記述子
...	
if (cu_mts_flag[x0][y0] && (cIdx == 0) &&	
!transform_skip_flag[x0][y0][cIdx] &&	
((CuPredMode[x0][y0] == MODE_INTRA && numSigCoeff > 2)	10
(CuPredMode[x0][y0] == MODE_INTER)) {	
if (cbHeight == cbWidth) {	
mts_idx[x0][y0]	ae(v)
} else {	
mts_flag[x0][y0]	u(1)
}	
}	

20

【0 4 6 4】

現在のブロックの形状に応じて、変換タイプの組み合わせ候補のインデックスを再配置（または再並べ替え）することができる。例えば、現在のブロックが正方形である場合に変換タイプの組み合わせ候補に割り当てるインデックスと、現在のブロックが非正方形である場合に変換タイプの組み合わせ候補に割り当てるインデックスが異なることができる。例えば、現在のブロックが正方形である場合、次の表 1 2に基づいて、変換タイプの組み合わせを選択し、現在のブロックが非正方形である場合、次の表 1 3に基づいて、変換タイプの組み合わせを選択することができる。

30

【表 1 2】

m t s __ i	イントラ		インター	
d x				
m t s __ i	水平方向変換	垂直方向変換	水平方向変換	垂直方向変換
d x	コア	コア	コア	コア
0	D S T 7	D S T 7	D C T 8	D C T 8
1	D C T 8	D S T 7	D S T 7	D C T 8
2	D S T 7	D C T 8	D C T 8	D S T 7
3	D C T 8	D C T 8	D S T 7	D S T 7

40

50

【表 1 3】

m t s _ i d x	イントラ		インター	
m t s _ i d x	水平方向変換 コア	垂直方向変換 コア	水平方向変換 コア	垂直方向変換 コア
0	D C T 8	D S T 7	D S T 7	D C T 8
1	D S T 7	D C T 8	D C T 8	D S T 7
2	D S T 7	D S T 7	D S T 7	D S T 7
3	D C T 8	D C T 8	D S T 7	D S T 7

10

【0 4 6 5】

現在のブロックの水平方向非ゼロ係数の数または垂直方向非ゼロ係数の数に基づいて、変換タイプを決定することができる。ここで、水平方向非ゼロ係数の数は、 $1 \times N$ （ここで、Nは現在のブロックの幅である）に含まれる非ゼロ係数の数を示し、垂直方向非ゼロ係数の数は、 $N \times 1$ （ここで、Nは現在のブロックの高さである）に含まれる非ゼロ係数の数を示す。水平方向非ゼロ係数の最大値が閾値より小さいか等しい場合、水平方向に一次変換タイプを適用し、水平方向非ゼロ係数の最大値が閾値を超える場合、水平方向に二次変換タイプを適用することができる。垂直方向非ゼロ係数の最大値が閾値より小さいか等しい場合、垂直方向に一次変換タイプを適用し、垂直方向非ゼロ係数の最大値が閾値を超える場合、垂直方向に二次変換タイプを適用することができる。

20

【0 4 6 6】

図40は、現在のブロックの変換タイプを決定する例を説明するための図である。

【0 4 6 7】

30

例えば、イントラ予測で現在のブロックを符号化し、現在のブロックの水平方向非ゼロ係数の最大値が2より小さいか等しい場合（図40（a）参照）、D S T 7と水平方向の変換タイプとして決定することができる。

【0 4 6 8】

イントラ予測で現在のブロックを符号化し、現在のブロックの垂直方向非ゼロ係数の最大値が2を超える場合（図40（b）を参照）、D C T 2またはD C T 8を垂直方向の変換タイプとして決定することができる。

【0 4 6 9】

残差係数を変換ユニットまたはサブ変換ユニットで符号化することができる。ここで、残差係数とは、変換によって生成された変換係数、変換スキップによって生成された変換スキップ係数または前記変換係数または係数を量子化して生成された量子化された係数を指す。

40

【0 4 7 0】

変換ユニットは、一次変換または二次変換が実行されたブロックを示すことができる。サブ変換ユニットは、変換ユニットより小さいブロックを示す。例えば、サブ変換ユニットは、 4×4 、 2×8 または 8×2 サイズのブロックであり得る。

【0 4 7 1】

現在のブロックのサイズまたは形状に基づいて、サブ変換ユニットのサイズまたは形状のうちの少なくとも1つを決定することができる。例えば、現在のブロックが幅が高さより大きい非正方形である場合、サブ変換ユニットも、幅が高さより大きい非正方形（例え

50

ば、 8×2)に設定し、現在のブロックが、高さが幅より大きい非正方形である場合、サブ変換ユニットも、高さが幅より大きい非正方形(例えば、 2×8)に設定することができる。現在のブロックが正方形である場合、サブ変換ユニットも、正方形(例えば、 4×4)に設定することができる。

【0472】

現在のブロックが複数のサブ変換ユニットを含む場合、サブ変換ユニットを順次符号化 / 復号化することができる。算術符号化 (Arithmetic Coding)などのエントロピー符号化を使用して、残差係数を符号化することができる。以下、図面を参照して、残差係数の符号化 / 復号化方法について詳細に説明する。

【0473】

図41は、残差係数を符号化する方法のフロー チャートである。

【0474】

本実施例では、現在のブロックが1つ以上のサブ変換ユニットを含むと仮定する。さらに、サブ変換ユニットのサイズが 4×4 であると仮定する。ただし、サブ変換ユニットのサイズがこれと異なる場合、またはサブ変換ユニットの形状がこれと異なる場合にも、本実施例をそのまま適用することができる。

【0475】

現在のブロックに非ゼロ係数が存在するか否かを決定することができる(S4101)。非ゼロ係数は、絶対値が0を超える残差係数を示す。現在のブロックに非ゼロ係数が存在するか否かを示す情報を符号化してシグナリングすることができる。例えば、前記情報は、1ビットのフラグCBF(Coded Block Flag)であり得る。

【0476】

現在のブロックに非ゼロ係数が存在する場合、各サブ変換ユニットに非ゼロ係数が存在するか否かを決定することができる(S4102)。各サブ変換ユニットに非ゼロ係数が存在するか否かを示す情報を符号化してシグナリングすることができる。例えば、前記情報は、1ビットのフラグ、符号化サブブロックフラグ(CSBF:Coded Sub Block Flag)であり得る。サブ変換ユニットは、選択された走査順序に従って符号化できる。

【0477】

サブ変換ユニットに非ゼロ係数がある場合、サブ変換ユニットの残差係数を符号化するためには、サブ変換ユニット内の残差係数を一次元的に配列することができる(S4103)。残差係数は、選択された走査順序に従って一次元的に配列されることがある。

【0478】

走査順序は、対角走査、水平走査、垂直走査、またはこれらの逆走査のうちの少なくとも1つを含み得る。

【0479】

図42および図43は、異なる走査順序における残差係数の配列順序を示す図である。

【0480】

図42(a)から図42(c)は、対角走査(Diagonal Scan)、水平走査(Horizontal Scan)、および垂直走査(Vertical Scan)を示し、図43(a)から図43(c)は、これらの逆走査を示す。

【0481】

選択された走査順序に従って、残差係数を一次元的に配列することができる。

【0482】

現在のブロックのサイズ、形状、イントラ予測モード、一次変換に使用される変換コアまたは二次変換を適用するか否かのうちの少なくとも1つを考慮して、走査順序を決定することができる。例えば、現在のブロックが幅が高さより大きい非正方形である場合、逆水平走査を使用して残差係数を符号化することができる。一方、現在のブロックが、高さが幅より大きい非正方形である場合、逆垂直走査を使用して残差係数を符号化することができる。

10

20

30

40

50

【0483】

あるいは、複数の走査順序のそれぞれのレート歪みの最適化（RDO：Rate Distortion Optimization）を計算し、RDOが最も低い走査順序を、現在のブロックの走査順序として決定することができる。この場合、現在のブロックの走査順序を示す情報を符号化してシグナリングすることができる。

【0484】

変換スキップ係数を符号化する場合に使用できる走査順序候補と、変換係数を符号化する場合に使用できる走査順序候補が異なることができる。例えば、変換係数を符号化する場合に使用できる走査順序候補の逆方向を、変換スキップ係数を符号化場合に使用できる走査順序候補として設定することができる。

10

【0485】

例えば、変換係数が逆対角走査、逆水平走査または逆垂直走査のいずれかを使用して符号化される場合、変換スキップ係数は、対角走査、水平走査または垂直走査のいずれかを使用して符号化できる。

【0486】

その後、変換ブロック内の走査順序が最後に位置する非ゼロ係数の位置を符号化することができる（S4103）。最後の非ゼロ係数のx軸位置およびy軸位置をそれぞれ符号化することができる。

【0487】

図44は、最後の非ゼロ系数の位置を符号化する例を示す図である。

20

【0488】

図44に示されるように、対角走査の場合、変換ブロックの右上隅に位置する残差係数を最後の非ゼロ係数として設定することができる。前記残差係数のx軸座標LastXは3であり、y軸座標LastYは0である。LastXは、プレフィックス部分last_sig_coeff_x_prefixとサフィックス部分last_sig_coeff_x_suffixに分離されて符号化されることができる。LastYも、プレフィックス部分last_sig_coeff_y_prefixとサフィックス部分last_sig_coeff_y_suffixに分離されて符号化されることができる。

【0489】

最後の非ゼロ係数の位置を決定した場合、最後の非ゼロ係数よりも走査順序が先立つ各残差係数について、残差係数が非ゼロ係数であるか否かを示す情報を符号化することができる（S4104）。前記情報は、1ビットのフラグ、sig_coeff_flagであり得る。残差係数が0である場合、非ゼロ係数フラグsig_coeff_flagの値を0に設定し、残差係数が1である場合、非ゼロ係数フラグsig_coeff_flagの値を1に設定することができる。最後の非ゼロ係数については、非ゼロ係数フラグsig_coeff_flagの符号化を省略することができる。非ゼロ係数フラグsig_coeff_flagの符号化を省略する場合、前記残差係数は0ではないと見なされる。

30

【0490】

変換スキップ係数については、最後の非ゼロ係数の位置に関する情報を符号化しないことができる。この場合、ブロック内のすべての変換スキップ係数について、非ゼロ係数フラグsig_coeff_flagを符号化することができる。

40

【0491】

あるいは、最後の非ゼロ係数の位置の情報が符号化されるか否かを示す情報を符号化してシグナリングすることができる。前記情報は、1ビットのフラグであり得る。復号化器は、前記フラグの値に基づいて、最後の非ゼロ係数の位置に関する情報を復号化するか否かを決定することができる。最後の非ゼロ係数の位置が復号化されない場合、すべての変換スキップ係数について、非ゼロ係数のフラグを復号化することができる。一方、最後の非ゼロ係数の位置が復号化される場合、最後の非ゼロ係数よりも走査順序が先立つ各変換スキップ係数について、変換スキップ係数が非ゼロ係数であるか否かを示す情報を符号化

50

することができる。

【0492】

変換係数または変換スキップ係数のDC成分は、ゼロに設定されないことができる。ここで、DC成分は、走査順序が最後に位置するサンプルまたはブロックの左上に位置するサンプルを示し得る。DC成分については、残差係数が非ゼロ係数フラグ`sig_coeff_flag_f1ag`の符号化を省略することができる。非ゼロ係数フラグ`sig_coeff_flag_f1ag`の符号化を省略する場合、前記残差係数は0ではないと見なされる。

【0493】

残差係数が0でない場合、残差係数の絶対値を示す情報および残差係数の符号を示す情報を符号化することができる(S4105)。残差係数の絶対値を示す情報の符号化プロセスについては、図45を参照して、より詳細に説明する。

10

【0494】

最後の非ゼロ係数の後のすべての残差係数と、現在のブロックに含まれるサブ変換ユニットのそれぞれについて、残差係数の符号化を順次実行することができる(S4106、S4107)。

【0495】

図45は、残差係数の絶対値を符号化するプロセスのフローチャートである。

【0496】

残差係数の絶対値が0を超える場合、残差係数または残差係数が偶数であるか奇数であるかを示す情報を符号化することができる(S4501)。ここで、残差係数は、残差係数の絶対値から1を差し引いた値として定義することができる。前記情報は、1ビットのパリティフラグであり得る。例えば、構文要素`par_level_flag_f1ag`の値が0であることは、残差係数または残余係数が偶数であることを示し、構文要素`par_level_f1ag`の値が1であることは、残差係数または残余係数が奇数であることを示す。当構文要素`par_level_f1ag`の値は、下記式30に基づいて決定できる。

20

【数30】

[式30]

$$par_level_flag = (abs(Tcoeff) - 1) \% 2$$

30

【0497】

式30において、`Tcoeff`は残差係数を表し、`abs()`は絶対値関数を表す。

【0498】

残差係数または残余係数を2で除算するか、または残差係数または残余係数にビットシフト演算を適用して、調整残余係数を導出することができる(S4502)。具体的には、残差係数を2で除算した商を調整残余係数として設定するか、または残余係数を右に1シフトした値を調整残余係数として設定することができる。

【0499】

40

例えば、下記式31に基づいて調整残余係数を導出することができる。

【数31】

[式31]

$$ReRemLevel = RemLevel \gg 1$$

【0500】

式31において、`ReRemLevel`は調整残余係数を表し、`RemLevel`は残余係数を表す。

50

【0501】

その後、調整残余係数のサイズを示す情報を符号化することができる。調整残余係数のサイズを示す情報は、調整残余係数がNより大きいか否かを示す情報を含み得る。ここで、Nは1、2、3、4などの整数であり得る。

【0502】

例えば、調整残余係数の値が1より大きいか否かを示す情報を符号化することができる(S4503)。前記情報は、1ビットのフラグrem_abs_g_t1_flagであり得る。

【0503】

残差係数が0または1である場合、rem_abs_g_t1_flagの値を0に設定できる。つまり、残差係数が1または2である場合、rem_abs_g_t1_flagの値を0に設定できる。この場合、残差係数が0である場合、par_level_flagを0に設定し、残差係数が1である場合、par_level_flagを1に設定することができる。

10

【0504】

調整残余係数が2以上である場合(S4504)、rem_abs_g_t1_flagの値を1に設定し、調整残余係数の値が2より大きいか否かを示す情報を符号化することができる(S4505)。前記情報は、1ビットのフラグrem_abs_g_t2_flagであり得る。

20

【0505】

残差係数が1または2である場合、rem_abs_g_t2_flagの値を0に設定できる。つまり、残差係数が3または4である場合、rem_abs_g_t2_flagの値を0に設定できる。この場合、残差係数が2である場合、par_level_flagを0に設定し、残差係数が3である場合、par_level_flagを1に設定することができる。

【0506】

調整残余係数が2を超える場合、調整残余係数から2を差し引いた残差値情報を符号化することができる(S4506)。つまり、残差係数の絶対値から5を差し引いた後、前記結果値を2で除算したものを残差値情報として符号化することができる。

30

【0507】

図示されていないが、調整残余係数が3を超えるか否かを示すフラグ(例えば、rem_abs_g_t3_flag)または調整残余係数が4を超えるか否かを示すフラグ(rem_abs_g_t4_flag)などをさらに使用して、残差係数を符号化することができる。この場合、調整残余係数から最大値を差し引いた値を残差値として設定できる。最大値は、rem_abs_g_tN_flagのうち最大N値を示す。

【0508】

調整残余係数の値を指定値と比較するフラグの代わりに、調整係数または残差係数の絶対値を指定値と比較するフラグを使用することもできる。例えば、rem_abs_g_t1_flagの代わりに、残差係数の絶対値が2を超えるか否かを示すgr2_flagを使用し、rem_abs_g_t2_flagの代わりに、残差係数の絶対値が4を超えるか否かを示すgr4_flagを使用することができる。

40

【0509】

表14は、構文要素を使用して、残差係数を符号化するプロセスを簡略的に示したものである。

50

【表 1 4】

```

Sig_coeff_flag = (abs(TCoeff) != 0) ? 1 : 0
Encode Sig_coeff_flag
if (Sig_coeff_flag) {
    RemLevel = abs (TCoeff) - 1
    Parity_level_flag = RemLevel%2
    Encode parity_level_flag
    ReRemLevel = RemLevel >> 1
    Rem_abs_gt1_flag = (ReRemLevel > 1) ? 1 : 0
    Encode rem_abs_gt1_flag
    if (Rem_abs_gt1_flag == 1) {
        Rem_abs_gt2_flag = absLevel > 4 ? 1 : 0
        if (Rem_abs_gt1_flag == 1) {
            Encode rem_abs_gt2_flag
            Rem = (absLevel - 5) >> 1
            Encode Rem
        }
    }
}
Sign_Coeff = TCoeff > 0 ? 0 : 1
Encode Sign_Coeff
}

```

10

20

30

40

【0510】

上述した例のように、`par_level_flag` および少なくとも 1 つの `rem_abs_gtN_flag`（ここで、N は 1、2、3、4 などの整数である）を使用して、残差係数を符号化することができる。

【0511】

`rem_abs_gtN_flag` は、残差係数が $2N$ を超えるか否かを示す。 $2N - 1$ または $2N$ の残差係数については、`rem_abs_gt(N-1)_flag` を `true` に設定し、`rem_abs_gtN_flag` を `false` に設定する。また、 $2N - 1$ の残差係数については、`par_level_flag` 値を 0 に設定し、 $2N$ の残差係数については、`par_level_flag` 値を 1 に設定することができます。つまり、 $2N$ 以下の残差係数について、`rem_abs_gtN_flag` と `par_level_flag` を使用して符号化することができる。

【0512】

50

残差係数が $2 \text{MAX} + 1$ 以上の残差係数について、 2MAX との差値を 2 で除算した残差情報を符号化することができる。ここで、 MAX は N の最大値を示す。例えば、`rem_abs_gt1_flag` と `rem_abs_gt2_flag` を使用する場合、 MAX は 2 であり得る。

【0513】

復号化器も、図 4-5 に示される順序に応じて残差係数を復号化することができる。具体的には、復号化器は、最後の非ゼロ係数の位置を決定し、最後の非ゼロ係数よりも走査順序が先立つ各残差係数について、`sig_coeff_flag` を復号化することができる。

【0514】

`sig_coeff_flag` が `true` である場合、前記残差係数の `par_level_flag` を復号化することができる。さらに、前記残差係数の `rem_abs_gt1_flag` を復号化することができる。この場合、`rem_abs_gt(N-1)_flag` の値に基づいて、`rem_abs_gtN_flag` をさらに復号化することができる。例えば、`rem_abs_gt(N-1)_flag` の値が 1 である場合、`rem_abs_gtN_flag` を復号化することができる。例えば、残差係数の `rem_abs_gt1_flag` の値が 1 である場合、前記残差係数の `rem_abs_gt2_flag` をさらに解析することができる。`rem_abs_gt(MAX)_flag` の値が 1 である場合、残差値情報を復号化することができる。

【0515】

残差係数の `rem_abs_gtN_flag` の値が 0 である場合、`par_level_flag` の値に基づいて、残差係数の値を $2N - 1$ または $2N$ に決定することができる。具体的には、`par_level_flag` が 0 である場合、残差係数を $2N - 1$ に設定し、`par_level_flag` が 1 である場合、残差係数を $2N$ に設定できる。

【0516】

例えば、`rem_abs_gt1_flag` が 0 である場合、`par_level_flag` 値に基づいて、残差係数の絶対値を 1 または 2 に設定できる。具体的には、`par_level_flag` 値が 0 である場合、残差係数の絶対値は 1 になり、`par_level_flag` 値が 1 である場合、残差係数の絶対値は 2 になる。

【0517】

例えば、`rem_abs_gt2_flag` が 0 である場合、`par_level_flag` 値に基づいて、残差係数の絶対値を 3 または 4 に設定できる。具体的には、`par_level_flag` 値が 0 である場合、残差係数の絶対値は 3 になり、`par_level_flag` 値が 1 である場合、残差係数の絶対値は 4 になる。

【0518】

残差値情報を復号化する場合、`par_level_flag` の値に基づいて、残差係数を $2(\text{MAX} + R) - 1$ または $2(\text{MAX} + R)$ に設定できる。ここで、 R は残差値情報で示される値を示す。例えば、`par_level_flag` が 0 である場合、残差係数を $2(\text{MAX} + R) - 1$ に設定し、`par_level_flag` が 1 である場合、残差係数を $2(\text{MAX} + R)$ に設定できる。例えば、 MAX が 2 である場合、下記式 3-2 に基づいて残差係数を導出することができる。

【数32】

[式3-2]

$$\text{parity_level_flag} + \text{Rem} \ll 1 + 5$$

【0519】

図 4-5 に示される例に基づいて、残差係数を符号化する場合、すべての非ゼロ係数のパ

10

20

30

40

50

リティフラグを符号化する必要がある。例えば、残差係数が 1 である場合にも、`par_level_flag` と `rem_abs_gt1_flag` を符号化する必要がある。上記のような符号化方法は、絶対値が 1 である残差係数を符号化するのに必要なビット数を増加させるという問題を引き起こす。このような問題を回避するために、残差係数が 1 を超えるか否かを示す情報を先に符号化した後、残差係数が 1 を超える場合に、パリティフラグを符号化することができる。

【0520】

図 46 は、残差係数の絶対値を符号化するプロセスのフローチャートである。

【0521】

非ゼロ残差係数については、残差係数の絶対値が 1 を超えるか否かを示す情報 `gr1_flag` を符号化することができる (S4601)。残差係数が 1 である場合、`gr1_flag` を 0 に設定し、残差係数が 1 を超える場合、`gr1_flag` を 1 に設定できる。

10

【0522】

残差係数の絶対値が 1 を超える場合、残差係数または残余係数是が偶数であるか奇数であるかを示すパリティフラグを符号化することができる (S4602, S4603)。ここで、残差係数を、残差係数から 2 を差し引いた値に設定できる。例えば、下記式 33 に基づいて `par_level_flag` を導出することができる。

【数33】

[式33]

20

$$par_level_flag = (abs(Tcoeff) - 2) \% 2$$

【0523】

残差係数または残余係数を 2 で除算するか、または残差係数または残余係数を右に 1 ビットシフトして調整残余係数を導出し、調整残余係数が 1 を超えるか否かを示す情報を符号化することができる (S4604)。例えば、`gr1_flag` が 1 である残差係数については、調整残余係数が 1 を超えるか否かを示す `rem_abs_gt1_flag` を符号化することができる。

30

【0524】

残差係数が 0 または 1 である場合、`rem_abs_gt1_flag` の値を 0 に設定できる。つまり、残差係数が 2 または 3 である場合、`rem_abs_gt1_flag` の値を 0 に設定できる。この場合、調整残余係数が 0 である場合、`par_level_flag` を 0 に設定し、残差係数が 1 である場合、`par_level_flag` を 1 に設定できる。

【0525】

調整残余係数が 2 以上である場合 (S4605)、`rem_abs_gt1_flag` の値を 1 に設定し、調整残余係数が 2 を超えるか否かを示す情報を符号化することができる (S4606)。例えば、`rem_abs_gt1_flag` が 1 である残差係数については、調整残余係数が 2 を超えるか否かを示す `rem_abs_gt2_flag` を符号化することができる。

40

【0526】

残差係数が 2 または 3 である場合、`rem_abs_gt2_flag` の値を 1 に設定できる。つまり、残差係数が 4 または 5 である場合、`rem_abs_gt2_flag` の値を 0 に設定できる。この場合、残差係数が 2 である場合、`par_level_flag` を 0 に設定し、残差係数が 3 である場合、`par_level_flag` を 1 に設定することができる。

【0527】

調整残余係数が 2 を超える場合 (S4607)、調整残余係数から 2 を差し引いた残差

50

値情報を符号化することができる（S 4 6 0 8）。つまり、残差係数の絶対値から 6 を差し引いた後、その結果値を 2 で除算したものを残差値情報として符号化することができる。

【0 5 2 8】

調整残余係数が 3 を超えるか否かを示すフラグ（例えば、r e m _ a b s _ g t 3 _ f l a g）または調整残余係数が 4 を超えるか否かを示すフラグ（r e m _ a b s _ g t 4 _ f l a g）などをさらに使用して、残差係数を符号化することができる。この場合、調整残余係数から最大値を差し引いた値を残差値として設定できる。最大値は、r e m _ a b s _ g t N _ f l a g のうち最大 N 値を示す。

【0 5 2 9】

調整残余係数の値を指定値と比較するフラグの代わりに、調整係数または残差係数の絶対値を指定値と比較するフラグを使用することもできる。例えば、r e m _ a b s _ g t 1 _ f l a g の代わりに、残差係数の絶対値が 3 を超えるか否かを示す g r 3 _ f l a g を使用し、r e m _ a b s _ g t 2 _ f l a g の代わりに、残差係数の絶対値が 5 を超えるか否かを示す g r 5 _ f l a g を使用することができる。

10

【0 5 3 0】

表 15 は、構文要素を使用して、残差係数を符号化するプロセスを簡略的に示したものである。

20

30

40

50

【表 1 5】

```

Sig_coeff_flag = (abs(TCoeff) != 0) ? 1 : 0
Encode Sig_coeff_flag
if (Sig_coeff_flag) {
    Gr1_flag = abs(TCoeff) > 1 ? 1 : 0
    Encode Gr1_flag
    if (Gr1_flag) {
        RemLevel = abs(Tcoeff) - 2
        Parity_level_flag = RemLevel % 2
        Encode Parity_level_flag
        ReRemLevel = RemLevel >> 1
        Rem_abs_gt1_flag = (ReRemLevel > 1) ? 1 : 0
        Encode Rem_abs_gt1_flag
        if (Rem_abs_gt1_flag == 1) {
            Rem_abs_gt2_flag = absLevel > 5 ? 1 : 0
            if (Rem_abs_gt1_flag == 1) {
                Encode Rem_abs_gt2_flag
                Rem = (absLevel - 6) >> 1
                Encode Rem
            }
        }
    }
}
Sign_Coeff = TCoeff > 0 ? 0 : 1
Encode Sign_Coeff
}

```

10

20

30

40

【0 5 3 1】

復号化器も、図 4 6 に示される順序に応じて残差係数を復号化することができる。具体的には、復号化器は、最後の非ゼロ係数の位置を決定し、最後の非ゼロ係数よりも走査順序が先立つ各残差係数について、`sig_coeff_flag` を復号化することができる。

【0 5 3 2】

`sig_coeff_flag` が `true` である場合、前記残差係数の `gr1_flag` を復号化することができる。`gr1_flag` が 0 である場合、前記残差係数の絶対値を 1 に決定し、`gr1_flag` が 1 である場合、前記残差係数の `parity_level`

50

`f1ag` を復号化することができる。その後、前記残差係数の `rem_abs_gt1_f1ag` を復号化することができる。この場合、`rem_abs_gt(N-1)_f1ag` の値に基づいて、`rem_abs_gtN_f1ag` をさらに復号化することができる。例えば、`rem_abs_gt(N-1)_f1ag` の値が 1 である場合、`rem_abs_gtN_f1ag` を復号化することができる。例えば、残差係数の `rem_abs_gt1_f1ag` の値が 1 である場合、`rem_abs_gt1_f1ag` をさらに解析することができる。`rem_abs_gt(MAX)_f1ag` の値が 1 である場合、残差値情報を復号化することができる。

【0533】

残差係数の `rem_abs_gtN_f1ag` の値が 0 である場合、`par_level1_f1ag` の値に基づいて、残差係数の値を $2N - 1$ または $2N$ に決定することができる。具体的には、`par_level1_f1ag` が 0 である場合、残差係数を $2N$ に設定し、`par_level1_f1ag` が 1 である場合、残差係数を $2N + 1$ に設定できる。

【0534】

例えば、`rem_abs_gt1_f1ag` が 0 である場合、`par_level1_f1ag` 値に基づいて、残差係数の絶対値を 2 または 3 に設定できる。具体的には、`par_level1_f1ag` 値が 0 である場合、残差係数の絶対値は 2 になり、`par_level1_f1ag` 値が 1 である場合、残差係数の絶対値は 3 になる。

【0535】

例えば、`rem_abs_gt2_f1ag` が 0 である場合、`par_level1_f1ag` 値に基づいて、残差係数の絶対値を 4 または 5 に設定できる。具体的には、`par_level1_f1ag` 値が 0 である場合、残差係数の絶対値は 4 になり、`par_level1_f1ag` 値が 1 である場合、残差係数の絶対値は 5 になる。

【0536】

残差値情報を復号化する場合、`par_level1_f1ag` の値に基づいて、残差係数を $2(MAX + R)$ または $2(MAX + R) + 1$ に設定できる。ここで、R は残差値情報の値を示す。例えば、`par_level1_f1ag` が 0 である場合、残差係数を $2(MAX + R)$ に設定し、`par_level1_f1ag` が 1 である場合、残差係数を $2(MAX + R) + 1$ に設定できる。

【0537】

別の例として、残差係数の値が 2 を超える場合にのみ、パリティフラグを符号化することができる。例えば、残差係数の値が 1 を超えるか否かを示す情報および残差係数の値が 2 を超えるか否かを示す情報を符号化した後、残差係数の値が 2 を超えると決定した場合、前記残差係数についてパリティフラグを符号化することができる。

【0538】

図 47 は、残差係数の絶対値を符号化するプロセスのフローチャートである。

【0539】

非ゼロ残差係数については、残差係数の絶対値が 1 を超えるか否かを示す情報 `gr1_f1ag` を符号化することができる (S4701)。残差係数が 1 である場合、`gr1_f1ag` を 0 に設定し、残差係数が 1 を超える場合、`gr1_f1ag` を 1 に設定できる。

【0540】

絶対値が 1 を超える残差係数については、残差係数の絶対値が 2 を超えるか否かを示す情報 `gr2_f1ag` を符号化することができる (S4702, S4703)。残差係数が 2 である場合、`gr2_f1ag` を 0 に設定し、残差係数が 2 を超える場合、`gr2_f1ag` を 1 に設定できる。

【0541】

絶対値が 2 を超える場合、残差係数または残余係数は偶数であるか奇数であるかを示すパリティフラグを符号化することができる (S4704, S4705)。ここで、残差係数を、残差係数から 3 を差し引いた値に設定できる。例えば、下記式 34 に基づいて `par_level1_f1ag` を導出することができる。

10

20

30

40

50

【数34】

[式34]

$$par_level_flag = (abs(Tcoeff) - 3) \% 2$$

【0542】

残差係数または残余係数を2で除算するか、または残差係数または残余係数を右に1ビットシフトして調整残余係数を導出し、調整残余係数が1を超えるか否かを示す情報を符号化することができる。例えば、`gr1_flg`が1である残差係数については、調整残余係数が1を超えるか否かを示す`rem_abs_gt1_flg`を符号化することができる(S4706)。

10

【0543】

残差係数が0または1である場合、`rem_abs_gt1_flg`の値を0に設定できる。つまり、残差係数が3または4である場合、`rem_abs_gt1_flg`の値を0に設定できる。この場合、調整残余係数が0である場合、`par_level_flg`を0に設定し、残差係数が1である場合、`par_level_flg`を1に設定できる。

20

【0544】

調整残余係数が2以上である場合(S4707)、`rem_abs_gt1_flg`の値を1に設定し、調整残余係数が2を超えるか否かを示す情報を符号化することができる(S4708)。例えば、`rem_abs_gt1_flg`が1である残差係数については、調整残余係数が2を超えるか否かを示す`rem_abs_gt2_flg`を符号化することができる。

30

【0545】

残差係数が2または3である場合、`rem_abs_gt2_flg`の値を1に設定できる。つまり、残差係数が5または6である場合、`rem_abs_gt2_flg`の値を0に設定できる。この場合、残差係数が2である場合、`par_level_flg`を0に設定し、残差係数が3である場合、`par_level_flg`を1に設定することができる。

30

【0546】

調整残余係数が2を超える場合、調整残余係数から2を差し引いた残差値を符号化することができる(S4709、S4710)。つまり、残差係数の絶対値から7を差し引いた後、その結果値を2で除算したものを残差値として符号化することができる。

【0547】

調整残余係数が3を超えるか否かを示すフラグ(例えば、`rem_abs_gt3_flg`)または調整残余係数が4を超えるか否かを示すフラグ(`rem_abs_gt4_flg`)などをさらに使用して、残差係数を符号化することができる。この場合、調整残余係数から最大値を差し引いた値を残差値として設定できる。最大値は、`rem_abs_gtN_flg`のうち最大N値を示す。

40

【0548】

調整残余係数の値を指定値と比較するフラグの代わりに、調整係数または残差係数の絶対値を指定値と比較するフラグを使用することもできる。例えば、`rem_abs_gt1_flg`の代わりに、残差係数の絶対値が4を超えるか否かを示す`gr4_flg`を使用し、`rem_abs_gt2_flg`の代わりに、残差係数の絶対値が6を超えるか否かを示す`gr6_flg`を使用することができる。

50

【0549】

復号化器も、図47に示される順序と同じ順序で、残差係数を復号化することができる。具体的には、復号化器は、最後の非ゼロ係数の位置を決定し、最後の非ゼロ係数よりも

走査順序が先立つ各残差係数について、`sig_coeff_flag`を復号化することができる。

【0550】

`sig_coeff_flag`が`true`である場合、前記残差係数の`gr1_flag`を復号化することができる。`gr1_flag`が0である場合、前記残差係数の絶対値を1に決定し、`gr1_flag`が1である場合、`gr2_flag`を復号化することができる。`gr2_flag`が0である場合、前記残差係数の絶対値を2に決定し、`gr2_flag`が1である場合、前記残差係数の`par_level_flag`を復号化することができる。その後、前記残差係数の`rem_abs_gt1_flag`を復号化することができる。この場合、`rem_abs_gt(N-1)_flag`の値に基づいて、`rem_abs_gtN_flag`をさらに復号化することができる。例えば、`rem_abs_gt(N-1)_flag`の値が1である場合、`rem_abs_gtN_flag`を復号化することができる。例えば、残差係数の`rem_abs_gt1_flag`の値が1である場合、前記残差係数の`rem_abs_gt2_flag`をさらに解析することができる。`rem_abs_gt(MAX)_flag`の値が1である場合、残差値情報を復号化することができる。

10

【0551】

残差係数の`rem_abs_gtN_flag`の値が0である場合、`par_level_flag`の値に基づいて、残差係数の値を $2N + 1$ または $2(N + 1)$ に決定することができる。具体的には、`par_level_flag`が0である場合、残差係数を $2N + 1$ に設定し、`par_level_flag`が1である場合、残差係数を $2(N + 1)$ に設定できる。

20

【0552】

例えば、`rem_abs_gt1_flag`が0である場合、`par_level_flag`値に基づいて、残差係数の絶対値を3または4に設定できる。具体的には、`par_level_flag`値が0である場合、残差係数の絶対値は3になり、`par_level_flag`値が1である場合、残差係数の絶対値は4になる。

【0553】

例えば、`rem_abs_gt2_flag`が0である場合、`par_level_flag`値に基づいて、残差係数の絶対値を5または6に設定できる。具体的には、`par_level_flag`値が0である場合、残差係数の絶対値は5になり、`par_level_flag`値が1である場合、残差係数の絶対値は6になる。

30

【0554】

残差値情報を復号化する場合、`par_level_flag`の値に基づいて、残差係数を $2(MAX + R)$ または $2(MAX + R) + 1$ に設定できる。ここで、Rは残差値情報の値を示す。例えば、`par_level_flag`が0である場合、残差係数を $2(MAX + R)$ に設定し、`par_level_flag`が1である場合、残差係数を $2(MAX + R) + 1$ に設定できる。

【0555】

現在のブロックのサイズ、形状、変換スキップするか否か、変換コア、非ゼロ係数の数、または最後の非ゼロ係数の位置のうちの少なくとも1に基づいて、調整残余係数を指定値と比較する比較フラグの数または種類のうちの少なくとも1を決定することができる。例えば、変換係数を符号化する場合、`rem_abs_gt1_flag`のみを使用することができる。一方、変換スキップ係数を符号化する場合、`rem_abs_gt1_flag`と`rem_abs_gt2_flag`を使用することができる。

40

【0556】

あるいは、 4×4 サイズのサブ変換ユニット内では、`rem_abs_gt1_flag`の数を最大8に設定し、`rem_abs_gt2_flag`の数を最大1に設定できる。あるいは、非ゼロ係数フラグの数が $(16 - N)$ である場合、`rem_abs_gt1_flag`の数を最大 $8 + (N / 2)$ に設定し、`rem_abs_gt2_flag`の数

50

を最大 $1 + (N - (N / 2))$ に設定できる。

【0557】

現在のブロックの復元ブロックを取得した場合、インループフィルタリング (In-loop filtering) を介して量子化と符号化プロセスで生成される情報の損失を減らすことができる。インループフィルタは、デブロッキングフィルタ (Deblocking filter)、サンプル適応オフセットフィルタ (SAO: Sample Adaptive Offset filter) または適応ループフィルタ (ALF: Adaptive Loop Filter) のうちの少なくとも 1 つを含み得る。以下、インループフィルタが適用される前の復元ブロックを第 1 復元ブロックと呼び、インループフィルタが適用された後の復元ブロックを第 2 復元ブロックと呼ぶ。

10

【0558】

デブロッキングフィルタ、SAO または ALF のうちの少なくとも 1 つを第 1 復元ブロックに適用して、第 2 復元ブロックを取得することができる。この場合、デブロッキングフィルタが適用された後に SAO または ALF を適用することができる。

【0559】

デブロッキングフィルタは、量子化がブロック単位で実行されるときにブロックの境界で生じる画質の低下 (ブロッキングアーティファクト、Blocking Artifacts) を軽減するために使用される。デブロッキングフィルタを適用するためには、第 1 復元ブロックと隣接する復元ブロックとの間のブロッキング強度 (BS: Blocking Strength) を決定することができる。

20

【0560】

図 48 は、ブロック強度を決定するプロセスのフローチャートである。

【0561】

図 48 に示される例では、P は第 1 復元ブロックを表し、Q は隣接する復元ブロックを表す。ここで、隣接復元ブロックは、現在のブロックの左側または上部に隣接するブロックであり得る。

【0562】

図 48 に示される例では、P と Q の予測符号化モード、非ゼロ変換係数が含まれるか否か、同じ参照画像を使用してインター予測が実行されるか否か、または動きベクトルの差値が閾値以上であるか否かを考慮して、ブロック強度を決定することを示す。

30

【0563】

ブロック強度に基づいて、デブロッキングフィルタを適用する否かを決定することができる。例えば、ブロック強度が 0 である場合、フィルタリングが実行されない場合がある。

【0564】

SAO は、周波数領域で量子化を実行するときに生じるリングアーティファクト (Ringing Artifacts) を軽減するために使用される。第 1 復元画像のパターンを考慮して決定されるオフセットを加算または減算することによって、SAO を実行することができる。オフセットの決定方法は、エッジオフセット (EO: Edge Offset) またはバンドオフセット (Band Offset) を含む。EO は、周辺画素のパターンに基づいて、現在のサンプルのオフセットを決定する方法を示す。BO は、領域内の同様の輝度値を持つ画素のグループに対して共通のオフセットを適用する方法を示す。具体的には、画素輝度を 32 個の均等な区間に分けて、同様の輝度値を持つ画素を 1 つのグループとして設定することができる。例えば、32 個のバンドのうち、隣接する 4 つのバンドを 1 つのグループに設定し、4 つのバンドに属するサンプルに同じオフセット値を適用することができる。

40

【0565】

ALF は、第 1 復元画像またはデブロッキングフィルタが適用された復元画像に、事前定義されたサイズ / 形状のフィルタを適用して第 2 復元画像を生成する方法である。下記式 35 は、ALF の適用例を示す。

【数 35】

50

[式35]

$$R'(i, j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} f(k, l) \cdot R(i+k, j+l)$$

【0566】

画像、符号化ツリーユニット、符号化ブロック、予測ブロックまたは変換ブロック単位で、事前定義されたフィルタ候補のいずれかを選択することができる。各フィルタ候補のサイズまたは形状のいずれかが異なることができる。

【0567】

図49は、事前定義されたフィルタ候補を示す図である。

【0568】

図49に示される例では、 5×5 、 7×7 、および 9×9 サイズの菱形のうちの少なくとも1つを選択することができる。

【0569】

彩度成分については、 5×5 サイズの菱形のみを使用することができる。

【0570】

復号化プロセスまたは符号化プロセスに焦点を合わせて説明された実施例を、符号化プロセスまたは復号化プロセスに適用することは、本発明の範囲に含まれるものである。所定の順序で説明された実施例を、説明された順序とは異なる順序で変更したものも、本発明の範囲に含まれる。

【0571】

上述した実施例は、一連のステップまたはフローチャートに基づいて説明されたが、これは本発明の時系列的順序を限定するものではなく、前記ステップは、必要に応じて同時に実行するか、または他の順序で実行することができる。また、上述した実施例では、ブロック図を構成する各構成要素（例えば、ユニット、モジュールなど）は、ハードウェア装置またはソフトウェアによって実装することができ、複数の構成要素を組み合わせて、単一のハードウェア装置またはソフトウェアとして実装できる。上述した実施例は、様々なコンピュータ構成要素をによって実行できるプログラム命令の形で実装でき、前記プログラム命令は、コンピュータ可読記憶媒体に記憶できる。前記コンピュータ可読記憶媒体は、プログラム命令、データファイル、データ構造などを単独または組み合わせて含むことができる。コンピュータ可読記憶媒体の例には、ハードディスク、フロッピーディスクおよび磁気テープなどの磁気媒体、CD-ROM、DVDなどの光学記憶媒体、フロピカルディスク(floptical disk)などの磁気・光媒体(magnetic-optical media)、およびROM、RAM、フラッシュメモリなど、プログラム命令を記憶および実行するように特別に構成されるハードウェア装置が含まれることができる。前記ハードウェア装置は、本発明に係る処理を実行するための1つまたは複数のソフトウェアモジュールとして動作するように構成されることができ、その逆も同様である。

【産業上の利用可能性】

【0572】

本発明は、ビデオを符号化／復号化する電子機器に適用されることがある。

10

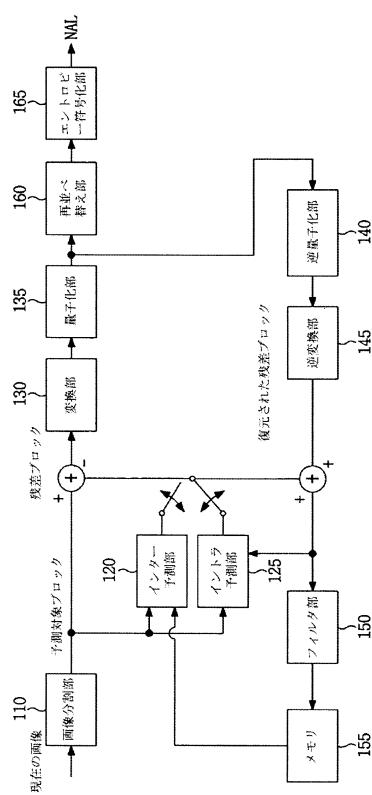
20

30

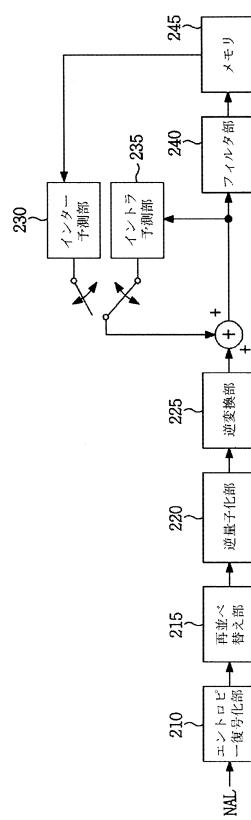
40

50

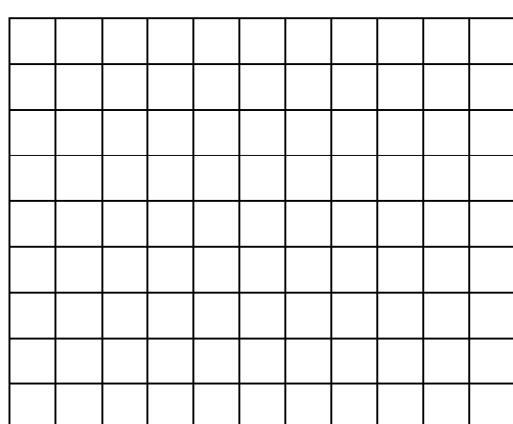
【図面】
【図 1】



【図 2】

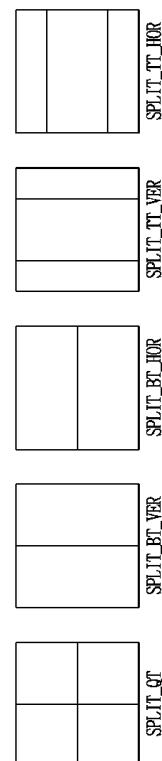


【図 3】



[E3]

【図 4】



10

20

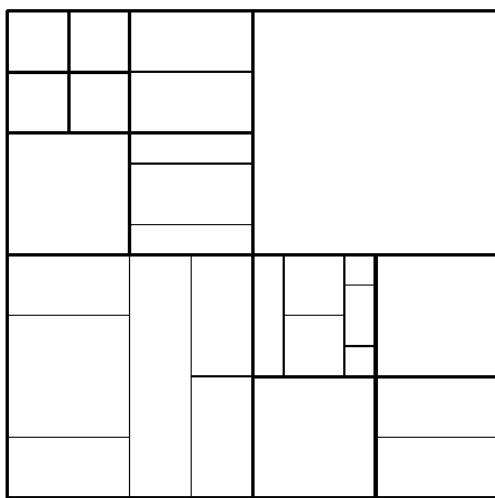
30

40

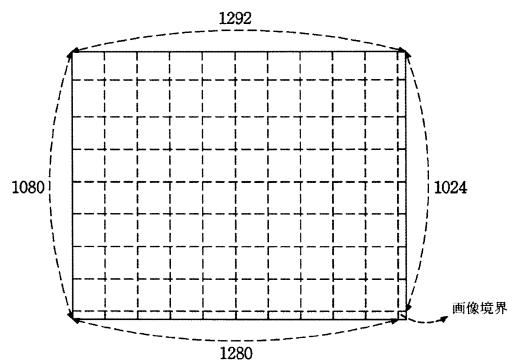
50

【図 5】

[도5]

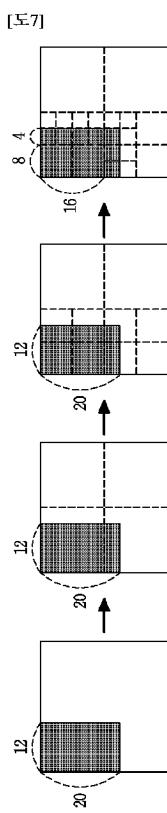


【図 6】



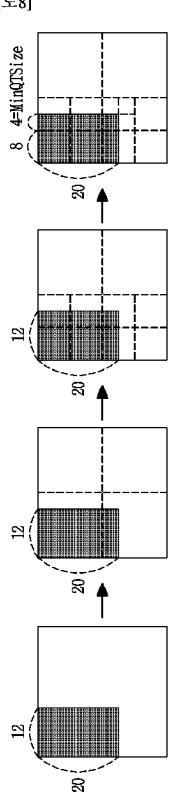
10

【図 7】



[도7]

【図 8】



20

[도8]

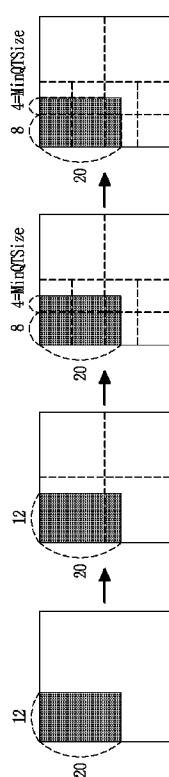
30

40

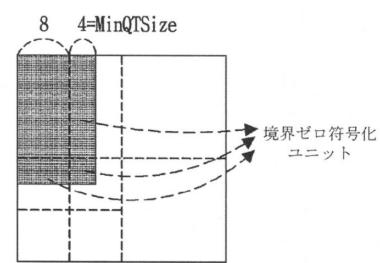
50

【図 9】

[E9]



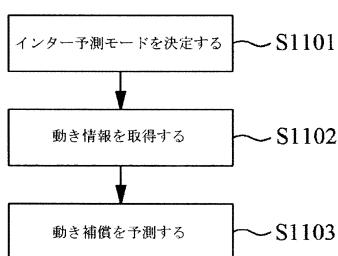
【図 10】



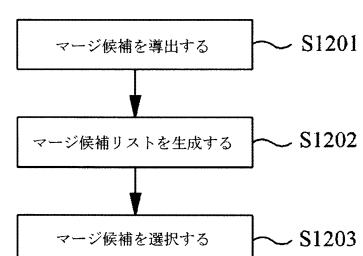
10

20

【図 11】



【図 12】

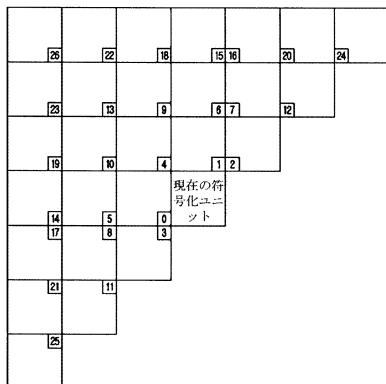


30

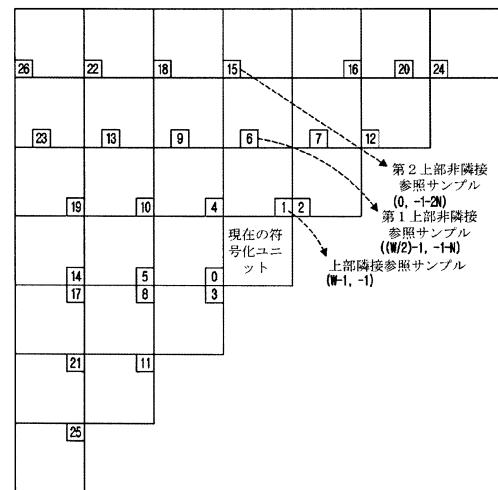
40

50

【図 1 3】

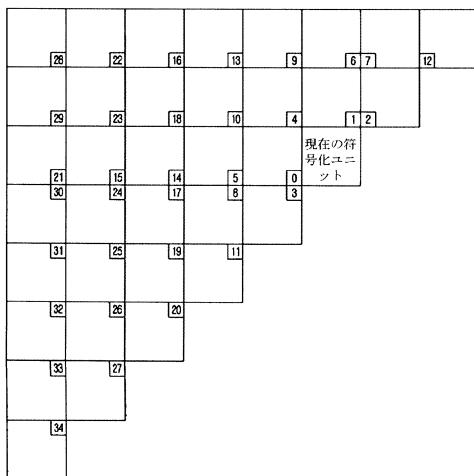


【図 1 4】

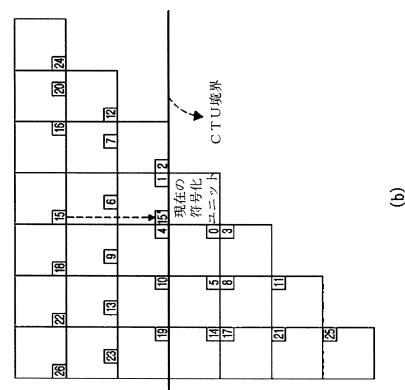


10

【図 1 5】

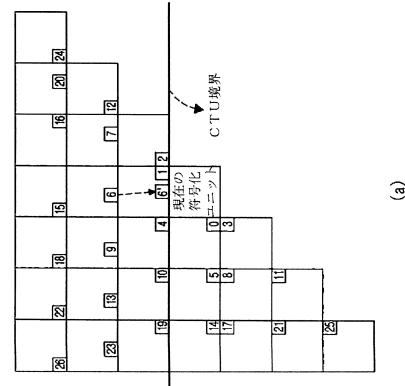


【図 1 6】



20

(b)



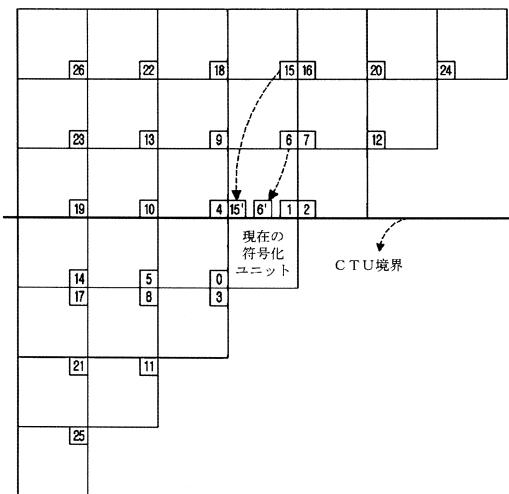
30

(a)

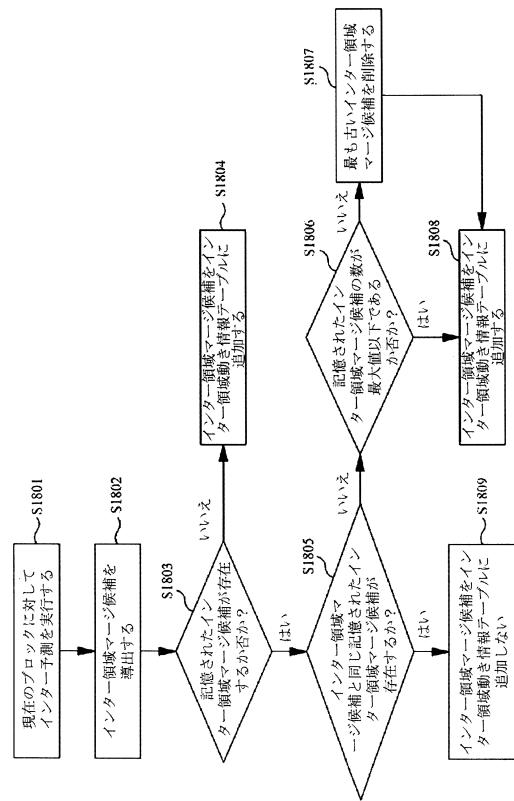
40

50

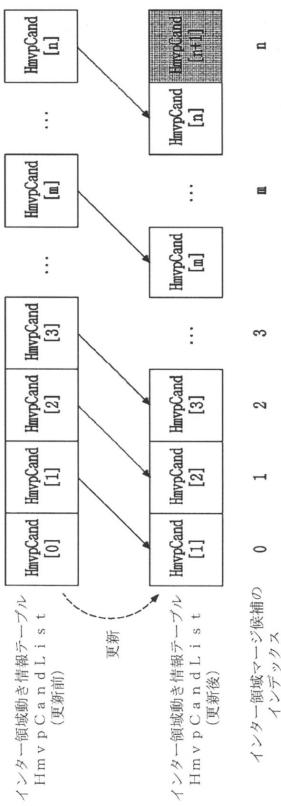
【図 17】



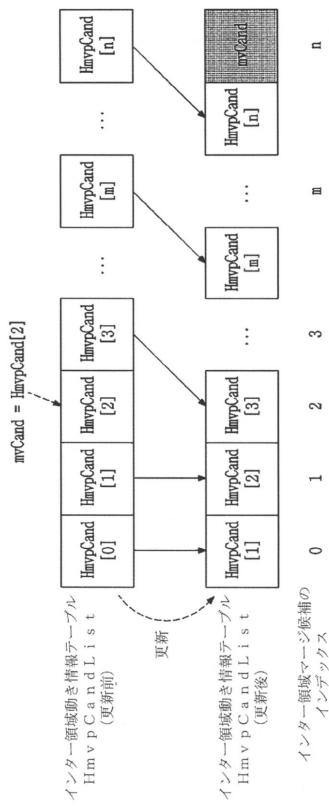
【図 18】



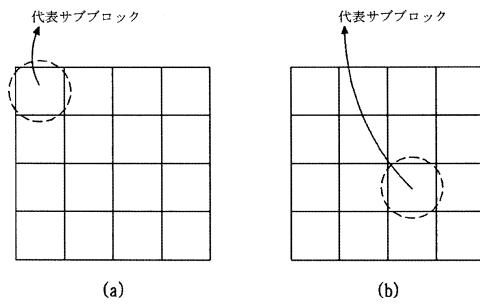
【図 19】



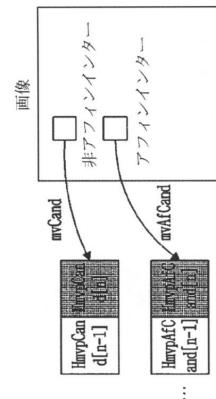
【図 20】



【図 2 1】

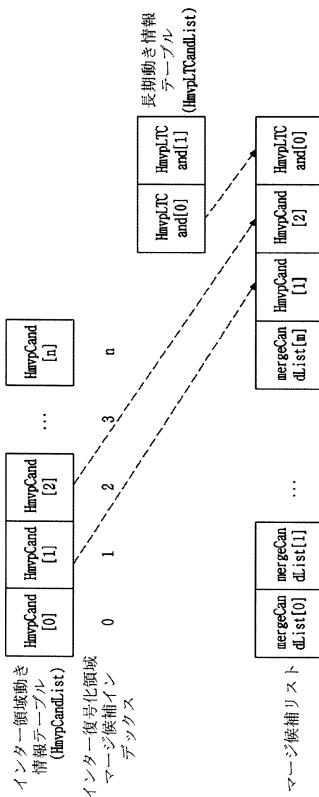


【図 2 2】

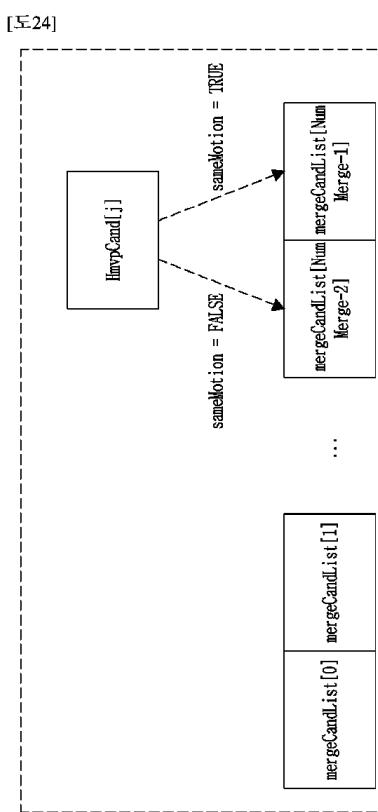


10

【図 2 3】



【図 2 4】

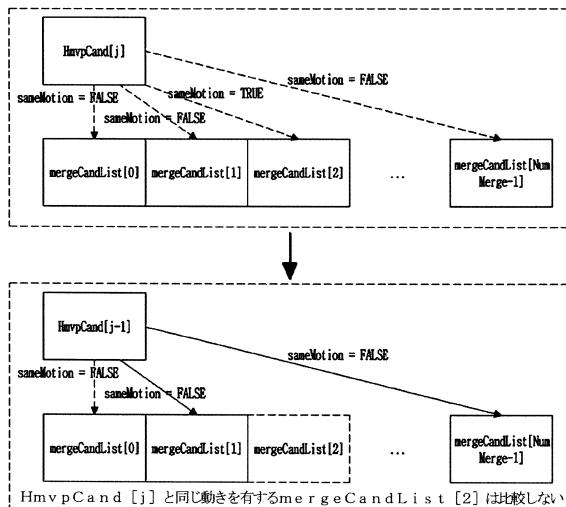


30

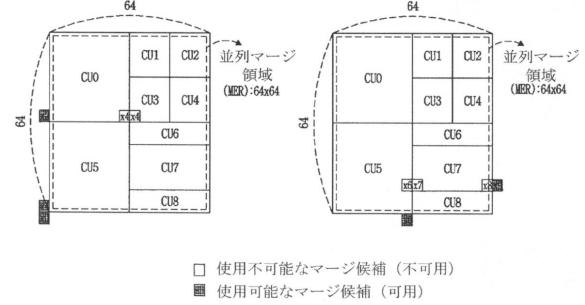
40

50

【図 2 5】

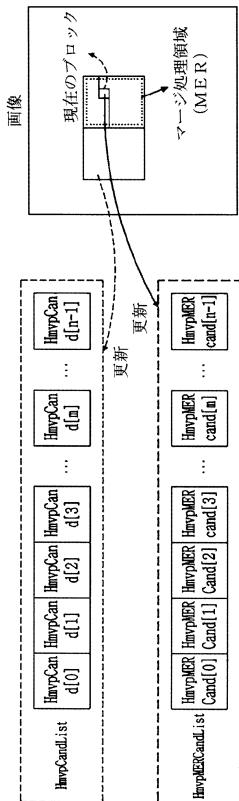


【図 2 6】

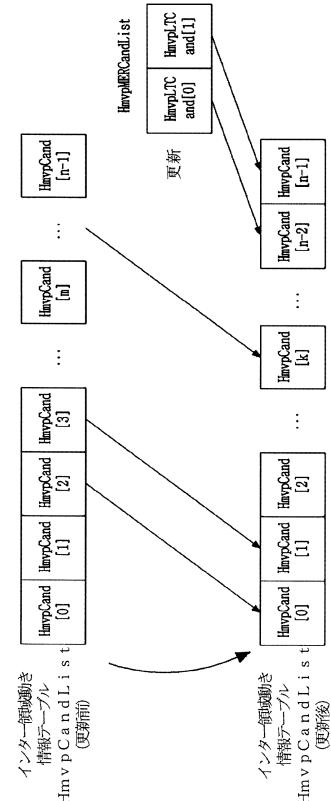


10

【図 2 7】



【図 2 8】



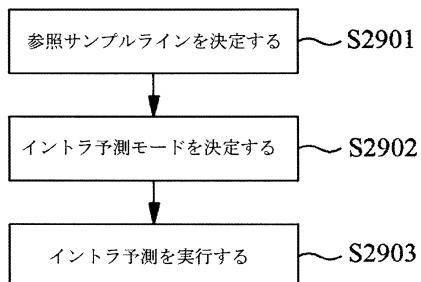
20

30

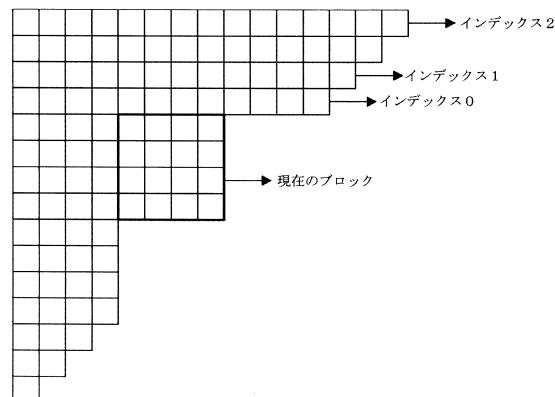
40

50

【図 2 9】

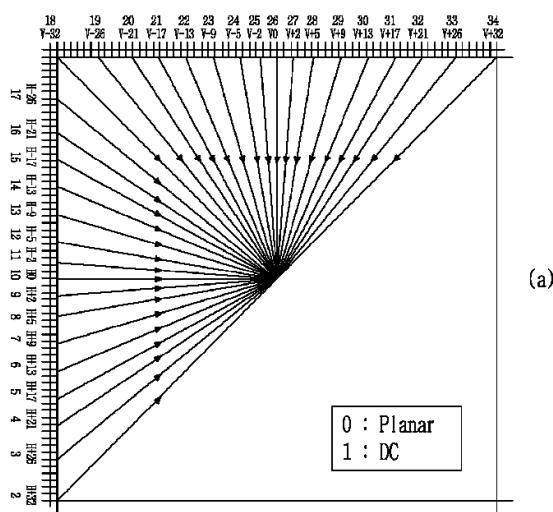


【図 3 0】



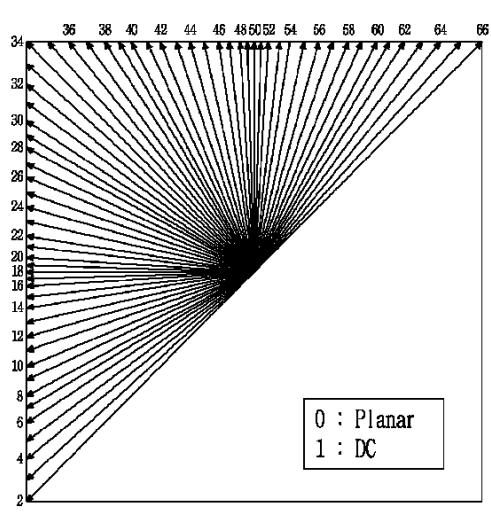
10

【図 3 1 (a)】



(a)

【図 3 1 (b)】



(b)

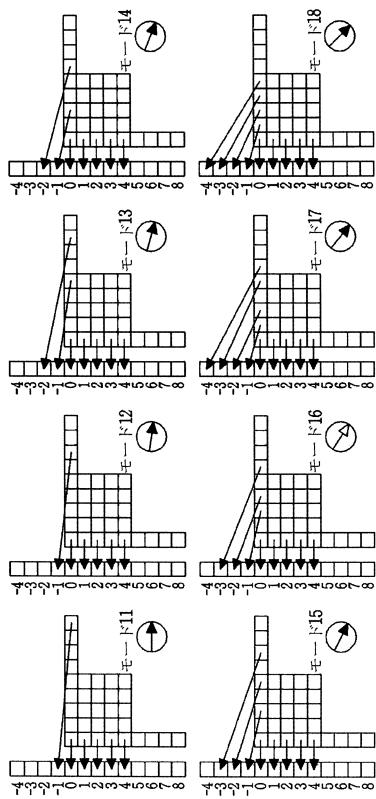
20

30

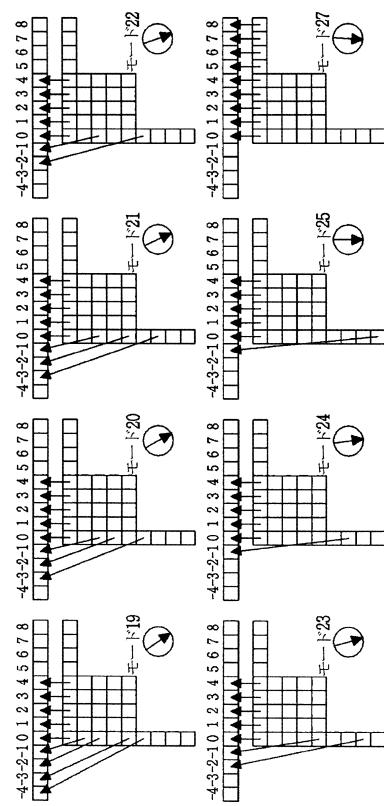
40

50

【図 3 2】



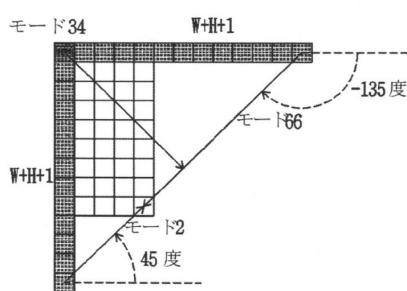
【図 3 3】



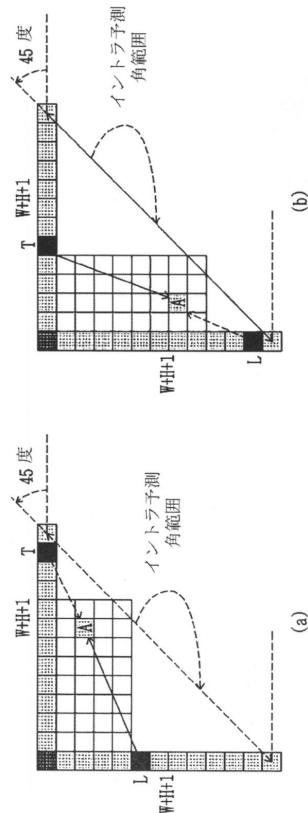
10

20

【図 3 4】



【図 3 5】



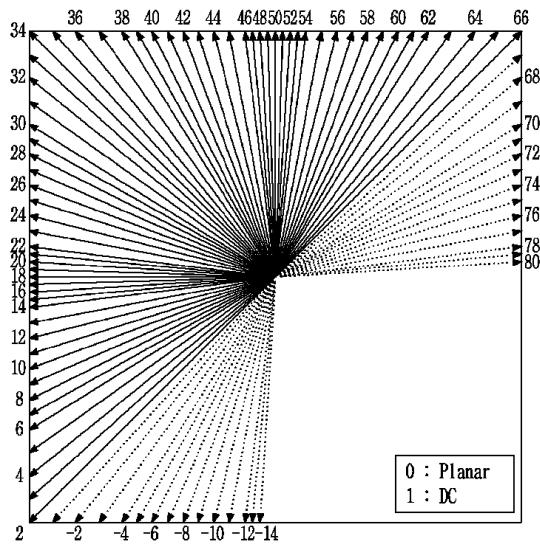
30

40

50

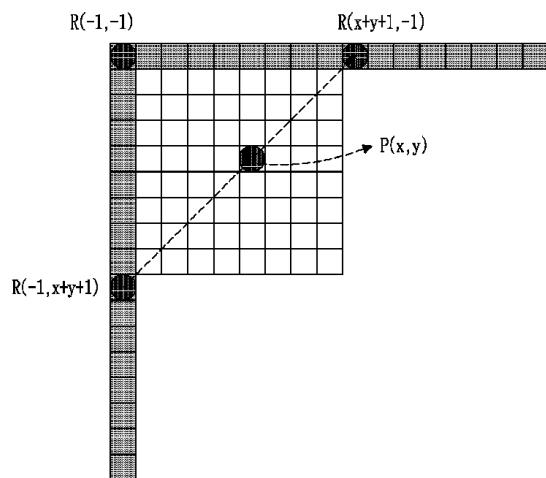
【図 3 6】

[도36]



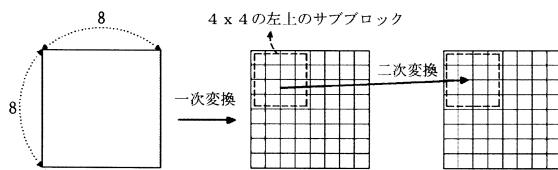
【図 3 7】

[도37]

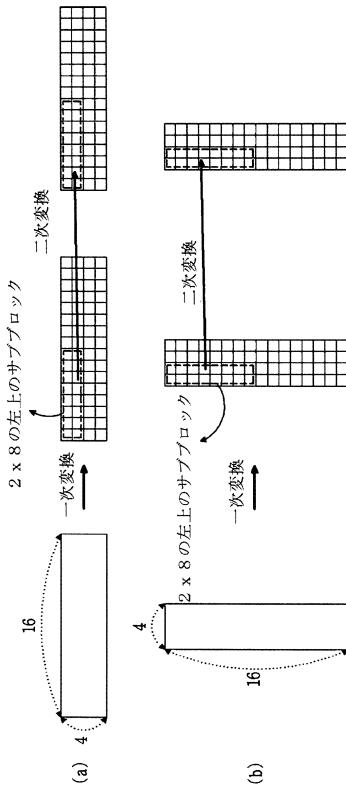


10

【図 3 8】



【図 3 9】



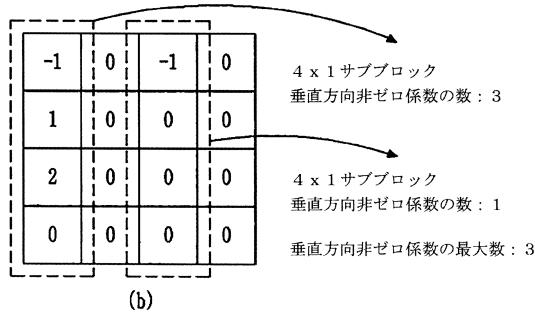
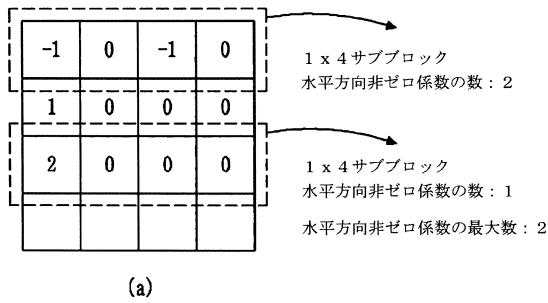
20

30

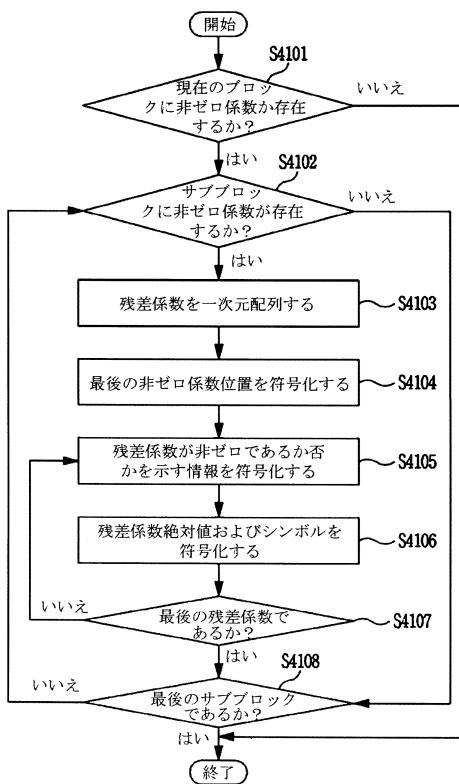
40

50

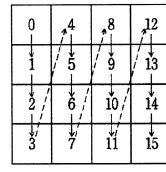
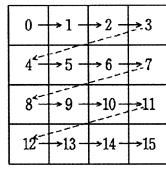
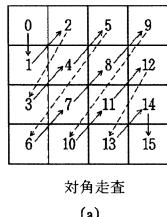
【図 4 0】



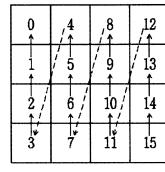
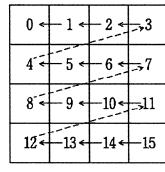
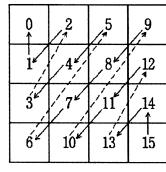
【図 4 1】



【図 4 2】

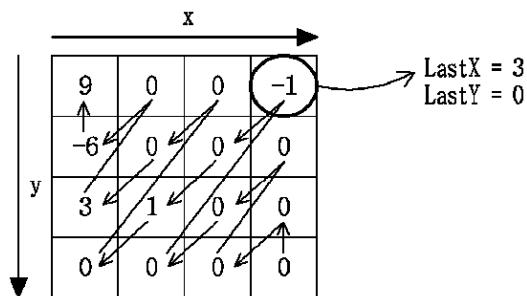


【図 4 3】

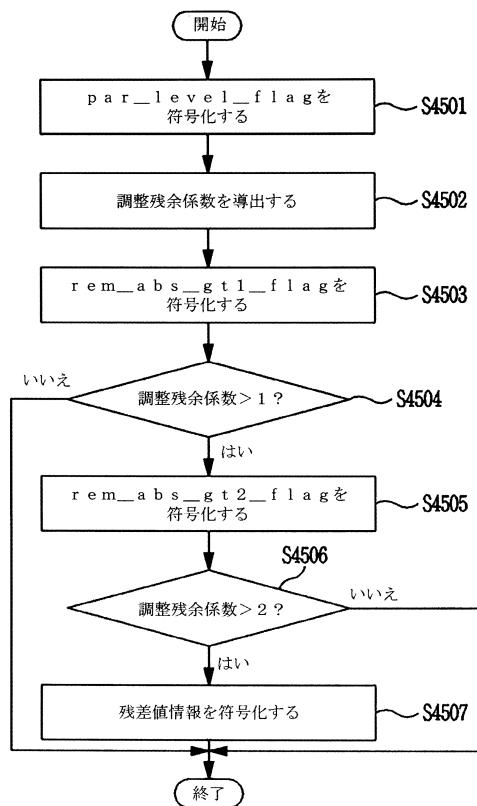


【図 4 4】

[図44]



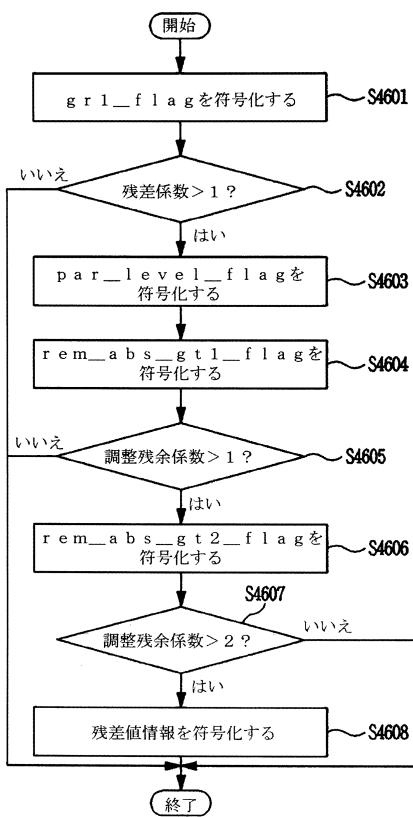
【図 4 5】



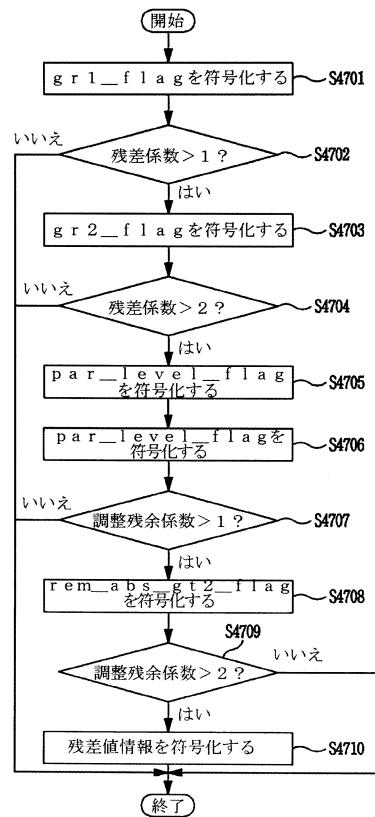
10

20

【図 4 6】



【図 4 7】

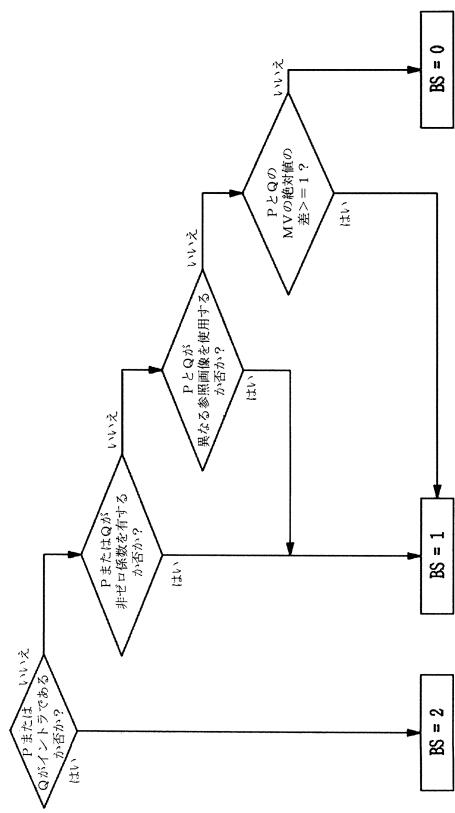


30

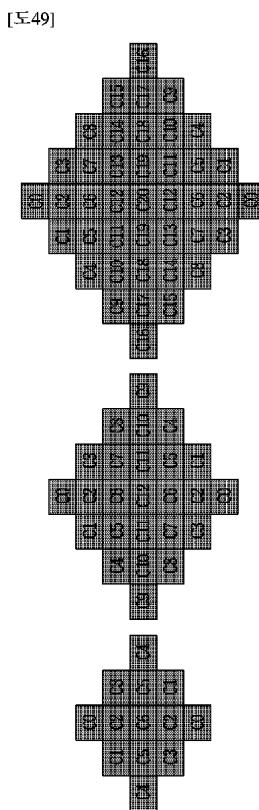
40

50

【図 4 8】



【図 4 9】



10

20

30

40

50

フロントページの続き

(33) 優先権主張国・地域又は機関

韓国(KR)

(31) 優先権主張番号 10-2018-0114346

(32) 優先日 平成30年9月21日(2018.9.21)

(33) 優先権主張国・地域又は機関

韓国(KR)

(31) 優先権主張番号 10-2018-0114347

(32) 優先日 平成30年9月21日(2018.9.21)

(33) 優先権主張国・地域又は機関

韓国(KR)

(31) 優先権主張番号 10-2019-0022754

(32) 優先日 平成31年2月26日(2019.2.26)

(33) 優先権主張国・地域又は機関

韓国(KR)

(74) 代理人 100107582

弁理士 関根 肇

(74) 代理人 100152205

弁理士 吉田 昌司

(74) 代理人 100137523

弁理士 出口 智也

(74) 代理人 100120385

弁理士 鈴木 健之

(72) 発明者 リー、ベゲン

大韓民国キヨンギ - ド、ソンナム - シ、ブンダン - グ、パンギョ - 口、393、214 - 1202

審査官 鉢呂 健

(56) 参考文献 国際公開第2020/060282(WO, A1)

特表2021-503247(JP, A)

国際公開第2020/007785(WO, A1)

SCHWARZ, Heiko et al. , CE7: Transform coefficient coding with reduced number of regular-coded bins (tests 7.1.3a, 7.1.3b) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 3-12 Oct. 2018, [JVET-L0274-v3] , JVET-L0274 (version 3) , ITU-T , 2018年10月03日 , < URL:https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0274-v3.zip > : JVET-L0274-v3.docx: pp.1-19, JVET-L0274_DraftTextForVVC-versionB.docx: pp.1-7

BROSS, Benjamin et al. , Versatile Video Coding (Draft 2) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting: Ljubljana, SI, 10-18 July 2018, [JVET-K1001-v5] , JVET-K1001 (version 5) , ITU-T , 2018年09月18日 , < URL: https://jvet-experts.org/doc_end_user/documents/11_Ljubljana/wg11/JVET-K1001-v5.zip > : JVET-K1001-v5.docx: pp.32-34,47-49

(58) 調査した分野 (Int.Cl. , D B名)

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