

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7560077号
(P7560077)

(45)発行日 令和6年10月2日(2024.10.2)

(24)登録日 令和6年9月24日(2024.9.24)

(51)国際特許分類

F I

H 0 4 N 19/70 (2014.01)

H 0 4 N 19/70

H 0 4 N 19/593 (2014.01)

H 0 4 N 19/593

請求項の数 11 (全41頁)

(21)出願番号	特願2023-206713(P2023-206713)	(73)特許権者	512297583
(22)出願日	令和5年12月7日(2023.12.7)		ヒョンダイ モーター カンパニー
(62)分割の表示	特願2021-577013(P2021-577013)		大韓民国, ソウル 0 6 7 9 7 , ソチョ
	)の分割		- グ, ホンルン路, 1 2
原出願日	令和2年6月24日(2020.6.24)	(73)特許権者	512297594
(65)公開番号	特開2024-19428(P2024-19428A)		キア コーポレーション
(43)公開日	令和6年2月9日(2024.2.9)		大韓民国, ソウル 0 6 7 9 7 , ソチョ
審査請求日	令和5年12月7日(2023.12.7)		- グ, ホンルン路, 1 2
(31)優先権主張番号	10-2019-0075163	(73)特許権者	518357128
(32)優先日	令和1年6月24日(2019.6.24)		イファ ユニバーシティ - インダストリー
(33)優先権主張国・地域又は機関	韓国(KR)		コラボレーション ファウンデーション
(31)優先権主張番号	10-2019-0080749		EWHA UNIVERSITY - I
(32)優先日	令和1年7月4日(2019.7.4)		NDUSTRY COLLABORAT
(33)優先権主張国・地域又は機関			ION FOUNDATION
	最終頁に続く		大韓民国 ソウル 0 3 7 6 0、ソデムン
			最終頁に続く

(54)【発明の名称】 動画データのイントラ予測コーディングのための方法及び装置

(57)【特許請求の範囲】

【請求項1】

ビデオデータを復号化する方法であって、

ビットストリームからビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを復号化するステップであって、前記イントラ予測タイプは、行列ベースのイントラ予測 (matrix based intra prediction: MIP) と正規イントラ予測 (regular intra prediction) を含むステップと、

前記シンタックスエレメントが指示する前記現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的に実行し、前記現在ブロックに対する予測ブロックを生成するステップと、

を含み、

前記正規イントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップは、

前記現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいてMPM (Most Probable Mode) 候補を導出して前記現在ブロックに対するMPMリストを構成するステップと、

前記MPMリストに基づいて前記現在ブロックに対する正規イントラ予測モードを導出するステップと、を含み、

前記行列ベースのイントラ予測を実行して前記現在ブロックに対する予測ブロックを生

成するステップは、

前記ビットストリームから前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを復号化するステップと、

前記現在ブロックの幅及び高さに基づいて前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップと、

前記行列ベースのイントラ予測モードに対して予め定義した行列と前記境界ベクトルとの間の行列 - ベクトル乗算に基づき、前記現在ブロックに対する予測サンプルを生成するステップと、

前記予測サンプルに基づいて前記現在ブロックに対する前記予測ブロックを導出するステップと、を含み、

10

前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップは、

前記ビットストリームから前記現在ブロックに隣接する左側周辺サンプルから導出した前記境界ベクトルの第 1 のエントリと前記現在ブロックに隣接する上側周辺サンプルから導出した前記境界ベクトルの第 2 のエントリとの間の接合順序を指示するシンタックスエレメントを復号化するステップと、

前記接合順序に従って前記第 1 のエントリと前記第 2 のエントリを接合して前記境界ベクトルを生成するステップと、を含むことを特徴とする方法。

【請求項 2】

前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントは、

20

前記現在ブロックの幅及び高さに対して許容される複数の行列ベースのイントラ予測モードのうちの 1 つを特定する切捨バイナリコード (`truncated binary code`) であることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記現在ブロックに隣接する周辺サンプルを利用して境界ベクトルを導出するステップは、

前記現在ブロックに隣接する左側周辺サンプルのセット及び上側周辺サンプルのセットのうち、前記現在ブロックに対する行列ベースのイントラ予測モードに依存して、選択された 1 つあるいは 2 つのセットから前記境界ベクトルを導出するステップを含むことを特徴とする請求項 2 に記載の方法。

30

【請求項 4】

前記境界ベクトルのエントリは、

前記現在ブロックの幅及び高さによって、前記現在ブロックに隣接する左側周辺サンプルからダウンサンプリングした値、又は前記現在ブロックに隣接する左側周辺サンプルで満たされることを特徴とする請求項 2 に記載の方法。

【請求項 5】

前記予測サンプルに基づいて前記現在ブロックに対する予測ブロックを導出するステップは、

前記予測サンプルを前記予測ブロック内の位置に割り当てるステップと、

40

前記予測サンプル、前記現在ブロックに隣接する左側周辺サンプル、及び前記現在ブロックに隣接する上側周辺サンプルに対する線形補間を利用して、前記予測ブロック内の前記予測サンプルが割り当てられていない位置についての予測サンプル値を作成するステップと、を含むことを特徴とする請求項 2 に記載の方法。

【請求項 6】

ビデオデータを符号化する方法であって、

ビットストリームにビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを符号化するステップであって、前記イントラ予測タイプは、行列ベースのイントラ予測と正規イントラ予測を含むステップと、前記現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的にを行い、

50

前記現在ブロックに対する予測ブロックを生成するステップと、

を含み、

前記正規イントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップは、

前記現在ブロックの正規イントラ予測モードを決定するステップと、

前記現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいてMPM (Most Probable Mode) 候補を導出して前記現在ブロックに対するMPM リストを構成するステップと、

前記MPM リストに基づいて前記現在ブロックの正規イントラ予測モードを指示する少なくとも1つのシンタックスエレメントを前記ビットストリームに符号化するステップと、
を含み、

前記行列ベースのイントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップは、

前記ビットストリームに前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを符号化するステップと、

前記現在ブロックの幅及び高さに基づいて前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップと、

前記行列ベースのイントラ予測モードに対して予め定義した行列と前記境界ベクトルとの間の行列 - ベクトル乗算に基づき、前記現在ブロックに対する予測サンプルを生成するステップと、

前記予測サンプルに基づいて前記現在ブロックに対する予測ブロックを導出するステップと、を含み、

前記現在ブロックに隣接する周辺サンプルを利用して境界ベクトルを導出するステップは、

前記ビットストリームに前記現在ブロックに隣接する左側周辺サンプルから導出した前記境界ベクトルの第1のエントリと前記現在ブロックに隣接する上側周辺サンプルから導出した前記境界ベクトルの第2のエントリとの間の接合順序を指示するシンタックスエレメントを符号化するステップと、

前記接合順序に従って前記第1のエントリと前記第2のエントリを接合して前記境界ベクトルを生成するステップと、を含むことを特徴とする、方法。

【請求項7】

前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントは、

前記現在ブロックの幅及び高さに対して許容される複数の行列ベースのイントラ予測モードのうちの1つを特定する切捨バイナリコード (truncated binary code) であることを特徴とする請求項6に記載の方法。

【請求項8】

前記現在ブロックに隣接する周辺サンプルを利用して境界ベクトルを導出するステップは、

前記現在ブロックに隣接する左側周辺サンプルのセット及び上側周辺サンプルのセットのうち、前記現在ブロックに対する行列ベースのイントラ予測モードに依存して、選択された1つ又は2つのセットから前記境界ベクトルを導出するステップを含むことを特徴とする請求項7に記載の方法。

【請求項9】

前記境界ベクトルのエントリは、

前記現在ブロックの幅及び高さによって、前記現在ブロックに隣接する左側周囲サンプルからダウンサンプリングした値、又は前記現在ブロックに隣接する左側周囲サンプルで満たされることを特徴とする請求項7に記載の方法。

【請求項10】

前記予測サンプルに基づいて前記現在ブロックに対する予測ブロックを導出するステッ

10

20

30

40

50

ブは、

前記予測サンプルを前記予測ブロック内の位置に割り当てるステップと、

前記予測サンプル、前記現在ブロックに隣接する左側周辺サンプル、及び前記現在ブロックに隣接する上側周辺サンプルに対する線形補間を利用して、前記予測ブロック内の前記予測サンプルが割り当てられていない位置についての予測サンプル値を作成するステップを含むことを特徴とする請求項 7 に記載の方法。

【請求項 11】

プロセッサにビデオデータブロックの符号化データを含むビットストリームを送信させるための方法が記憶されたコンピュータ読み取り可能な記録媒体であって、

前記方法は、

ビデオデータブロックをビットストリームに符号化するステップと、

前記ビットストリームをビデオ復号化装置に送信するステップと、

を含み、

前記ビデオデータブロックをビットストリームに符号化するステップは、

前記ビットストリームにビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを符号化するステップであって、前記イントラ予測タイプは、行列ベースのイントラ予測と正規イントラ予測を含むステップと、

前記現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的にを行い、前記現在ブロックに対する予測ブロックを生成するステップと、

を含み、

前記正規イントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップは、

前記現在ブロックの正規イントラ予測モードを決定するステップと、

前記現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいて MPM (Most Probable Mode) 候補を導出して前記現在ブロックに対する MPM リストを構成するステップと、

前記 MPM リストに基づいて、前記現在ブロックの正規イントラ予測モードを指示する少なくとも 1 つのシンタックスエレメントを前記ビットストリームに符号化するステップと、を含み、

前記行列ベースのイントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップは、

前記ビットストリームに前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを符号化するステップと、

前記現在ブロックの幅及び高さに基づいて前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップと、

前記行列ベースのイントラ予測モードに対して予め定義した行列と前記境界ベクトルとの間の行列 - ベクトル乗算に基づき、前記現在ブロックに対する予測サンプルを生成するステップと、

前記予測サンプルに基づいて前記現在ブロックに対する予測ブロックを導出するステップと、を含み、

前記現在ブロックに隣接する周辺サンプルを利用して境界ベクトルを導出するステップは、

前記ビットストリームに前記現在ブロックに隣接する左側周辺サンプルから導出した前記境界ベクトルの第 1 のエントリと前記現在ブロックに隣接する上側周辺サンプルから導出した前記境界ベクトルの第 2 のエントリとの間の接合順序を指示するシンタックスエレメントを符号化するステップと、

前記接合順序に従って前記第 1 のエントリと前記第 2 のエントリを接合して前記境界ベクトルを生成するステップと、を含むことを特徴とする、記録媒体。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】

【0001】

本発明は、動画データの符号化及び復号化に関する。

【背景技術】

【0002】

動画データは、音声データや静止画データなどに比べて多くのデータ量を有するため、圧縮のための処理なしにそれ自体を保存又は伝送するためにはメモリを含む多くのハードウェアリソースを必要とする。

【0003】

したがって、通常、動画データを保存又は伝送する際には、エンコーダを用いて動画データを圧縮して保存又は伝送し、デコーダでは圧縮した動画データを受信して圧縮を解除して再生する。このような動画圧縮技術としては、H.264/AVCをはじめ、H.264/AVCに比べて約40%程度の符号化効率を向上させたHEVC(High Efficiency Video Coding)が存在する。

10

【0004】

しかし、ピクチャのサイズ及び解像度、フレームレートが徐々に増加しており、これに伴い符号化すべきデータ量も増加しているため、従来の圧縮技術よりも符号化効率が良く、画質改善効果も高い新たな圧縮技術が求められている。

【発明の概要】

【発明が解決しようとする課題】

20

【0005】

本発明の目的は、動画データのブロックをイントラ予測コーディングするための改善された技術を提供することにある。

【課題を解決するための手段】

【0006】

本発明の一態様による方法は、ビデオデータを復号化する方法であって、ビットストリームからビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを復号化するステップであって、前記イントラ予測タイプは行列ベースのイントラ予測(matrix based intra prediction: MIP)と正規イントラ予測(regular intra prediction)を含むステップと、前記シンタックスエレメントが指示する前記現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的に実行し、前記現在ブロックに対する予測ブロックを生成するステップと、を含むことを特徴とする。

30

【0007】

前記方法は、前記正規イントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップの一部であって、前記方法は、前記現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいてMPM(Most Probable Mode)候補を導出して前記現在ブロックに対するMPMリストを構成するステップと、前記MPMリストに基づいて前記現在ブロックに対する正規イントラ予測モードを導出するステップを含む。前記MPM候補を導出する際に、前記周辺ブロックのイントラ予測タイプが行列ベースのイントラ予測である場合に、前記周辺ブロックの正規イントラ予測モードはプラナー(PLANAR)モードに設定される。

40

【0008】

前記方法は、前記行列ベースのイントラ予測を実行して前記現在ブロックに対する予測ブロックを生成するステップの一部として、前記ビットストリームから前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを復号化するステップと、現在ブロックのサイズ及び形状に基づいて前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップと、前記行列ベースのイントラ予測モードに対して予め定義した行列と前記境界ベクトルとの間の行列・ベクトル乗算に基づき、前記現在ブロックに対する予測サンプルを生成するステップ、及び、前記予測サン

50

ブルに基づいて前記現在ブロックに対する予測ブロックを導出するステップと、を含む。

【 0 0 0 9 】

本発明の一態様による装置は、ビデオデータを復号化する装置であって、ビットストリームからビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを復号化する復号化部であって、前記イントラ予測タイプは行列ベースのイントラ予測と正規イントラ予測を含む復号化部と、前記シンタックスエレメントが指示する現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的に実行して前記現在ブロックに対する予測ブロックを生成するイントラ予測部と、を含むことを特徴とする。

【 0 0 1 0 】

前記イントラ予測部は、前記正規イントラ予測を実行して前記現在ブロックに対する予測ブロックを生成することの一部として、前記現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいてM P M候補を導出して前記現在ブロックに対するM P Mリストを構成し、前記M P Mリストに基づいて前記現在ブロックに対する正規イントラ予測モードを導出する。前記M P M候補を導出するに際して、前記イントラ予測部は、前記周辺ブロックのイントラ予測タイプが行列ベースのイントラ予測である場合に、前記周辺ブロックの正規イントラ予測モードをプラナー (P L A N A R) モードに設定する (みなす) 。

【 0 0 1 1 】

前記行列ベースのイントラ予測を実行して前記現在ブロックに対する予測ブロックを生成することの一部として、前記イントラ予測部は、前記ビットストリームから前記現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを復号化するステップと、前記現在ブロックのサイズ及び形状に基づいて前記現在ブロックに隣接する周囲サンプルを利用して境界ベクトルを導出するステップと、前記行列ベースのイントラ予測モードに対して予め定義した行列と前記境界ベクトルとの間の行列 - ベクトル乗算に基づいて前記現在ブロックに対する予測サンプルを生成するステップと、前記予測サンプルに基づいて前記現在ブロックに対する予測ブロックを導出するステップと、を実行する。

【 発明の効果 】

【 0 0 1 2 】

本発明によれば、イントラ予測の複雑さを低減し、符号化効率が向上したコーディング技術を提供することができる。

【 図面の簡単な説明 】

【 0 0 1 3 】

【 図 1 】 本発明の技術を具現することができる映像符号化装置の例示的なブロック図である。

【 図 2 】 Q T B T T 構造を用いてブロックを分割する方法を説明するための図である。

【 図 3 a 】 複数のイントラ予測モードを示す図である。

【 図 3 b 】 広角イントラ予測モードを含む複数のイントラ予測モードを示す図である。

【 図 4 】 本発明の技術を具現できる映像復号化装置の例示的なブロック図である。

【 図 5 】 本発明の技法で使用されるM I P技術の主要なプロセスを図式化した概念図である。

【 図 6 】 本発明の一実施例によるM I Pベースでブロックを予測する概略的なプロセスを示すフローチャートである。

【 図 7 a 】 左隣のサンプルを用いて行列 - ベクトル乗算演算に inputs する境界ベクトルを構成する例示的な方法を示す概念図である。

【 図 7 b 】 左隣のサンプルを用いて行列 - ベクトル乗算演算に inputs する境界ベクトルを構成する例示的な方法を示す概念図である。

【 図 7 c 】 左隣のサンプルを用いて行列 - ベクトル乗算演算に inputs する境界ベクトルを構成する例示的な方法を示す概念図である。

10

20

30

40

50

【図 8 a】上隣のサンプルを使用して行列 - ベクトル乗算演算に入力する境界ベクトルを構成する例示的な方法を示す概念図である。

【図 8 b】上隣のサンプルを使用して行列 - ベクトル乗算演算に入力する境界ベクトルを構成する例示的な方法を示す概念図である。

【図 8 c】上隣のサンプルを使用して行列 - ベクトル乗算演算に入力する境界ベクトルを構成する例示的な方法を示す概念図である。

【図 9】左隣のサンプル、境界ベクトル、及びそれから予測した予測サンプルを示す。

【図 10】上隣のサンプル、境界ベクトル、及びそれから予測した予測サンプルを示す。

【図 11】本発明の一実施例によるビデオデータを復号化する方法を例示するフローチャートである。

10

【発明を実施するための形態】

【0014】

以下、本発明の一部の実施例を例示的な図面を通して詳しく説明する。各図面の構成要素に識別符号を付加する際には、同一の構成要素に対しては、たとえ他の図面上に表示されても可能な限り同一の符号を有するようにしている。なお、本発明を説明するにあたり、関連する公知の構成又は機能についての具体的な説明が本発明の要旨を曖昧にすると判断した場合には、その詳しい説明は省く。

【0015】

図 1 は、本発明の技術を具現する映像符号化装置の例示的なブロック図である。以下では、図 1 を参照して映像符号化装置とこの装置の下位構成について説明する。

20

【0016】

映像符号化装置は、ピクチャ分割部 110、予測部 120、減算器 130、変換部 140、量子化部 145、並べ替え部 150、エントロピー符号化部 155、逆量子化部 160、逆変換部 165、加算器 170、フィルタ部 180、及びメモリ 190 を含んでなる。

【0017】

映像符号化装置の各構成要素は、ハードウェア又はソフトウェアで具現してもよいし、ハードウェア及びソフトウェアの組み合わせで具現してもよい。さらに、各構成要素の機能がソフトウェアで具現され、マイクロプロセッサが各構成要素に対応するソフトウェアの機能を実行するように具現してもよい。

【0018】

30

1 つの映像（ビデオ）は複数のピクチャから構成される。各ピクチャは複数の領域に分割され、各領域毎に符号化が行われる。例えば、1 つのピクチャは、1 つ以上のタイル（Tile）又は / 及びスライス（Slice）に分割される。ここで、1 つ以上のタイルをタイルグループ（Tile Group）と定義する。各タイル又は / 及びスライスは、1 つ以上の CTU（Coding Tree Unit）に分割される。そして、各 CTU は、ツリー構造によって 1 つ以上の CU（Coding Unit）に分割される。各 CU に適用する情報は CU のシンタックスとして符号化され、1 つの CTU に含まれる CU に共通に適用する情報は CTU のシンタックスとして符号化される。また、1 つのスライス内の全てのブロックに共通に適用する情報は、スライスヘッダのシンタックスとして符号化され、1 つのピクチャを構成する全てのブロックに適用する情報は、ピクチャパラメータセット（PPS、Picture Parameter Set）又はピクチャヘッダに符号化する。さらに、複数のピクチャが共通に参照する情報は、シーケンスパラメータセット（SPS、Sequence Parameter Set）に符号化する。そして、1 つ以上の SPS が共通に参照する情報は、ビデオパラメータセット（VPS、Video Parameter Set）に符号化する。さらに、1 つのタイル又はタイルグループに共通に適用する情報は、タイル又はタイルグループヘッダのシンタックスとして符号化してもよい。

40

【0019】

ピクチャ分割部 110 は、CTU（Coding Tree Unit）の大きさを決定する。CTU のサイズに関する情報（CTU size）は SPS 又は PPS のシンタッ

50

クスとして符号化し、映像復号化装置に伝達する。

【0020】

ピクチャ分割部110は、映像を構成する各ピクチャを予め決まった大きさを有する複数のCTU(Coding Tree Unit)に分割した後に、ツリー構造(tree structure)を用いてCTUを繰り返し(recursively)分割する。ツリー構造におけるリーフノード(leaf node)が符号化の基本単位であるCU(coding unit)となる。

【0021】

ツリー構造としては、上位ノード(あるいは親ノード)が同じサイズの4つの下位ノード(あるいは子ノード)に分割されるクワッドツリー(Quad Tree、QT)、又は上位ノードが2つの下位ノードに分割されるバイナリツリー(Binary Tree、BT)、又は上位ノードが1:2:1の比率で3つの下位ノードに分割されるターナリツリー(Ternary Tree、TT)、又はこれらのQT構造、BT構造、及びTT構造のうちの2つ以上を混用した構造であってもよい。例えば、QTBT(Quad Tree plus Binary Tree)構造を使用するか、又はQTBT TT(Quad Tree plus Binary Tree Ternary Tree)構造を使用する。ここで、BT TTを合わせてMTT(Multiple-Type Tree)と呼ぶ。

【0022】

図2は、QTBT TT分割ツリー構造を示す。図2に示すように、CTUは最初にQT構造に分割される。クワッドツリー分割は、分割ブロック(splitting block)のサイズがQTで許容されるリーフノードの最小ブロックサイズ(Min QTS size)に達するまで繰り返される。QT構造の各ノードが下位レイヤーの4つのノードに分割されるか否かを指示する第1のフラグ(QT_split_flag)は、エントロピー符号化部155によって符号化され、映像復号化装置でシグナリングされる。QTのリーフノードがBTで許容されるルートノードの最大ブロックサイズ(Max BTS size)よりも大きくない場合、BT構造又はTT構造のうちのいずれか1つ以上にさらに分割される。BT構造及び/又はTT構造では、複数の分割方向が存在してもよい。例えば、該当ノードのブロックが水平に分割される方向と垂直に分割される方向の2つが存在し得る。図2に示すように、MTT分割を開始すると、ノードを分割したか否かを示す第2のフラグ(MTT_split_flag)と、分割した場合には追加で分割方向(vertical又はhorizontal)を示すフラグ及び/又は分割タイプ(BinaryもしくはTernary)を示すフラグが、エントロピー符号化部155によって符号化され、映像復号化装置にシグナリングされる。対案として、各ノードが下位レイヤーの4つのノードに分割されるか否かを示す第1のフラグ(QT_split_flag)を符号化する前に、そのノードを分割するか否かを指示するCU分割フラグ(split_cu_flag)を符号化してもよい。CU分割フラグ(split_cu_flag)の値が分割しないことを指示する場合、該当ノードのブロックが分割ツリー構造におけるリーフノード(leaf node)となり、符号化の基本単位であるCU(coding unit)となる。CU分割フラグ(split_cu_flag)の値が分割することを指示する場合、映像符号化装置は、上述したように第1のフラグから符号化を開始する。

【0023】

ツリー構造の他の例としてQTBTを使用する場合、当該ノードのブロックを同一サイズの2つのブロックに横に分割するタイプ(すなわち、symmetric horizontal splitting)と縦に分割するタイプ(すなわち、symmetric vertical splitting)の2つの方法がある。BT構造の各ノードを下位レイヤーのブロックに分割するか否かを指示する分割フラグ(split_flag)及び、分割するタイプを指示する分割タイプ情報がエントロピー符号化部155によって符号化されて映像復号化装置に伝達される。一方、該当ノードのブロックを互いに非対称型の2つのブロックに分割するタイプがさらに存在してもよい。非対称型には、該当ノード

ドのブロックを 1 : 3 のサイズ比を有する 2 つの長方形ブロックに分割する形態を含んでもよく、あるいは、該当ノードのブロックを対角線方向に分割する形態を含んでもよい。

【 0 0 2 4 】

C U は、C T U からの Q T B T 又は Q T B T T T 分割によって様々なサイズを有する。以下では、符号化又は復号化する C U (すなわち、Q T B T T T T のリーフノード) に該当するブロックを「現在ブロック」と称する。Q T B T T T 分割の採用によって、現在ブロックの形状は正方形だけでなく長方形でもよい。

【 0 0 2 5 】

予測部 1 2 0 は、現在ブロックを予測して予測ブロックを生成する。予測部 1 2 0 は、イントラ予測部 1 2 2 とインター予測部 1 2 4 を含む。

10

【 0 0 2 6 】

一般に、ピクチャ内の現在ブロックはそれぞれ予測的にコーディングされる。一般に、現在ブロックの予測は、(現在ブロックを含むピクチャからのデータを使用する)イントラ予測技術、又は(現在ブロックを含むピクチャの前にコーディングしたピクチャからのデータを使用する)インター予測技術を使用して実行される。インター予測は、一方向予測と双方向予測の両方を含む。

【 0 0 2 7 】

イントラ予測部 1 2 2 は、現在ブロックを含む現在ピクチャ内で現在ブロックの周辺に位置するピクセル(参照ピクセル)を用いて現在ブロック内のピクセルを予測する。予測方向によって複数のイントラ予測モードが存在する。例えば、図 3 a に示すように、複数のイントラ予測モードは、プラナー(p l a n a r)モード及び D C モードを含む 2 つの非方向性モードと、6 5 の方向性モードを含む。各予測モードによって使用する周辺ピクセルと演算式が異なるように定義される。下表は、イントラ予測モードの番号と名称を列挙する。

20

【 0 0 2 8 】

【表 1】

イントラ予測モード	名称
0	INTRA_PLANAR
1	INTRA_DC
2 ... 66	INTRA_ANGULAR2 ... INTRA_ANGULAR66

30

【 0 0 2 9 】

長方形の現在ブロックに対する効率的な方向性予測のために、図 3 b に点線矢印で示す方向性モード(6 7 ~ 8 0 番、- 1 ~ - 1 4 番イントラ予測モード)をさらに使用する。これらは「広角イントラ予測モード(w i d e a n g l e i n t r a - p r e d i c t i o n m o d e s)」と称する。図 3 b で、矢印は、予測に使用した対応する参照サンプルを指すものであり、予測方向を示すものではない。予測方向は矢印が指す方向とは反対である。広角イントラ予測モードは、現在ブロックが長方形のときに追加のビット伝送なしに特定の方向性モードを反対方向に予測するモードである。この場合、広角イントラ予測モードのうち、長方形の現在ブロックの幅と高さの比率によって、現在ブロックに利用可能な一部の広角イントラ予測モードを決定する。例えば、4 5 度よりも小さい角度を有する広角イントラ予測モード(6 7 ~ 8 0 番イントラ予測モード)は、現在ブロックが、高さが幅よりも小さい長方形である場合に利用可能であり、- 1 3 5 度よりも大きい角度を有する広角イントラ予測モード(- 1 ~ - 1 4 番イントラ予測モード)は、現在ブロックが、高さが幅よりも大きい長方形の形である場合に利用可能である。

40

【 0 0 3 0 】

イントラ予測部 1 2 2 は、現在ブロックを符号化するために使用するイントラ予測モードを決定する。一部の例では、イントラ予測部 1 2 2 は、いくつかのイントラ予測モードを使用して現在ブロックを符号化し、テストしたモードから使用する適切なイントラ予測

50

モードを選択してもよい。例えば、イントラ予測部 122 は、いくつかのテストしたイントラ予測モードに対するレート歪み (rate-distortion) 分析を用いてレート歪み値を計算し、テストしたモードの中で最良のレート歪み特徴を有するイントラ予測モードを選択してもよい。

【0031】

イントラ予測部 122 は、複数のイントラ予測モードの中から 1 つのイントラ予測モードを選択し、選択したイントラ予測モードによって決定される周辺ピクセル (参照ピクセル) と演算式を用いて現在ブロックを予測する。選択したイントラ予測モードに関する情報は、エントロピー符号化部 155 によって符号化されて映像復号化装置に伝達される。

【0032】

また、イントラ予測部 122 は、後述する行列ベースのイントラ予測 (Matrix-based Intra Prediction: MIP) を用いて、現在ブロックに対する予測ブロックを生成してもよい。イントラ予測部 122 は、現在ブロックの左側に再構成したサンプルと現在ブロックの上側に再構成したサンプルから導出した境界ベクトルと、予め定義した行列、及びオフセットベクトルを用いて現在ブロックに対する予測ブロックを生成してもよい。

【0033】

インター予測部 124 は、動き補償過程を通じて現在ブロックに対する予測ブロックを生成する。インター予測部 124 は、現在ピクチャよりも先に符号化及び復号化した参照ピクチャ内で現在ブロックに最も類似したブロックを探索し、その探索したブロックを用いて現在ブロックに対する予測ブロックを生成する。そして、現在ピクチャ内の現在ブロックと参照ピクチャ内の予測ブロックとの間の変位 (displacement) に該当する動きベクトル (motion vector) を生成する。一般に、動き推定は、ルマ (luma) 成分に対して行われ、ルマ成分に基づいて計算したモーションベクトルはルマ成分及びクロマ (chroma) 成分の両方に対して用られる。現在ブロックを予測するために用いられる参照ピクチャに関する情報及び動きベクトルに関する情報を含む動き情報は、エントロピー符号化部 155 によって符号化されて映像復号化装置に伝達される。

【0034】

減算器 130 は、現在ブロックからイントラ予測部 122 又はインター予測部 124 によって生成された予測ブロックを減算して残差ブロックを生成する。

【0035】

変換部 140 は、残差ブロックを 1 つ以上の変換ブロックに分割し、変換を 1 つ以上の変換ブロックに適用し、変換ブロックの残差値をピクセルドメインから周波数ドメインに変換する。周波数ドメインにおいて、変換されたブロックは、1 つ以上の変換係数値を含む係数ブロックと呼ばれる。変換には 2 次元変換カーネルを使用し、横方向変換と縦方向変換にそれぞれ 1 次元変換カーネルを使用する。変換カーネルは、離散コサイン変換 (DCT)、離散サイン変換 (DST) などに基づく。

【0036】

変換部 140 は、残差ブロックの全体サイズを変換単位として用いて残差ブロック内の残差信号を変換する。また、変換部 140 は、残差ブロックを横方向又は縦方向に 2 つのサブブロックに分割し、変換を 2 つのサブブロックのうちの 1 つにのみ行う。したがって、変換ブロックのサイズは、残差ブロックのサイズ (したがって予測ブロックのサイズ) と異なってもよい。変換が実行されないサブブロックには、ノンゼロ残差サンプル値が存在しないか、又は非常に希少である。変換が実行されないサブブロックの残差サンプルはシグナリングされず、画像復号化装置によってすべて「0」と見なされる。分割方向と分割比率によって、複数のパーティションタイプが存在する。変換部 140 は、残差ブロックのコーディングモード (又は変換モード) に関する情報 (例えば、残差ブロックを変換したか、又は残差サブブロックを変換したかを示す情報、残差ブロックをサブブロックに分割するために選択したパーティションタイプを示す情報、変換が行われるサブブロック

10

20

30

40

50

を識別する情報などを含む情報)をエントロピー符号化部155に提供する。エントロピー符号化部155は、残差ブロックのコーディングモード(あるいは変換モード)に関する情報を符号化する。

【0037】

量子化部145は、変換部140から出力する変換係数を量子化し、量子化した変換係数をエントロピー符号化部155に出力する。量子化部145は、任意のブロック又はフレームに対し、変換なしに、関連する残差ブロックを直ちに量子化してもよい。

【0038】

並べ替え部150は、量子化した残差値に対して係数値の並べ替えを行う。並べ替え部150は、係数スキニング(coefficient scanning)を介して2次元の係数アレイを1次元の係数シーケンスに変更する。例えば、並べ替え部150では、千鳥状スキャン(Zig-Zag Scan)又は対角線スキャン(Diagonal Scan)を用いてDC係数から高周波領域の係数までスキャンして1次元の係数シーケンスを出力する。変換単位のサイズ及びイントラ予測モードによって、千鳥状スキャンの代わりに2次元の係数アレイを列方向にスキャンする縦方向スキャン、2次元のブロック形態係数を行方向にスキャンする横方向スキャンを使用してもよい。すなわち、変換単位のサイズ及びイントラ予測モードによって、千鳥状スキャン、対角線スキャン、縦方向スキャン、及び横方向スキャンの中で使用するスキャン方法を決定してもよい。

【0039】

エントロピー符号化部155は、CABAC(Context-based Adaptive Binary Arithmetic Code)、指数ゴロム(Exponential Golomb)などの様々な符号化方式を用いて、並べ替え部150から出力された1次元の量子化した変換係数のシーケンスを符号化することによってビットストリームを生成する。

【0040】

また、エントロピー符号化部155は、ブロック分割に関するCTUサイズ、CU分割フラグ、QT分割フラグ、MTT分割タイプ、MTT分割方向などの情報を符号化し、映像復号化装置が映像符号化装置と同様にブロックを分割できるようにする。また、エントロピー符号化部155は、現在ブロックがイントラ予測によって符号化したのか、それともインター予測によって符号化したのかを指示する予測タイプに関する情報を符号化し、予測タイプに応じたイントラ予測情報(すなわち、イントラ予測モードに関する情報)又はインター予測情報(参照ピクチャ及び動きベクトルに関する情報)を符号化する。

【0041】

逆量子化部160は、量子化部145から出力される量子化した変換係数を逆量子化して変換係数を生成する。逆変換部165は、逆量子化部160から出力された変換係数を周波数ドメインから空間ドメインに変換して残差ブロックを復元する。

【0042】

加算部170は、復元された残差ブロックと予測部120によって生成された予測ブロックを加算して現在ブロックを復元する。復元された現在ブロック内のピクセルは、次の順序のブロックをイントラ予測するときの参照ピクセルとして使用される。

【0043】

フィルタ部180は、ブロックベースの予測及び変換/量子化によって発生するブロッキングアーチファクト(blocking artifacts)、リングングアーチファクト(ringing artifacts)、ぼかしアーチファクト(blurring artifacts)等を低減するために、復元したピクセルに対するフィルタリングを実行する。フィルタ部180は、デブロッキングフィルタ182とSAO(Sample Adaptive Offset)フィルタ184を含む。

【0044】

デブロッキングフィルタ180は、ブロック単位の符号化/復号化によって発生するブロッキング現象(blocking artifact)を除去するために復元されたプロ

10

20

30

40

50

ック間の境界をフィルタリングし、S A Oフィルタ184はデブロッキングフィルタリングされた画像に対して追加のフィルタリングを実行する。S A Oフィルタ184は、損失符号化(l o s s y c o d i n g)によって発生する復元されたピクセルと元本ピクセルとの間の差を補償するために使用するフィルタである。

【0045】

デブロッキングフィルタ182及びS A Oフィルタ184を介してフィルタリングされた復元ブロックはメモリ190に保存される。1つのピクチャ内のすべてのブロックが復元されると、復元されたピクチャは、後で符号化したいピクチャ内のブロックをインター予測するための参照ピクチャとして使用する。

【0046】

図4は、本発明の技術を具現する映像復号化装置の例示的な機能ブロック図である。以下では、図4を参照して、映像復号化装置とこの装置の下位構成について説明する。

【0047】

映像復号化装置は、エントロピー復号化部410、並べ替え部415、逆量子化部420、逆変換部430、予測部440、加算器450、フィルタ部460、及びメモリ470を含んでなる。

【0048】

図1の映像符号化装置と同様に、映像復号化装置の各構成要素は、ハードウェア又はソフトウェアで具現するか、あるいはハードウェア及びソフトウェアの組み合わせで具現する。さらに、各構成要素の機能がソフトウェアで具現され、マイクロプロセッサが各構成要素に対応するソフトウェアの機能を実行するように具現してもよい。

【0049】

エントロピー復号化部410は、映像符号化装置によって生成されたビットストリームを復号化してブロック分割に関連する情報を抽出することにより復号化しようとする現在ブロックを決定し、現在ブロックを復元するために必要な予測情報と残差信号に関する情報などを抽出する。

【0050】

エントロピー復号化部410は、S P S (S e q u e n c e P a r a m e t e r S e t) 又はP P S (P i c t u r e P a r a m e t e r S e t) からC T Uサイズに関する情報を抽出してC T Uのサイズを決定し、ピクチャを決定したサイズのC T Uに分割する。そして、C T Uをツリー構造の最上位レイヤー、すなわちルートノードとして決定し、C T Uについての分割情報を抽出することにより、ツリー構造を用いてC T Uを分割する。

【0051】

例えば、Q T B T T T T構造を用いてC T Uを分割する場合、まずQ Tの分割に関連する第1のフラグ(Q T _ s p l i t _ f l a g)を抽出して各ノードを下位レイヤーの4つのノードに分割する。そして、Q Tのリーフノードに該当するノードに対しては、M T Tの分割に関連する第2のフラグ(M T T _ s p l i t _ f l a g)及び分割方向(v e r t i c a l / h o r i z o n t a l)及び/又は分割タイプ(b i n a r y / t e r n a r y)情報を抽出して当該リーフノードをM T T構造に分割する。これにより、Q Tのリーフノード以下の各ノードをB T又はT T構造に、繰り返し(r e c u r s i v e l y)分割する。

【0052】

また他の例として、Q T B T T T T構造を用いてC T Uを分割する場合、まずC Uを分割するか否かを指示するC U分割フラグ(s p l i t _ c u _ f l a g)を抽出し、該当ブロックが分割された場合、第1のフラグ(Q T _ s p l i t _ f l a g)を抽出してもよい。分割の過程で、各ノードは、0回以上の繰り返しのQ T分割後に0回以上の繰り返しのM T T分割が発生する。例えば、C T UはすぐにM T T分割が発生するか、あるいは逆に複数回のQ T分割のみが発生する。

【0053】

10

20

30

40

50

他の例として、Q T B T 構造を用いてC T Uを分割する場合、Q Tの分割に関連する第1のフラグ(Q T _ s p l i t _ f l a g)を抽出して各ノードを下位レイヤーの4つのノードに分割する。そして、Q Tのリーフノードに該当するノードに対しては、B Tにさらに分割するか否かを指示する分割フラグ(s p l i t _ f l a g)及び分割方向情報を抽出する。

【0054】

一方、エントロピー復号化部410は、ツリー構造の分割を通じて復号化しようとする現在ブロックを決定すると、現在ブロックがイントラ予測したかインター予測したかを指示する予測タイプに関する情報を抽出する。予測タイプ情報がイントラ予測を指示する場合、エントロピー復号化部410は、現在ブロックのイントラ予測情報(イントラ予測モード)についてのシンタックスエレメントを抽出する。予測タイプ情報がインター予測を指示する場合、エントロピー復号化部410は、インター予測情報についてのシンタックスエレメント、すなわち動きベクトル及びその動きベクトルが参照する参照ピクチャを示す情報を抽出する。

10

【0055】

一方、エントロピー復号化部410は、残差ブロックのコーディングモードに関する情報(例えば、残差ブロックが符号化したか残差ブロックのサブブロックのみが符号化したかに関する情報、残差ブロックをサブブロックに分割するために選択したパーティションタイプを示す情報、符号化した残差サブブロックを識別する情報、量子化パラメータなど)をビットストリームから抽出する。また、エントロピー復号化部410は、残差信号に関する情報として現在ブロックの量子化した変換係数に関する情報を抽出する。

20

【0056】

並べ替え部415は、映像符号化装置により実行した係数スキャニング順序の逆順で、エントロピー復号化部410でエントロピー復号化した1次元の量子化した変換係数のシーケンスを再び2次元の係数アレイ(すなわち、ブロック)に変更する。

【0057】

逆量子化部420は、量子化した変換係数を逆量子化し、逆変換部430は、残差ブロックのコーディングモードに関する情報に基づいて逆量子化した変換係数を周波数ドメインから空間ドメインへ逆変換して残差信号を復元することによって、現在ブロックに対する復元した残差ブロックを生成する。

30

【0058】

逆変換部430は、残差ブロックのコーディングモードに関する情報が映像符号化装置で現在ブロックの残差ブロックが符号化したと指示する場合に、逆量子化した変換係数に対して現在ブロックのサイズ(したがって、復元する残差ブロックのサイズ)を変換単位として使用して逆変換を実行し、現在ブロックに対する復元残差ブロックを生成する。

【0059】

また、逆変換部430は、残差ブロックのコーディングモードに関する情報が映像符号化装置において残差ブロックの1つのサブブロックのみが符号化されたと指示する場合に、逆量子化した変換係数について、変換したサブブロックのサイズを変換単位として使用して、逆変換を実行して変換したサブブロックの残差信号を復元し、非変換したサブブロックの残差信号を「0」値で埋めることによって、現在ブロックに復元した残差ブロックを作成する。

40

【0060】

予測部440は、イントラ予測部442及びインター予測部444を含む。イントラ予測部442は、現在ブロックの予測タイプがイントラ予測であるときに活性化され、インター予測部444は、現在ブロックの予測タイプがインター予測であるときに活性化される。

【0061】

イントラ予測部442は、エントロピー復号化部410から抽出されたイントラ予測モードに対するシンタックスエレメントに基づいて複数のイントラ予測モードの中から、現

50

在ブロックのイントラ予測モードを決定し、イントラ予測モードによって現在ブロック周囲の参照ピクセルを使用して現在ブロックを予測する。また、イントラ予測部 442 は、後述する行列ベースのイントラ予測 (Matrix-based Intra Prediction: MIP) を用いて、現在ブロックに対する予測ブロックを生成してもよい。イントラ予測部 422 は、現在ブロックの左側に再構成したサンプルと現在ブロックの上側に再構成したサンプルから導出した境界ベクトルと、予め定義した行列及びオフセットベクトルを用いて現在ブロックに対する予測ブロックを生成してもよい。

【0062】

インター予測部 444 は、エントローピー復号化部 410 から抽出されたイントラ予測モードに対するシンタックスエレメントを用いて現在ブロックの動きベクトルとその動きベクトルが参照する参照ピクチャを決定し、動きベクトルと参照ピクチャを使用して現在ブロックを予測する。

10

【0063】

加算器 450 は、逆変換部から出力される残差ブロックと、インター予測部又はイントラ予測部から出力される予測ブロックとを加算して現在ブロックを復元する。復元した現在ブロック内のピクセルは、後で復号化するブロックをイントラ予測する際の参照ピクセルとして利用する。

【0064】

フィルタ部 460 は、デブロッキングフィルタ 462 及び SAO フィルタ 464 を含む。デブロッキングフィルタ 462 は、ブロック単位の復号化によって発生するブロッキング現象 (blocking artifact) を除去するために、復元されたブロック間の境界をデブロッキングフィルタリングする。SAO フィルタ 464 は、損失符号化 (lossy coding) によって発生する復元されたピクセルと元本ピクセルとの間の差を補償するために、デブロッキングフィルタリング後の復元されたブロックに対して追加のフィルタリングを実行する。デブロッキングフィルタ 462 及び SAO フィルタ 464 を介してフィルタリングされた復元ブロックはメモリ 470 に保存される。1つのピクチャ内のすべてのブロックが復元されると、復元されたピクチャは、後で符号化したいピクチャ内のブロックをインター予測するための参照ピクチャとして使用される。

20

【0065】

本発明の技術は一般にイントラ予測符号化に関連する。以下の説明は、主に復号化技術、すなわちビデオ復号化器の動作に焦点を当てており、符号化技術についての説明は、包括的に説明した復号化技術とは逆であることから、簡略化する。

30

【0066】

次世代ビデオコーディング標準 (VVC; Versatile Video Coding) についての議論で、HEV (High Efficiency Video Coding) 標準と比較してより良いコーディング性能を可能にするいくつかの新しいコーディングツールが導入された。VVC に導入された多重変換選択 (Multiple Transform Selection; MTS) 技法は、DCT-2、DST-7、及び DCT-8 を含む3つの三角関数変換 (Trigonometric Transforms) に依存し、ビデオ符号化器は、率-歪みコスト (rate-distortion cost) を最大化する水平及び垂直変換を選択する。

40

【0067】

【表 2】

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

10

【0068】

MTSを適用するブロックに対し、横方向と縦方向でどの変換カーネルを使用するかをシグナリングするために、1つあるいはそれ以上のシンタックスエレメントをコーディングする。

20

【0069】

一例として、表3のような予め定義したマッピングテーブルと共に、3つのフラグ(MTS_CU_flag、MTS_Hor_flag、MTS_Ver_flag)を使用する。

【0070】

【表 3】

MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	イントラ／インター	
			横方向	縦方向
0	-	-	DCT2	
1	0	0	DST7	DST7
	0	1	DCT8	DST7
	1	0	DST7	DCT8
	1	1	DCT8	DCT8

30

【0071】

まず、DCT-2が横方向と縦方向の両方に使用されているか否かを示す1つのフラグ(MTS_CU_flag)がシグナリングされる。DCT-2を両方向に適用しない場合は、DST-7とDCT-8のうち、どの変換カーネルを横方向に適用するかを示す1つのフラグ(MTS_Hor_flag)がシグナリングされ、DST-7とDCT-8のうち、どの変換カーネルを縦方向に適用するかを示す1つのフラグ(MTS_Ver_flag)がシグナリングされる。DST-7及びDCT-8は最大32×32のブロックサイズに対して使用され、DCT-2は最大64×64のブロックサイズに対して使用される。したがって、MTS_CU_flagは、ブロックのサイズ(幅及び高さ)が32よりも小さいか同じであり、ブロックが少なくとも1つの非ゼロ残差信号を有する場合にシグナリングされる。MTS_CU_flagが存在しない場合、MTS_CU_f1

40

50

ag は 0 と推論される。

【 0 0 7 2 】

対案として、表 4 のような予め定義したマッピングテーブルと共に、MTS カーネルインデックスを表すシンタックスエレメント (tu_mts_idx) を使用する。tu_mts_idx は、MTS カーネルインデックスで表され、各インデックス値によって、横方向及び縦方向のカーネルが選択される。表 4 で、trTypeHor と trTypeVer は、横方向と縦方向のカーネルの種類を示す。0 は DCT - 2、1 は DST - 7、2 は DCT - 8 を意味する。例えば、tu_mts_idx = 2 の場合、横方向には DCT - 8 を、縦方向には DST - 7 を用いる。

【 0 0 7 3 】

【表 4】

tu_mts_idx	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

【 0 0 7 4 】

tu_mts_idx は、SPS で sps_explicit_mts_enabled_flag が 1 のときにのみシグナリングされる。明示的に MTS インデックスをシグナリングするため、これを Explicit MTS という。一方、tu_mts_idx は明示的に示されず、符号化器及び復号化器から暗黙的に導出してもよく、これを Implicit MTS という。Implicit MTS は、SPS で sps_explicit_mts_enabled_flag が 0 のときに実行される。Implicit MTS を使用するとき、変換カーネルの選択は、ブロックに使用するイントラ予測モード、インター予測モード、ブロックサイズなどのような情報に基づいて実行される。たとえば、ブロックの幅が 4 以上で 16 以下の場合、横方向のカーネルとして DST - 7 を使用し、そうでない場合は DCT - 2 を使用する。また、ブロックの高さが 4 以上で 16 以下の場合、縦方向のカーネルとして DST - 7 を使用し、そうでない場合は DCT - 2 を使用する。つまり、ブロックのサイズが小さい場合は、予測残差信号の方向性が大きいため、DST - 7 を使用し、相対的に大きなブロックの場合、より均一な特性を有する残差信号の特性が強く、DCT - 2 カーネルを使用する。

【 0 0 7 5 】

表 5 は、VVC ドラフト 5 で規定している変換ユニットシンタックスの一部を示す。以下のシンタックスで、エレメントの灰色強調は理解を助けるために使用する。

【 0 0 7 6 】

10

20

30

40

50

【表 5】

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	記述子
(中略)	
if(tu_cbf_huma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_huma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
(後略)	

10

20

【 0 0 7 7 】

変換ブロックは、変換動作なしにコーディングされ、これは、少なくとも1つの非ゼロ係数を有する変換ブロックに対してシグナリングされたシンタックスエレメント `transform_skip_flag` によって指示される。変換ブロックのコーディングモードの上位レベル制御は、SPS (Sequential Parameter Set) などのようなHLS (High - Level Syntax) エレメント `transform_skip_enabled_flag` 及び `log2_transform_skip_max_size_minus_2` によって達成される。`transform_skip_enabled_flag` が1の場合、高さと幅が「 $1 < (\log 2_transform_skip_max_size_minus_2 + 2)$ 」以下の変換ブロックに対して `transform_skip_flag` がコーディングされる。`transform_skip_flag` が1の場合、関連変換ブロックは変換省略モードにコーディングされ、そうでなければ関連変換ブロックにMTSベースの変換を適用する。

30

【 0 0 7 8 】

一方、行列ベースのイントラ予測 (Matrix - based Intra Prediction : MIP) は、VTM5.0に導入された新しいイントラ予測技術である。元のアイデアは、ニューラルネットワークに基づくイントラ予測技術、すなわち隣接する再構成したピクセルに基づいて現在のPUピクセル値を予測するために多層ニューラルネットワークを使用することである。しかしながら、ニューラルネットワークを用いた予測方法の高度な複雑さにより、事前に訓練した行列を用いたアフィン線形変換に基づくイントラ予測技術を導入した。

40

【 0 0 7 9 】

幅W及び高さHを有する長方形ブロックPUを予測するために、MIPはブロックの左側に再構成したH個のサンプルとブロック上側に再構成したW個のサンプルを入力として

50

受け取る。そして、最後に予測されるピクセルは、平均化 (Averaging)、行列 - ベクトル乗算 (Matrix-Vector Multiplication)、及び線形補間 (Linear Interpolation) などによって得られる。

【 0 0 8 0 】

MIPを適用するブロックのサイズは、次のように3つのカテゴリに分類される。

【 0 0 8 1 】

$$idx(W, H) = \begin{cases} 0 & \text{for } W = H = 4 \\ 1 & \text{for } \max(W, H) = 8 \\ 2 & \text{for } \max(W, H) > 8 \end{cases}$$

10

【 0 0 8 2 】

$idx(W, H)$ によって、MIPモードの個数 ($numModes$)、境界サイズ ($boundarySize$)、予測ブロックのサイズ ($predW$ 、 $predH$ 、 $predC$)を次のように定義する。下表で、 $MipSizeId = idx(W, H)$ である。

【 0 0 8 3 】

【表 6】

MipSizeId	numModes	boundarySize	predW	predH	predC
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	$\text{Min}(nTbW, 8)$	$\text{Min}(nTbH, 8)$	8

20

【 0 0 8 4 】

図5は、本発明の技法で使用されるMIP技術の主要なプロセスを図式化した概念図である。

【 0 0 8 5 】

(1) 平均演算 (Averaging)

このステップの主な目的は、参照サンプルを正規化することである。ブロックサイズ及び形状によって(すなわち、 $MipSizeId$ によって)、4つ又は8つのサンプルが得られる。現在ブロックの幅と高さの両方が4の場合 ($W = H = 4$)、左から2つと上から2つを含む合計4つのサンプルが得られる。 ($boundarySize = 2$)。残りの場合には、左から4つ、上から4つを含む合計8つのサンプルが得られる ($boundarySize = 4$)。

30

【 0 0 8 6 】

図5に示すように、上隣サンプルは、 $bdry^{top}$ と表記し、左隣サンプルは、 $bdry^{left}$ と表記する。 $bdry^{top}$ と $bdry^{left}$ に対してそれぞれ平均演算を行い、ダウンサンプリングしたサンプルセット $bdry_{red}^{top}$ と $bdry_{red}^{left}$ が得られる。平均演算は、次のようなダウンサンプリングプロセスである。

40

【 0 0 8 7 】

$$redS[x] = (\sum_{i=0}^{bDwn-1} refS[x * bDwn + i] + (1 \ll (\text{Log2}(bDwn) - 1))) \gg \text{Log2}(bDwn)$$

【 0 0 8 8 】

上記の式で、 $bDwn$ はダウンサンプリングのスケール値 ($nTbs / boundarySize$)であり、 $refS$ はオリジナル参照サンプルを指す。

50

計算したredSは、左隣の場合 $bdry_{red}^{left}$ に保存され、上隣の場合 $bdry_{red}^{top}$ に保存される。

【 0 0 8 9 】

ダウンサンプリングした参照サンプルは、長さ 4 又は 8 のベクトルにステッチング (s t i t c h i n g) される。ベクトル行列乗算演算に入力する縮小した境界ベクトル

$bdry_{red}$

は、下記の式のように定義する。例えば、 $W = H = 4$ で M I P モードが 1 8 未満の場合、 $bdry_{red}^{top}$ と $bdry_{red}^{left}$ の順序で接合 (s t i t c h) して境界ベクトルを構成し、

10

$W = H = 4$ で M I P モードが 1 8 以上であれば、

$bdry_{red}^{left}$ と $bdry_{red}^{top}$ の順序で接合する。

下記の式で、「mode」は M I P モードを意味する。

【 0 0 9 0 】

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } W = H = 4 \text{ and } mode < 18 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6 \end{cases}$$

20

【 0 0 9 1 】

(2) 行列 - ベクトル乗算 (M a t r i x - V e c t o r M u l t i p l i c a t i o n)

このステップで、縮小した境界ベクトル $bdry_{red}$ から現在ブロックのダウンサンプリングされた予測信号 $pred_{red}$ を生成する。 $pred_{red}$ は、行列-ベクトルの積とオフセットとの和であり、次のように計算される。

【 0 0 9 2 】

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

40

【 0 0 9 3 】

$pred_{red}$ の大きさは $W_{red} \times H_{red}$ であり、 W_{red} と H_{red} は現在ブロックの大きさと形状によって以下のように定義される。

行列 A は、 $W_{red} * H_{red}$ だけの行 (r o w) を有し、 $W = H = 4$ の場合は4つの列 (c o l u m n) を有し、それ以外は8つの列を有する。オフセットベクトル b は、 $W_{red} * H_{red}$ サイズのベクトルである。

【 0 0 9 4 】

50

$$W_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(W, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(H, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

【 0 0 9 5 】

ブロックに使用する行列 A とオフセットベクトル b のセット S₀、S₁、S₂ は、ブロックのサイズのカテゴリ別に予め定義される。セット S のインデックス (0、1、2) は、
上述の M i p S i z e I d (すなわち、i d x (W , H)) によって選択され、行列 A と
オフセットベクトル b は、セット S₀、S₁、S₂ のうちの 1 つのセットからの現在ブロック
に対して適用する M I P モードによって抽出される。

10

【 0 0 9 6 】

セット S₀ は、それぞれ 1 6 行と 4 列を有する 1 8 個の行列 A₀ と 1 8 個の 1 6 次元オフ
セットベクトル b₀ から構成され、4 × 4 のブロックに使用される。セット S₁ は、そ
れぞれ 1 6 行と 8 列を有する 1 0 個の行列 A₁ と 1 0 個の 1 6 次元オフセットベクトル b₁
から構成され、4 × 8、8 × 4、及び 8 × 8 サイズのブロックに使用される。最後に、セ
ット S₂ は、それぞれ 6 4 行と 8 列を有する 6 つの行列 A₂ と 6 つの 6 4 次元オフセット
ベクトル b₂ とから構成され、残りのすべてのブロック形状に使用される。

20

【 0 0 9 7 】

(3) ピクセル補間 (i n t e r p o l a t i o n)

補間はアップサンプリングプロセスである。上述のように、

pred_{red}

は元のブロックのダウンサンプリングした予測信号である。このとき、サイズ p r e d W
と p r e d H を有するダウンサンプリングした予測ブロックは次のように定義される。

【 0 0 9 8 】

p r e d r e d [x] [y] , w i t h x = 0 . . p r e d W - 1 , y = 0 . . p r e
d H - 1

30

【 0 0 9 9 】

残りの位置での予測信号を各方向に沿って線形補間して生成する元ブロックサイズ (n
T b W、n T b H) の予測ブロックは、次のように定義される。

【 0 1 0 0 】

p r e d S a m p l e s [x] [y] , w i t h x = 0 . . n T b W - 1 , y = 0 .
. n T b H - 1

【 0 1 0 1 】

水平及び垂直アップサンプリングスケールファクタである u p H o r (= n T b W / p
r e d W) と u p V e r (= n T b H / p r e d H) に依存して、次のように、

40

pred_{red}

から p r e d S a m p l e s の一部あるいは全てが満たされる。

【 0 1 0 2 】

p r e d S a m p l e s [(x + 1) * u p H o r - 1] [(y + 1) * u p V e r
- 1] = p r e d r e d [x] [y]

【 0 1 0 3 】

u p H o r = 1 であれば、

50

$pred_{red}$

から $predSamples$ の横方向の全ての位置が満たされ、 $upVer = 1$ であれば、 $pred_{red}$

から $predSamples$ の縦方向の全ての位置が満たされる。

【0104】

この後、バイリニア (bilinear) 補間法により、 $predSamples$ の残りの空きサンプルが満たされる。横方向の補間と縦方向の補間はアップサンプリングプロセスである。 $predSamples$ 内の左側及び上側のサンプルを補間するために、ダウンサンプリングしたサンプル

$bdry_{red}^{top}$

は、 $predSamples[x][- 1]$ 値に割り当てられ、左側の元の参照サンプル

$bdry_{red}^{left}$

は、 $predSamples[- 1][y]$ 値に割り当てられる。

【0105】

(4) MIPイントラ予測モードのシグナリング

イントラ予測コーディングする各コーディングユニットCUに対して、行列ベースのイントラ予測モード (すなわち、MIPモード) を適用するか否かを示すフラグが伝送される。VVC5ドラフトでは、MIPモードのシグナリングのために、行列ベースのイントラ予測ではなく既存のイントラ予測モード (以下「正規イントラ予測モード」) と同様に、MPMリストを使用する。例えば、MIPモードのシグナリングのために、 $intra_mip_mpm_flag$ 、 $intra_mip_mpm_idx$ 、及び $intra_mip_mpm_remainder$ が使用される。 $intra_mip_mpm_idx$ は切り捨てバイナリコード (truncated binary code) にコーディングされ、 $intra_mip_mpm_remainder$ は固定長コード (fixed length code) にコーディングされる。

【0106】

コーディングブロックCUのサイズによって、最大35個のMIPモードがサポートされる。たとえば、 $\max(W, H) \leq 8$ 、及び $W * H < 32$ のCUに対し、35個のモードが利用可能である。そして、それぞれ $\max(W, H) = 8$ 、 $\max(W, H) > 8$ のCUに対し、それぞれ19個及び11個の予測モードが使用される。さらに、一対のモード (2つのモード) は、メモリ要件を減らすために行列及びオフセットベクトルを共有する。具体的な共有モードは次のように計算される。例えば、 4×4 コーディングブロックに対し、モード19は、モード2に割り当てられた行列の転置行列 (transposed matrix) が使用される。

【0107】

$$m = \begin{cases} mode & \text{for } W = H = 4 \text{ and } mode < 18 \\ mode - 17 & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ mode & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ mode - 9 & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ mode & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ mode - 5 & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6 \end{cases}$$

10

20

30

40

50

【 0 1 0 8 】

行列ベースのイントラ予測 (MIP) ではなく正規イントラ予測モードを適用するブロック (以下「正規ブロック」と呼ぶ) の周囲にMIPが適用となるブロックがある場合、正規ブロックのMPM誘導のためにMIPモードと正規モードとの間で定義されたマッピングテーブルを使用する。マッピングテーブルは、隣接MIPを適用するブロックのMIPモードから類似の特性の正規モードを導出するために使用される。このように導出された正規モードは、正規ブロックのMPM導出に使用される。同様に、クロマDM誘導で使用する同一位置の輝度ブロックがMIPを適用する場合にも、マッピングテーブルを用いて正規モードを導出し、クロマDM誘導に使用する。下記の式は、マッピングテーブルを使用した正規モードとMIPモードのマッピングを表現する。

10

【 0 1 0 9 】

$$predmode_{MIP} = map_regular_to_mip_{idx}[predmode_{regular}]$$

$$predmode_{regular} = map_mip_to_regular_{idx}[predmode_{MIP}]$$

【 0 1 1 0 】

以上で説明したように、VVCドラフト5では、ブロックをMIPベースで予測するとき、常にブロック左側の再構成した周辺サンプル

$$bdry^{left}$$

20

とブロック上段の再構成した周辺サンプル

$$bdry^{top}$$

が用いられる。このようなアプローチは、ブロックのテクスチャが方向特性を有するブロックの場合に予測性能を阻害し得る。また、VVCドラフト5では、predSamples内の上側サンプルの補間のために、図5に示すように、ダウンサンプリングしたサンプルセット

30

$$bdry_{red}^{top}$$

をpredSamples[x][- 1]値に割り当てて使用することで、補間プロセスが必要以上に複雑になる。さらに、VVCドラフト5では、MIPモードと正規モードのシグナリングのためにそれぞれMPMリストを使用されるため、MIPモードと正規モードとの間のマッピングなどが要求されるなど、多くの検査と条件によって具現が非常に複雑になり得る。

【 0 1 1 1 】

この観点から、本発明は、MPMコーディングに対するいくつかの改良した技法を提示する。本発明の一側面によると、MIPモードの具現の複雑さを低減し、ブロックのテクスチャが有する方向特性を考慮したアプローチを提示する。

40

【 0 1 1 2 】

図6は、本発明の一実施例によるMIPベースでブロックを予測する概略プロセスを示すフローチャートである。

【 0 1 1 3 】

1. MIPモードの復号化ステップ (S610)

ビデオ復号化器は、イントラ予測モードでコーディングしたコーディングユニット(CU)に対し、イントラ予測タイプが行列ベースのイントラ予測(MIP)であるかどうかを示すフラグを復号化する。MIPを適用する場合、利用可能な複数のMIPモードのう

50

ち、現在のコーディングユニットで用いる M I P モードを指示するシンタックスエレメントを復号化する。

【 0 1 1 4 】

既存のイントラ予測モード（正規イントラ予測モード）とは異なり、M I P モードのシグナリングのために M P M リストを使用しない場合もある。例えば、複数の M I P モードの中で、現在のコーディングユニットで用いる M I P モードを指示する切捨バイナリコード（`truncated binary code`）でコーディングされ得る、1つのシンタックスエレメント（例えば、`intra__mip__mode`）を使用する。

【 0 1 1 5 】

V V C 5 ドラフトに基づいて提案された例示的な変換ユニットシンタックスの一部を以下に提供する。下記のシンタックスで、エレメントの灰色強調は理解を助けるために用いる。

【 0 1 1 6 】

【表 7】

<code>if(CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>if(sps_bdpem_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)</code>	
<code>intra_bdpem_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_bdpem_flag[x0][y0])</code>	
<code>intra_bdpem_dir_flag[x0][y0]</code>	<code>ae(v)</code>
<code>else {</code>	
<code>if(sps_mip_enabled_flag && (Abs(Log2(cbWidth) - Log2(cbHeight)) <= 2) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)</code>	
<code>intra_mip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(intra_mip_flag[x0][y0])</code>	
<code>intra_mip_mode[x0][y0]</code>	<code>ae(v)</code>

【 0 1 1 7 】

`intra__mip__flag[ x0 ][ y0 ]` が 1 の場合は、現在ブロックのイントラ予測タイプが行列ベースのイントラ予測（M I P）であることを指す。`intra__mip__flag[ x0 ][ y0 ]` が 0 の場合は、現在ブロックのイントラ予測タイプが行列ベースのイントラ予測ではなく正規のイントラ予測であることを指す。`intra__mip__flag[ x0 ][ y0 ]` が存在しない場合は、0 に等しいと推論する。`intra__mip__mode[ x0 ][ y0 ]` は、行列ベースのイントラ予測（M I P）で現在のブロックに対して使用する M I P モードを指定する。

【 0 1 1 8 】

予測信号生成に使用する現在コーディングブロックの隣接サンプルは、M I P モードによって異なるように決定される。一例で、 4×4 サイズのコーディングブロックに対し、例えば、35 個の M I P モードが利用可能であり、モードの範囲によって予測信号生成に使用する隣接サンプルを以下のように決定する。

【 0 1 1 9 】

左隣のサンプル	$for\ mode < 12$
上隣のサンプル	$for\ mode \geq 24$
左隣のサンプル及び上隣のサンプル	その他

【 0 1 2 0 】

現在のコーディングブロックのピクセルが横方向特性を有する場合、左隣のサンプルを

使用して予測信号を生成することが有利である。同様に、現在のコーディングブロックのピクセルが縦方向特性を有する場合、上隣のサンプルを使用して予測信号を生成することが有利である。

【 0 1 2 1 】

ブロックが有する方向特性に適した M I P モードを選択するために、ビデオ符号化器は、イントラ予測モードの正規モードを適用し、各モードによるブロックの歪み値 (*d i s t o r t i o n*) を計算する。横方向の予測モード (例えば、 *I N T R A _ A N G U L A R _ 1 4 ~ I N T R A _ A N G U L A R _ 2 2*) の歪み値が他のモードと比較して少ない場合、横方向は現在のコーディングブロックの主方向として決定される。逆に、縦方向の予測モード (例えば、 *I N T R A _ A N G U L A R _ 4 6 ~ I N T R A _ A N G U L A R _ 5 4*) の歪み値が他のモードと比較して少ない場合、縦方向は現在のコーディングブロックの主方向として決定される。

10

【 0 1 2 2 】

横方向が主方向である場合、符号化器は左隣のサンプルのみを使用する M I P モードのうちの 1 つを現在のコーディングブロックに割り当てる。逆に、縦方向が主方向である場合、符号化器は、上隣のサンプルのみを使用する M I P モードのうちの 1 つを現在のコーディングブロックに割り当てる。

【 0 1 2 3 】

以下では、説明の便宜のために、左隣のサンプルを用いる場合と上隣のサンプルを用いる場合とを区別し、境界ベクトル及びそれから予測サンプルを決定する方法を説明する。左隣及び上隣のサンプルの両方を使用する場合のプロセスは、図 5 を参照して説明したものと実質的に同じである。

20

【 0 1 2 4 】

2 - 1 . 境界ベクトルの作成 - 左隣のサンプルを使用する (*S 6 2 0*)

図 7 a ~ 図 7 c は、左隣のサンプルを用いて行列 - ベクトル乗算演算に入力する境界ベクトルを構成する例示的な方法を示す概念図である。

【 0 1 2 5 】

一例として、図 7 a に例示するように、現在のコーディングブロックのサイズ (高さ) と境界ベクトル

$bdry_{red}$

30

のサイズが等しい場合、左隣のサンプルセット

$bdry^{left}$

を使用して同じサイズの境界ベクトル

$bdry_{red}$

40

を埋める。例えば、左隣のサンプルのそれぞれを境界ベクトルのエントリに含める。

【 0 1 2 6 】

他の例として、図 7 b に示すように、左側に隣接する 2 つの列を使用して各行の 2 つのピクセル間の平均値を計算することによって境界ベクトル

$bdry_{red}$

を埋める。

【 0 1 2 7 】

また他の一例として、図 7 c に示すように、左隣のサンプルセット

50

$bdry^{left}$

から得られたダウンサンプリングしたサンプルセット

$bdry_{red}^{left}$

を使用して境界ベクトル $bdry_{red}$ を埋める。

$bdry_{red}^{left}$ は、 $bdry^{left}$ を 2 つのサンプルずつ平均して得る。

【 0 1 2 8 】

10

コーディングブロックの大きさによって、適応的に $bdry^{left}$ と $bdry_{red}^{left}$ を使用する。例えば、下の式のように、 $H \leq 8$ であれば $bdry^{left}$ を使用し、その他は $bdry_{red}^{left}$ を使用する。

【 0 1 2 9 】

$$bdry_{red} = \begin{cases} [bdry^{left}] & \text{for } H \leq 8 \\ [bdry_{red}^{left}] & \text{その他} \end{cases}$$

【 0 1 3 0 】

20

2 - 2 . 境界ベクトルの生成 - 上隣のサンプルの利用 (S 6 2 0)

図 8 a ~ 図 8 c は、上隣のサンプルを使用して行列 - ベクトル乗算演算に入力する境界ベクトルを構成する例示的な方法を示す概念図である。

【 0 1 3 1 】

一例として、図 8 a に例示するように、現在のコーディングブロックのサイズ (幅) と境界ベクトル

$bdry_{red}$

のサイズが等しい場合、上隣のサンプルセット

30

$bdry^{top}$

を使用して同じサイズの境界ベクトル

$bdry_{red}$

を埋める。例えば、上隣のサンプルのそれぞれを境界ベクトルのエントリに含める。

【 0 1 3 2 】

他の例として、図 8 b に示すように、上部に隣接する 2 つの行を用いて、各列の 2 つのピクセル間の平均値を計算することによって、境界ベクトル

40

$bdry_{red}$

を埋める。

【 0 1 3 3 】

また他の一例として、図 8 c に示すように、上隣のサンプルセット $bdry^{top}$ から得られたダウンサンプリングしたサンプルセット $bdry_{red}^{top}$ を使用して境界ベクトル $bdry_{red}$ を埋めてもよい。 $bdry_{red}^{top}$ は、 $bdry^{top}$ を 2 つのサンプルずつ平均して得る。

50

【 0 1 3 4 】

コーディングブロックの大きさによって、適応的に $bdry_{red}^{top}$ と $bdry_{red}^{top}$ を使用してもよい。
例えば、下の式のように、 $H \leq 8$ であれば $bdry_{red}^{top}$ を使用し、その他は $bdry_{red}^{top}$ を使用する。

【 0 1 3 5 】

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}] & \text{for } H \leq 8 \\ [bdry_{red}^{top}] & \text{その他} \end{cases}$$

10

【 0 1 3 6 】

3 . 行列 - ベクトル乗算及びオフセット加算 (S 6 3 0)

ビデオ復号化器は、境界ベクトル

$bdry_{red}$

に対し、行列 - ベクトル積演算を実行し、オフセットベクトルを加算する。行列 A 及びオフセットベクトル b は、復号化した M I P モード k に従って決定される。

その結果、元のブロックの予測信号又はサブサンプリングした予測信号 $pred_{red}$ が生成される。 $pred_{red}$ は次のように計算する。

20

【 0 1 3 7 】

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

【 0 1 3 8 】

ここで、 $pred_{red}$ の大きさは $W_{red} \times H_{red}$ である。

30

【 0 1 3 9 】

左隣のサンプルを用いて行列 - ベクトル乗算演算に入力する境界ベクトルを構成した場合、 W_{red} と H_{red} は、コーディングブロックのサイズによって次のように決定される。

【 0 1 4 0 】

$$W_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(W, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{for } H \leq 8 \\ 8 & \text{for } H > 8 \end{cases}$$

40

【 0 1 4 1 】

上隣のサンプルを用いて行列 - ベクトル乗算演算に入力する境界ベクトルを構成した場合、 W_{red} と H_{red} は、コーディングブロックの大きさによって次のように決定される。

【 0 1 4 2 】

50

$$W_{red} = \begin{cases} 4 & \text{for } W \leq 8 \\ 8 & \text{for } W > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(H, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

【 0 1 4 3 】

生成した

$pred_{red}$

10

のサイズがコーディングブロックのサンプル数より小さい場合、補間が必要である。

【 0 1 4 4 】

4 . 線形補間 (S 6 4 0)

図 9 は、左隣のサンプル、境界ベクトル、及びそれから予測した予測サンプルを示す。

図 9 で、現在のコーディングブロックに対する予測ブロック $predSamples[x]$

$[y]$ 内に三角形で示したピクセルは、生成した

$pred_{red}$

20

から割り当てられた値である。

【 0 1 4 5 】

図 9 を参照すると、予測ブロック $predSamples[x][y]$ 内の空のピクセルを決定するために、様々な補間方法を使用する。

【 0 1 4 6 】

例えば、1 番ピクセルのように、左隣のサンプル、上隣のサンプル及び下部 $pred_{red}$ を用いて空のピクセル値を決定する。他の例として、2 番ピクセルのように、上隣のサンプル及び下側 $pred_{red}$ を使用して空のピクセル値を決定する。また他の例として、3 番ピクセルのように、上側 $pred_{red}$ と下側 $pred_{red}$ を使用して空のピクセル値を決定してもよい。また他の例として、4 番ピクセルのように、左隣のサンプル、上側 $pred_{red}$ 及び下側 $pred_{red}$ を使用して空のピクセル値を決定してもよい。

30

【 0 1 4 7 】

図 10 は、上隣のサンプル、境界ベクトル、及びそれから得られた予測サンプルを示す。図 10 で、現在のコーディングブロックに対する予測ブロック $predSamples[x][y]$ 内に三角形で示したピクセルは、生成した

$pred_{red}$

40

から割り当てられた値である。

【 0 1 4 8 】

図 10 を参照すると、予測ブロック $predSamples[x][y]$ 内の空のピクセルを決定するために、様々な補間方法を使用する。

例えば、1 番ピクセルのように、左隣のサンプル、上隣のサンプル、及び下側 $pred_{red}$ を使用して空のピクセル値を決定する。他の例として、2 番ピクセル 2 のように、左隣のサンプル、上側 $pred_{red}$ 及び下側 $pred_{red}$ を使用して空のピクセル値を決定する。また他の例として、3 番ピクセルのように、上側 $pred_{red}$ と下側 $pred_{red}$ を使用して空のピクセル値を決定してもよい。また他の例として、4 番ピクセルのように、上隣のサンプル及び下側 $pred_{red}$ を使用して空のピクセル値を決定してもよい。

【 0 1 4 9 】

図 9 及び図 10 に例示したものと異なり、予測ブロック $predSamples[x][y]$ 内の空のピクセルを決定するために、上段のオリジナル参照サンプル

10

$bdry^{top}$

が $predSamples[x][-1]$ 値に割り当てられ、左側の元の参照サンプル

$bdry^{left}$

が $predSamples[-1][y]$ 値に割り当てられてもよい。

【 0 1 5 0 】

以上の実施例は、MIP ベースで予測ブロックを生成する際にブロックの方向性が考慮されるように、MIP モードによって現在ブロックの隣接サンプルを選択的に使用する方式を導入した。

20

【 0 1 5 1 】

本発明の他側面によると、MIP モードによって境界ベクトルを生成するために使用する隣接サンプルセットを変える代わりに、境界ベクトルに対して適用する行列 A とオフセット b の値を修正する方法を使用してもよい。

【 0 1 5 2 】

境界ベクトルに対する行列 - ベクトル乗算演算とオフセット加算演算に適用する行列 A とオフセット b を次のように 2 つの部分に区分する。

【 0 1 5 3 】

30

$$pred_{red} = A \begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + (b)$$

$$pred_{red} = (A1|A2) \begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \end{pmatrix}$$

【 0 1 5 4 】

A 2 及び b 2 が有する値をゼロに近似すると、予測信号

$pred_{red}$

40

は主に左隣のピクセル

$bdry_{red}^{left}$

によって影響を受け、これはブロックのテクスチャが横方向性を有するブロックに対して有用である。A 1 と b 1 が有する値を 0 に近似すると、予測信号

$pred_{red}$

50

は主に上隣のピクセル

$bdry_{red}^{left}$

に影響を受け、これは、ブロックのテクスチャが縦方向を有するブロックに対して有用である。

【 0 1 5 5 】

ビデオ符号化器及び復号化器は、現在ブロックのMIPモードによって境界ベクトルに対して適用する行列A及びオフセットbの値を修正して使用する。一例として、 4×4 サイズのコーディングユニットに対し、例えば、35個のMIPモードが利用可能であり、モードの範囲によって境界ベクトルに適用する行列A及びオフセットbの値を修正して使用する。

10

【 0 1 5 6 】

$$\begin{cases} A2 \text{ and } b2 \cong 0 & \text{for mode} < 12 \\ A1 \text{ and } b2 \cong 0 & \text{for mode} \geq 24 \\ A \text{ 及び } b \text{ を使用} & \text{その他} \end{cases}$$

【 0 1 5 7 】

上述したように、VVCドラフト5は、コーディングブロックCUのサイズ及び形状によって最大35個のMIPモードがサポートする。たとえば、 $\max(W, H) \leq 8$ & $W * H < 32$ のCUに対して35個のモードが利用可能で、それぞれ $\max(W, H) = 8$ 、及び $\max(W, H) > 8$ のCUに対してそれぞれ19個及び11個の予測モードを使用する。さらに、一对のモード(2つのモード)は、メモリ要件を減らすために行列及びオフセットベクトルを共有する。例えば、 4×4 コーディングブロックに対し、モード19は、モード2に対して割り当てられた行列の転置行列(transposed matrix)を使用する。

20

さらに、MIPモードとブロックのサイズによって決定する順序で $bdry_{red}^{top}$ と $bdry_{red}^{left}$ を接合することで、ベクトル行列乗算演算に入力する境界ベクトル $bdry_{red}$ を生成する。

30

【 0 1 5 8 】

複雑さを下げながらも実質的に同等のレベルのコーディング効率を達成する改善したアプローチを使用する。本発明の他側面によると、1つのモードがもう1つのモードが使用する行列を転置して使用する従来の方式の代わりに、

各モードに対し、ベクトル-行列乗算に使用する境界ベクトル $bdry_{red}$ を構成する $bdry_{red}^{top}$ と $bdry_{red}^{left}$ を接合する順序を変更可能に構成した新しい方式を使用する。ビデオ符号化器は、各モードに対し、境界ベクトル $bdry_{red}$ を構成する $bdry_{red}^{top}$ と $bdry_{red}^{left}$ を接合する順序を指示するシンタックスエレメントをシグナリングする。

この方式によると、利用可能なMIPモードの数は、既存の方式と比較して半分のレベルに減らすことができ、境界ベクトルの生成及びベクトル行列乗算演算のためのコーディング複雑さを低減できる。

40

【 0 1 5 9 】

一方、表5のシンタックス構造に関連して説明したように、特定条件を満たす変換ユニットについては、毎回変換省略関連シンタックス要素であるtransform_skip_flagをシグナリングし、transform_skip_flagが真でない場合、SPSにおけるフラグ(sps_explicit_mts_inter_enabled_flag, sps_explicit_mts_intra_enabled_flag)によって、多重変換選択関連シンタックス要素であるtu_mts_idxを伝送する必要があるため、ビットストリームで伝送するビット数に負担をかける。

50

【 0 1 6 0 】

しかしながら、このような多重変換選択及び変換省略は、残差信号の特性と密接に関連しており、したがってブロックに使用する予測モードとは無関係に、変換関連シンタックス要素を常にシグナリングすることは効率的ではない場合もある。本発明者は、平均値演算と補間演算を含む行列ベースのイントラ予測 (M I P) 技法のプロセスを考慮すると、M I Pを適用したブロックの残差信号は、変換ドメインで低周波数成分が優勢な特性を有し、複雑な動きとテクスチャを含むブロックのように、ピクセル間の相関関係が少ないブロックには、M I Pが適用されない可能性が高いことに注目する。したがって、M I Pを使用したブロックに対しては変換省略 (t r a n s f o r m s k i p) が適用されない可能性が高い。

10

【 0 1 6 1 】

この観点から、ブロックにM I Pを適用するか否かにかかわらず、多重変換選択関連シンタックスエレメントである t u _ m t s _ i d x と変換省略関連シンタックスエレメントである t r a n s f o r m _ s k i p _ f l a g を常にシグナリングするアプローチはそれほど効率的ではない場合がある。

【 0 1 6 2 】

本発明の一側面によると、M I P適用如何によって変換関連シンタックス要素のシグナリングを省略することができ、省略したシンタックス要素はビデオ復号化器によって暗黙的に推論する。

【 0 1 6 3 】

20

一実施例で、M I Pフラグによって多重変換選択 (M T S) 技法を暗黙的に適用する。下表のように、i n t r a _ m i p _ f l a g が真でない場合にのみ、t r a n s f o r m _ s k i p _ f l a g と t u _ m t s _ i d x がコーディングされる。i n t r a _ m i p _ f l a g が真の場合、t r a n s f o r m _ s k i p _ f l a g は0と推論され、t u _ m t s _ i d x も0と推論され、i m p l i c i t M T Sを適用すると推論される。したがって、横方向及び縦方向の変換カーネルは、ブロックサイズによって決定される。下記のシンタックスで、エレメントの灰色強調は、シンタックスでの潜在的な変化を示すため又は理解を助けるために用いられる。

【 0 1 6 4 】

30

40

50

【表 8】

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	記述子
(中略)	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]) && (!intra_mip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
(後略)	

10

20

【 0 1 6 5 】

他の実施例では、MIPフラグによって多重変換選択 (MTS) 技法を明示的に適用する。下表のように、intra_mip_flag が真でない場合にのみ、transform_skip_flag と tu_mts_idx がコーディングされる。intra_mip_flag が真の場合、transform_skip_flag は 0 と推論され、tu_mts_idx は明示的にシグナリングされ、explicit MTS が適用される。下記のシンタックスで、エレメントの灰色強調は、シンタックスでの潜在的な変化を示すため又は理解を助けるために用いられる。

30

【 0 1 6 6 】

40

50

【表 9】

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	記述子
(中略)	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
(後略)	

10

20

【 0 1 6 7 】

可能な他の実施例で、複雑な動きやテクスチャを含む小さなサイズのブロックのような、ピクセル間の相関関係が少ないブロックにはMIPを明示的に適用しない場合がある。MIPを適用した4×4CUには変換省略(TS)を適用しない場合もある。また、変換省略(TS)を適用した4×4CUにはMIPを適用しない場合もある。さらに、MIPはTSと共に4×4CUに適用しない場合もある。

30

【 0 1 6 8 】

従来のアプローチでは、MPM(Most Probable Mode)を使用するイントラ予測コーディングが使用される。例えば、HEVCで、3つのMPMのリストは、左及び上のブロックのイントラ予測モードから構成される。このような方法の欠点は、より多くのモード(MPMではなくイントラモード)がより多くのビットでコーディングすべき非MPMに属することである。MPMの数を3つ以上のエントリ(例えば、6つのMPMモード)に拡張するためのいくつかの方法が提案された。ただし、より多くのエントリを含むこのようなMPMリストを構成するには、より多くの検査と条件が必要になることから、これにより具現がより複雑になる。

40

【 0 1 6 9 】

上述したように、VVCドラフト5では、MIPモードと正規モードのシグナリングのためにそれぞれMPMリストを使用することにより、MIPリストを構成するためにMIPモードと正規モードとの間のマッピングなどが要求されるなど多くの検査と条件によって具現が非常に複雑になる。

【 0 1 7 0 】

MPMリスト構成の複雑さを低く保つために、現在ブロックに隣接する左側ブロック及

50

び上側ブロックのイントラ予測モードを用いて6つのMPM候補を含むMPMリストを構成する。MPM候補は、デフォルトイントラ予測モード（例えば、PLANARモード）、周辺ブロックのイントラ予測モード、周辺ブロックのイントラ予測モードから派生したイントラ予測モードから構成される。周辺ブロックのイントラ予測モードが使用されない場合（例えば、周辺ブロックをインター予測した場合、あるいは周辺ブロックが他のスライス又は他のタイルに位置する場合）、周辺ブロックのイントラ予測モードのイントラ予測モードはプラナー（Planar）に設定される。

【0171】

左側ブロックのモード（Left）及び上側ブロックのモード（Above）のイントラ予測モードのタイプによって大きく4つの場合に区分され、「Left」と「Above」が互いに異なり、両方のモードが方向性モードの場合、「Left」と「Above」の違いによって、4つのケースでさらに区別してMPMリストを作成する。下表で、「Max」は「Left」と「Above」のうち、大きなモードを指し、「MIN」は「Left」と「Above」のうち、小さなモードを指す。

【0172】

【表10】

条件	細部条件	MPM モード
左側モードと上側モードが方向性モードであり、互いに同一の場合		{Planar, Left, Left-1, Left+1, Left-2, Left+2}
左側モードと上側モードが互いに異なり、2つのモードが方向性モードである場合	Max-Min = 1	{Planar, Left, Above, Min-1, Max+1, Min-2}
	Max-Min ≥ 62	{Planar, Left, Above, Min+1, Max-1, Min+2}
	Max-Min ≥ 2	{Planar, Left, Above, Min+1, Min-1, Max+1}
	2 < Max-Min < 62	{Planar, Left, Above, Min-1, Min+1, Max-1}
左側モードと上側モードが互いに異なり、2つのうち、1つのみが方向性モードである場合		{Planar, Max, Max-1, Max+1, Max-2, Max+2}
左側モードと上側モードがすべて方向性モードでない場合（つまり、PlanarまたはDCである場合）		{Planar, DC, Angular50, Angular18, Angular46, Angular54}

【0173】

平均値演算と補間演算を含む行列ベースのイントラ予測（MIP）技法の特性上、MIPを適用したブロックの残差信号は変換ドメインで低周波数成分が優勢である。このような残差特性は、プラナーモード又はDCモードを適用したブロックの残差信号と類似することに留意したい。したがって、正規のイントラ予測モードでコーディングするブロックのMPMリストを導出する際に、このような残差信号の類似性を考慮することが有用である。

【0174】

本発明の一側面によると、正規のイントラ予測モードでコーディングしたブロック（すなわち、正規ブロック）に対してMPMリストを導出する際に、周辺ブロックがMIPモードでコーディングした場合に、周辺ブロックのイントラ予測モードはプラナーモード（又はDCモード）であると見なす。一例として、周辺ブロックにMIPモードを適用した場合、その周辺ブロックのMIPモードの代わりにプラナーモード（又はDCモード）をMPMリストに追加する。

【0175】

同様に、クロマDM（direct mode）誘導時にも、同一位置のルマブロックにMIPを適用した場合、MIPモードと正規モードとの間のマッピングテーブルを使用する代わりに、そのルマブロックのイントラ予測モードはプラナーモード（又はDCモード）と見なす。したがって、符号化器は、クロマブロックに対するイントラ予測モードを

特定するシンタックスエレメントをパージングし、そのシンタックスエレメントによって、クロマブロックのイントラ予測モードが同一位置のルマブロックのイントラ予測モードをそのまま利用することを指示され、その同一位置のルマブロックにMIPを適用した場合、そのルマブロックのイントラ予測モードは、プラナーモード（又はDCモード）と見なす。すなわち、クロマDM（direct mode）で同一位置のルマブロックにMIPを適用した場合に、クロマブロックのイントラ予測モードはプラナーモード（あるいはDCモード）と決定される。

【0176】

ビデオ符号化器は、現在ブロックのイントラ予測モードがMPMに該当するか否かを示す1ビットフラグ（例えば、mpm_flag）をシグナリングする。典型的には、現在ブロックのイントラ予測モードがMPMに該当する場合、6つのMPMのうちの1つ（すなわち、プラナーモード）を指示するMPMインデックスが追加でシグナリングされる。上表で、プラナーモードが常にMPMリストに含まれることに注意する。したがって、符号化器は、現在ブロックのイントラ予測モードがプラナーモードである場合、現在ブロックのイントラ予測モードがプラナーモードであるか如何を（例えば、1ビットフラグを使用して）明示的にシグナリングし、現在ブロックのイントラ予測モードが残りの5つのMPMのうちの1つと同じである場合、残りの5つのMPMのうちの1つを指示するMPMインデックスを追加でシグナリングすることが効率的である。現在ブロックのイントラ予測モードがMPMに該当しない場合、6つのMPMを除く残りの61個の非MPMのうちの1つを指示するシンタックスエレメントを切捨てたバイナリコード（truncated binary code）を利用して符号化する。

【0177】

図11は、本発明の一実施例によるビデオデータを復号化する方法を例示するフローチャートである。

【0178】

ビデオ復号化器は、ビットストリームからビデオデータの現在ブロックのイントラ予測タイプを指示するシンタックスエレメントを復号化する（S1110）。イントラ予測タイプは、行列ベースのイントラ予測（matrix based intra prediction：MIP）と正規イントラ予測（regular intra prediction）を含む。上記シンタックスエレメントは、現在ブロックのサイズ及び形状に対して許容される複数の行列ベースのイントラ予測モードのうちの1つを特定する切捨てバイナリコード（truncated binary code）である。

【0179】

ビデオ復号化器は、現在ブロックのイントラ予測タイプに基づいて行列ベースのイントラ予測又は正規イントラ予測を選択的に実行し、現在ブロックに対する予測ブロックを生成する。

【0180】

ビデオ復号化器は、正規イントラ予測を実行して現在ブロックに対する予測ブロックを生成することの一部として、次のようなステップ（S1120～S1140）を実行する。ビデオ復号化器は、現在ブロックに隣接する周辺ブロックの正規イントラ予測モードに基づいてMPM（Most Probable Mode）候補を導出して現在ブロックに対するMPMリストを構成し（S1120）、MPMリストに基づいて現在ブロックに対する正規イントラ予測モードを導出する（S1130）。周辺ブロックの正規イントラ予測モードに基づいてMPM候補を導出するにあたり、ビデオ復号化器は、周辺ブロックのイントラ予測タイプが行列ベースのイントラ予測である場合に、周辺ブロックの正規イントラ予測モードはプラナーモードに設定する（みなす）。復号化器は、現在ブロックの正規イントラ予測モードを利用して現在ブロックに対する予測ブロックを生成する（S1140）。

【0181】

ビデオ復号化器は、行列ベースのイントラ予測を実行して現在ブロックに対する予測ブ

ロックを生成することの一部として、次のようなステップ（S 1 1 2 1 ~ S 1 1 5 1）を実行する。ビデオ復号化器は、現在ブロックに対する行列ベースのイントラ予測モードを決定するために、ビットストリームから現在ブロックに対する行列ベースのイントラ予測モードを指示するシンタックスエレメントを復号化する（S 1 1 2 1）。ビデオ復号化器は、現在ブロックのサイズ及び形状に基づいて現在ブロックに隣接する周辺サンプルを利用して境界ベクトルを導出し（S 1 1 3 1）、現在ブロックに対する行列ベースのイントラ予測モードに対して予め定義した行列と境界ベクトルとの間の行列・ベクトル乗算に基づき、現在ブロックに対する予測サンプルを生成する（S 1 1 4 1）。ビデオ復号化器は、予測サンプルに基づいて線形補間、クリッピングなどを実行して現在ブロックに対する予測ブロックを導出する（S 1 1 5 1）。

10

【0 1 8 2】

ビデオ復号化器は、現在ブロックに隣接する左側周辺サンプルのセット及び上側周辺サンプルのセットのうち、現在ブロックに対する行列ベースのイントラ予測モードに依存して、選択した1つ又は2つのセットから上記境界ベクトルを導出する。

【0 1 8 3】

ビデオ復号化器は、ビットストリームから、現在ブロックに隣接する左側周辺サンプルから導出した境界ベクトルの第1のエントリと現在ブロックに隣接する上側周辺サンプルから導出した境界ベクトルの第2のエントリとの間の接合順序を指示するシンタックスエレメントを復号化する。ビデオ復号化器は、指示された接合順序に従って第1のエントリと第2のエントリを接合して境界ベクトルを生成する。

20

【0 1 8 4】

境界ベクトルのエントリは、現在ブロックの大きさ及び形状によって、現在ブロックに隣接する左側周辺サンプルからダウンサンプリングした値、又は現在ブロックに隣接する左側周辺サンプルで満たされる。

【0 1 8 5】

ビデオ復号化器は、予測サンプルに基づいて現在ブロックに対する予測ブロックを導出するために、予測サンプルを予測ブロック内の位置に割り当てる。予測ブロック内の予測サンプルが割り当てられていない位置に対する予測サンプル値を生成するために、予測サンプル、現在ブロックに隣接する左側周辺サンプル、及び現在ブロックに隣接する上側周辺サンプルに対する線形補間を実行する。

30

【0 1 8 6】

以上の説明で例示的な実施例は、多くの他の方式で具現される。1つ以上の例示で説明した機能又は方法は、ハードウェア、ソフトウェア、ファームウェア、又はそれらの任意の組み合わせで具現される。本明細書で説明する機能的コンポーネントは、それらの具現独立性を特に強調するために「...部（Unit）」とラベル付けした。

【0 1 8 7】

一方、本明細書で説明した様々な機能又は方法は、1つ以上のプロセッサによって読み取られ実行される非一時的記録媒体に保存した命令語で具現してもよい。非一時的記録媒体は、例えば、コンピュータシステムによって読み取り可能な形態でデータを保存するあらゆる種類の記録装置を含む。例えば、非一時的な記録媒体は、EPROM（erasable programmable read only memory）、フラッシュドライブ、光学ドライブ、磁気ハードドライブ、ソリッドステートドライブ（SSD）のような記憶媒体を含む。

40

【0 1 8 8】

以上の説明は、本発明の技術的思想によるいくつかの実施例を例示的に説明したものに過ぎず、本発明が属する技術分野で通常の知識を有する者であれば、本発明の本質的な特性から逸脱しない範囲で多様な修正及び変形が可能である。したがって、例示した実施例は、本発明の技術思想を限定するものではなく説明するためのものであり、そのような実施例によって本発明の技術的思想の範囲を限定するものではない。

【0 1 8 9】

50

CROSS - REFERENCE TO RELATED APPLICATION

本特許出願は、本明細書にその全体が参考として含まれる、2019年6月24日付で韓国に出願した特許出願番号第10-2019-0075163号、2019年7月4日付で韓国へ出願した特許出願番号第10-2019-0080749号及び2020年6月24日付で韓国に出願した特許出願番号第10-2020-0077014号に対して優先権を主張する。

【符号の説明】

【0190】

110 ピクチャ分割部
120、440 予測部
122、442 イントラ予測部
124、444 インター予測部
130 減算器
140 変換部
145 量子化部
150、415 並べ替え部
155 エントロピー符号化部
160、420 逆量子化部
165、430 逆変換部
170、450 加算器
180、460 フィルタ部
182、462 デブロッキングフィルタ
184、464 SAOフィルタ
190、470 メモリ
410 エントロピー復号化部

10

20

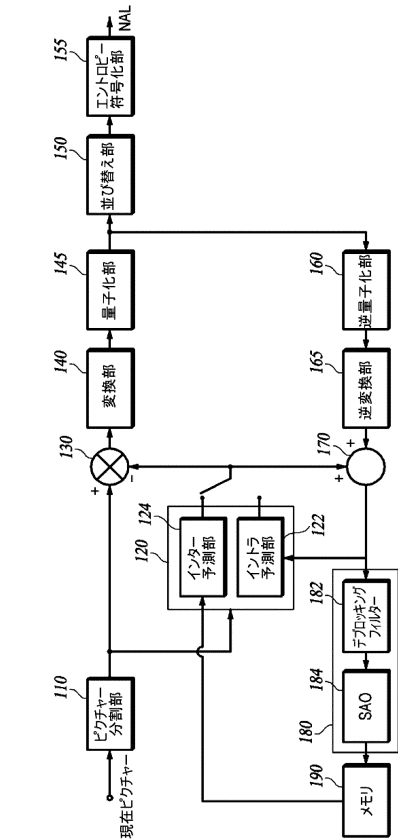
30

40

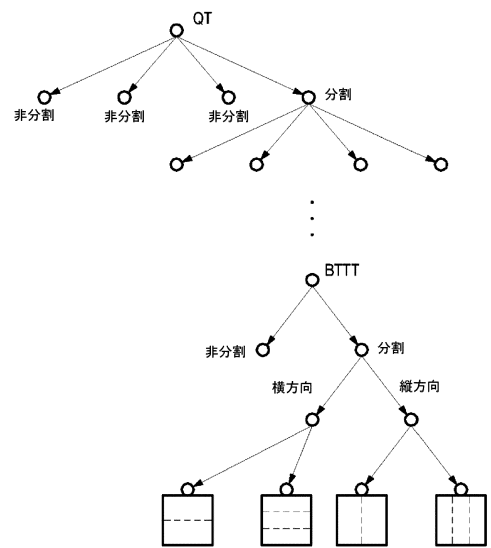
50

【図面】

【図 1】



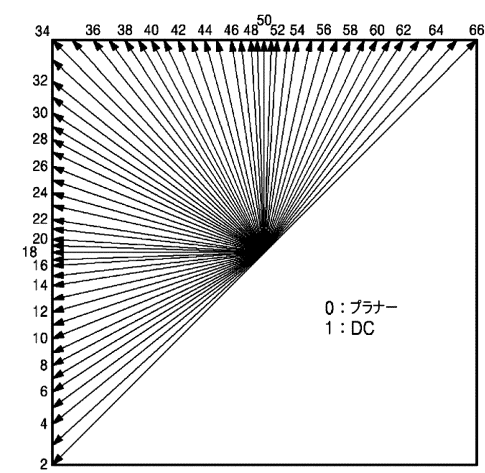
【図 2】



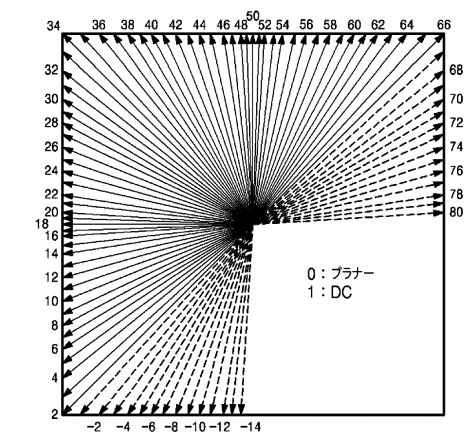
10

20

【図 3 a】



【図 3 b】

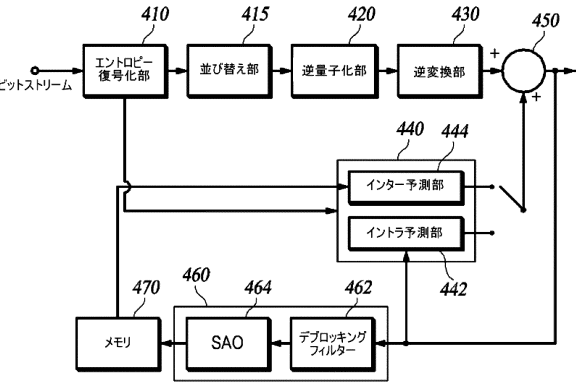


30

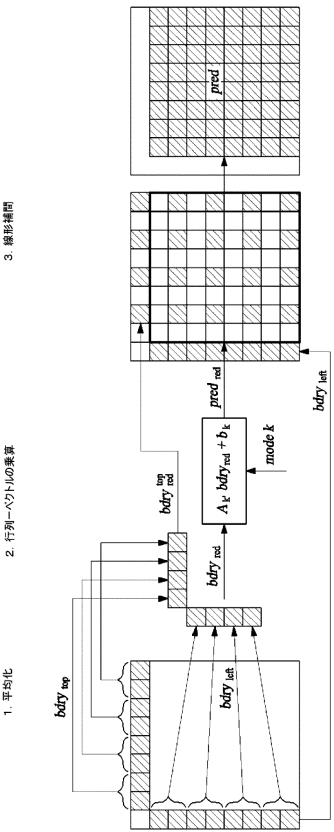
40

50

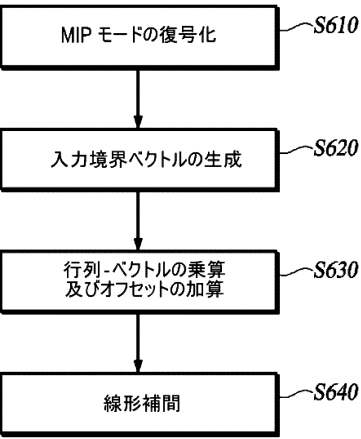
【図 4】



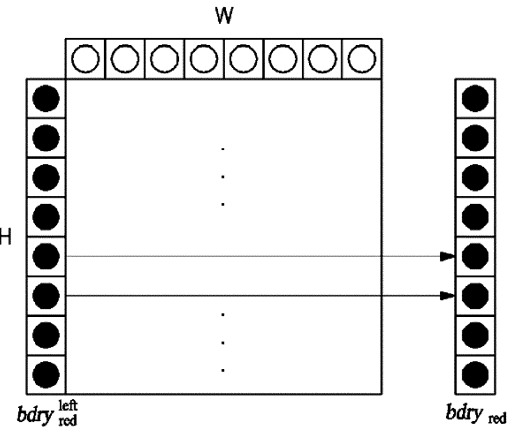
【図 5】



【図 6】



【図 7 a】



10

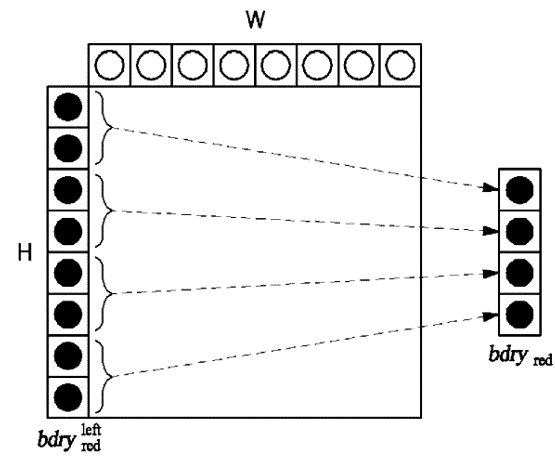
20

30

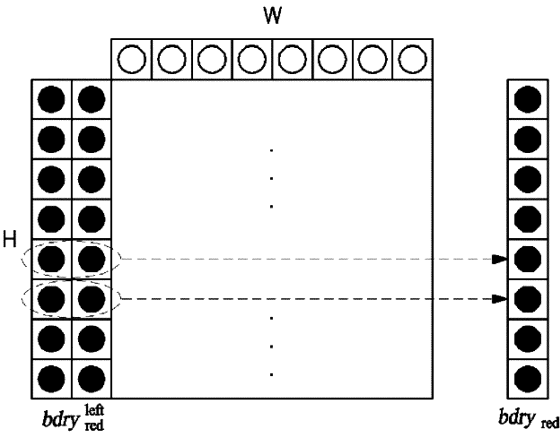
40

50

【 7 b 】

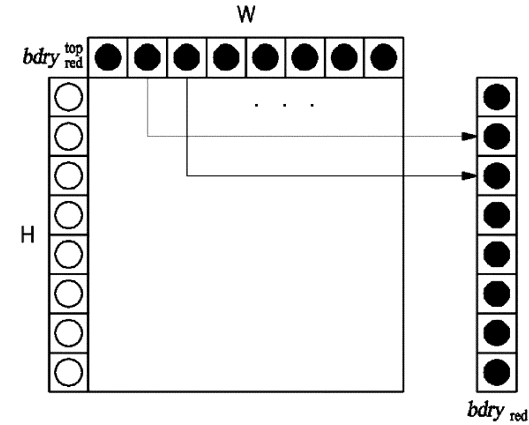


【 7 c 】

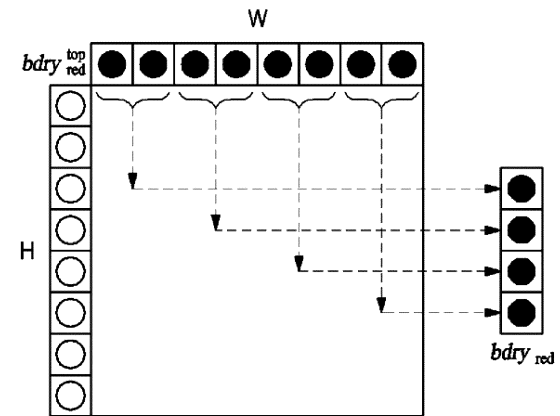


10

【 8 a 】



【 8 b 】



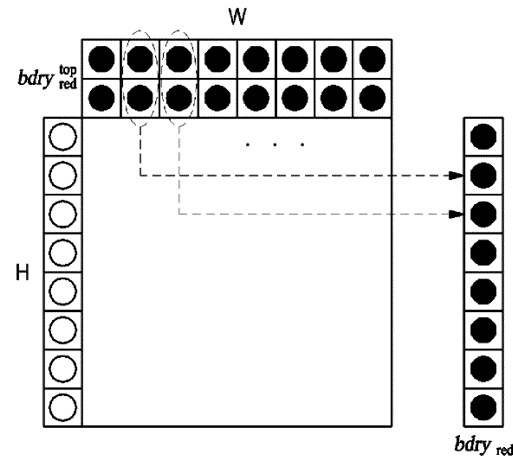
20

30

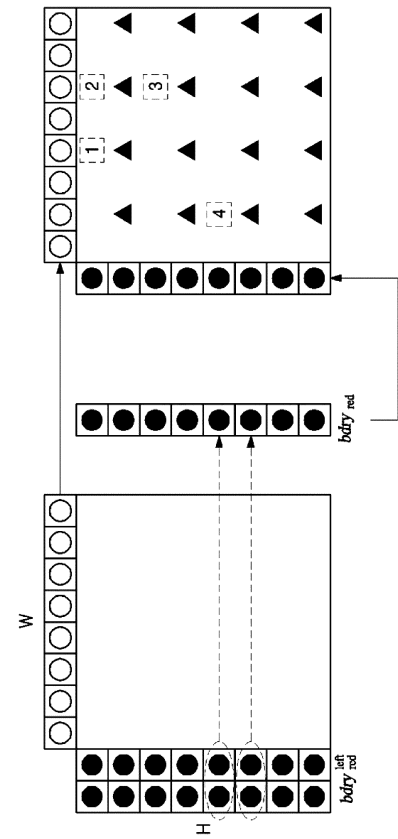
40

50

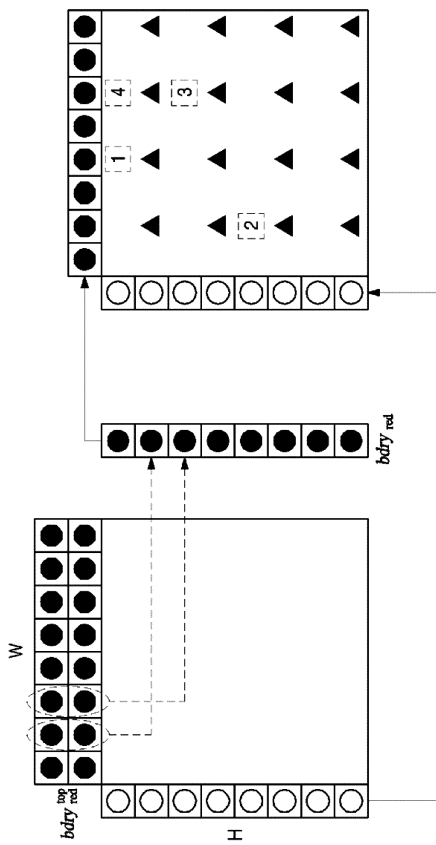
【図 8 c】



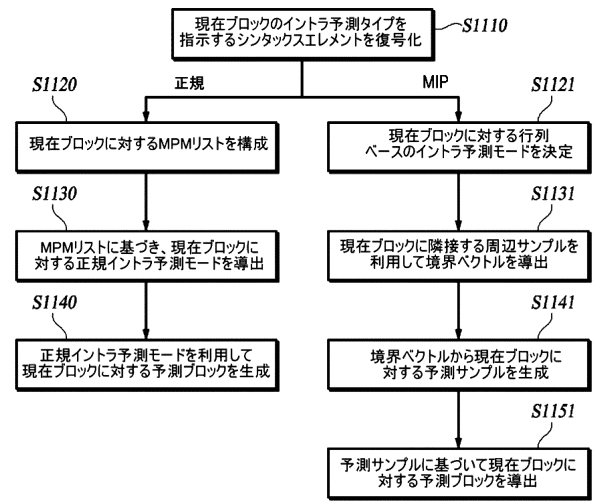
【図 9】



【図 10】



【図 11】



10

20

30

40

50

フロントページの続き

韓国(KR)

(31)優先権主張番号 10-2020-0077014

(32)優先日 令和2年6月24日(2020.6.24)

(33)優先権主張国・地域又は機関

韓国(KR)

- グ、イファヨデ - ギル 52

52, Ewhayeodae - gil Seodaemun - gu Seoul 03760, Republic of Korea

(74)代理人 110000051

弁理士法人共生国際特許事務所

(72)発明者 カン, ゼ ウォン

大韓民国 04069 ソウル マポ - グ トクマク - ロ 145, 104 - 904

(72)発明者 バク, サン ヒョ

大韓民国 07782 ソウル ガンソ - グ ガンソ - ロ 5ラ - ギル 77, 403

(72)発明者 バク, スン ウク

大韓民国 16803 ギョンギ - ド ヨンイン - シ スジ - グ テボン - ロ 17, 403 - 302

(72)発明者 イム, ファ ピョン

大韓民国 18271 ギョンギ - ド ファソン - シ ナムヤン - ウプ ヨクゴル - ロ 49ボン - ギル 29, 202

審査官 岩井 健二

(56)参考文献 国際公開第2020/251330(WO, A1)

国際公開第2020/249633(WO, A1)

国際公開第2020/221203(WO, A1)

国際公開第2020/207502(WO, A1)

Jonathan Pfaff, et al., CE3: Affine linear weighted intra prediction (CE3-4.1, CE3-4.2), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N0217, 14th Meeting: Geneva, CH, 2019年03月, pp.1-17

Benjamin Bross, Jianle Chen, and Shan Liu, Versatile Video Coding (Draft 5), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N1001 (version 8), 14th Meeting: Geneva, CH, 2019年06月11日, pp.53-57,109-114,127-142

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00 - 19/98