

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6203755号
(P6203755)

(45) 発行日 平成29年9月27日 (2017.9.27)

(24) 登録日 平成29年9月8日 (2017.9.8)

(51) Int. Cl. F I
HO 4 N 19/436 (2014.01) HO 4 N 19/436
HO 4 N 19/70 (2014.01) HO 4 N 19/70

請求項の数 10 (全 52 頁)

(21) 出願番号	特願2014-553301 (P2014-553301)	(73) 特許権者	595020643
(86) (22) 出願日	平成24年12月19日 (2012.12.19)		クォアルコム・インコーポレイテッド
(65) 公表番号	特表2015-507907 (P2015-507907A)		QUALCOMM INCORPORATED
(43) 公表日	平成27年3月12日 (2015.3.12)		アメリカ合衆国、カリフォルニア州 92
(86) 国際出願番号	PCT/US2012/070687		121-1714、サン・ディエゴ、モア
(87) 国際公開番号	W02013/109383		ハウス・ドライブ 5775
(87) 国際公開日	平成25年7月25日 (2013.7.25)	(74) 代理人	100108855
審査請求日	平成27年11月20日 (2015.11.20)		弁理士 蔵田 昌俊
(31) 優先権主張番号	61/588,096	(74) 代理人	100109830
(32) 優先日	平成24年1月18日 (2012.1.18)		弁理士 福原 淑弘
(33) 優先権主張国	米国 (US)	(74) 代理人	100103034
(31) 優先権主張番号	13/718,945		弁理士 野河 信久
(32) 優先日	平成24年12月18日 (2012.12.18)	(74) 代理人	100075672
(33) 優先権主張国	米国 (US)		弁理士 峰 隆司

最終頁に続く

(54) 【発明の名称】 ビデオコーディングにおける波面並列処理のためのサブストリーム

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを符号化する方法であって、前記方法は、

ピクチャを複数のタイルに区分することと、前記ピクチャは、前記ビデオデータのビデオピクチャのシーケンス中にあり、前記ピクチャは、複数のコード化ツリーブロック (CTB) 行を有し、前記 CTB 行の各々は 1 CTB 上であり、前記複数の CTB 行の各それぞれの CTB 行は、前記複数のタイルのそれぞれのタイルを形成し、

複数のサブストリームを生成するために波面並列処理 (WPP) を実行することと、前記複数のサブストリームの各それぞれのサブストリームは、前記複数のタイルのうちの 1 つのタイルの連続する一連の符号化ビットを含み、前記サブストリームのいずれも 2 つ以上のタイルのコード化ビットを含まず、前記サブストリームのうちの少なくともいくつかは、前記サブストリームのバイト整合を保証するパディングビットを含み、

ビットストリームで、特定の値を有するシンタックス要素を信号伝達することと、前記特定の値を有する前記シンタックス要素は、前記ピクチャを符号化するために WPP が使用されること、及び前記複数の CTB 行の各それぞれの CTB 行が前記複数のタイルのそれぞれのタイルを形成することを示し、

スライスヘッダ及びスライスデータを含むそれぞれのコード化スライスネットワークアブストラクションレイヤ (NAL) 単位を生成することと、前記スライスヘッダは、ある数のエントリオフセットシンタックス要素及びエントリオフセットシンタックス要素のセットを含み、エントリオフセットシンタックス要素の前記数は、前記エントリオフセット

10

20

シンタックス要素のセット中のエントリオフセットシンタックス要素の総数を指定し、前記スライスデータは、前記複数のサブストリームの中の、エントリオフセットシンタックス要素の前記総数 + 1 個のサブストリームにより構成され、前記複数のエントリオフセットシンタックス要素の各それぞれのエントリオフセットシンタックス要素は、それぞれのエントリポイントオフセットをバイトで指定する、

を備える、方法。

【請求項 2】

前記シンタックス要素を信号伝達することは、前記ピクチャを符号化するために W P P が使用されることを示す前記シンタックス要素を含むピクチャパラメータセット (P P S) を生成することを備え、又は、前記シンタックス要素を信号伝達することは、前記ピクチャを符号化するために W P P が使用されることを示す前記シンタックス要素を含むシーケンスパラメータセット (S P S) を生成することを備える、請求項 1 に記載の方法。

【請求項 3】

前記複数のサブストリームを生成するために W P P を実行することは、

第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと、

前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー符号化することと、を備え、前記第 2 の C T B が、前記ピクチャの前記左境界に隣接し、C T B の 1 行が前記第 1 の C T B よりも低い、請求項 1 に記載の方法。

【請求項 4】

ビデオデータを復号する方法であって、前記方法は、

スライスヘッダ及びスライスデータを含むコード化スライスネットワークアブストラクションレイヤ (N A L) 単位を含むビットストリームを受信することと、前記スライスヘッダは、ある数のエントリオフセットシンタックス要素及びオフセットシンタックス要素のセットを含み、エントリオフセットシンタックス要素の前記数は、前記オフセットシンタックス要素のセット中のオフセットシンタックス要素の総数を指定し、前記スライスデータは、複数のサブストリームの中の、オフセットシンタックス要素の前記総数 + 1 個のサブストリームにより構成され、前記複数のオフセットシンタックス要素の各それぞれのエントリオフセットシンタックス要素は、それぞれのエントリポイントオフセットをバイトで指定し、前記サブストリームの各々は、複数のタイルの中の 1 つのタイルの連続する一連の符号化ビットを含み、前記ビデオデータのピクチャは、前記複数のタイルに区分され、前記ピクチャは、複数のコード化ツリーブロック (C T B) 行を有し、前記 C T B 行の各々は 1 C T B 上であり、前記サブストリームのうちの少なくともいくつかは、前記サブストリームのバイト整合を保証するパディングビットを含み、

特定の値を有する前記ビットストリーム中のシンタックス要素に基づいて、波面並列処理 (W P P) を使用して前記ピクチャが符号化され、及び前記複数の C T B 行の各それぞれの C T B 行が前記複数のタイルのそれぞれのタイルを形成すると決定することと、

W P P を使用して前記スライスデータを復号することと

を備える、方法。

【請求項 5】

前記ビットストリームが前記ピクチャを符号化するために W P P が使用されることを示す前記シンタックス要素を含むピクチャパラメータセット (P P S) を含む、請求項 4 に記載の方法。

【請求項 6】

前記ビットストリームが前記ピクチャを符号化するために W P P が使用されることを示す前記シンタックス要素を含むシーケンスパラメータセット (S P S) を含む、請求項 4 に記載の方法。

【請求項 7】

W P P を使用して前記スライスデータを復号することは、

第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと、

前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー復号することと、前記第 2 の C T B が、前記ピクチャの前記左境界に隣接し、C T B の 1 行が前記第 1 の C T B よりも低い、を備える、請求項 4 に記載の方法。

【請求項 8】

前記方法がモバイルコンピュータ機器によって実行される、請求項 4 に記載の方法。

【請求項 9】

コンピュータ機器の 1 つ以上のプロセッサによって実行されたとき、請求項 1 ないし 8 のいずれか一項に記載の方法を行うように前記コンピュータ機器を構成する命令を記憶したコンピュータ可読記憶媒体。

【請求項 10】

請求項 1 ないし 8 のいずれか一項に記載の方法を行うように構成された 1 つ以上のプロセッサを備えるコンピュータ機器。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本出願は、その内容全体が参照により本明細書に組み込まれる、2012年1月18日に提出された米国仮特許出願第61/588,096号の利益を主張する。

【0002】

[0002]本開示は、ビデオコード化(video coding)(即ち、ビデオデータの符号化(encoding)及び/又は復号(decoding))に関する。

【背景技術】

【0003】

[0003]デジタルビデオ機能は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップ又はデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録機器、デジタルメディアプレーヤ、ビデオゲーム機器、ビデオゲームコンソール、セルラー又は衛星無線電話、所謂「スマートフォン」、ビデオ遠隔会議機器、ビデオストリーミング機器などを含む、広範囲にわたる機器に組み込まれ得る。デジタルビデオ機器は、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、Part 10、アドバンスドビデオコード化(AVC)、現在開発中の高効率ビデオコード化(HEVC)規格によって定義された規格、及びそのような規格の拡張に記載されているビデオ圧縮技法などのビデオ圧縮技法を実装する。ビデオ機器は、そのようなビデオ圧縮技法を実装することによって、デジタルビデオ情報をより効率的に送信、受信、符号化、復号、及び/又は記憶し得る。

【0004】

[0004]ビデオ圧縮技法は、ビデオシーケンスに固有の冗長性を低減又は除去するために空間的(イントラピクチャ)予測及び/又は時間的(インターピクチャ)予測を実行する。ブロックベースのビデオコード化の場合、ビデオスライス(即ち、ビデオフレーム又はビデオフレームの一部)はビデオブロックに区分され、これらのビデオブロックは、ツリーブロック、コード化単位(CU)及び/又はコード化ノードと呼ばれることがある。ピクチャのイントラコード化(I)スライス中のビデオブロックは、同じピクチャ中の隣接ブロックにおける参照サンプルに対する空間的予測を使用して符号化される。ピクチャのインターコード化(P又はB)スライス中のビデオブロックは、同じピクチャ中の隣接ブロック中の参照サンプルに対する空間的予測、又は他の参照ピクチャ中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

【0005】

[0005]空間的予測又は時間的予測は、コード化されるべきブロックのための予測ブロックを生じる。残差データは、コード化されるべき元のブロックと予測ブロックとの間の画素差分を表す。インターコード化ブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトル、及びコード化ブロックと予測ブロックとの間の差分を示す残差データに従って符号化される。イントラコード化ブロックは、イントラコード化モードと残差データとに従って符号化される。更なる圧縮のために、残差データは、画素領域から変換領域に変換されて、残差係数が得られ得、その残差係数は、次いで量子化され得る。量子化された係数は、最初は2次元アレイで構成され、係数の1次元ベクトルを生成するために走査され得、なお一層の圧縮を達成するためにエントロピーコード化が適用され得る。

10

【発明の概要】

【0006】

[0006]概して、本開示では、タイル及び波面並列処理(WPP: wavefront parallel processing)を伴うビデオコード化のための技法について説明する。より詳細には、ビデオエンコーダは、ビデオピクチャのシーケンスのピクチャを符号化するためにWPPが使用されるかどうかを信号伝達(signaling)し得る。ピクチャを符号化するためにWPPが使用される場合、ビデオエンコーダは複数のサブストリームを生成する。サブストリームの各々は、ピクチャのスライス中のコード化ツリーブロック(CTB: coding tree block)の1つの符号化された行を表す連続する一連のビットを含み得る。ビデオエンコーダは、複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ(NAL: network abstraction layer)単位を生成し得る。ビデオデコーダは、コード化スライスNAL単位を含むビットストリームを受信し得る。更に、ビデオデコーダは、ビットストリーム中のシンタックス要素に基づいて、WPPを使用してスライスが符号化されると決定し得、WPPを使用してスライスを復号し得る。

20

【0007】

[0007]一態様では、本開示は、ビデオデータを符号化するための方法について説明する。本方法は、ビデオピクチャのシーケンスのピクチャを符号化するためにWPPが使用されることを信号伝達することを備える。本方法はまた、複数のサブストリームを生成するためにWPPを実行することであって、サブストリームの各々が、ピクチャのスライス中のCTBの1つの符号化された行を表す連続する一連のビットを含む、実行することを備える。本方法はまた、複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ(NAL)単位を生成することを備える。

30

【0008】

[0008]別の態様では、本開示は、ビデオピクチャのシーケンスのピクチャを符号化するためにWPPが使用されることを信号伝達するように構成された1つ又は複数のプロセッサを備えるコンピュータ機器について説明する。1つ又は複数のプロセッサはまた、複数のサブストリームを生成するためにWPPを実行することであって、サブストリームの各々が、ピクチャのスライス中のCTBの1つの符号化された行を表す連続する一連のビットを含む、実行することを行うように構成される。1つ又は複数のプロセッサはまた、複数のサブストリームを含むコード化スライスNAL単位を生成するように構成される。

40

【0009】

[0009]別の態様では、本開示は、ピクチャを符号化するためにWPPが使用されることを信号伝達するための手段を備えるコンピュータ機器について説明する。コンピュータ機器はまた、複数のサブストリームを生成するためにWPPを実行するための手段であって、サブストリームの各々が、ピクチャのスライス中のCTBの1つの符号化された行を表す連続する一連のビットを含む、実行するための手段を備える。更に、本方法は、複数のサブストリームを含むコード化スライスNAL単位を生成するための手段を備える。

【0010】

[0010]別の態様では、本開示は、コンピュータ機器の1つ又は複数のプロセッサによって実行されたとき、ピクチャを符号化するためにWPPが使用されることを信号伝達する

50

ようにコンピュータ機器を構成する命令を記憶したコンピュータ可読記憶媒体について説明する。命令はまた、複数のサブストリームを生成するためにWPPを実行することであって、サブストリームの各々が、ピクチャのスライス中のCTBの1つの符号化された行を表す連続する一連のビットを含む、実行することを行うようにコンピュータ機器を構成する。更に、命令は、複数のサブストリームを含むコード化スライスNAL単位を生成するようにコンピュータ機器を構成する。

【0011】

【0011】別の態様では、本開示は、ビデオデータを復号するための方法について説明する。本方法は、コード化スライスNAL単位を含むビットストリームを受信することを備え、コード化スライスNAL単位が複数のサブストリームを含み、サブストリームの各々が、ピクチャのスライス中のコード化ツリーブロック（CTB）の1つの行を表す連続する一連のビットを含む。本方法はまた、ビットストリーム中のシンタックス要素に基づいて、WPPを使用してスライスが符号化されると決定することを備える。更に、本方法は、WPPを使用してスライスを復号することを備える。

10

【0012】

【0012】別の態様では、本開示は、コード化スライスNAL単位を含むビットストリームを受信するように構成された1つ又は複数のプロセッサを備えるコンピュータ機器であって、コード化スライスNAL単位が複数のサブストリームを含み、サブストリームの各々が、ピクチャのスライス中のCTBの1つの行を表す連続する一連のビットを含む、コンピュータ機器について説明する。1つ又は複数のプロセッサはまた、ビットストリーム中の1つ又は複数のシンタックス要素に基づいて、WPPを使用してスライスが符号化されると決定するように構成される。更に、1つ又は複数のプロセッサは、WPPを使用してスライスを復号するように構成される。

20

【0013】

【0013】別の態様では、本開示は、コード化スライスNAL単位を含むビットストリームを受信するための手段を備えるコンピュータ機器であって、コード化スライスNAL単位が複数のサブストリームを含み、サブストリームの各々が、ピクチャのスライス中のCTBの1つの行を表す連続する一連のビットを含む、コンピュータ機器について説明する。コンピュータ機器はまた、ビットストリーム中の1つ又は複数のシンタックス要素に基づいて、WPPを使用してスライスが符号化されると決定するための手段を備える。コンピュータ機器はまた、WPPを使用してスライスを復号するための手段を備える。

30

【0014】

【0014】別の態様では、本開示は、コンピュータ機器の1つ又は複数のプロセッサによって実行されたとき、コード化スライスNAL単位を含むビットストリームを受信するようにコンピュータ機器を設定する命令を記憶したコンピュータ可読記憶媒体であって、コード化スライスNAL単位が複数のサブストリームを含み、サブストリームの各々が、ピクチャのスライス中のCTBの1つの行を表す連続する一連のビットを含む、コンピュータ可読記憶媒体について説明する。命令はまた、ビットストリーム中の1つ又は複数のシンタックス要素に基づいて、WPPを使用してスライスが符号化されると決定するようにコンピュータ機器を構成する。更に、命令は、WPPを使用してスライスを復号するようにコンピュータ機器を構成する。

40

【0015】

【0015】本開示の1つ又は複数の例の詳細を添付の図面及び以下の説明に記載する。他の特徴、目的、及び利点は、説明、図面、及び特許請求の範囲から明らかになる。

【図面の簡単な説明】

【0016】

【図1】【0016】本開示で説明する技法を利用し得る例示的なビデオコード化システムを示すブロック図。

【図2】【0017】本開示で説明する技法を実装し得る例示的なビデオエンコーダを示すブロック図。

50

【図 3】[0018]本開示で説明する技法を実装し得る例示的なビデオデコードを示すブロック図。

【図 4】[0019]本開示の 1 つ又は複数の態様による、単一のピクチャ内のタイルと波面並列処理 (WPP) との組合せが許可されない、ビデオデータを符号化するためのビデオエンコーダの例示的な動作を示すフローチャート。

【図 5】[0020]本開示の 1 つ又は複数の態様による、単一のピクチャ内のタイルと WPP との組合せが許可されない、ビデオデータを復号するためのビデオデコードの例示的な動作を示すフローチャート。

【図 6】[0021]本開示の 1 つ又は複数の態様による、単一のピクチャ内のタイルと WPP との組合せが許可されない、ビデオデータを復号するためのビデオデコードの別の例示的な動作を示すフローチャート。

10

【図 7】[0022]本開示の 1 つ又は複数の態様による、ピクチャのコード化ツリーブロック (CTB: coding tree block) の各行が別個のサブストリーム中にある、ビデオデータを符号化するためのビデオエンコーダの例示的な動作を示すフローチャート。

【図 8】[0023]本開示の 1 つ又は複数の態様による、ピクチャの CTB の各行が別個のサブストリーム中にある、ビデオデータを復号するためのビデオデコードの例示的な動作を示すフローチャート。

【図 9 A】[0024]本開示の 1 つ又は複数の態様による、スライスデータを構文解析するための例示的なコンテキスト適応型バイナリ算術コード化 (CABAC: Context-Adaptive Binary Arithmetic Coding) 構文解析プロセスの第 1 の部分を示すフローチャート。

20

【図 9 B】[0025]図 9 A の例示的な CABAC 構文解析プロセスの続きを示すフローチャート。

【図 10】[0026]WPP の一例を示す概念図。

【図 11】[0027]ピクチャが複数のタイルに区分されるときに例示的なコード化順序を示す概念図。

【発明を実施するための形態】

【0017】

[0028]ビデオコード化中に、ピクチャは、複数のタイル、波面並列処理 (WPP) 波、及び / 又はエントロピースライスに区分され得る。ピクチャのタイルは、ピクチャを通して水平及び / 又は垂直タイル境界によって画定される。ピクチャのタイルはラスト走査順序に従ってコード化され、各タイル内のコード化ツリーブロック (CTB) もラスト走査順序に従ってコード化される。WPP では、ピクチャ中の CTB の各行が「WPP 波」である。ビデオコード化がピクチャをコード化するために WPP を使用するとき、ビデオコード化は、直接により高い WPP 波の 2 つ以上の CTB をコード化した後に、ビデオコード化は、WPP 波の CTB を左から右にコード化し始め得る。エントロピースライスは、ラスト走査順序に従って一連の連続する CTB を含み得る。エントロピースライス境界を越えて来る情報の使用は、エントロピーコード化コンテキストの選択における使用のためには禁止されるが、他の目的のためには許可され得る。

30

【0018】

[0029]既存のビデオコード化システムでは、ピクチャは、タイルと、WPP 波と、エントロピースライスとの任意の組合せを有し得る。例えば、ピクチャは複数のタイルに区分され得る。この例では、タイルのうちの幾つかの中の CTB は、ラスト走査順序に従ってコード化され得るが、タイルのうちの他の幾つかの中の CTB は、WPP を使用してコード化され得る。ピクチャがタイルと WPP 波とエントロピースライスとの組合せを含むことを可能にすることは、そのようなビデオコード化システムの実装の複雑さ及びコストを不必要に増加させ得る。

40

【0019】

[0030]本開示の技法はこの問題を矯正し得る。即ち、本開示の技法によれば、2 つ以上のタイルと、WPP 波と、エントロピースライスとのいずれかのピクチャ内の組合せは許可されない。例えば、ビデオエンコーダは、ピクチャが第 1 のコード化モードに従って符

50

号化されるか第2のコード化モードに従って符号化されるかを示すシンタックス要素を含むビットストリームを生成し得る。第1のコード化モードでは、ピクチャは、WPPを使用して完全に符号化される。第2のコード化モードでは、ピクチャは1つ又は複数のタイルを有し、ピクチャの各タイルは、WPPを使用せずに符号化される。

【0020】

[0031]更に、この例では、ビデオデコーダは、ピクチャのコード化表現を含むビットストリームからシンタックス要素を構文解析し得る。シンタックス要素が特定の値を有すると決定したことに応答して、ビデオデコーダは、WPPを使用してピクチャを完全に復号し得る。シンタックス要素が特定の値を有しないと決定したことに応答して、ビデオデコーダは、WPPを使用せずにピクチャの各タイルを復号し得る。ピクチャは1つ又は複数のタイルを有し得る。

10

【0021】

[0032]添付の図面は例を示す。添付の図面中の参照番号によって示される要素は、以下の説明において同様の参照番号によって示される要素に対応する。本開示では、序数語（例えば、「第1の」、「第2の」、「第3の」など）で始まる名前を有する要素は、必ずしもそれらの要素が特定の順序を有することを暗示するとは限らない。むしろ、そのような序数語は、同じ又は同様のタイプの異なる要素を指すために使用されるにすぎない。

【0022】

[0033]図1は、本開示の技法を利用し得る例示的なビデオコード化システム10を示すブロック図である。本明細書で使用する「ビデオコーダ」という用語は、総称的にビデオエンコーダとビデオデコーダの両方を指す。本開示では、「ビデオコード化」又は「コード化」という用語は、総称的にビデオ符号化又はビデオ復号を指すことがある。

20

【0023】

[0034]図1に示すように、ビデオコード化システム10は、発信源機器12と宛先機器14とを含む。発信源機器12は、符号化ビデオデータを生成する。従って、発信源機器12は、ビデオ符号化機器又はビデオ符号化装置と呼ばれることがある。宛先機器14は、発信源機器12によって生成された符号化ビデオデータを復号し得る。従って、宛先機器14は、ビデオ復号機器又はビデオ復号装置と呼ばれることがある。発信源機器12及び宛先機器14は、ビデオコード化機器又はビデオコード化装置の例であり得る。発信源機器12及び宛先機器14は、デスクトップコンピュータ、モバイルコンピュータ機器、ノートブック（例えば、ラップトップ）コンピュータ、タブレットコンピュータ、セットトップボックス、所謂「スマート」フォンなどの電話ハンドセット、テレビジョン、カメラ、表示装置、デジタルメディアプレーヤ、ビデオゲームコンソール、車内コンピュータなどを含む、広範囲の機器を備え得る。

30

【0024】

[0035]宛先機器14は、チャンネル16を介して発信源機器12から符号化ビデオデータを受信し得る。チャンネル16は、発信源機器12から宛先機器14に符号化ビデオデータを移動することが可能な1つ又は複数の媒体及び/又は機器を備え得る。一例では、チャンネル16は、発信源機器12が符号化ビデオデータを宛先機器14にリアルタイムで直接送信することを可能にする1つ又は複数の通信媒体を備え得る。この例では、発信源機器12は、ワイヤレス通信プロトコルなどの通信規格に従って符号化ビデオデータを変調し得、変調されたビデオデータを宛先機器14に送信し得る。1つ又は複数の通信媒体は、無線周波数（RF）スペクトル若しくは1つ又は複数の物理伝送線路など、ワイヤレス及び/又はワイヤード通信媒体を含み得る。1つ又は複数の通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、又はグローバルネットワーク（例えば、インターネット）など、パケットベースネットワークの一部を形成し得る。1つ又は複数の通信媒体は、発信源機器12から宛先機器14への通信を可能にする、ルータ、スイッチ、基地局、又は他の機器を含み得る。

40

【0025】

[0036]別の例では、チャンネル16は、発信源機器12によって生成された符号化ビデオ

50

データを記憶する記憶媒体を含み得る。この例では、宛先機器 14 は、ディスクアクセス又はカードアクセスを介して記憶媒体にアクセスし得る。記憶媒体は、ブルーレイ（登録商標）ディスク、DVD、CD-ROM、フラッシュメモリ、又は符号化ビデオデータを記憶するための他の好適なデジタル記憶媒体など、様々なローカルにアクセスされるデータ記憶媒体を含み得る。

【0026】

[0037]更なる例では、チャンネル 16 は、発信源機器 12 によって生成された符号化ビデオを記憶するファイルサーバ又は別の中間記憶装置を含み得る。この例では、宛先機器 14 は、ストリーミング又はダウンロードを介して、ファイルサーバ又は他の中間記憶装置に記憶された符号化ビデオデータにアクセスし得る。ファイルサーバは、符号化ビデオデータを記憶することと、符号化ビデオデータを宛先機器 14 に送信することとが可能なタイプのサーバであり得る。例示的なファイルサーバとしては、（例えば、ウェブサイト用の）ウェブサーバ、ファイル転送プロトコル（FTP）サーバ、ネットワーク接続記憶（NAS）機器、及びローカルディスクドライブがある。

【0027】

[0038]宛先機器 14 は、インターネット接続などの標準的なデータ接続を通して符号化ビデオデータにアクセスし得る。データ接続の例示的なタイプとしては、ファイルサーバに記憶された符号化ビデオデータにアクセスするのに好適な、ワイヤレスチャンネル（例えば、Wi-Fi（登録商標）接続）、ワイヤード接続（例えば、DSL、ケーブルモデムなど）、又は両方の組合せがある。ファイルサーバからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、又は両方の組合せであり得る。

【0028】

[0039]本開示の技法は、ワイヤレス適用例又は設定に限定されない。本技法は、オーバーエアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、例えばインターネットを介したストリーミングビデオ送信、データ記憶媒体に記憶するためのビデオデータの符号化、データ記憶媒体に記憶されたビデオデータの復号、又は他の適用例など、様々なマルチメディア適用例をサポートするビデオコード化に適用され得る。幾つかの例では、ビデオコード化システム 10 は、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、及び/又はビデオテレフォニーなどの適用例をサポートするために、単方向又は双方向のビデオ送信をサポートするように構成され得る。

【0029】

[0040]図 1 の例では、発信源機器 12 は、ビデオ発信源 18 と、ビデオエンコーダ 20 と、出力インターフェース 22 とを含む。幾つかの例では、出力インターフェース 22 は、変調器/復調器（モデム）及び/又は送信機を含み得る。ビデオ発信源 18 は、例えばビデオカメラなどの撮像装置、以前に撮影されたビデオデータを含んでいるビデオアーカイブ、ビデオコンテンツプロバイダからビデオデータを受信するためのビデオフィードインターフェース、及び/又はビデオデータを生成するためのコンピュータグラフィックスシステム、若しくはビデオデータのそのような発信源の組合せを含み得る。

【0030】

[0041]ビデオエンコーダ 20 は、ビデオ発信源 18 からのビデオデータを符号化し得る。幾つかの例では、発信源機器 12 は、出力インターフェース 22 を介して宛先機器 14 に符号化ビデオデータを直接送信する。符号化ビデオデータはまた、復号及び/又は再生のための宛先機器 14 による後のアクセスのために記憶媒体又はファイルサーバ上に記憶され得る。

【0031】

[0042]図 1 の例では、宛先機器 14 は、入力インターフェース 28 と、ビデオデコーダ 30 と、表示装置 32 とを含む。幾つかの例では、入力インターフェース 28 は、受信機及び/又はモデムを含む。入力インターフェース 28 は、チャンネル 16 を介して符号化ビデオデータを受信し得る。表示装置 32 は、宛先機器 14 と一体化され得るか又はその外部にあり得る。概して、表示装置 32 は、復号ビデオデータを表示する。表示装置 32 は

、液晶表示器（LCD）、プラズマ表示器、有機発光ダイオード（OLED）表示器、又は別のタイプの表示装置など、様々な表示装置を備え得る。

【0032】

[0043]ビデオエンコーダ20及びビデオデコーダ30は、現在開発中の高効率ビデオコード化（HEVC）規格など、ビデオ圧縮規格に従って動作し得、HEVCテストモデル（HM）に準拠し得る。「HEVC Working Draft 5」又は「WD 5」と呼ばれる次のHEVC規格のドラフトは、2012年10月10日現在、http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-v3.zipからダウンロード可能である、Brossら、「WD5: Working Draft 5 of High-Efficiency Video Coding」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコード化共同研究部会（JCT-VC）、第7回会合、ジュネーブ、スイス、2011年11月に記載されており、その内容全体は参照により本明細書に組み込まれる。「HEVC Working Draft 9」と呼ばれる次のHEVC規格の別のドラフトは、2012年11月7日現在、http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zipからダウンロード可能である、Brossら、「High Efficiency Video Coding (HEVC) text specification draft 9」、ITU-T SG16 WP3とISO/IEC JTC1/SC29/WG11とのビデオコード化共同研究部会（JCT-VC）、第11回会合、上海、中国、2012年10月に記載されており、その内容全体は参照により本明細書に組み込まれる。

10

【0033】

[0044]代替として、ビデオエンコーダ20及びビデオデコーダ30は、ITU-T H.261、ISO/IEC MPEG-1 Visual、ITU-T H.262又はISO/IEC MPEG-2 Visual、ITU-T H.263、ISO/IEC MPEG-4 Visual、及びスケーラブルビデオコード化（SVC）拡張とマルチビュービデオコード化（MVC）拡張とを含む、（ISO/IEC MPEG-4 AVCとしても知られる）ITU-T H.264を含む、他のプロプライエタリ規格又は業界規格に従って動作し得る。しかしながら、本開示の技法は、いかなる特定のコード化規格又はコード化技法にも限定されない。

20

【0034】

[0045]この場合も、図1は一例にすぎず、本開示の技法は、符号化機器と復号機器との間のデータ通信を必ずしも含むとは限らないビデオコード化設定（例えば、ビデオ符号化又はビデオ復号）に適用され得る。他の例では、データがローカルメモリから取り出されること、ネットワークを介してストリーミングされることなどが行われる。符号化機器はデータを符号化し、メモリに記憶し得、及び/又は復号機器はメモリからデータを取り出し、復号し得る。多くの例では、符号化及び復号は、互いに通信しないが、メモリにデータを符号化し、及び/又はメモリからデータを取り出して復号するだけである機器によって実行される。

30

【0035】

[0046]ビデオエンコーダ20及びビデオデコーダ30はそれぞれ、1つ又は複数のマイクロプロセッサ、デジタル信号プロセッサ（DSP）、特定用途向け集積回路（ASIC）、フィールドプログラマブルゲートアレイ（FPGA）、ディスクリート論理、ハードウェアなど、様々な好適な回路のいずれか、若しくはそれらの任意の組合せとして実装され得る。本技法が部分的にソフトウェアで実装される場合、機器は、好適な非一時的コンピュータ可読記憶媒体にソフトウェアの命令を記憶し得、1つ又は複数のプロセッサを使用してその命令をハードウェアで実行して、本開示の技法を実行し得る。（ハードウェア、ソフトウェア、ハードウェアとソフトウェアの組合せなどを含む）上記のいずれも、1つ又は複数のプロセッサであると見なされ得る。ビデオエンコーダ20及びビデオデコーダ30の各々は1つ又は複数のエンコーダ又はデコーダ中に含まれ得、そのいずれも、それぞれの機器において複合エンコーダ/デコーダ（コーデック）の一部として統合され得る。

40

50

【 0 0 3 6 】

[0047]本開示では、概して、ビデオエンコーダ 2 0 が、ある情報をビデオデコーダ 3 0 などの別の機器に「信号伝達(signaling)」することに言及することがある。「信号伝達」という用語は、概して、符号化ビデオデータを表すシンタックス要素及び/又は他のデータの通信を指し得る。そのような通信は、リアルタイム又はほぼリアルタイムで行われ得る。代替的に、そのような通信は、符号化時に符号化ビットストリーム中でシンタックス要素をコンピュータ可読記憶媒体に記憶するときに行われることがあるなど、ある時間期間にわたって行われ得、次いで、これらの要素は、この媒体に記憶された後の任意の時間に復号機器によって取り出され得る。

【 0 0 3 7 】

[0048]上記で手短かに述べたように、ビデオエンコーダ 2 0 はビデオデータを符号化する。ビデオデータは 1 つ又は複数のピクチャを備え得る。ピクチャの各々は静止画像であり得る。幾つかの例では、ピクチャはビデオ「フレーム」と呼ばれることがある。ビデオエンコーダ 2 0 は、ビデオデータのコード化表現を形成するビットのシーケンスを含むビットストリームを生成し得る。ビットストリームは、コード化されたピクチャと、関連データとを含み得る。コード化されたピクチャは、ピクチャのコード化表現である。関連データは、シーケンスパラメータセット (S P S : sequence parameter set) と、ピクチャパラメータセット (P P S : picture parameter set) と、他のシンタックス構造とを含み得る。S P S は、ピクチャの 0 個以上のシーケンスに適用可能なパラメータを含んでいることがある。P P S は、0 個以上のピクチャに適用可能なパラメータを含んでいることがある。

【 0 0 3 8 】

[0049]ピクチャの符号化表現を生成するために、ビデオエンコーダ 2 0 は、ピクチャをコード化ツリーブロック (C T B) のグリッドに区分し得る。幾つかの事例では、C T B は、「ツリーブロック (tree block)」、「最大コード化単位 (largest coding unit)」(L C U) 又は「コード化ツリー単位 (coding tree unit)」と呼ばれることがある。H E V C の C T B は、H . 2 6 4 / A V C など、以前の規格のマクロブロックに広い意味で類似し得る。しかしながら、C T B は、必ずしも特定のサイズに限定されとは限らず、1 つ又は複数のコード化単位 (C U) を含み得る。

【 0 0 3 9 】

[0050]C T B の各々は、ピクチャ内の画素の様々な等しいサイズのブロックに関連付けられ得る。各画素は、ルミナンス (ルーマ) サンプルと、2 つのクロミナンス (クロマ) サンプルとを備え得る。従って、各 C T B は、ルミナンスサンプルのブロックと、クロミナンスサンプルの 2 つのブロックとに関連付けられ得る。説明を簡単にするために、本開示では、画素の 2 次元アレイを画素ブロックと呼び、サンプルの 2 次元アレイをサンプルブロックと呼ぶことがある。ビデオエンコーダ 2 0 は、クワッドツリー区分 (quad-tree partitioning) を使用して、C T B に関連する画素ブロックを、C U に関連する画素ブロックに区分し得、従って「コード化ツリーブロック (coding tree blocks) 」という名前がある。

【 0 0 4 0 】

[0051]ピクチャの C T B は、1 つ又は複数のスライスにグループ化され得る。幾つかの例では、スライスの各々は整数個の C T B を含む。ピクチャを符号化することの一部として、ビデオエンコーダ 2 0 は、ピクチャの各スライスの符号化表現 (即ち、コード化スライス) を生成し得る。コード化スライスを生成するために、ビデオエンコーダ 2 0 は、スライスの各 C T B を符号化してスライスの C T B の各々の符号化表現 (即ち、コード化 C T B) を生成し得る。

【 0 0 4 1 】

[0052]コード化 C T B を生成するために、ビデオエンコーダ 2 0 は、C T B に関連する画素ブロックに対してクワッドツリー区分を再帰的に実行して、画素ブロックを徐々により小さい画素ブロックに分割し得る。より小さい画素ブロックの各々は C U に関連付けら

れ得る。区分されたCUは、その画素ブロックが、他のCUに関連する画素ブロックに区分された、CUであり得る。区分されていないCUは、その画素ブロックが、他のCUに関連する画素ブロックに区分されていない、CUであり得る。

【0042】

[0053]ビデオエンコーダ20は、区分されていないCUごとに1つ又は複数の予測単位(PU)を生成し得る。CUのPUの各々は、CUの画素ブロック内の異なる画素ブロックに関連付けられ得る。ビデオエンコーダ20は、CUのPUごとに予測画素ブロックを生成し得る。PUの予測画素ブロックは画素のブロックであり得る。

【0043】

[0054]ビデオエンコーダ20は、イントラ予測又はインター予測を使用して、PUの予測画素ブロックを生成し得る。ビデオエンコーダ20がイントラ予測を使用してPUの予測画素ブロックを生成する場合、ビデオエンコーダ20は、PUに関連付けられたピクチャの復号画素に基づいて、PUの予測画素ブロックを生成し得る。ビデオエンコーダ20がインター予測を使用してPUの予測画素ブロックを生成する場合、ビデオエンコーダ20は、PUに関連付けられたピクチャ以外の1つ又は複数のピクチャの復号画素に基づいて、PUの予測画素ブロックを生成し得る。

10

【0044】

[0055]ビデオエンコーダ20は、CUのPUの予測画素ブロックに基づいて、CUの残差画素ブロックを生成し得る。CUの残差画素ブロックは、CUのPUのための予測画素ブロック中のサンプルと、CUの元の画素ブロック中の対応するサンプルとの間の差分を示し得る。

20

【0045】

[0056]更に、区分されていないCUに対して符号化することの一部として、ビデオエンコーダ20は、CUの残差画素ブロックに対して再帰的クワッドツリー区分化(recursive quad-tree partitioning)を実行して、CUの残差画素ブロックを、CUの変換単位(TU)に関連付けられた残差データの1つ又は複数のより小さい残差画素ブロックに区分し得る。TUに関連付けられた画素ブロック中の画素はそれぞれ、ルーマサンプルと2つのクロマサンプルとを含むので、TUの各々は、ルーマサンプルの残差サンプルブロックと、クロマサンプルの2つの残差サンプルブロックとに関連付けられ得る。

【0046】

30

[0057]ビデオエンコーダ20は、TUに関連付けられた残差サンプルブロックに1つ又は複数の変換を適用して、係数ブロック(即ち、係数のブロック)を生成し得る。ビデオエンコーダ20は、係数ブロックの各々に対して量子化プロセスを実行し得る。量子化は、概して、係数を表すために使用されるデータの量をできるだけ低減するために係数を量子化して、更なる圧縮を行うプロセスを指す。

【0047】

[0058]ビデオエンコーダ20は、量子化された係数ブロック中の係数を表すシンタックス要素のセットを生成し得る。ビデオエンコーダ20は、これらのシンタックス要素のうちの少なくとも幾つかに、コンテキスト適応型バイナリ算術コード化(CABAC)演算などのエントロピー符号化演算を適用し得る。エントロピー符号化演算を実行することの一部として、ビデオエンコーダ20はコード化コンテキストを選択し得る。CABACの場合、コード化コンテキストは、0値ピンと1値ピンとの確率を示し得る。

40

【0048】

[0059]ビデオエンコーダ20によって生成されるビットストリームは、一連のネットワークアブストラクションレイヤ(NAL: Network Abstraction Layer)単位を含み得る。NAL単位の各々は、NAL単位中のデータのタイプの指示と、データを含んでいるバイトとを含んでいるシンタックス構造であり得る。例えば、NAL単位は、SPS、PPS、コード化スライス、補足エンハンスメント情報(SEI)、アクセス単位デリミタ(access unit delimiter)、フィラーデータ(filler data)、又は別のタイプのデータを表すデータを含むことがある。コード化スライスのNAL単位は、コード化スライスを含むN

50

NAL単位である。

【0049】

[0060]ビデオデコーダ30はビットストリームを受信し得る。ビットストリームは、ビデオエンコーダ20によって符号化されたビデオデータのコード化表現を含み得る。ビデオデコーダ30は、ビットストリームからシンタックス要素を抽出するためにビットストリームを構文解析し得る。ビットストリームから幾つかのシンタックス要素を抽出することの一部として、ビデオデコーダ30は、ビットストリーム中のデータをエントロピー復号（例えば、CABAC復号、指数ゴロム復号など）し得る。ビデオデコーダ30は、ビットストリームから抽出されたシンタックス要素に基づいて、ビデオデータのピクチャを再構成し得る。

10

【0050】

[0061]シンタックス要素に基づいてビデオデータを再構成するためのプロセスは、概して、シンタックス要素を生成するためにビデオエンコーダ20によって実行されるプロセスの逆であり得る。例えば、ビデオデコーダ30は、CUに関連付けられた構文要素に基づいて、CUのPUのための予測画素ブロックを生成し得る。更に、ビデオデコーダ30は、CUのTUに関連付けられた係数ブロックを逆量子化し得る。ビデオデコーダ30は、係数ブロックに対して逆変換を実行して、CUのTUに関連付けられた残差画素ブロックを再構成し得る。ビデオデコーダ30は、予測画素ブロックと残差画素ブロックとに基づいて、CUの画素ブロックを再構築し得る。

【0051】

20

[0062]幾つかの例では、ビデオエンコーダ20は、ピクチャを複数のエントロピースライスに分割し得る。本開示では、スライスをエントロピースライスから区別するために「正規スライス」という用語を使用し得る。エントロピースライスは、正規スライスのCUのサブセットを含み得る。幾つかの例では、ビデオエンコーダ20は、エントロピースライスのいずれも、上限よりも多くのピン（例えば、エントロピーコード化ビット）を含まないように、CUをエントロピースライスの中で区分し得る。各エントロピースライスは別々のNAL単位中に含まれ得る。

【0052】

[0063]本開示では、ピクチャ内予測は、同じピクチャの第2の単位をコード化するための、ピクチャの第1の単位（例えば、CTB、CU、PUなど）に関連する情報の使用を指し得る。エントロピースライス境界を越えるピクチャ内予測は、エントロピーコード化の目的を除いて、可能にされる。例えば、ビデオコード（例えば、ビデオエンコーダ20又はビデオデコーダ30）が特定のPUに対してイントラ予測を実行している場合、ビデオコードは、隣接PUが特定のPUとは異なるエントロピースライス中にある場合でも、隣接PUからのサンプルを使用し得る。この例では、ビデオコードは、隣接PUが特定のPUとは異なるスライス中にある場合、隣接PUからのサンプルを使用することが不可能であり得る。

30

【0053】

[0064]しかしながら、ビデオコードが、特定のPUに関連するデータに対してエントロピーコード化を実行しているとき、ビデオコードは、特定のPUと隣接PUとが同じエントロピースライス中にある場合、隣接PUに関連する情報に基づいてコード化コンテキストを選択することのみを可能にされる。この制限のために、ビデオコードは、スライスの複数のエントロピースライスに対してエントロピーコード化（即ち、エントロピー符号化又は復号）演算を並列に実行することが可能であり得る。従って、ビデオデコーダ30は、複数のエントロピースライスのシンタックス要素を並列に構文解析することが可能であり得る。しかしながら、ビデオデコーダ30は、スライスの複数のエントロピースライスの画素ブロックを並列に再構成することが可能でない。

40

【0054】

[0065]上記のように、コード化スライスNAL単位はコード化スライスを含んでいることができる。このスライスは、エントロピースライス又は正規スライスのいずれかであり

50

得る。コード化スライスNAL単位中のスライスヘッダは、スライスがエントロピースライスであるか正規スライスであるかを示すシンタックス要素（例えば、`entropy_slice_flag`）を含み得る。例えば、シンタックス要素が1に等しい場合、コード化スライスNAL単位中のスライスはエントロピースライスであり得る。

【0055】

[0066]各コード化スライスは、スライスヘッダとスライスデータとを含み得る。エントロピースライスのスライスヘッダは、正規スライスのスライスヘッダとは異なり得る。例えば、エントロピースライスのスライスヘッダ中のシンタックス要素は、正規スライスのスライスヘッダ中のシンタックス要素のサブセットを含み得る。エントロピースライスのスライスヘッダは正規スライスのスライスヘッダよりも少数のシンタックス要素を含むので、エントロピースライスは、軽量スライス、ショートスライスヘッダをもつスライス、又はショートスライスと呼ばれることもある。エントロピースライスは、復号順序でエントロピースライスに先行する正規スライスのスライスヘッダから、エントロピースライスのスライスヘッダから省略されたシンタックス要素を継承し得る。

【0056】

[0067]従来、ビデオエンコーダは、エントロピースライスごとに別個のNAL単位を生成する。個々のNAL単位は、しばしば、別個のパケット中でネットワーク上でトランスポートされる。言い換えれば、ネットワークを介したNAL単位の送信中にパケット当たり1つのNAL単位があり得る。これは、エントロピースライスを含んでいるNAL単位にとって問題になり得る。正規スライスを含むNAL単位を含んでいるパケットが送信中に紛失した場合、ビデオデコーダ30は、正規スライスのスライスヘッダからシンタックス要素を継承するエントロピースライスを使用することが不可能になり得る。更に、第1のエントロピースライスの1つ又は複数のCTBが、ピクチャ内予測のために第2のエントロピースライスの1つ又は複数のCTBに依拠し、第2のエントロピースライスを含むNAL単位を含んでいるパケットが送信中に紛失した場合、ビデオエンコーダ30は、第1のエントロピースライスのCTBを復号することが不可能になり得る。

【0057】

[0068]幾つかの例では、ビデオコーダは、波面並列処理(WPP)を使用してピクチャの少なくとも一部をコード化し得る。以下で詳細に説明する図9は、WPPの一例を示す概念図である。ビデオコーダがWPPを使用してピクチャをコード化する場合、ビデオコーダは、ピクチャのCTBを複数の「WPP波」に分割し得る。WPP波の各々は、ピクチャ中のCTBの異なる行に対応し得る。ビデオコーダがWPPを使用してピクチャをコード化する場合、ビデオコーダは、CTBの最上行をコード化し始め得る。ビデオコーダが最上行の2つ以上のCTBをコード化した後に、ビデオコーダは、CTBの最上行をコード化することと並列にCTBの最上行から2番目の行をコード化し始め得る。ビデオコーダが最上行から2番目の行の2つ以上のCTBをコード化した後に、ビデオコーダは、CTBの上位行をコード化することと並列にCTBの最上行から3番目の行をコード化し始め得る。このパターンは、ピクチャ中のCTBの行を下って続き得る。

【0058】

[0069]ビデオコーダがWPPを使用している場合、ビデオコーダは、空間的に隣接するCUが現在CTBの左、左上、上、又は右上にある限り、現在CTB中の特定のCUに対してピクチャ内予測を実行するために現在CTB外の空間的に隣接するCUに関連する情報を使用し得る。現在CTBが一番上の行以外の行中の最左CTBである場合、ビデオコーダは、現在CTBの1つ又は複数のシンタックス要素をCABACコード化するためのコンテキストを選択するためにすぐ上の行の第2のCTBに関連する情報を使用し得る。そうではなく、現在CTBが上記行中の最左CTBでない場合、ビデオコーダは、現在CTBの1つ又は複数のシンタックス要素をCABACコード化するためのコンテキストを選択するために現在CTBの左側のCTBに関連する情報を使用し得る。このようにして、ビデオコーダは、すぐ上の行の2つ以上のCTBを符号化した後に、すぐ上の行のCABAC状態に基づいて、ある行のCABAC状態を初期化し得る。

【 0 0 5 9 】

[0070]従って、第1のCTBが単一のCTBによってピクチャの左境界から分離されると決定したことに応答して、ビデオコードは、第1のCTBに関連するコンテキスト変数を記憶し得る。ビデオコードは、第1のCTBに関連するコンテキスト変数に少なくとも部分的に基づいて、第2のCTBの1つ又は複数のシンタックス要素をエントロピーコード化（例えば、エントロピー符号化又はエントロピー復号）し得、第2のCTBは、ピクチャの左境界に隣接し、CTBの1行だけ第1のCTBよりも低い。

【 0 0 6 0 】

[0071]スライスのコード化CTBは通常、WPPが使用されるときでも、ラスト走査順序に従ってコード化スライスNAL単位に構成される。これは、WPPを実装するビデオコードの設計を複雑にし得る。WPP波の数が1よりも大きく、ピクチャのCTB行の数よりも小さいとき、CTBのためのコード化ビットのビットストリーム順序（即ち、コード化ピクチャが1つのデコードコアによって処理され、並列に復号されない場合の復号順序）は、以下のようにWPPが適用されないときと比較して変更される。ビットストリーム/復号順序でより後のコード化CTBは、ピクチャ内予測のために、復号順序でより前の別のコード化CTBによって必要とされ得る。これは、ビットストリーム/復号順序でより前のデータがより後に来るデータに依存しない、ビットストリーム因果律を破壊し得る。ビットストリーム因果律は、ビデオコード化規格を含むビデオコード化設計において概して従われている原理である。復号プロセスが動作している間、ビットストリーム中の現在位置を示すビットストリームポインタが、コード化スライスNAL単位に関連するビットストリームの部分内で前後に移動し得ると、復号プロセスはより複雑になり得る。

【 0 0 6 1 】

[0072]幾つかの例では、ビデオエンコーダ20は、ピクチャを1つ又は複数のタイルに分割し得る。タイルは、ピクチャのCTBの重複しないセットを備え得る。ビデオエンコーダ20は、2つ以上の垂直タイル境界と2つ以上の水平タイル境界とを画定することによってピクチャをタイルに分割し得る。ピクチャの各垂直辺が垂直タイル境界であり得る。現在ピクチャの各水平辺が水平タイル境界であり得る。例えば、ビデオエンコーダ20がピクチャのために4つの垂直タイル境界と3つの水平タイル境界とを画定する場合、現在ピクチャは6つのタイルに分割される。

【 0 0 6 2 】

[0073]ビデオエンコーダ20又はビデオデコーダ30などのビデオコードは、タイル走査順序に従ってピクチャのタイルのCTBをコード化し得る。タイル走査順序に従ってCTBをコード化するために、ビデオコードは、ラスト走査順序に従ってピクチャのタイルをコード化し得る。即ち、ビデオコードは、タイルの最上行から開始し、次いでピクチャを下に進んで、左から右への順序でタイルの行中の各タイルをコード化し得る。更に、ビデオコードは、ラスト走査順序に従ってタイル内の各CTBをコード化し得る。このようにして、ビデオコードは、ピクチャの別のタイルのいずれかのCTBをコード化する前に、ピクチャの所与のタイルの各CTBをコード化し得る。言い換えれば、タイル走査順序は、タイル内でCTBラスト走査順序でCTBを横断し、ピクチャ内でタイルラスト走査順序でタイルを横断する。従って、ピクチャが複数のタイルに区分される場合、ビデオコードがピクチャのCTBをコード化する順序は、ピクチャが複数のタイルに区分されない場合とは異なり得る。以下で説明する図10は、ピクチャが複数のタイルに区分されときの例示的なタイル走査順序を示す概念図である。

【 0 0 6 3 】

[0074]幾つかの事例では、ビデオコードは、スライス境界を越えないが、タイル境界を越えてピクチャ内予測を実行し得る。他の事例では、タイル境界とスライス境界とを越えるピクチャ内予測が禁止される。タイル境界とスライス境界とを越えるピクチャ内予測が禁止される事例では、ビデオコードは、複数のタイルを並列にコード化することが可能であり得る。

【 0 0 6 4 】

【0075】幾つかの例では、タイル境界を越えるピクチャ内予測はフラグ（例えば、「tile_boundary_independence_idc」）によって制御される。フラグが1に等しい場合、タイル境界を越えるピクチャ内予測はピクチャ内で可能にされない。そうでない場合、タイル境界を越えるピクチャ内予測は、ピクチャ境界又はスライス境界でもあるタイル境界を除いて、可能にされる。タイル境界を越えるピクチャ内予測が可能にされた場合、タイルの機能は、ピクチャがタイルを有しないか又は等価的にただ1つのタイルを有する場合と比較して、CTBの走査順序を変更することであり得る。タイル境界を越えるピクチャ内予測が可能にされない場合、CTBの走査順序を変更することのほかに、タイルは、タイルの並列コード化（符号化及び/又は復号）のために使用され得る独立した区分をも行い得る。従って、ピクチャが少なくとも第1のタイルと第2のタイルとに区分された場合、ビデオデコーダ30がWPPを使用せずにタイルを復号するとき、ビデオデコーダ30は、第1のタイルのCTBと第2のタイルのCTBとを並列に復号し得る。

10

【0065】

【0076】幾つかの事例では、ピクチャは、タイルと、WPP波と、エントロピースライスとの組合せに区分され得る。例えば、ピクチャは、タイルと、WPP波のセットとに区分され得る。別の例では、ピクチャは、2つのタイルと、エントロピースライスとに区分され得る。ピクチャ内でタイルとWPP波とエントロピースライスとの組合せを可能にすることは、ビデオエンコーダ及び/又はビデオデコーダの複雑さ及びコストを増加させ得るので、そのような組合せを可能にすることは問題になり得る。

20

【0066】

【0077】本開示の技法は、上記で説明した問題を解決し得る。本開示の技法によれば、ピクチャは、タイルとWPP波とエントロピースライスとのいかなる組合せにも区分されないことがある。言い換えれば、ピクチャは1つ又は複数のタイルに区分されるか、ピクチャはWPP波に区分されるか、若しくはピクチャは1つ又は複数のエントロピースライスに区分され得る。しかしながら、ピクチャは、以下の組合せ、即ち、(a)タイル、WPP波及びエントロピースライス、(b)タイル及びWPP波、(c)タイル及びエントロピースライス、又は(d)WPP波及びエントロピースライスのいずれにも区分されないことがある。

【0067】

30

【0078】これを達成するために、ビデオエンコーダ20は、ピクチャが第1のコード化モードに従って符号化されるか第2のコード化モードに従って符号化されるかを示すシンタックス要素をビットストリーム中に含み得る。第1のコード化モードでは、ピクチャは、WPPを使用して完全に符号化される。即ち、ピクチャ中のCTBの各行はWPP波として符号化され得る。第2のコード化モードでは、ピクチャは1つ又は複数のタイルを有し得る。更に、第2のコード化モードでは、ピクチャの各タイルは、WPPを使用せずに符号化され得る。例えば、第2のコード化モードでは、ビデオエンコーダ20は、ピクチャのタイルごとに、タイルのCTBの行を越えて、CTBの行に沿って、左から右への順序で連続的にタイル内のCTBを符号化し得る。説明を簡単にするために、このシンタックス要素は、本明細書ではコード化モードシンタックス要素と呼ぶことがある。

40

【0068】

【0079】ビデオデコーダ30は、ビデオデータ中にピクチャのコード化表現を含むビットストリームからシンタックス要素を構文解析し得る。シンタックス要素が特定の値を有すると決定したことに応答して、ビデオデコーダ30は、WPPを使用してピクチャを完全に復号し得る。シンタックス要素が特定の値を有しないと決定したことに応答して、ビデオデコーダ30は、WPPを使用せずにピクチャの各タイルを復号し得、ピクチャは1つ又は複数のタイルを有する。

【0069】

【0080】ビットストリームの様々な部分がコード化モードシンタックス要素を含み得る。例えば、ビデオエンコーダ20は、コード化モードシンタックス要素を含むSPSを生成

50

し得る。この例では、ビデオデコーダ30は、ビットストリームから、コード化モードシンタックス要素を含むSPSを構文解析し得る。別の例では、ビデオエンコーダ20は、コード化モードシンタックス要素を含むPPSを生成し得る。この例では、ビデオデコーダ30は、ビットストリームから、コード化モードシンタックス要素を含むPPSを構文解析し得る。更に、ピクチャが第2のコード化モードに従って符号化された場合、ビットストリームは、ピクチャのためにエントロピースライスが有効化されるかどうかを示す1つ又は複数のシンタックス要素を含み得る。ビットストリームの様々な部分は、ピクチャのためにエントロピースライスが有効化されるかどうかを示す1つ又は複数のシンタックス要素を含み得る。例えば、SPSは、SPSに関連するピクチャのためにエントロピースライスが有効化されることを示す1つ又は複数のシンタックス要素を含み得る。別の例では、PPSは、PPSに関連するピクチャのためにエントロピースライスが有効化されることを示す1つ又は複数のシンタックス要素を含み得る。例えば、この例では、PPSは、そのPPSを参照するコード化スライスがエントロピースライスからなり得るか否かを示す`entropy_slice_enabled_flag`シンタックス要素を含み得る。

10

【0070】

[0081]ピクチャが1つ又は複数のエントロピースライスを含む場合、ピクチャのスライスに関連する各エントロピースライスは、別々のNAL単位中に含まれるのではなく、単一のコード化スライスNAL単位中に含まれ得る。従って、エントロピースライスはスライスのサブセットとして定義され得、エントロピースライスのエントロピー復号プロセスは、同じスライス中の他のエントロピースライスとは無関係である。

20

【0071】

[0082]上記で手短かに説明したように、ビットストリームは、コード化スライスを含むコード化スライスNAL単位を含み得る。コード化スライスはスライスヘッダとスライスデータとを備え得る。スライスデータは1つ又は複数のサブストリームを含み得る。本開示の技法によれば、ピクチャが第1のコード化モードで符号化される（即ち、WPPを使用してピクチャが完全に符号化される）場合、スライスのCTBの各行は、サブストリームのうちのただ1つによって表される。ピクチャが第2のコード化モードで符号化される（即ち、WPPを使用せずにピクチャの各タイルが符号化される）場合、スライス中に1つ又は複数のCTBを有するピクチャの各タイルは、サブストリームのうちのただ1つによって表される。

30

【0072】

[0083]更に、本開示の技法によれば、コード化スライスのスライスヘッダは、コード化スライスNAL単位のスライスデータ内のタイル、WPP波、又はエントロピースライスのエントリポイントを示すシンタックス要素のセットを含み得る。サブストリームのエントリポイントはサブストリームの第1のビットであり得る。更に、コード化スライスNAL単位のスライスデータ内のタイル、WPP波、又はエントロピースライスは、タイル、WPP波、又はエントロピースライスがバイト整合されることを保証するパディングビット(padding bits)を含み得る。

40

【0073】

[0084]図2は、本開示の技法を実装するように構成された例示的なビデオエンコーダ20を示すブロック図である。図2は、説明のために与えられており、本開示で広く例示及び説明する技法を限定するものと見なされるべきではない。説明のために、本開示では、HEVCコード化のコンテキストにおいてビデオエンコーダ20について説明する。但し、本開示の技法は、他のコード化規格又は方法に適用可能であり得る。

【0074】

[0085]図2の例では、ビデオエンコーダ20は、予測処理ユニット100と、残差生成ユニット102と、変換処理ユニット104と、量子化ユニット106と、逆量子化ユニット108と、逆変換処理ユニット110と、再構成ユニット112と、フィルタユニット113と、復号ピクチャバッファ114と、エントロピー符号化ユニット116とを含

50

む。予測処理ユニット100は、インター予測処理ユニット121と、イントラ予測処理ユニット126とを含む。インター予測処理ユニット121は、動き推定ユニット122と、動き補償ユニット124とを含む。他の例では、ビデオエンコーダ20は、より多数の、より少数の、又は異なる機能構成要素を含み得る。

【0075】

[0086]ビデオエンコーダ20はビデオデータを受信し得る。ビデオデータを符号化するために、ビデオエンコーダ20は、ビデオデータの各ピクチャの各スライスを符号化し得る。スライスを符号化することの一部として、ビデオエンコーダ20は、スライス中の各CTBを符号化し得る。CTBを符号化することの一部として、予測処理ユニット100は、CTBに関連する画素ブロックに対してクワッドツリー区分化を実行して、画素ブロックを徐々により小さい画素ブロックに分割し得る。より小さい画素ブロックはCUに関連付けられ得る。例えば、予測処理ユニット100は、CTBの画素ブロックを4つの等しいサイズのサブブロックに区分し、サブブロックのうちの1つ又は複数を、4つの等しいサイズのサブサブブロックに区分し得、以下同様である。

【0076】

[0087]ビデオエンコーダ20は、CTBのCUを符号化して、CUの符号化表現（即ち、コード化CU）を生成し得る。ビデオエンコーダ20は、z走査順序に従ってCTBのCUを符号化し得る。言い換えれば、ビデオエンコーダ20は、その順序で、左上のCUと、右上のCUと、左下のCUと、次いで右下のCUとを符号化し得る。ビデオエンコーダ20が区分されたCUを符号化するとき、ビデオエンコーダ20は、z走査順序に従って、区分されたCUの画素ブロックのサブブロックに関連付けられたCUを符号化し得る。

【0077】

[0088]CUを符号化することの一部として、予測処理ユニット100は、CUの1つ又は複数のPUの中で、CUの画素ブロックを区分し得る。ビデオエンコーダ20及びビデオデコーダ30は、様々なPUサイズをサポートし得る。特定のCUのサイズが $2N \times 2N$ であると仮定すると、ビデオエンコーダ20及びビデオデコーダ30は、イントラ予測のための $2N \times 2N$ 又は $N \times N$ のPUサイズと、インター予測のための $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 、又は同様の対称PUサイズとをサポートし得る。ビデオエンコーダ20及びビデオデコーダ30はまた、インター予測のための $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、及び $nR \times 2N$ のPUサイズに対する非対称区分をサポートし得る。

【0078】

[0089]インター予測処理ユニット121は、CUの各PUに対してインター予測を実行することによってPUの予測データを生成し得る。PUの予測データは、PUに対応する予測画素ブロックと、PUの動き情報とを含み得る。スライスは、Iスライス、Pスライス、又はBスライスであり得る。インター予測ユニット121は、PUがIスライス中にあるか、Pスライス中にあるか、Bスライス中にあるかに応じて、CUのPUに対して異なる演算を実行し得る。Iスライス中では、全てのPUがイントラ予測される。従って、PUがIスライス中にある場合、インター予測ユニット121は、PUに対してインター予測を実行しない。

【0079】

[0090]PUがPスライス中にある場合、動き推定ユニット122は、PUの参照ブロックについて参照ピクチャのリスト（例えば、「リスト0」）中の参照ピクチャを探索し得る。PUの参照ブロックは、PUの画素ブロックに最も密接に対応する画素ブロックであり得る。動き推定ユニット122は、PUの参照ブロックを含んでいるリスト0中の参照ピクチャを示す参照ピクチャインデックス、及びPUの画素ブロックと参照ブロックとの間の空間変位を示す動きベクトルを生成し得る。動き推定ユニット122は、PUの動き情報として、参照ピクチャインデックスと動きベクトルとを出力し得る。動き補償ユニット124は、PUの動き情報によって示される参照ブロックに基づいてPUの予測画素ブロックを生成し得る。

【 0 0 8 0 】

[0091] P U が B スライス中にある場合、動き推定ユニット 1 2 2 は、P U について単方向インター予測又は双方向インター予測を実行し得る。P U について単方向インター予測を実行するために、動き推定ユニット 1 2 2 は、P U の参照ブロックについて第 1 の参照ピクチャリスト（「リスト 0」）又は第 2 の参照ピクチャリスト（「リスト 1」）の参照ピクチャを探索し得る。動き推定ユニット 1 2 2 は、P U の動き情報として、参照ブロックを含んでいる参照ピクチャのリスト 0 又はリスト 1 中の位置を示す参照ピクチャインデックス、P U の画素ブロックと参照ブロックとの間の空間変位を示す動きベクトル、及び参照ピクチャがリスト 0 中にあるかリスト 1 中にあるかを示す予測方向インジケータを出力し得る。

10

【 0 0 8 1 】

[0092] P U について双方向インター予測を実行するために、動き推定ユニット 1 2 2 は、P U の参照ブロックについてリスト 0 中の参照ピクチャを探索し得、また、P U の別の参照ブロックについてリスト 1 中の参照ピクチャを探索し得る。動き推定ユニット 1 2 2 は、参照ブロックを含んでいる参照ピクチャのリスト 0 及びリスト 1 中の位置を示す参照ピクチャインデックスを生成し得る。更に、動き推定ユニット 1 2 2 は、参照ブロックと P U の画素ブロックとの間の空間変位を示す動きベクトルを生成し得る。P U の動き情報は、P U の参照ピクチャインデックスと動きベクトルとを含み得る。動き補償ユニット 1 2 4 は、P U の動き情報によって示される参照ブロックに基づいて P U の予測画素ブロックを生成し得る。

20

【 0 0 8 2 】

[0093] イントラ予測処理ユニット 1 2 6 は、P U に対してイントラ予測を実行することによって P U の予測データを生成し得る。P U の予測データは、P U の予測画素ブロックと、様々なシンタックス要素とを含み得る。イントラ予測処理ユニット 1 2 6 は、I スライス、P スライス、及び B スライス中の P U に対してイントラ予測を実行し得る。

【 0 0 8 3 】

[0094] P U に対してイントラ予測を実行するために、イントラ予測処理ユニット 1 2 6 は、複数のイントラ予測モードを使用して、P U のための予測データの複数のセットを生成し得る。イントラ予測モードを使用して P U の予測データのセットを生成するために、イントラ予測処理ユニット 1 2 6 は、イントラ予測モードに関連する方向に隣接 P U のサンプルブロックから P U のサンプルブロックを越えてサンプルを拡張し得る。隣接 P U は、P U、C U、及び C T B について左から右、上から下の符号化順序を仮定すると、P U の上、右上、左上、又は左にあり得る。イントラ予測処理ユニット 1 2 6 は、様々な数のイントラ予測モード、例えば、3 3 個の方向性イントラ予測モードを使用し得る。幾つかの例では、イントラ予測モードの数は、P U の画素ブロックのサイズに依存し得る。

30

【 0 0 8 4 】

[0095] 予測処理ユニット 1 0 0 は、P U についてインター予測処理ユニット 1 2 1 によって生成された予測データ、又は P U についてイントラ予測処理ユニット 1 2 6 によって生成された予測データの中から、C U の P U のための予測データを選択し得る。幾つかの例では、予測処理ユニット 1 0 0 は、予測データのセットのレート／歪みメトリックに基づいて C U の P U のための予測データを選択する。選択された予測データの予測画素ブロックは、本明細書では、選択された予測画素ブロックと呼ぶことがある。

40

【 0 0 8 5 】

[0096] 残差生成ユニット 1 0 2 は、C U の画素ブロックと、C U の P U の選択された予測画素ブロックとに基づいて、C U の残差画素ブロックを生成し得る。例えば、残差生成ユニット 1 0 2 は、残差画素ブロック中の各サンプルが、C U の画素ブロック中のサンプルと、C U の P U の選択された予測画素ブロック中の対応するサンプルとの間の差分に等しい値を有するように、C U の残差画素ブロックを生成し得る。

【 0 0 8 6 】

[0097] 予測処理ユニット 1 0 0 は、クワッドツリー区分を実行して、C U の残差画素ブ

50

ロックをサブブロックに区分し得る。分割されていない各残差画素ブロックは、C Uの異なるT Uに関連付けられ得る。C UのT Uに関連付けられた残差画素ブロックのサイズ及び位置は、C UのP Uの画素ブロックのサイズ及び位置に基づくことも、基づかないこともある。

【 0 0 8 7 】

[0098] T Uの残差画素ブロックの画素は、ルーマサンプルと2つのクロマサンプルとを備え得るので、T Uの各々は、ルーマサンプルのブロックと、クロマサンプルの2つのブロックとに関連付けられ得る。変換処理ユニット104は、T Uに関連付けられた残差サンプルブロックに1つ又は複数の変換を適用することによって、C UのT Uごとに係数ブロックを生成し得る。変換処理ユニット104は、T Uに関連付けられた残差サンプルブロックに様々な変換を適用し得る。例えば、変換処理ユニット104は、離散コサイン変換(DCT)、方向性変換、又は概念的に同様の変換を残差サンプルブロックに適用し得る。

10

【 0 0 8 8 】

[0099] 量子化ユニット106は、T Uに関連付けられた係数ブロック中の係数を量子化し得る。量子化プロセスは、係数の一部又は全部に関連するビット深度を低減し得る。例えば、量子化中にnビット係数がmビット係数に切り捨てられ得、但し、nはmよりも大きい。量子化ユニット106は、C Uに関連付けられた量子化パラメータ(QP)値に基づいて、C UのT Uに関連付けられた係数ブロックを量子化し得る。ビデオエンコーダ20は、C Uに関連付けられたQP値を調整することによって、C Uに関連付けられた係数ブロックに適用される量子化の程度を調整し得る。

20

【 0 0 8 9 】

[0100] 逆量子化ユニット108及び逆変換処理ユニット110は、それぞれ、係数ブロックに逆量子化及び逆変換を適用して、係数ブロックから残差サンプルブロックを再構成し得る。再構成ユニット112は、再構成された残差サンプルブロックを、予測処理ユニット100によって生成された1つ又は複数の予測サンプルブロックからの対応するサンプルに加算して、T Uに関連する再構成されたサンプルブロックを生成し得る。このようにC UのT Uごとにサンプルブロックを再構成することによって、ビデオエンコーダ20は、C Uの画素ブロックを再構成し得る。

【 0 0 9 0 】

30

[0101] フィルタユニット113は、デブロッキング動作を実行して、C Uに関連付けられた画素ブロック中のブロック歪みを低減し得る。復号ピクチャバッファ114は、フィルタユニット113が、再構成された画素ブロックに対して1つ又は複数のデブロッキング動作を実行した後に、再構成された画素ブロックを記憶し得る。インター予測ユニット121は、再構成された画素ブロックを含んでいる参照ピクチャを使用して、他のピクチャのP Uに対してインター予測を実行し得る。更に、イントラ予測処理ユニット126は、復号ピクチャバッファ114中の再構成された画素ブロックを使用して、C Uと同じピクチャの中の他のP Uに対してイントラ予測を実行し得る。

【 0 0 9 1 】

[0102] エントロピー符号化ユニット116は、ビデオエンコーダ20の他の機能構成要素からデータを受信し得る。例えば、エントロピー符号化ユニット116は、量子化ユニット106から係数ブロックを受信し得、予測処理ユニット100からシンタックス要素を受信し得る。エントロピー符号化ユニット116は、データに対して1つ又は複数のエントロピー符号化演算を実行してエントロピー符号化データを生成し得る。例えば、エントロピー符号化ユニット116は、コンテキスト適応型可変長コード化(CAVLC: context-adaptive variable length coding)演算、CABAC演算、変数-変数(V2V: variable-to-variable)レングスコード化演算、シンタックスベースコンテキスト適応型バイナリ算術コード化(SBAC: syntax-based context-adaptive binary arithmetic coding)演算、確率間隔区分エントロピー(PIPE: Probability Interval Partitioning Entropy)コード化演算、指数ゴロム符号化演算、又は別のタイプのエントロピー

40

50

符号化演算をデータに対して実行し得る。

【 0 0 9 2 】

[0103]ビデオエンコーダ20は、エントロピー符号化ユニット116によって生成されたエントロピー符号化データを含むビットストリームを出力し得る。ビットストリームは、一連のNAL単位を含み得る。NAL単位は、コード化スライスNAL単位、SPS NAL単位、PPS NAL単位などを含み得る。ピクチャがタイルとWPP波とエントロピースライスとの組合せを含まないことを保証するために、ビットストリームは、WPPを使用してピクチャが完全に符号化されるかどうか、又はWPPを使用せずにピクチャの各タイルが符号化されるかどうかを示すシンタックス要素を含み得る。

【 0 0 9 3 】

[0104]図3は、本開示の技法を実装するように構成された例示的なビデオデコーダ30を示すブロック図である。図3は、説明のために与えられており、本開示で広く例示及び説明する技法に対する限定ではない。説明のために、本開示では、HEVCコード化のコンテキストにおいてビデオデコーダ30について説明する。但し、本開示の技法は、他のコード化規格又は方法に適用可能であり得る。

【 0 0 9 4 】

[0105]図3の例では、ビデオデコーダ30は、エントロピー復号ユニット150と、予測処理ユニット152と、逆量子化ユニット154と、逆変換処理ユニット156と、再構成ユニット158と、フィルタユニット159と、復号ピクチャバッファ160とを含む。予測処理ユニット152は、動き補償ユニット162と、イントラ予測処理ユニット164とを含む。他の例では、ビデオデコーダ30は、より多数の、より少数の、又は異なる機能構成要素を含み得る。

【 0 0 9 5 】

[0106]ビデオデコーダ30はビットストリームを受信し得る。エントロピー復号ユニット150は、ビットストリームからシンタックス要素を抽出するためにビットストリームを構文解析し得る。ビットストリームを構文解析することの一部として、エントロピー復号ユニット150は、ビットストリーム中のエントロピー符号化されたシンタックス要素をエントロピー復号し得る。予測処理ユニット152、逆量子化ユニット154、逆変換処理ユニット156、再構成ユニット158、及びフィルタユニット159は、ビットストリームから抽出されたシンタックス要素に基づいて復号ビデオデータを生成し得る。

【 0 0 9 6 】

[0107]ビットストリームは、一連のNAL単位を備え得る。ビットストリームのNAL単位は、コード化スライスNAL単位を含み得る。ビットストリームを構文解析することの一部として、エントロピー復号ユニット150は、コード化スライスNAL単位からシンタックス要素を抽出し、エントロピー復号し得る。コード化スライスの各々は、スライスヘッダとスライスデータとを含み得る。スライスヘッダは、スライスに関するシンタックス要素を含んでいることがある。スライスヘッダ中のシンタックス要素は、スライスを含んでいるピクチャに関連付けられたPPSを識別するシンタックス要素を含み得る。

【 0 0 9 7 】

[0108]更に、ビデオデコーダ30は、区分されていないCUに対して再構成演算を実行し得る。区分されていないCUに対して再構成演算を実行するために、ビデオデコーダ30は、CUの各TUに対して再構成演算を実行し得る。CUの各TUについて再構成演算を実行することによって、ビデオデコーダ30は、CUに関連付けられた残差画素ブロックを再構成し得る。

【 0 0 9 8 】

[0109]CUのTUに対して再構成演算を実行することの一部として、逆量子化ユニット154は、TUに関連付けられた係数ブロックを逆量子化(inverse quantize)、即ち、逆量子化(de-quantize)し得る。逆量子化ユニット154は、量子化の程度を決定し、同様に、逆量子化ユニット154が適用すべき逆量子化の程度を決定するために、TUのCUに関連付けられたQP値を使用し得る。

10

20

30

40

50

【 0 0 9 9 】

[0110]逆量子化ユニット 1 5 4 が係数ブロックを逆量子化した後に、逆変換処理ユニット 1 5 6 は、T Uに関連付けられた残差サンプルブロックを生成するために、係数ブロックに 1 つ又は複数の逆変換を適用し得る。例えば、逆変換処理ユニット 1 5 6 は、逆 D C T、逆整数変換、逆カルーネンレーベ変換 (K L T)、逆回転変換、逆方向性変換、又は別の逆変換を係数ブロックに適用し得る。

【 0 1 0 0 】

[0111]イントラ予測を使用して P U が符号化された場合、イントラ予測処理ユニット 1 6 4 は、P U の予測サンプルブロックを生成するためにイントラ予測を実行し得る。イントラ予測処理ユニット 1 6 4 は、空間的に隣接する P U の画素ブロックに基づいて P U の予測画素ブロックを生成するためにイントラ予測モードを使用し得る。イントラ予測処理ユニット 1 6 4 は、ビットストリームから構文解析された 1 つ又は複数のシンタックス要素に基づいて P U のイントラ予測モードを決定し得る。

10

【 0 1 0 1 】

[0112]動き補償ユニット 1 6 2 は、ビットストリームから抽出されたシンタックス要素に基づいて、第 1 の参照ピクチャリスト (リスト 0) と第 2 の参照ピクチャリスト (リスト 1) とを構成し得る。更に、インター予測を使用して P U が符号化された場合、エンロピー復号ユニット 1 5 0 は、P U の動き情報を抽出し得る。動き補償ユニット 1 6 2 は、P U の動き情報に基づいて、P U のための 1 つ又は複数の参照ブロックを決定し得る。動き補償ユニット 1 6 2 は、P U のための 1 つ又は複数の参照ブロックに基づいて、P U の予測画素ブロックを生成し得る。

20

【 0 1 0 2 】

[0113]再構成ユニット 1 5 8 は、適用可能なとき、C U の T U に関連付けられた残差画素ブロックと、C U の P U の予測画素ブロックとを使用して、即ち、イントラ予測データ又はインター予測データのいずれかを使用して、C U の画素ブロックを再構成し得る。特に、再構成ユニット 1 5 8 は、C U の画素ブロックを再構成するために、予測画素ブロックの対応するサンプルに残差画素ブロックのサンプルを加算し得る。

【 0 1 0 3 】

[0114]フィルタユニット 1 5 9 は、C U の画素ブロックに関連するブロック歪み (blocking artifacts) を低減するためにデブロッキング動作を実行し得る。ビデオデコーダ 3 0 は、復号ピクチャバッファ 1 6 0 に C U の画素ブロックを記憶し得る。復号ピクチャバッファ 1 6 0 は、後続の動き補償、イントラ予測、及び図 1 の表示装置 3 2 などの表示装置上での提示のために、参照ピクチャを与え得る。例えば、ビデオデコーダ 3 0 は、復号ピクチャバッファ 1 6 0 中の画素ブロックに基づいて、他の C U の P U に対してイントラ予測演算又はインター予測演算を実行し得る。

30

【 0 1 0 4 】

[0115]上述のように、ビデオデコーダ 3 0 は、コード化モードシンタックス要素を含むビットストリームを受信し得る。コード化モードシンタックス要素が特定の値を有する場合、コード化モードシンタックス要素は、W P P を使用してピクチャが完全に符号化されることを示す。様々な例では、コード化モードシンタックス要素は、ビットストリームの様々な部分の中にあり得る。例えば、S P S がコード化モードシンタックス要素を含み得る。以下の表 1 に、コード化モードシンタックス要素 (「 t i l e _ m o d e 」) を含む S P S のための例示的なシンタックスを与える。

40

【表 1】

表1ーシーケンスパラメータセットRBSPシンタックス

seq_parameter_set_rbsp() {	記述子
...	
inter_4x4_enabled_flag	u(1)
tile_mode // 0: 1つのピクチャ中にただ1つのタイル、1: 均一離間、2: 不均一離間、3: WPP	u(2)
if(tile_mode == 1 tile_mode == 2) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
}	
if(tile_mode == 2) {	
for (i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for (i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
if(tile_mode == 1 tile_mode == 2) {	
tile_boundary_independence_flag	u(1)
if(tile_boundary_independence_flag)	
loop_filter_across_tile_flag	u(1)
}	
rbps_trailing_bits()	
}	

【 0 1 0 5 】

[0116] タイプ記述子 $ue(v)$ をもつシンタックス要素は、最初に左ビットとともに指数ゴロムコード化(exponential-Golomb coding)を使用して符号化される符号なし可変長値である。タイプ記述子 $u(1)$ 及び $u(2)$ をもつシンタックス要素は、それぞれ長さが1ビット又は2ビットである符号なし値である。表1の例示的なシンタックスでは、**inter_4x4_enabled_flag** シンタックス要素は、インター予測が、4x4ルーマサンプルのサイズを有するブロックに適用され得るかどうかを指定する。

【 0 1 0 6 】

[0117] 更に、表1の例示的なシンタックスでは、**tile_mode** シンタックス要素は、SPSに関連付けられたピクチャのタイルモードを指定する。**tile_mode** シンタックス要素が0に等しい場合、SPSに関連付けられたピクチャの各々中にただ1つのタイルがある。各ピクチャの単一のタイル中のCTBは、WPPを使用せずにラスタ走査順序に従ってコード化される。**tile_mode** シンタックス要素が1に等しい場合、SPSに関連付けられたピクチャは均一離間タイルモード(uniformly-spaced tile mode)にある。ピクチャが均一離間タイルモードにあるとき、タイル列境界及びタイル行境界は、SPSに関連付けられた各ピクチャ中で均一に分布する。その結果、ピクチャが均一離間タイルモードにあるとき、ピクチャのタイルは同じサイズを有する。均一に分布したタイルの各々内のCTBは、WPPを使用せずにラスタ走査順序に従って符号化され得る。**tile_mode** シンタックス要素が2に等しい場合、SPSに関連付けられたピクチャは不均一離間タイルモード(non-uniformly-spaced tile mode)にある。ピクチャが不均一離間タイルモードにあるとき、タイル列境界及びタイル行境界は、ピクチャにわたって均一に分布しないが、SPSの**column_width[i]** 及び **row_height[i]** シンタックス要素を使用して明示的に信号伝達され得る。不均一に離間したタ

イルの各々内のCTBは、WPPを使用せずにラスト走査順序に従って符号化され得る。

【0107】

[0118] `tile_mode` シンタックス要素が3に等しい場合、SPSに関連付けられたピクチャは、WPPモードを使用してコード化される。言い換えれば、`tile_mode` シンタックス要素が特定の値（例えば、3）を有する場合、SPSに関連付けられたピクチャは、WPPを使用して完全に符号化される。`tile_mode` シンタックス要素が3以外の値を有する場合、SPSに関連付けられたどのピクチャのタイルも、WPPを使用して符号化されない。更に、WPPを使用してピクチャがコード化されるとき、ピクチャのCTBの行のうちの2つのCTBを復号した後に特定の記憶プロセスが呼び出される。更に、ピクチャのCTBの行のうちの第1のCTBを復号する前に特定の同期プロセスが呼び出される。更に、行の最右CTBがコード化されたとき、内部変数の特定のCABAC状態再初期化プロセスが呼び出される。

【0108】

[0119] 上記の特定の記憶プロセスでは、ビデオコードは、第1のCTBが単一のCTBによってピクチャの左境界から分離されると決定したことに応答して、第1のCTBに関連する特定のコンテキスト変数を記憶し得る。特定の同期プロセスでは、ビデオコードは、第1のCTBに関連するコンテキスト変数に少なくとも部分的に基づいて、第2のCTBの1つ又は複数のシンタックス要素をエントロピーコード化（即ち、エントロピー符号化又はエントロピー復号）し、第2のCTBは、ピクチャの左境界に隣接して位置し、CTBの1行だけ第1のCTBよりも低く位置し得る。

【0109】

[0120] 更に、表1の例示的なシンタックスでは、`num_tile_columns_minus1` シンタックス要素は、SPSに関連付けられたピクチャの各々を区分するタイル列の数を指定する。`tile_mode` シンタックス要素が0又は3に等しいとき、`num_tile_columns_minus1` シンタックス要素の値は0に等しいと推論され得る。この理由は、`tile_mode` シンタックス要素が0に等しいとき、ピクチャ中に単一のタイルのみがあり、`tile_mode` シンタックス要素が3に等しいとき、ピクチャの各CTB行は単一のタイルであるからである。`num_tile_rows_minus1` シンタックス要素は、SPSに関連付けられたピクチャの各々を区分するタイル行の数を指定する。`tile_mode` シンタックス要素が0に等しいとき、`num_tile_rows_minus1` シンタックス要素の値は0に等しいと推論され得る。`tile_mode` シンタックス要素が3に等しいとき、ビデオデコード30は、`num_tile_rows_minus1` シンタックス要素の値がCTB中のピクチャの高さ-1に等しいと自動的に決定（即ち、推論）し得る。更に、`tile_mode` シンタックス要素が1又は2に等しいとき、`num_tile_columns_minus1` シンタックス要素及び`num_tile_rows_minus1` シンタックス要素のうちの少なくとも1つは0よりも大きい。

【0110】

[0121] ビデオデコード30は、`column_width[i]` シンタックス要素及び`row_height[i]` シンタックス要素に基づいて、SPSに関連付けられたピクチャのタイルの幅及び高さを決定し得る。`column_width[i]` シンタックス要素は、SPSに関連付けられたピクチャのタイル列の幅を示す。ビデオデコード30は、`column_width[i]` シンタックス要素に少なくとも部分的に基づいて、SPSに関連付けられたピクチャ中のタイルの列の幅を示す`columnWidth` ベクトルを生成し得る。ビデオデコード30は、SPSの`column_width[i]` シンタックス要素から`columnWidth` ベクトルを生成するために以下の擬似コードを使用し得る。

【数 1】

```

for ( i = 0; i <= num_tile_columns_minus1; i++ ) {
    if ( tile_mode != 2 )
        columnWidth[i] =
            ( ( i+1 ) * PicWidthInLCUs ) /
            ( num_tile_columns_minus1 + 1 ) -
            ( i * PicWidthInLCUs ) / ( num_tile_columns_minus1 + 1 )
    else
        columnWidth[i] = column_width[i]
}

```

10

【 0 1 1 1 】

[0122] ビデオデコーダ 30 は、SPS に関連付けられたピクチャ中のタイルの高さを示す `rowHeight` ベクトルを生成し得る。幾つかの例では、ビデオデコーダ 30 は、`rowHeight` ベクトルを生成するために以下の擬似コードを使用し得る。

【数 2】

20

```

for ( i = 0; i <= num_tile_rows_minus1; i++ ) {
    if ( tile_mode != 2 )
        rowHeight[i] =
            ( ( i+1 ) * PicHeightInLCUs ) /
            ( num_tile_rows_minus1 + 1 ) -
            ( i * PicHeightInLCUs ) / ( num_tile_rows_minus1 + 1 )
    else
        rowHeight[i] = row_width[i]
}

```

30

【 0 1 1 2 】

[0123] 更に、ビデオデコーダ 30 は、タイルの各列の最左列境界の SPS に関連付けられたピクチャ内のロケーションを示す `colBd` ベクトルを生成し得る。幾つかの例では、ビデオデコーダ 30 は、以下の擬似コードを使用して `colBd` ベクトルを決定し得る。

【数 3】

```

colBd[0] = 0
for ( i = 0; i <= num_tile_columns_minus1; i++ )
    colBd[i+1] = colBd[i] + columnWidth[i]

```

40

【 0 1 1 3 】

[0124] ビデオデコーダ 30 は、タイルの各行の最上行境界の SPS に関連付けられたピクチャ内のロケーションを示す `rowBd` ベクトルを生成し得る。幾つかの例では、ビデオデコーダ 30 は、以下の擬似コードを使用して `rowBd` ベクトルを決定し得る。

【数 4】

```

rowBd[0] = 0
for ( i = 0; i <= num_tile_rows_minus1; i++ )
    rowBd[i+1] = rowBd[i] + rowHeight[i]

```

【 0 1 1 4 】

[0125] 表 1 の例示的なシンタックスでは、`tile_boundary_independence_flag` シンタックス要素は、タイルが単独で復号可能かどうかを示す。例えば、`tile_boundary_independence_flag` が 1 に等しい場合、タイルは単独で復号可能である。例えば、`tile_boundary_independence_flag` が 1 に等しく、ビデオデコーダ 30 が特定の CTB を復号している場合、特定の CTB と同じタイル内にはない特定の CTB に隣接する全ての CTB は、ピクチャ内予測のために利用不可能であると決定される。更に、`tile_boundary_independence_flag` が 1 に等しい場合、ビデオデコーダ 30 は、タイル中の第 1 の CTB をエン트로ピー復号するより前に、エン트로ピーコード化コンテキストを再初期化する。

10

【 0 1 1 5 】

[0126] `tile_boundary_independence_flag` シンタックス要素が 0 に等しい場合、ピクチャ内予測のための CTB の利用可能性は、タイル境界によって影響を受けない。言い換えれば、`tile_boundary_independent_flag` シンタックス要素が 0 に等しい場合、ビデオデコーダ 30 は、タイル境界を越えてピクチャ内予測を実行し得る。更に、`tile_boundary_independence_flag` シンタックス要素が 0 に等しい場合、エン트로ピー復号ユニット 150 は、ピクチャ中の第 1 のツリーブロックを除いて、タイル中の第 1 の CTB を復号するときに同期プロセスを呼び出し得る。この同期プロセスでは、エン트로ピー復号ユニット 150 は、タイル中の第 1 の CTB の 1 つ又は複数のシンタックス要素をエン트로ピー復号するためのコード化コンテキストを選択するために、前のタイルの最後の CTB に関連する情報を使用し得る。更に、エン트로ピー復号ユニット 150 は、タイル中の第 2 の CTB 行の第 1 の CTB を復号するときに記憶プロセスを実行し得る。記憶プロセスは、CTB の次のより低い行の最左 CTB の 1 つ又は複数のシンタックス要素を CABAC コード化するためのコンテキストを選択する際に使用するコンテキスト変数を記憶し得る。

20

30

【 0 1 1 6 】

[0127] `tile_mode` シンタックス要素が 0 に等しい（即ち、ピクチャごとにただ 1 つのタイルがある）場合、SPS は、表 1 の例示的なシンタックスでは、`tile_boundary_independence_flag` シンタックス要素を含まない。しかしながら、`tile_mode` シンタックス要素が 0 に等しい場合、ビデオデコーダ 30 は、`tile_boundary_independence_flag` シンタックス要素の値が 1 に等しいと自動的に決定し得る。同様に、`tile_mode` シンタックス要素が 3 に等しい（即ち、WPP を使用してピクチャが完全に符号化される）場合、SPS は、表 1 の例示的なシンタックスでは、`tile_boundary_independence_flag` シンタックス要素を含まない。しかしながら、`tile_mode` シンタックス要素が 3 に等しい場合、ビデオデコーダ 30 は、`tile_boundary_independence_flag` シンタックス要素の値が 0 に等しいと自動的に決定し得る。

40

【 0 1 1 7 】

[0128] 表 1 の例示的なシンタックスでは、`loop_filter_across_t`

50

`tile_flag` シンタックス要素は、ビデオデコーダ 30 がタイル境界を越えてループ内フィルタ処理演算を実行すべきかどうかを指定する。例えば、`loop_filter_across_tile_flag` シンタックス要素が 1 に等しい場合、ビデオデコーダ 30 は、タイル境界を越えてループ内フィルタ処理演算を実行し得る。そうではなく、`loop_filter_across_tile_flag` シンタックス要素が 0 に等しい場合、ビデオデコーダ 30 は、タイル境界を越えてループ内フィルタ処理演算を実行し得ない。例示的なループ内フィルタ処理演算としては、デブロッキングフィルタ、サンプル適応オフセット、及び適応ループフィルタがあり得る。

【0118】

[0129] `tile_mode` シンタックス要素が 0 に等しい（即ち、ピクチャごとにただ 1 つのタイルがある）か又は 3 に等しい（即ち、SPS に関連付けられた各ピクチャが、WPP を使用して完全に符号化される）場合、SPS は、表 1 の例示的なシンタックスでは、`loop_filter_across_tile_flag` シンタックス要素を含まない。しかしながら、`tile_mode` シンタックス要素が 0 に等しい場合、ビデオデコーダ 30 は、`loop_filter_across_tile_flag` シンタックス要素の値が 0 に等しいと自動的に決定し得る。`tile_mode` シンタックス要素が 3 に等しい場合、ビデオデコーダ 30 は、`loop_filter_across_tile_flag` シンタックス要素の値が 1 に等しいと自動的に決定し得る。

【0119】

[0130] コード化モードシンタックス要素を含む SPS を受信することの代替又は追加として、ビデオデコーダ 30 は、コード化モードシンタックス要素を含む PPS を受信し得る。ビデオデコーダ 30 が、同じピクチャに適用される SPS 及び PPS を受信し、それらの SPS と PPS の両方がコード化モードシンタックス要素を含む、幾つかの例では、ビデオデコーダ 30 は、PPS によって指定されたコード化モードシンタックス要素を優先し得る。以下の表 2 に、コード化モードシンタックス要素（「`tile_mode`」）を含む PPS の例示的なシンタックスを提示する。

10

20

【表 2 A】

表2ーピクチャパラメータセットRBSPシンタックス

pic_parameter_set_rbsp() {	記述子	
pic_parameter_set_id	ue(v)	
seq_parameter_set_id	ue(v)	
num_temporal_layer_switching_point_flags	ue(v)	
for(i = 0; i < num_temporal_layer_switching_point_flags; i++)		
temporal_layer_switching_point_flag[i]	u(1)	
num_ref_idx_l0_default_active_minus1	ue(v)	10
num_ref_idx_l1_default_active_minus1	ue(v)	
pic_init_qp_minus26 /*26に対して 26 */	se(v)	
constrained_intra_pred_flag	u(1)	
slice_granularity	u(2)	
max_cu_qp_delta_depth	ue(v)	
weighted_pred_flag	u(1)	
weighted_bipred_idc	u(2)	
tile_partition_info_present_flag	u(1)	
if(tile_partition_info_present_flag == 1) {		20
tile_mode // 0: 1つのピクチャ中にただ1つのタイル; 1: 均一離間; 2: 不均一離間; 3: WPP	u(2)	
if(tile_mode == 1 tile_mode == 2) {		
num_tile_columns_minus1	ue(v)	
num_tile_rows_minus1	ue(v)	
}		
if(tile_mode == 2) {		
for(i = 0; i < num_tile_columns_minus1; i++)		
column_width[i]	ue(v)	
for(i = 0; i < num_tile_rows_minus1; i++)		30
row_height[i]	ue(v)	
}		

【表 2 B】

}	
if(tile_mode == 1 tile_mode == 2) {	
tile_control_info_present_flag	u(1)
if(tile_control_info_present_flag == 1) {	
tile_boundary_independence_flag	u(1)
if(tile_boundary_independence_flag)	
loop_filter_across_tile_flag	u(1)
}	
}	
if(tile_mode == 0)	
entropy_slice_enabled_flag	u(1)
rbsp_trailing_bits()	
}	

10

【 0 1 2 0 】

[0131]表 2 の例示的なシンタックスでは、`tile_partition_info_present_flag`シンタックス要素が 1 に等しい場合、`tile_mode`シンタックス要素が存在する。更に、`tile_partition_info_present_flag`シンタックス要素が 1 に等しい場合、`num_tile_columns_minus_1`、`num_tile_rows_minus_1`、`column_width[i]`、及び`row_height[i]`シンタックス要素が PPS 中に存在し得る。`tile_mode`シンタックス要素と、`num_tile_columns_minus_1`シンタックス要素と、`num_tile_rows_minus_1`シンタックス要素と、`column_width`シンタックス要素と、`row_height`シンタックス要素とのセマンティクスは、表 1 の例示的な SPS シンタックスに関して上記で説明したものと同じであり得る。`tile_partition_info_present_flag`シンタックス要素が 0 に等しい場合、`tile_mode`、`num_tile_columns_minus_1`、`num_tile_rows_minus_1`、`column_width[i]`、及び`row_height[i]`シンタックス要素は PPS 中に存在しない。

20

30

【 0 1 2 1 】

[0132]このようにして、ビデオデコーダ 30 は、WPP を使用してピクチャの CTB が符号化されないことを示す値を有するコード化モードシンタックス要素（例えば、`tile_mode`）に少なくとも部分的に基づいて、パラメータセット（例えば、SPS 又は PPS）がタイル列数シンタックス要素とタイル行数シンタックス要素とを含むと決定し得る。ビデオデコーダ 30 はまた、タイル列数シンタックス要素に基づいて、タイル列の数を決定し得る。パラメータセットに関連付けられた各ピクチャのタイルの列の数は、タイル列の数に等しくなり得る。ビデオデコーダ 30 はまた、タイル行数シンタックス要素に基づいて、タイル行の数を決定し得る。パラメータセットに関連付けられた各ピクチャのタイルの行の数は、タイル行の数に等しくなり得る。更に、ビデオデコーダ 30 は、パラメータセット（例えば、SPS 又は PPS）が、一連の 1 つ又は複数の列幅シンタックス要素と、一連の 1 つ又は複数のタイル高さシンタックス要素とを含むと決定し得る。更に、ビデオデコーダ 30 は、列幅シンタックス要素に少なくとも部分的に基づいて、パラメータセットに関連付けられた各ピクチャのタイルの列の幅を決定し得る。更に、ビデオデコーダ 30 は、タイル高さシンタックス要素に少なくとも部分的に基づいて、パラメータセットに関連付けられた各ピクチャのタイルの高さを決定し得る。

40

【 0 1 2 2 】

50

[0133]同様に、ビデオエンコーダ20は、タイル列数シンタックス要素とタイル行数シンタックス要素とを含むパラメータセットを生成し得る。パラメータセットは、ピクチャパラメータセット(PPS)又はシーケンスパラメータセット(SPS)であり得る。タイル列の数はタイル列数シンタックス要素に基づいて決定可能であり、パラメータセットに関連付けられた各ピクチャのタイルの列の数はタイル列の数の数に等しい。タイル行の数はタイル行数シンタックス要素に基づいて決定可能であり、パラメータセットに関連付けられた各ピクチャのタイルの行の数はタイル行の数の数に等しい。ビデオエンコーダ20がパラメータセットを生成するとき、ビデオエンコーダ20は、一連の1つ又は複数の列幅シンタックス要素と、一連の1つ又は複数の行高さシンタックス要素とを生成し得る。パラメータセットに関連付けられた各ピクチャのタイルの列の幅は、列幅シンタックス要素に少なくとも部分的に基づいて決定可能であり得る。パラメータセットに関連付けられた各ピクチャのタイルの行の高さは、行高さシンタックス要素に少なくとも部分的に基づいて決定可能であり得る。

10

【0123】

[0134]更に、表2の例示的なシンタックスでは、`tile_control_info_present_flag`シンタックス要素が1に等しい場合、`tile_boundary_independence_flag`及び`loop_filter_across_tile_flag`シンタックス要素がPPS中に存在し得る。`tile_control_info_present_flag`シンタックス要素が0に等しい場合、`tile_boundary_independence_flag`及び`loop_filter_across_tile_flag`シンタックス要素はPPS中に存在しない。

20

【0124】

[0135]表2の例示的なシンタックスでは、`entropy_slice_enabled_flag`が1に等しい場合、PPSを参照するコード化スライス、1つ又は複数のエントロピースライスを含み(及びそれらからなり)得る。`entropy_slice_enabled_flag`シンタックス要素が0に等しい場合、PPSを参照するコード化スライスはエントロピースライスを含んでいない。`entropy_slice_enabled_flag`シンタックス要素が存在しないとき、ビデオデコーダ30は、`entropy_slice_enabled_flag`シンタックス要素が0に等しいと自動的に決定(即ち、推論)し得る。PPSの他のシンタックス要素のためのセマンティクスは、HEVCWD5において定義されているセマンティクスと同じであり得る。

30

【0125】

[0136]表2の例示的なシンタックスでは、`tile_mode`シンタックス要素が0に等しい場合、PPSは、`entropy_slice_enabled_flag`シンタックス要素のみを含む。上記で説明したように、ビデオデコーダ30は、`tile_mode`シンタックス要素に基づいて、ピクチャの各タイルのCTBを復号するためにWPPを使用すべきかどうかを決定し得る。従って、ビデオデコーダ30は、特定の値を有するコード化モードシンタックス要素(例えば、`tile_mode`)に基づいて、コード化モードシンタックス要素と追加のシンタックス要素とを含むパラメータセット(例えば、SPS又はPPS)を参照するピクチャの符号化表現のためにエントロピースライスが有効化されるかどうかを示す追加のシンタックス要素(例えば、`entropy_slice_enabled_flag`)をビットストリームが含むと決定し得る。

40

【0126】

[0137]上記で説明したように、コード化スライスNAL単位はスライスのコード化表現を含み得る。スライスのコード化表現は、スライスデータと、それに後続するスライスヘッダとを含み得る。幾つかの例では、ビデオデコーダ30は、コード化モードシンタックス要素(例えば、`tile_mode`)に少なくとも部分的に基づいて、スライスデータ中のサブストリームのエントリポイントがそれから決定可能である複数のエントリオフセットシンタックス要素をスライスヘッダが含むかどうかを決定し得る。スライスヘッダがエントリオフセットシンタックス要素を含むと決定したことに応答して、ビデオデコーダ

50

30は、複数のエントリオフセットシンタックス要素を使用して、スライスデータ中のサブストリームのエントリポイントを決定し得る。言い換えれば、ビデオデコーダ30は、オフセットシンタックス要素に少なくとも部分的に基づいて、サブストリームのメモリ中の位置を決定し得る。コード化モードシンタックス要素がある値（例えば、3）を有する場合、ピクチャのCTBの各行は、サブストリームのうちのただ1つによって表される。コード化モードシンタックス要素が異なる値（例えば、0、1、又は2）を有する場合、スライス中に1つ又は複数のCTBを有するピクチャの各タイルは、サブストリームのうちのただ1つによって表される。スライスヘッダは、以下の表3の例示的なシンタックスに準拠し得る。

【表3A】

10

表3—スライスヘッダシンタックス

slice_header() {	記述子
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
slice_type	ue(v)
pic_parameter_set_id	ue(v)
if(sample_adaptive_offset_enabled_flag adaptive_loop_filter_enabled_flag)	
aps_id	ue(v)
frame_num	u(v)
if(IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type == 0)	
pic_order_cnt_lsb	u(v)
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification()	
ref_pic_list_combination()	
if(nal_ref_flag)	
dec_ref_pic_marking()	
if(slice_type != I)	
cabac_init_idc	ue(v)
slice_qp_delta	se(v)
disable_deblocking_filter_flag	u(1)

20

30

40

【表 3 B】

if(slice_type == B)	
collocated_from_l0_flag	u(1)
if(adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag) {	
byte_align()	
alf_cu_control_param()	
byte_align()	
}	
if((weighted_pred_flag && slice_type == P) (weighted_bipred_idc == 1 && slice_type == B))	
pred_weight_table()	
if(slice_type == P slice_type == B)	
5_minus_max_num_merge_cand	ue(v)
if(tile_mode != 0 entropy_slice_enabled_flag) {	
num_entry_offsets	ue(v)
if(num_entry_offsets) {	
offset_len_minus8	ue(v)
for(i = 0; i < num_entry_offsets; i++)	
entry_offset[i]	u(v)
}	
}	
if(entropy_slice_enabled_flag && num_entry_offsets)	
for(i = 0; i < num_entry_offsets; i++) {	
entropy_slice_address[i]	u(v)
if(slice_type != I)	
entropy_slice_cabac_init_idc[i]	ue(v)
}	
}	

【 0 1 2 7 】

[0138]表 3 の例示的なシンタックスでは、スライスヘッダシンタックス要素「pic_parameter_set_id」、「frame_num」、「idr_pic_id」、「pic_order_cnt_lsb」、「delta_pic_order_cnt[0]」、及び「delta_pic_order_cnt[1]」の値は、コード化されるピクチャの全てのスライスヘッダ中で同じである。更に、表 3 の例示的なシンタックスでは、first_slice_in_pic_flagシンタックス要素は、スライスが、ピクチャの左上ルーマサンプルをカバーするCUを含むかどうかを示す。first_slice_in_pic_flagシンタックス要素が 1 に等しい場合、ビデオデコーダ 30 は、変数 SliceAddress と変数 LCUAddress の両方を 0 に設定し得、ビデオデコーダ 30 は、ピクチャ中の第 1 の CTB で復号を開始し得る。

【 0 1 2 8 】

[0139]更に、表 3 の例示的なシンタックスでは、slice_addressシンタックス要素は、スライスグラニュラリティ解像度で、スライスが開始するアドレスを指定する。スライスグラニュラリティ解像度は、スライスがそれを用いて定義されるグラニュラリティである。slice_addressシンタックス要素のビット数は $(\text{Ceil}(\text{Log}_2(\text{NumLCUsInPicture})) + \text{SliceGranularity})$ に等しくなり得、但し、「NumLCUsInPicture」はピクチャ中の CTB の数である。

【 0 1 2 9 】

[0140]表3の例示的なシンタックスでは、ビデオデコーダ30は、LCUAddress変数を(slice__address>>SliceGranularity)に設定する。LCUAddress変数は、ラスタ走査順序でスライスのスライスアドレスのLCU部分を示す。ビデオデコーダ30は、GranularityAddress変数を(slice__address-(LCUAddress<<SliceGranularity))に設定する。GranularityAddress変数は、スライスアドレスのサブLCU部分を表す。GranularityAddress変数はz走査順序で表現される。

【 0 1 3 0 】

10

[0141]ビデオデコーダ30は、SliceAddress変数を(LCUAddress<<(log2__diff__max__min__coding__block__size<<1))+ (GranularityAddress<<((log2__diff__max__min__coding__block__size<<1)-SliceGranularity))に設定する。値log2__diff__max__min__coding__block__sizeは、最大CUサイズと最小CUサイズとの間の差分を指定する。ビデオデコーダ30は、スライス開始座標において可能な限り大きいCUを用いてスライスを復号し始め得る。スライス開始座標は、スライスの第1のCUの左上画素の座標であり得る。

【 0 1 3 1 】

20

[0142]更に、表3の例示的なシンタックスでは、cabac__init__idcシンタックスは、コンテキスト変数のための初期化プロセスにおいて使用される初期化テーブルを決定するためのインデックスを指定する。cabac__init__idcシンタックス要素の値は、両端値を含む0~2の範囲内であり得る。

【 0 1 3 2 】

[0143]表3の例示的なシンタックスでは、num__entry__offsetsシンタックス要素は、スライスヘッダ中のentry__offset[i]シンタックス要素の数を指定する。言い換えれば、複数のエントリオフセットシンタックス要素中のエントリオフセットシンタックス要素の数は、num__entry__offsetsシンタックス要素に基づいて決定可能である。num__entry__offsetsシンタックス要素が存在しないとき、ビデオデコーダ30は、num__entry__offsetsシンタックス要素の値が0に等しいと決定し得る。このようにして、ビデオデコーダ30は、num__entry__offsetsシンタックス要素に基づいて、複数のエントリオフセットシンタックス要素中に何個のオフセットシンタックス要素があるかを決定し得る。offset__len__minus8シンタックス要素+8は、entry__offset[i]シンタックス要素のビットでの長さを指定する。言い換えれば、エントリオフセットシンタックス要素の各々のビットでの長さは、offset__len__minus8シンタックス要素に基づいて決定可能である。このようにして、ビデオデコーダ30は、offset__len__minus8シンタックス要素に基づいて、オフセットシンタックス要素のビットでの長さを決定し得る。entry__offset[i]シンタックス要素は、i番目のエントリオフセットをバイトで指定する。

30

40

【 0 1 3 3 】

[0144]ビデオデコーダ30は、複数のオフセットシンタックス要素中に何個のオフセットシンタックス要素があるかと、オフセットシンタックス要素のビットでの長さにと少なくとも部分的に基づいて、ビットストリームからオフセットシンタックス要素を構文解析し得る。コード化スライスNAL単位中のサブストリームの数は、num__entry__offsets+1に等しくなり得る。サブストリームのインデックス値は、両端値を含む0~num__entry__offsetsの範囲内であり得る。コード化スライスNAL単位のサブストリーム0は、コード化スライスNAL単位のスライスデータの、両端値を含むバイト0~entry__offset[0]-1からなり得る。両端値を含む1~

50

`num__entry__offsets - 1`の範囲内に`k`をもつ、コード化スライスNAL単位のサブストリーム`k`は、コード化スライスNAL単位のスライスデータの、両端値を含むバイト`entry__offset[k - 1] ~ entry__offset[k] - 1`からなり得る。`(num__entry__offsets`に等しいサブストリームインデックスをもつ)コード化スライスNAL単位の最後のサブストリームは、コード化スライスNAL単位のスライスデータのバイトの残りからなり得る。

【0134】

[0145]表3の例示的なシンタックスでは、`tile__mode`シンタックス要素が0よりも大きい場合、`1 ~ num__entry__offsets - 1`の範囲内にサブストリームインデックスをもつ各サブストリームは、1つのタイルの各コード化ビットを含んでおり、サブストリームインデックス0をもつサブストリームは、タイルの各コード化ビット、又はタイルの終端コード化ビットの数のいずれかを含んでいる。タイルの終端コード化ビットは、タイルの終端においてコード化されるコード化ビットである。更に、`tile__mode`シンタックス要素が0よりも大きい場合、最後のサブストリーム(即ち、`num__entry__offsets`に等しいサブストリームインデックスをもつサブストリーム)は、タイルの全てのコード化ビット、又はタイルの始端コード化ビットの数のいずれかを含んでいる。タイルの始端コード化ビットは、タイルの始端においてコード化されるコード化ビットである。サブストリームは、2つ以上のタイルのコード化ビットを含んでいない。表3の例示的なシンタックスでは、コード化スライスNAL単位のNAL単位ヘッダとスライスヘッダとは、常にサブストリーム0中に含まれる。`tile__mode`シンタックス要素が0に等しく、`entropy__slice__enabled__flag`シンタックス要素が1に等しい場合、各サブストリームは、1つのエントロピースライスの各コード化ビットを含んでおり、別のエントロピースライスのコード化ビットを含んでいない。

【0135】

[0146]表3の例示的なシンタックスでは、`entropy__slice__address[i]`シンタックス要素は、コード化スライスNAL単位中の $(i + 1)$ 番目のエントロピースライスの開始アドレスをスライスグラニュラリティ解像度で指定する。`entropy__slice__address[i]`シンタックス要素の各々のビットでのサイズは $(\text{Ceil}(\text{Log}_2(\text{NumLCUsInPicture})) + \text{SliceGranularity})$ に等しくなり得る。

【0136】

[0147]更に、表3の例示的なシンタックスでは、「`entropy__slice__cabac__init__idc[i]`」シンタックス要素は、コード化スライスNAL単位中の $(i + 1)$ 番目のエントロピースライスのためのコンテキスト変数の初期化プロセスにおいて使用される初期化テーブルを決定するためのインデックスを指定する。`entropy__slice__cabac__init__idc[i]`の値は、両端値を含む0~2の範囲内である。スライスヘッダの他のシンタックス要素のためのセマンティクスは、HEVC WD5において定義されているセマンティクスと同じであり得る。

【0137】

[0148]幾つかの例では、`entry__offset[i]`シンタックス要素は、サブストリームのオフセットをビットで示す。その上、幾つかの例では、スライスヘッダは、「`entry__offset[i]`」の単位がバイトであるか(1に等しいとき)、ビットであるか(0に等しいとき)を示すフラグを含み得る。このフラグは、`offset__len__minus 8`シンタックス要素の後にスライスヘッダ中に配置され得る。

【0138】

[0149]更に、幾つかの例では、スライスヘッダは、サブストリーム0を含む各サブストリームのサブストリームタイプを示すための、それぞれのサブストリームのシンタックス要素を含み得る。この例では、サブストリームのシンタックス要素が第1の値を有する場合、サブストリームはタイルである。サブストリームのシンタックス要素が第2の値を有

10

20

30

40

50

する場合、サブストリームはエントロピースライスである。

【0139】

[0150] 上述のように、コード化表現はスライスヘッダとスライスデータとを含み得る。スライスデータは1つ又は複数のサブストリームを含み得る。コード化モードシンタックス要素が第1の値（例えば、3）を有する場合、ピクチャのCTBの各行は、サブストリームのうちのただ1つによって表される。シンタックス要素が第2の値（例えば、0、1、又は2）を有する場合、スライス中に1つ又は複数のCTBを有するピクチャの各タイルは、サブストリームのうちのただ1つによって表される。WPP、又はスライスのタイルを並列に復号することを可能にするために、スライスデータ中のサブストリームは、サブストリームのバイト整合を保証するパディングビットを含み得る。但し、ピクチャ中にただ1つのタイルがあり、エントロピースライスが有効化されない事例では、そのようなパディングビットを含む必要がないことがある。従って、ビデオデコーダ30は、コード化モードシンタックス要素（例えば、`tile_mode`）に少なくとも部分的に基づいて、スライスデータ中のサブストリームが、サブストリームのバイト整合を保証するパディングビットを含むかどうかを決定し得る。

【0140】

[0151] スライスデータは、以下の表4の例示的なシンタックスに準拠し得る。

【表4】

表4—スライスデータシンタックス

slice_data() {	記述子
CurrTbAddr = LCUAddress	
moreDataFlag = 1	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlagIdx = -1	
subStreamIdx = 0	
do {	
xCU = HorLumaLocation(CurrTbAddr)	
yCU = VerLumaLocation(CurrTbAddr)	
moreDataFlag = coding_tree(xCU, yCU, Log2TbSize, 0)	
CurrTbAddr = NextTbAddress(CurrTbAddr)	
if(tile_mode != 0 entropy_slice_enabled_flag) {	
byteIdx = byte_index()	
if(byte_aligned() && byteIdx ==	
entry offset[subStreamIdx])	
subStreamIdx++	
else if(!byte_aligned() && byteIdx ==	
entry offset[subStreamIdx] - 1) {	
while(!byte_aligned())	
bit_equal_to_one	f(1)
subStreamIdx++	
}	
moreDataFlag = moreDataFlag && (subStreamIdx ==	
num_entry_offsets)	
}	
} while(moreDataFlag)	
}	

【0141】

[0152] 表4の例示的なシンタックスでは、スライスデータは`coding_tree()`関数を含む。ビデオデコーダ30がスライスデータを構文解析するとき、ビデオデコー

ダ30はループを実行し得る。ループの各反復中に、ビデオデコーダ30は、スライスデータ中のコード化CTBを構文解析するために`coding_tree()`関数を呼び出す。ビデオデコーダ30が特定のコード化CTBを構文解析するために`coding_tree()`関数を呼び出すと、ビデオデコーダ30は、スライスデータから`end_of_slice_flag`シンタックス要素を構文解析し得る。`end_of_slice_flag`シンタックス要素が0に等しい場合、スライス又はエントロピースライス中に特定のコード化CTBに続いて別のCTBがある。`end_of_slice_flag`シンタックス要素が1に等しい場合、特定のコード化CTBは、スライス又はエントロピースライスの最後のコード化CTBである。

【0142】

10

[0153]更に、表4の例示的なシンタックスは`byte_index()`関数を含む。`byte_index()`関数は、NAL単位のビット内の現在位置のバイトインデックスを戻し得る。NAL単位のビット内の現在位置は、NAL単位の第1の構文解析されていないビットであり得る。ビットストリーム中の次のビットがNAL単位ヘッダの第1のバイトのいずれかのビットである場合、`byte_index()`関数は0に等しい値を戻す。

【0143】

[0154]表4のスライスデータシンタックスは一例である。スライスデータシンタックスの他の例では、表4の条件「`if (tile_mode != 0 || entropy_slice_enabled_flag)`」は、条件「`if (tile_mode == 1 || tile_mode == 2 || entropy_slice_enabled_flag)`」と置き換えられる。

20

【0144】

[0155]図4は、本開示の1つ又は複数の態様による、単一のピクチャ内のタイルとWPP波との組合せが許可されない、ビデオデータを符号化するためのビデオエンコーダ20の例示的な動作200を示すフローチャートである。図4は一例として与えられる。他の例では、本開示の技法は、図4の例に示すステップに対して、より多数の、より少数の、又は異なるステップを使用して実装され得る。

【0145】

[0156]図4の例では、ビデオエンコーダ20は、第1のコード化モードに従ってピクチャを符号化することによって第1のコード化ピクチャを生成する(202)。ビデオエンコーダ20が第1のコード化モードに従ってピクチャを符号化するとき、ピクチャは、WPPを使用して完全に符号化される。更に、ビデオエンコーダ20は、第2のコード化モードに従ってピクチャを符号化することによって第2のコード化ピクチャを生成する(204)。ビデオエンコーダ20が第2のコード化モードに従ってピクチャを符号化するとき、ビデオエンコーダ20は、ピクチャを1つ又は複数のタイルに区分し得る。ビデオエンコーダ20は、WPPを使用せずにピクチャの各タイルを符号化し得る(即ち、タイルの各々中の各CTBを符号化し得る)。例えば、ビデオエンコーダ20は、WPPを使用せずにラスタ走査順序に従ってタイルの各々のCTBを符号化し得る。ビデオエンコーダ20は、次いで、第1のコード化ピクチャ又は第2のコード化ピクチャを選択する(206)。幾つかの例では、ビデオエンコーダ20は、第1のコード化ピクチャ及び第2のコード化ピクチャのレート/歪み分析に基づいて第1のコード化ピクチャ又は第2のコード化ピクチャを選択し得る。ビデオエンコーダ20は、選択されたコード化ピクチャと、ピクチャが第1のコード化モードに従って符号化されるか第2のコード化モードに従って符号化されるかを示すシンタックス要素とを含むビットストリームを生成する(208)。

30

40

【0146】

[0157]図5は、本開示の1つ又は複数の態様による、単一のピクチャ内のタイルとWPPとの組合せが許可されない、ビデオデータを復号するためのビデオデコーダ30の例示的な動作220を示すフローチャートである。図5は一例として与えられる。

【0147】

50

[0158]図 5 の例では、ビデオデコーダ 30 は、ビデオデータ中にピクチャのコード化表現を含むビットストリームからシンタックス要素を構文解析する(222)。ビデオデコーダ 30 は、シンタックス要素が特定の値を有するかどうかを決定する(224)。シンタックス要素が特定の値を有すると決定したことに応答して(224の「YES」)、ビデオデコーダ 30 は、WPP を使用してピクチャを完全に復号する(226)。シンタックス要素が特定の値を有しないと決定したことに応答して(224の「NO」)、ビデオデコーダ 30 は、WPP を使用せずにピクチャの各タイルを復号し、ピクチャは 1 つ又は複数のタイルを有する(228)。

【0148】

[0159]図 6 は、本開示の 1 つ又は複数の態様による、単一のピクチャ内のタイルと WPP との組合せが許可されない、ビデオデータを復号するためのビデオデコーダ 30 の例示的な動作 230 を示すフローチャートである。図 6 は一例として与えられる。他の例では、本開示の技法は、図 6 の例に示すステップに対して、より多数の、より少数の、又は異なるステップを使用して実装され得る。図 6 は、図 5 の動作 220 のより具体的な例であり得る。

【0149】

[0160]図 6 の例では、ビデオデコーダ 30 は、ビットストリームを受信する(231)。ビデオデコーダ 30 は、ビットストリームからシンタックス要素を構文解析する(232)。幾つかの例では、ビットストリームは、シンタックス要素を含む SPS を含む。他の例では、ビットストリームは、シンタックス要素を含む PPS を含む。

【0150】

[0161]その後、ビデオデコーダ 30 は、シンタックス要素が第 1 の値、例えば、0 を有するかどうかを決定する(234)。図 6 の例では、シンタックス要素が第 1 の値を有する場合(234の「YES」)、ピクチャは単一のタイルを有し、ビデオデコーダ 30 は、WPP を使用せずにピクチャの単一のタイルを復号する(236)。

【0151】

[0162]しかしながら、シンタックス要素が第 1 の値を有しない場合(234の「NO」)、ビデオデコーダ 30 は、シンタックス要素が第 2 の値、例えば、1 を有するかどうかを決定する(238)。シンタックス要素が第 2 の値を有すると決定したことに応答して(238の「YES」)、ビデオデコーダ 30 は、ピクチャが複数の均一に離間したタイルを有すると決定し、ビデオデコーダ 30 は、WPP を使用せずに均一に離間したタイルの各々を復号する(238)。

【0152】

[0163]一方、シンタックス要素が第 2 の値を有しない場合(238の「NO」)、ビデオデコーダ 30 は、シンタックス要素が第 3 の値、例えば、2 を有するかどうかを決定する(242)。シンタックス要素が第 3 の値を有すると決定したことに応答して(242の「YES」)、ビデオデコーダ 30 は、ピクチャが複数の不均一に離間したタイルを有すると決定し、ビデオデコーダ 30 は、WPP を使用せずにピクチャの不均一に離間したタイルを復号する(244)。しかしながら、シンタックス要素が第 3 の値を有しないと決定したことに応答して(242の「NO」)、ビデオデコーダ 30 は、WPP を使用してピクチャを完全に復号する(246)。このようにして、シンタックス要素が第 1 の値(例えば、3)を有する場合、ピクチャは、WPP を使用して完全に符号化され、シンタックス要素が第 1 の値とは異なる第 2 の値(例えば、0、1、又は 2)を有する場合、ピクチャは 1 つ又は複数のタイルに区分され、ピクチャは、WPP を使用せずに符号化される。

【0153】

[0164]図 7 は、本開示の 1 つ又は複数の態様による、ピクチャの CTB の各行が別個のサブストリーム中にある、ビデオデータを符号化するためのビデオエンコーダ 20 の例示的な動作 270 を示すフローチャートである。幾つかのビデオコード化システムでは、タイルと WPP 波とのためのエントリポイントを信号伝達する異なる方法がある。これは、

これらのビデオコード化システムに複雑さを加え得る。本開示の技法は、図 7 及び図 8 に関して説明したように、タイルと、W P P 波と、幾つかの例では、エントロピースライスとのエントリポイントを示すための統合シンタックスを与えることによって、これらの問題を解決し得る。

【 0 1 5 4 】

[0165] 図 7 の例では、ビデオエンコーダ 2 0 は、ビデオピクチャのシーケンスのピクチャを符号化するために W P P が使用されることを信号伝達する (2 7 2)。ビデオエンコーダ 2 0 は、ピクチャを符号化するために W P P が使用されることを様々な方法で信号伝達し得る。例えば、ビデオエンコーダ 2 0 は、ピクチャを完全に復号するために W P P が使用されるべきかどうかを示すシンタックス要素 (例えば、「`tile__mode`」) を含む S P S を生成し得る。別の例では、ビデオエンコーダ 2 0 は、ピクチャを復号するために W P P が使用されるべきかどうかを示すシンタックス要素 (例えば、「`tile__mode`」) を含む P P S を生成し得る。

10

【 0 1 5 5 】

[0166] 更に、ビデオエンコーダ 2 0 は、複数のサブストリームを生成するために W P P を実行する (2 7 4)。サブストリームの各々は、ピクチャのスライス中の C T B の 1 つの符号化された行を表す連続する一連のビットを含み得る。従って、C T B の各行は 1 つのサブストリームとして符号化される。ビデオエンコーダ 2 0 は、複数のサブストリームを含むコード化スライス N A L 単位を生成する (2 7 6)。コード化スライス N A L 単位は、上記の表 3 及び表 4 の例示的なシンタックスに準拠するスライスヘッダ及びスライスデータを含み得る。

20

【 0 1 5 6 】

[0167] 図 8 は、本開示の 1 つ又は複数の態様による、ピクチャの C T B の各行が別個のサブストリーム中にある、ビデオデータを復号するためのビデオデコーダ 3 0 の例示的な動作 2 8 0 を示すフローチャートである。図 8 の例では、ビデオデコーダ 3 0 は、コード化スライス N A L 単位を含むビットストリームを受信する (2 8 2)。コード化スライス N A L 単位は複数のサブストリームを含む。サブストリームの各々は、ピクチャのスライス中の C T B の 1 つの行を表す連続する一連のビットを含み得る。更に、図 8 の例では、ビデオデコーダ 3 0 は、ビットストリーム中の 1 つ又は複数のシンタックス要素に基づいて、W P P を使用してスライスが符号化されると決定する (2 8 4)。例えば、ビデオデコーダ 3 0 は、3 に等しい `tile__mode` シンタックス要素に基づいて、W P P を使用してスライスが符号化されると決定し得る。この例では、`tile__mode` シンタックス要素が 3 に等しくない場合、ビデオデコーダ 3 0 は、W P P を使用せずにピクチャの 1 つ又は複数のタイルの各々を復号し得る。

30

【 0 1 5 7 】

[0168] 次に、ビデオデコーダ 3 0 は、W P P を使用してスライスを復号する (2 8 6)。ビデオデコーダ 3 0 がスライスを復号すると、ビデオデコーダ 3 0 は、スライスの C T B に関連付けられたシンタックス要素を構文解析し得る。ビデオデコーダ 3 0 は、C T B に関連付けられたシンタックス要素を構文解析することの一部として、シンタックス要素のうちの幾つかに対して C A B A C 構文解析プロセスを実行し得る。

40

【 0 1 5 8 】

[0169] 図 9 A は、本開示の 1 つ又は複数の態様による、スライスデータを構文解析するための例示的な C A B A C 構文解析プロセス 3 0 0 の第 1 の部分を示すフローチャートである。ビデオデコーダ 3 0 は、スライスデータ中で及びコード化ツリーシンタックス中で記述子 `ae (v)` をもつシンタックス要素を構文解析するとき、図 9 A のプロセスを実行し得る。図 9 A のプロセスはシンタックス要素の値を出力し得る。

【 0 1 5 9 】

[0170] 図 9 A の例では、ビデオデコーダ 3 0 のエントロピー復号ユニット 1 5 0 は、C A B A C 構文解析プロセスの初期化を実行する (3 0 2)。幾つかの例では、C A B A C 構文解析プロセスの初期化は、H E V C W D 5 の 9 . 2 . 1 節に記載されている初期化

50

と同じである。

【 0 1 6 0 】

[0171]更に、エントロピー復号ユニット150は、隣接CTBのアドレスを決定する(304)。隣接CTBは、現在CTB(即ち、ビデオデコーダ30が現在復号しているCTB)の左、左上、上、又は右上に隣接するブロックを含んでいるCTBであり得る。幾つかの例では、エントロピー復号ユニット150は、次式のように隣接CTBのアドレスを決定し得る。

【 数 5 】

$$tbAddrT = cuAddress(x0 + 2 * (1 \ll \text{Log2MaxCUsSize}) - 1, y0 - 1)$$

10

【 0 1 6 1 】

上記の式において、tbAddrTは隣接CTBのアドレスを示し、x0は現在CTBの左上ルーマサンプルのx座標を示し、y0は現在CTBの左上ルーマサンプルのy座標を示し、Log2MaxCUsizeはCUの最大サイズの対数、底2を示す。関数cuAddressは、第1のパラメータによって指定されたx座標と、第2のパラメータによって指定されたy座標とを含む、CUのアドレスを戻す。

【 0 1 6 2 】

[0172]次に、エントロピー復号ユニット150は、隣接CTBのアドレスを使用して、ピクチャ内予測のための隣接CTBの利用可能性を決定する(306)。言い換えれば、エントロピー復号ユニット150は、CABACコンテキストを選択する際に使用するために、隣接CTBに関連する情報が利用可能であるかどうかを決定し得る。

20

【 0 1 6 3 】

[0173]エントロピー復号ユニット150は、ピクチャ内予測のための隣接CTBの利用可能性を様々な方法で決定し得る。例えば、エントロピー復号ユニット150は、ピクチャ内予測のための隣接CTBの利用可能性を決定するために、tbAddrTを入力として用いて、WD5の6.4.3節に記載されているプロセスを実行し得る。別の例では、エントロピー復号ユニット150は、以下の条件のうちの1つが当てはまらない限り、CTBがピクチャ内予測のために利用可能であると決定し得る。以下の条件のうちの1つが当てはまる場合、エントロピー復号ユニット150は、CTBがピクチャ内予測のために利用不可能であると決定し得る。第1に、CTBのアドレスが0よりも小さい場合、エントロピー復号ユニット150は、CTBがピクチャ内予測のために利用不可能であると決定し得る。第2に、CTBのアドレスが、エントロピー復号ユニット150が現在構文解析しているCTBのアドレスよりも大きい場合、エントロピー復号ユニット150は、CTBがピクチャ内予測のために利用不可能であると決定し得る。第3に、特定のCTBが、エントロピー復号ユニット150が現在構文解析しているCTBとは異なるスライスに属する場合、エントロピー復号ユニット150は、その特定のCTBがピクチャ内予測のために利用不可能であると決定し得る。例えば、特定のCTBのアドレスがtbAddrとして示され、エントロピー復号ユニット150が現在構文解析しているCTBのアドレスがCurrTbAddrとして示された場合、エントロピー復号ユニット150は、アドレスtbAddrをもつCTBが、アドレスCurrTbAddrをもつCTBとは異なるスライスに属すると決定し得る。第4に、ビデオデコーダ30が現在復号しているピクチャのタイルが単独で復号可能であり、CTBが、エントロピー復号ユニット150が現在構文解析しているCTBとは異なるタイル中にあることを、ビットストリーム中の1つ又は複数のシンタックス要素が示す場合、エントロピー復号ユニット150は、CTBがピクチャ内予測のために利用不可能であると決定し得る。例えば、表1の例示的なシンタックスのtile_boundary_independence_flagシンタックス要素が1に等しく、アドレスtbAddrをもつCTBが、アドレスCurrTbAddrをもつCTBとは異なるタイル中に含まれている場合、エントロピー復号ユニット

30

40

50

150は、CTBがピクチャ内予測のために利用不可能であると決定し得る。

【0164】

[0174]更に、エントロピー復号ユニット150は、エントロピー復号ユニット150が現在構文解析しているシンタックス要素（即ち、現在シンタックス要素）がコード化ツリーシンタックス構造中にあるかどうかを決定する（308）。現在シンタックス要素がコード化ツリーシンタックス構造中ではない場合（308の「NO」）、エントロピー復号ユニット150は、図9Bに示されているCABAC構文解析プロセス300の部分を実行し得る。一方、現在シンタックス要素がコード化ツリー構造中にある場合（308の「YES」）、エントロピー復号ユニット150は、現在ピクチャ（即ち、現在CTBを含むピクチャ）のタイルが単独で復号可能であるかどうかを決定する（310）。例えば、表1の例示的なSPSシンタックスでは、現在ピクチャに関連付けられたSPSが、1に等しい`tile__boundary__independence__flag`シンタックス要素を含む場合、エントロピー復号ユニット150は、現在ピクチャのタイルが単独で復号可能であると決定し得る。現在ピクチャのタイルが単独で復号可能であると決定したことに応答して（310の「YES」）、エントロピー復号ユニット150は、図9Bに示されているCABAC構文解析プロセス300の部分を実行し得る。

10

【0165】

[0175]しかしながら、現在ピクチャのタイルが単独で復号可能でないと決定したことに応答して（310の「NO」）、エントロピー復号ユニット150は、`tbAddr%picWidthInLCUs`が0に等しいかどうかを決定し、但し、`tbAddr`は隣接CTBのアドレスであり、`%`はモジュロ演算子を示し、`picWidthInLCUs`はCTB（即ち、LCU）中の現在ピクチャの幅を示す（312）。

20

【0166】

[0176]`tbAddr%picWidthInLCUs`が0に等しいと決定したことに応答して（312の「YES」）、エントロピー復号ユニット150は、隣接CTBがピクチャ内予測のために利用可能であるかどうかを決定する（314）。幾つかの例では、エントロピー復号ユニット150は、アクション306において、隣接CTBがピクチャ内予測のために利用可能であるかどうかを示す変数`availableFlagT`の値を決定するためのプロセスを実行し得る。変数`availableFlagT`が1に等しい場合、隣接CTBはピクチャ内予測のために利用可能である。アクション314において、エントロピー復号ユニット150は、変数`availableFlagT`が1に等しいかどうかを決定し得る。

30

【0167】

[0177]隣接CTBがピクチャ内予測のために利用可能であると決定したことに応答して（314の「YES」）、エントロピー復号ユニット150は、CABAC構文解析プロセスの同期プロセスを実行する（316）。幾つかの例では、エントロピー復号ユニット150は、HEVCWD5の9.2.1.3節に記載されている同期プロセスを実行し得る。同期プロセスを実行した後に、又は隣接CTBがピクチャ内予測のために利用可能でないと決定したことに応答して（314の「NO」）、エントロピー復号ユニット150は、終了前の二分決定（binary decision）のための復号プロセスを実行する（318）。概して、終了前の二分決定のための復号プロセスは、`end_of_slice_flag`及び`pcm_flag`シンタックス要素をエントロピー復号するための特殊な復号プロセスである。ビデオデコード30は、スライスデータを構文解析するプロセスの終了前に二分決定を行うために、`end_of_slice_flag`及び`pcm_flag`を使用し得る。幾つかの例では、エントロピー復号ユニット150は、HEVCWD5の9.2.3.2.4節において指定されているように、終了前に二分決定のための復号プロセスを実行し得る。

40

【0168】

[0178]終了前に二分決定のための復号プロセスを実行した後に（318）、エントロピー復号ユニット150は、算術復号エンジンのための初期化プロセスを実行する（320

50

）。幾つかの例では、エントロピー復号ユニット150は、HEVC WD5の9.2.1.4節において定義されている初期化プロセスを実行し得る。算術復号エンジンのための初期化プロセスを実行した後に、エントロピー復号ユニット150は、図9Bに示されているCABAC構文解析プロセス300の部分を実行し得る。

【0169】

[0179] `tbAddrPicWidthInLCUs` が0に等しくない場合(312の「NO」)、エントロピー復号ユニット150は、`tbAddrPicWidthInLCUs` が2に等しいかどうかを決定する(322)。言い換えれば、エントロピー復号ユニット150は、隣接CTBのCTBアドレス `mod CTB` 中の現在ピクチャの幅が2に等しいかどうかを決定し得る。`tbAddrPicWidthInLCUs` が2に等しくないとは決定したことに応答して、エントロピー復号ユニット150は、図9Bに示されているCABAC構文解析プロセス300の部分を実行し得る。しかしながら、`tbAddrPicWidthInLCUs` が2に等しいとは決定したことに応答して(322の「YES」)、エントロピー復号ユニット150は記憶プロセスを実行する(324)。概して、記憶プロセスは、`end_of_slice_flag` シンタックス要素以外のシンタックス要素に割り当てられたコンテキスト変数の初期化プロセスにおいて使用される変数を出力する。幾つかの例では、エントロピー復号ユニット150は、HEVC WD5の9.2.1.2節において定義されている記憶プロセスを実行し得る。記憶プロセスを実行した後に、エントロピー復号ユニット150は、図9Bに示されているCABAC構文解析プロセス300の部分を実行し得る。

【0170】

[0180] 図9Bは、図9Aの例示的なCABAC構文解析プロセス300の続きを示すフローチャートである。図9Bに示すように、エントロピー復号ユニット150は、現在シンタックス要素を2値化する(330)。言い換えれば、エントロピー復号ユニット150は、現在シンタックス要素の2値化を導出し得る。シンタックス要素の2値化は、シンタックス要素の全ての可能な値のためのピンストリングのセットであり得る。ピンストリングは、シンタックス要素の2値化からのシンタックス要素の値の中間表現であるピンのストリングである。幾つかの例では、エントロピー復号ユニット150は、現在シンタックス要素の2値化を導出するために、HEVC WD5の9.2.2節において定義されているプロセスを実行し得る。

【0171】

[0181] 更に、エントロピー復号ユニット150は、コード化プロセスフローを決定する(332)。エントロピー復号ユニット150は、現在シンタックス要素の2値化と、構文解析されたピンのシーケンスとに基づいてコード化プロセスフローを決定し得る。幾つかの例では、エントロピー復号ユニット150は、HEVC WD5の9.2.2.9節に記載されているようにコード化プロセスフローを決定し得る。

【0172】

[0182] 更に、エントロピー復号ユニット150は、現在シンタックス要素の2値化の各ピンのためのコンテキストインデックスを決定する(334)。現在シンタックス要素の2値化のピンの各々は変数 `binIdx` によってインデックス付けされ、現在シンタックス要素の2値化のピンのためのコンテキストインデックスは `ctxIdx` として示され得る。幾つかの例では、エントロピー復号ユニット150は、HEVC WD5の9.2.3.1節において指定されているように現在シンタックス要素の2値化のピンのためのコンテキストインデックスを決定し得る。

【0173】

[0183] エントロピー復号ユニット150は、コンテキストインデックスごとに算術復号プロセスを実行する(336)。幾つかの例では、エントロピー復号ユニット150は、HEVC WD5の9.2.3.2節において指定されているようにコンテキストインデックスごとに算術復号プロセスを実行し得る。コンテキストインデックスごとに算術復号プロセスを実行することによって、エントロピー復号ユニット150は、構文解析された

ピンのシーケンスを生成し得る。

【 0 1 7 4 】

[0184] エントロピー復号ユニット 1 5 0 は、構文解析されたピンのシーケンスが、現在シンタックス要素の 2 値化によって生成されたピンストリングのセット中のピンストリングに一致するかどうかを決定する (3 4 0)。構文解析されたピンのシーケンスが、現在シンタックス要素の 2 値化によって生成されたピンストリングのセット中のピンストリングに一致する場合 (3 4 0 の「 Y E S 」)、エントロピー復号ユニット 1 5 0 は、対応する値を現在シンタックス要素に割り当てる (3 4 2)。対応する値を現在シンタックス要素に割り当てた後に、又は、構文解析されたピンのシーケンスが、現在シンタックス要素の 2 値化によって生成されたピンストリングのセット中のどのピンストリングにも一致しないと決定したことに応答して (3 4 0 の「 N O 」)、エントロピー復号ユニット 1 5 0 は、現在シンタックス要素を構文解析するのを完了する。

10

【 0 1 7 5 】

[0185] 幾つかの例では、現在シンタックス要素が `mb__type` シンタックス要素であり、`mb__type` シンタックス要素の復号された値が `I__PCM` に等しい場合、`HEVC_WD5` の 9 . 2 . 1 . 2 節において指定されているように `pcm__alignment__zero__bit` シンタックス要素ならびに全ての `pcm__sample__luma` 及び `pcm__sample__chroma` データを復号した後に、エントロピー復号ユニット 1 5 0 は初期化され得る。

【 0 1 7 6 】

20

[0186] 図 1 0 は、W P P の一例を示す概念図である。上記で説明したように、ピクチャは画素ブロックに区分され、画素ブロックの各々は C T B に関連付けられ得る。図 1 0 は、C T B に関連付けられた画素ブロックを白い矩形のグリッドとして示している。ピクチャは C T B 行 3 5 0 A ~ 3 5 0 E (総称して、「 C T B 行 3 5 0 」) を含む。

【 0 1 7 7 】

[0187] (例えば、複数の並列処理コアのうちの 1 つによって実行される) 第 1 の並列処理スレッドが、C T B 行 3 5 0 A 中の C T B をコード化していることがある。同時に、(例えば、他の並列処理コアによって実行される) 他のスレッドが、C T B 行 3 5 0 B、3 5 0 C、及び 3 5 0 D 中の C T B をコード化していることがある。図 1 0 の例では、第 1 のスレッドは C T B 3 5 2 A を現在コード化しており、第 2 のスレッドは C T B 3 5 2 B を現在コード化しており、第 3 のスレッドは C T B 3 5 2 C を現在コード化しており、第 4 のスレッドは C T B 3 5 2 D を現在コード化している。本開示では、C T B 3 5 2 A、3 5 2 B、3 5 2 C、及び 3 5 2 D を「現在 C T B 3 5 2」と総称することがある。ビデオコードは、すぐ上の行の 3 つ以上の C T B がコード化された後、C T B 行をコード化し始め得るので、現在 C T B 3 5 2 は、2 つの C T B の幅だけ互いから水平方向に変位される。

30

【 0 1 7 8 】

[0188] 図 1 0 の例では、スレッドは、現在 C T B 3 5 2 中の C U のためのイントラ予測又はインター予測を実行するために、太い灰色の矢印によって示される C T B からのデータを使用し得る。(スレッドはまた、C U のためのインター予測を実行するために、1 つ又は複数の参照フレームからのデータを使用し得る。) 所与の C T B をコード化するために、スレッドは、前にコード化された C T B に関連する情報に基づいて 1 つ又は複数の C A B A C コンテキストを選択し得る。スレッドは、所与の C T B の第 1 の C U に関連付けられたシンタックス要素に対して C A B A C コード化を実行するために、1 つ又は複数の C A B A C コンテキストを使用し得る。所与の C T B が行の最左 C T B でない場合、スレッドは、所与の C T B の左側の C T B の最後の C U に関連する情報に基づいて 1 つ又は複数の C A B A C コンテキストを選択し得る。所与の C T B が行の最左 C T B である場合、スレッドは、所与の C T B の上の及び 2 C T B 右の C T B の最後の C U に関連する情報に基づいて 1 つ又は複数の C A B A C コンテキストを選択し得る。スレッドは、現在 C T B 3 5 2 の第 1 の C U のための C A B A C コンテキストを選択するために、細い黒い矢印に

40

50

よって示される C T B の最後の C U からのデータを使用し得る。

【 0 1 7 9 】

[0189] 図 1 1 は、複数のタイル 4 0 2 A、4 0 2 B、4 0 2 C、4 0 2 D、4 0 2 E、及び 4 0 2 F（総称して、「タイル 4 0 2」）に区分されるピクチャ 4 0 0 のための例示的な C T B コード化順序を示す概念図である。ピクチャ 4 0 0 中の各矩形ブロックは、C T B に関連付けられた画素ブロックを表す。太い破線は、例示的なタイル境界を示す。異なるタイプのクロスハッチングは、異なるスライスに対応する。

【 0 1 8 0 】

[0190] 画素ブロックにおける数字は、ピクチャ 4 0 0 のタイルコード化順序で対応する C T B（L C U）の位置を示す。図 1 1 の例に示すように、タイル 4 0 2 A 中の C T B が最初にコード化され、タイル 4 0 2 B 中の C T B が続き、タイル 4 0 2 C 中の C T B が続き、タイル 4 0 2 D 中の C T B が続き、タイル 4 0 2 E 中の C T B が続き、タイル 4 0 2 F 中の C T B が続く。タイル 4 0 2 の各々の内で、C T B は、ラスト走査順序に従ってコード化される。

【 0 1 8 1 】

[0191] ビデオエンコーダは、ピクチャ 4 0 0 のために 4 つのコード化スライス N A L 単位を生成し得る。第 1 のコード化スライス N A L 単位は C T B 1 ~ 1 8 の符号化表現を含み得る。第 1 のコード化スライス N A L 単位のスライスデータは 2 つのサブストリームを含み得る。第 1 のサブストリームは C T B 1 ~ 9 の符号化表現を含み得る。第 2 のサブストリームは C T B 1 0 ~ 1 8 の符号化表現を含み得る。従って、第 1 のコード化スライス N A L 単位は、複数のタイルを含んでいるスライスの符号化表現を含み得る。

【 0 1 8 2 】

[0192] 第 2 のコード化スライス N A L 単位は C T B 1 9 ~ 2 2 の符号化表現を含み得る。第 2 のコード化スライス N A L 単位のスライスデータは単一のサブストリームを含み得る。第 3 のコード化スライス N A L 単位は C T B 2 3 ~ 2 7 の符号化表現を含み得る。第 3 のコード化スライス N A L 単位のスライスデータは単一のサブストリームのみを含み得る。従って、タイル 4 0 2 C は複数のスライスを含んでいることがある。

【 0 1 8 3 】

[0193] 第 4 のコード化スライス N A L 単位は C T B 2 8 ~ 4 5 の符号化表現を含み得る。第 4 のコード化スライス N A L 単位のスライスデータは、それぞれタイル 4 0 2 D、4 0 2 E、及び 4 0 2 F のための 3 つのサブストリームを含み得る。従って、第 4 のコード化スライス N A L 単位は、複数のタイルを含んでいるスライスの符号化表現を含み得る。

【 0 1 8 4 】

[0194] 1 つ又は複数の例では、説明した機能は、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合せで実装され得る。ソフトウェアで実装される場合、機能は、1 つ又は複数の命令又はコードとしてコンピュータ可読媒体上に記憶されるか、若しくはコンピュータ可読媒体を介して送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体を含み得、又は、例えば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を可能にする任意の媒体を含む通信媒体を含み得る。このようにして、コンピュータ可読媒体は、概して、(1) 非一時的である有形コンピュータ可読記憶媒体、若しくは (2) 信号又は搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明した技法の実装のための命令、コード及び / 又はデータ構造を取り出すために 1 つ又は複数のコンピュータ若しくは 1 つ又は複数のプロセッサによってアクセスされ得る任意の利用可能な媒体であり得る。コンピュータプログラム製品はコンピュータ可読媒体を含み得る。

【 0 1 8 5 】

[0195] 限定ではなく例として、そのようなコンピュータ可読記憶媒体は、R A M、R O M、E E P R O M（登録商標）、C D - R O M 又は他の光ディスク記憶、磁気ディスク記憶、又は他の磁気記憶装置、フラッシュメモリ、若しくは命令又はデータ構造の形態の所

10

20

30

40

50

望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る、任意の他の媒体を備えることができる。また、いかなる接続もコンピュータ可読媒体と適切に呼ばれる。例えば、命令が、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、又は赤外線、無線、及びマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、又は他のリモート発信源から送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、又は赤外線、無線、及びマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。但し、コンピュータ可読記憶媒体及びデータ記憶媒体は、接続、搬送波、信号、又は他の一時媒体を含まないが、代わりに非一時的有形記憶媒体を対象とすることを理解されたい。本明細書で使用するディスク(disk)及びディスク(disc)は、コンパクトディスク(disc)(CD)、レーザディスク(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)及びブルーレイ(登録商標)ディスク(disc)を含み、ディスク(disk)は、通常、データを磁氣的に再生し、ディスク(disc)は、データをレーザで光学的に再生する。上記の組合せもコンピュータ可読媒体の範囲内に含まれるべきである。

【0186】

[0196]命令は、1つ又は複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、若しくは他の等価な集積回路又はディスクリット論理回路など、1つ又は複数のプロセッサによって実行され得る。従って、本明細書で使用する「プロセッサ」という用語は、前述の構造のいずれか、又は本明細書で説明した技法の実装に好適な任意の他の構造を指し得る。更に、幾つかの態様では、本明細書で説明した機能は、符号化及び復号のために構成された専用ハードウェア及び/又はソフトウェアモジュール内に提供され得、若しくは複合コーデックに組み込まれ得る。また、本技法は、1つ又は複数の回路又は論理要素中に十分に実装され得る。

【0187】

[0197]本開示の技法は、ワイヤレスハンドセット、集積回路(IC)又はICのセット(例えば、チップセット)を含む、多種多様な機器又は装置において実装され得る。本開示では、開示する技法を実行するように構成された機器の機能的態様を強調するために様々な構成要素、モジュール、又はユニットについて説明したが、それらの構成要素、モジュール、又はユニットを、必ずしも異なるハードウェアユニットによって実現する必要があるとは限らない。むしろ、上記で説明したように、様々なユニットが、好適なソフトウェア及び/又はファームウェアとともに、上記で説明した1つ又は複数のプロセッサを含めて、コーデックハードウェアユニットにおいて組み合わせられるか、又は相互動作ハードウェアユニットの集合によって与えられ得る。

【0188】

[0198]様々な例について説明した。これら及び他の例は以下の特許請求の範囲内に入る。

以下に本件出願当初の特許請求の範囲に記載された発明を付記する。

[1] ビデオデータを符号化する方法であって、前記方法は]ビデオピクチャのシーケンスのピクチャを符号化するために波面並列処理(WPP)が使用されることを信号伝達することと]複数のサブストリームを生成するためにWPPを実行することと、前記サブストリームの各々が、前記ピクチャのスライス中のコード化ツリーブロック(CTB)の1つの符号化された行を表す連続する一連のビットを含む]前記複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ(NAL)単位を生成することとを備える、方法。

[2] 前記コード化スライスNAL単位を生成することは、前記コード化スライスNAL単位のスライスヘッダと、前記コード化スライスNAL単位のスライスデータとを生成することとを備え、前記スライスデータが前記サブストリームを含み、前記スライスヘッダは、前記サブストリームのエントリポイントがそれから決定可能である複数のオフセット

シンタックス要素を含む、[1]に記載の方法。

[3] 前記スライスヘッダが第 1 のシンタックス要素と第 2 のシンタックス要素とを更に含み、前記複数のオフセットシンタックス要素中のオフセットシンタックス要素の数が前記第 1 のシンタックス要素に基づいて決定可能であり、前記オフセットシンタックス要素の各々のビットでの長さが前記第 2 のシンタックス要素に基づいて決定可能である、[2]に記載の方法。

[4] 前記ピクチャを符号化するために W P P が使用されることを信号伝達することは、前記ピクチャを符号化するために W P P が使用されることを示すシンタックス要素を含むピクチャパラメータセット (P P S) を生成することを備える、[1]に記載の方法。

[5] 前記ピクチャを符号化するために W P P が使用されることを信号伝達することは、前記ピクチャを符号化するために W P P が使用されることを示すシンタックス要素を含むシーケンスパラメータセット (S P S) を生成することを備える、[1]に記載の方法。

[6] 前記複数のサブストリームを生成するために W P P を実行することは] 第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと] 前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー符号化することと、を備え、前記第 2 の C T B が、前記ピクチャの前記左境界に隣接し、C T B の 1 行だけ前記第 1 の C T B よりも低い、[1]に記載の方法。

[7] 前記方法がモバイルコンピュータ機器によって実行される、[1]に記載の方法。

[8] ビデオピクチャのシーケンスのピクチャを符号化するために波面並列処理 (W P P) が使用されることを信号伝達することと] 複数のサブストリームを生成するために W P P を実行することと、前記サブストリームの各々が、前記ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの符号化された行を表す連続する一連のビットを含む] 前記複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ (N A L) 単位を生成することとを行うように構成された 1 つ以上のプロセッサを備えるコンピュータ機器。

[9] 前記コード化スライス N A L 単位を生成することは、前記コード化スライス N A L 単位のスライスヘッダと、前記コード化スライス N A L 単位のスライスデータとを生成することを備え、前記スライスデータが前記サブストリームを含み、前記スライスヘッダは、前記サブストリームのエントリポイントがそれから決定可能である複数のオフセットシンタックス要素を含む、[8]に記載のコンピュータ機器。

[1 0] 前記スライスヘッダが第 1 のシンタックス要素と第 2 のシンタックス要素とを更に含み、前記複数のオフセットシンタックス要素中のオフセットシンタックス要素の数が前記第 1 のシンタックス要素に基づいて決定可能であり、前記オフセットシンタックス要素の各々のビットでの長さが前記第 2 のシンタックス要素に基づいて決定可能である、[9]に記載のコンピュータ機器。

[1 1] 前記 1 つ以上のプロセッサは、前記ピクチャを符号化するために W P P が使用されることを示すシンタックス要素を含むピクチャパラメータセット (P P S) を生成するように構成された、[8]に記載のコンピュータ機器。

[1 2] 前記 1 つ以上のプロセッサは、前記ピクチャを符号化するために W P P が使用されることを示すシンタックス要素を含むシーケンスパラメータセット (S P S) を生成するように構成された、[8]に記載のコンピュータ機器。

[1 3] 前記 1 つ以上のプロセッサは] 第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと] 前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー符号化することと、を行うように構成され、前記第 2 の C T B が、前記ピクチャの前記

10

20

30

40

50

左境界に隣接し、C T B の 1 行だけ前記第 1 の C T B よりも低い、[8] に記載のコンピュータ機器。

[1 4] 前記コンピュータ機器がモバイルコンピュータ機器である、[8] に記載のコンピュータ機器。

[1 5] ピクチャを符号化するために波面並列処理 (W P P) が使用されることを信号伝達するための手段と] 複数のサブストリームを生成するために W P P を実行するための手段と、前記サブストリームの各々が、前記ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの符号化された行を表す連続する一連のビットを含む] 前記複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ (N A L) 単位を生成するための手段とを備えるコンピュータ機器。

10

[1 6] コンピュータ機器の 1 つ以上のプロセッサによって実行されたとき] ピクチャを符号化するために波面並列処理 (W P P) が使用されることを信号伝達することと] 複数のサブストリームを生成するために W P P を実行することと、前記サブストリームの各々が、前記ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの符号化された行を表す連続する一連のビットを含む] 前記複数のサブストリームを含むコード化スライスネットワークアブストラクションレイヤ (N A L) 単位を生成することとを行うように前記コンピュータ機器を設定する命令を記憶したコンピュータ可読記憶媒体。

[1 7] ビデオデータを復号する方法であって、前記方法は] コード化スライスネットワークアブストラクションレイヤ (N A L) 単位を含むビットストリームを受信することと、前記コード化スライス N A L 単位が複数のサブストリームを含み、前記サブストリームの各々が、ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの行を表す連続する一連のビットを含む] 前記ビットストリーム中のシンタックス要素に基づいて、波面並列処理 (W P P) を使用して前記スライスが符号化されると決定することと] W P P を使用して前記スライスを復号することとを備える、方法。

20

[1 8] 前記コード化スライス N A L 単位がスライスヘッダとスライスデータとを含み、前記方法は、W P P を使用して前記スライスが符号化されることを示す前記スライスヘッダ中のシンタックス要素に少なくとも部分的に基づいて、前記コード化スライス N A L 単位が、前記サブストリームの前記スライスデータ内の位置を示す複数のオフセットシンタックス要素を含むと決定することを更に備える、[1 7] に記載の方法。

[1 9] 前記スライスヘッダが第 1 のシンタックス要素と第 2 のシンタックス要素とを含み、前記方法は] 前記第 1 のシンタックス要素に基づいて、前記複数のオフセットシンタックス要素中に何個のオフセットシンタックス要素があるかを決定することと] 前記第 2 のシンタックス要素に基づいて、前記オフセットシンタックス要素のビットでの長さを決定することと] 前記複数のオフセットシンタックス要素中に何個のオフセットシンタックス要素があるかと、前記オフセットシンタックス要素の前記ビットでの長さにと少なくとも部分的に基づいて、前記ビットストリームから前記オフセットシンタックス要素を構文解析することとを更に備える、[1 8] に記載の方法。

30

[2 0] 前記オフセットシンタックス要素に少なくとも部分的に基づいて、前記サブストリームのメモリ中の位置を決定することを更に備える、[1 8] に記載の方法。

[2 1] 前記ビットストリームがピクチャパラメータセット (P P S) を含み、前記方法は、前記 P P S 中の 1 つ以上のシンタックス要素に基づいて、前記ピクチャを符号化するために W P P が使用されると決定することを更に備える、[1 7] に記載の方法。

40

[2 2] 前記ビットストリームがシーケンスパラメータセット (S P S) を含み、前記方法は、前記 S P S 中の 1 つ以上のシンタックス要素に基づいて、前記ピクチャを符号化するために W P P が使用されると決定することを更に備える、[1 7] に記載の方法。

[2 3] W P P を使用して前記スライスを復号することは] 第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと] 前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー復号することと、前記第 2 の C T B が、前記ピクチャの前記左境界

50

に隣接し、C T B の 1 行だけ前記第 1 の C T B よりも低い、を備える、[1 7] に記載の方法。

[2 4] 前記方法がモバイルコンピュータ機器によって実行される、[1 7] に記載の方法。

[2 5] コード化スライスネットワークアブストラクションレイヤ (N A L) 単位を含むビットストリームを受信することと、前記コード化スライス N A L 単位が複数のサブストリームを含み、前記サブストリームの各々が、ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの行を表す連続する一連のビットを含む] 前記ビットストリーム中の 1 つ以上のシンタックス要素に基づいて、波面並列処理 (W P P) を使用して前記スライスが符号化されると決定することと] W P P を使用して前記スライスを復号することとを行うように構成された 1 つ以上のプロセッサを備えるコンピュータ機器。

10

[2 6] 前記コード化スライス N A L 単位がスライスヘッダとスライスデータとを含み、前記 1 つ以上のプロセッサは、W P P を使用して前記スライスが符号化されることを示す前記スライスヘッダ中のシンタックス要素に少なくとも部分的に基づいて、前記コード化スライス N A L 単位が、前記サブストリームの前記スライスデータ内の位置を示す複数のオフセットシンタックス要素を含むと決定するように更に構成された、[2 5] に記載のコンピュータ機器。

[2 7] 前記スライスヘッダが第 1 のシンタックス要素と第 2 のシンタックス要素とを含み、前記第 1 のシンタックス要素は、前記複数のオフセットシンタックス要素中に何個のオフセットシンタックス要素があるかを示し、前記第 2 のシンタックス要素が、前記オフセットシンタックス要素のビットでの長さを示し] 前記 1 つ以上のプロセッサが、前記第 1 のシンタックス要素と前記第 2 のシンタックス要素とに少なくとも部分的に基づいて、前記ビットストリームから前記オフセットシンタックス要素をパースするように構成された、[2 6] に記載のコンピュータ機器。

20

[2 8] 前記ビットストリームがピクチャパラメータセット (P P S) を含み、前記 1 つ以上のプロセッサは、前記 P P S 中の 1 つ以上のシンタックス要素に基づいて、前記ピクチャを符号化するために W P P が使用されると決定するように更に構成された、[2 5] に記載のコンピュータ機器。

[2 9] 前記ビットストリームがシーケンスパラメータセット (S P S) を含み、前記 1 つ以上のプロセッサは、前記 S P S 中の 1 つ以上のシンタックス要素に基づいて、前記ピクチャを符号化するために W P P が使用されると決定するように更に構成された、[2 5] に記載のコンピュータ機器。

30

[3 0] 1 つ以上のプロセッサは] 第 1 の C T B が単一の C T B によって前記ピクチャの左境界から分離されると決定したことに応答して、前記第 1 の C T B に関連するコンテキスト変数を記憶することと] 前記第 1 の C T B に関連する前記コンテキスト変数に少なくとも部分的に基づいて、第 2 の C T B の 1 つ以上のシンタックス要素をエントロピー復号することと、前記第 2 の C T B が、前記ピクチャの前記左境界に隣接し、C T B の 1 行だけ前記第 1 の C T B よりも低い、を行うように構成された、[2 5] に記載のコンピュータ機器。

[3 1] 前記コンピュータ機器がモバイルコンピュータ機器である、[2 5] に記載のコンピュータ機器。

40

[3 2] コード化スライスネットワークアブストラクションレイヤ (N A L) 単位を含むビットストリームを受信するための手段と、前記コード化スライス N A L 単位が複数のサブストリームを含み、前記サブストリームの各々が、ピクチャのスライス中のコード化ツリーブロック (C T B) の 1 つの行を表す連続する一連のビットを含む] 前記ビットストリーム中の 1 つ以上のシンタックス要素に基づいて、波面並列処理 (W P P) を使用して前記スライスが符号化されると決定するための手段と] W P P を使用して前記スライスを復号するための手段とを備えるコンピュータ機器。

[3 3] コンピュータ機器の 1 つ以上のプロセッサによって実行されたとき] コード化スライス N A L 単位を含むビットストリームを受信することと、前記コード化スライスネ

50

ットワークアブストラクションレイヤ（NAL）単位が複数のサブストリームを含み、前記サブストリームの各々が、ピクチャのスライス中のコード化ツリーブロック（CTB）の１つの行を表す連続する一連のビットを含む」前記ビットストリーム中の１つ以上のシンタックス要素に基づいて、波面並列処理（WPP）を使用して前記スライスが符号化されると決定することと」WPPを使用して前記スライスを復号することとを行うように前記コンピュータ機器を設定する命令を記憶したコンピュータ可読記憶媒体。

【図 1】

図 1

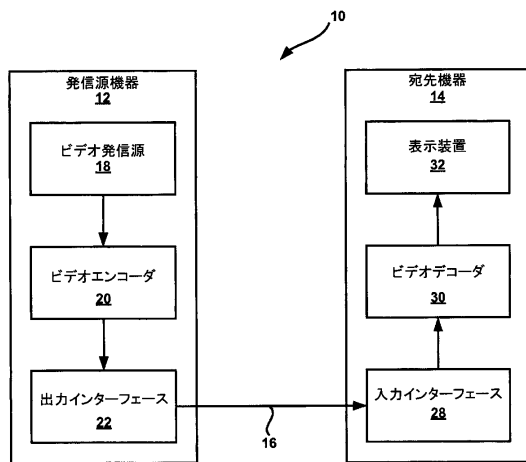


FIG. 1

【図 2】

図 2

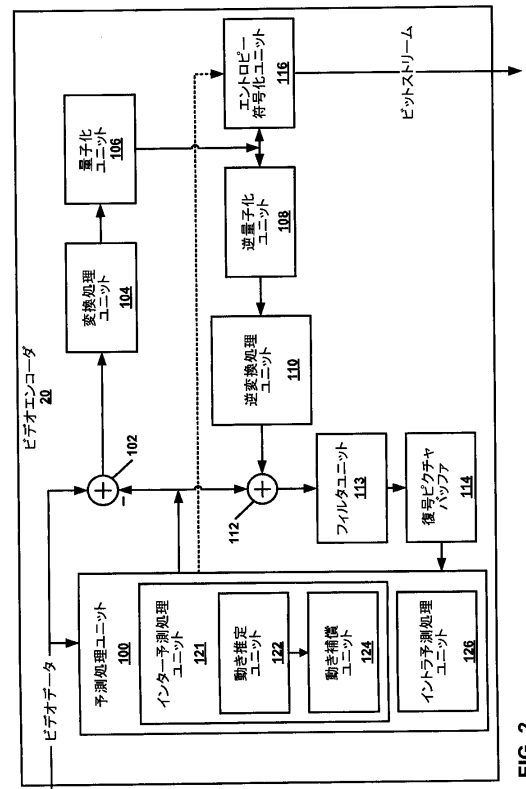


FIG. 2

【図 3】

図 3

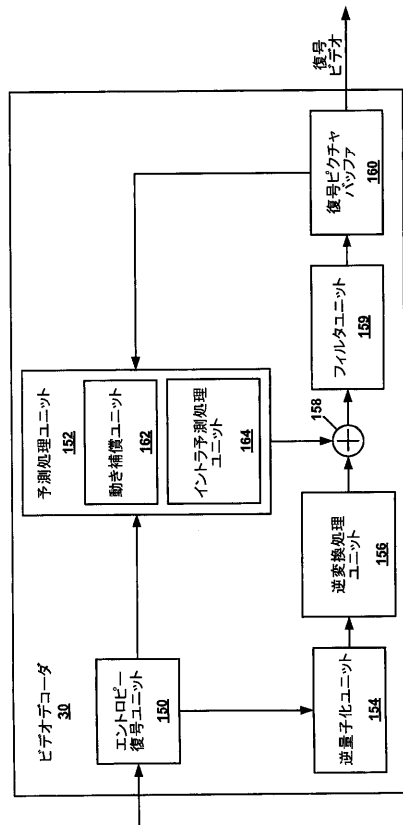


FIG. 3

【図 4】

図 4

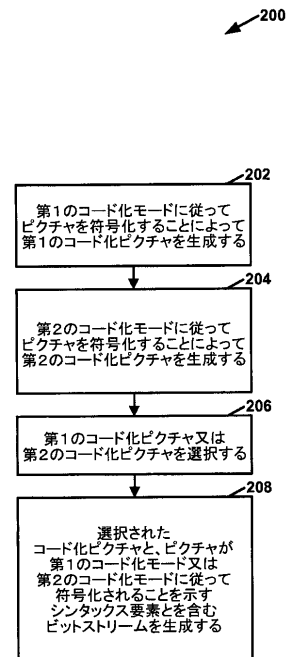


FIG. 4

【図 5】

図 5

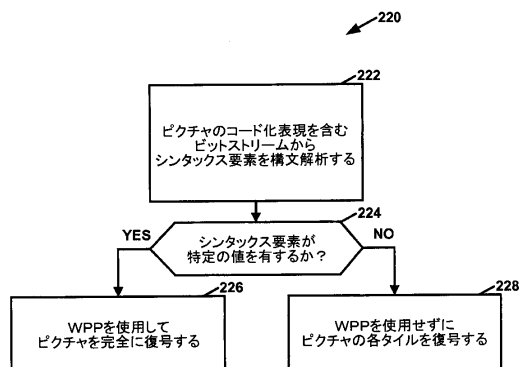


FIG. 5

【図 6】

図 6

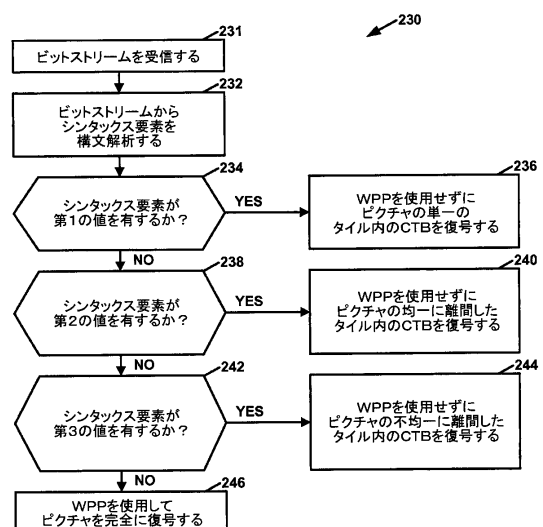


FIG. 6

【図 7】

図 7

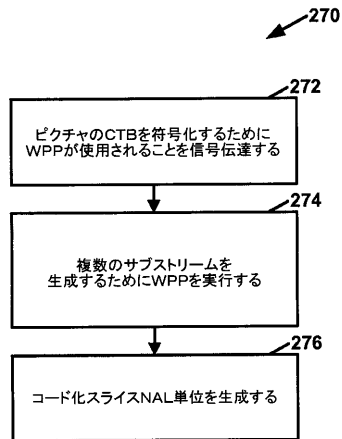


FIG. 7

【図 8】

図 8

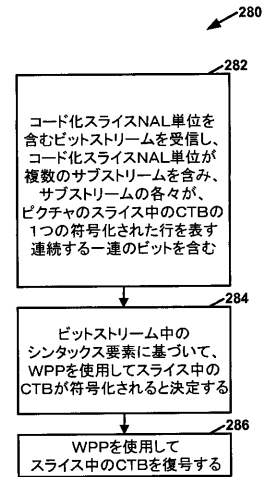


FIG. 8

【図 9 A】

図 9A

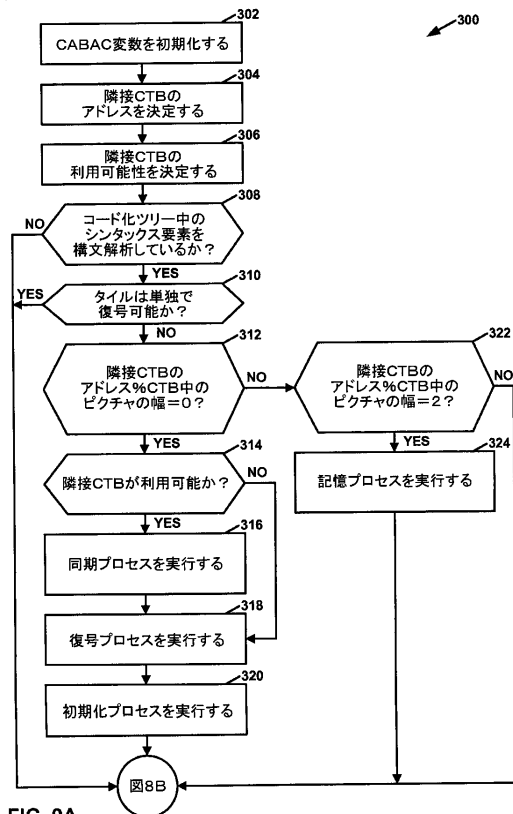


FIG. 9A

【図 9 B】

図 9B

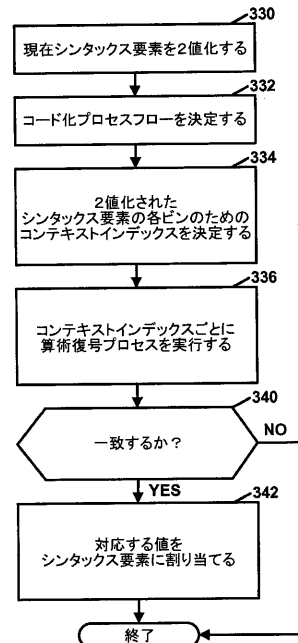


FIG. 9B

【 図 1 0 】

図 10

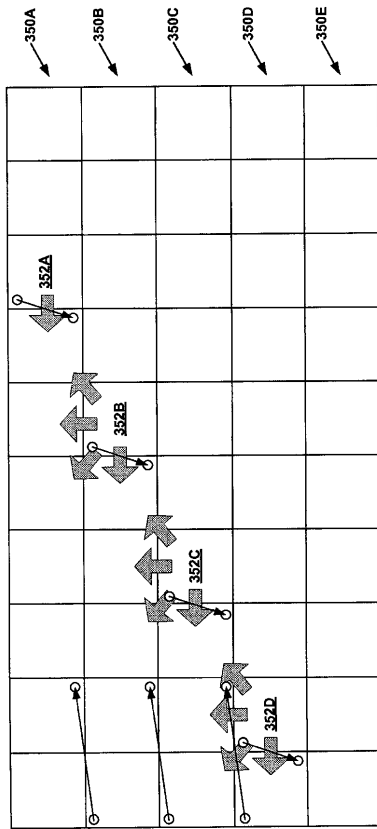


FIG. 10

【 図 1 1 】

図 11

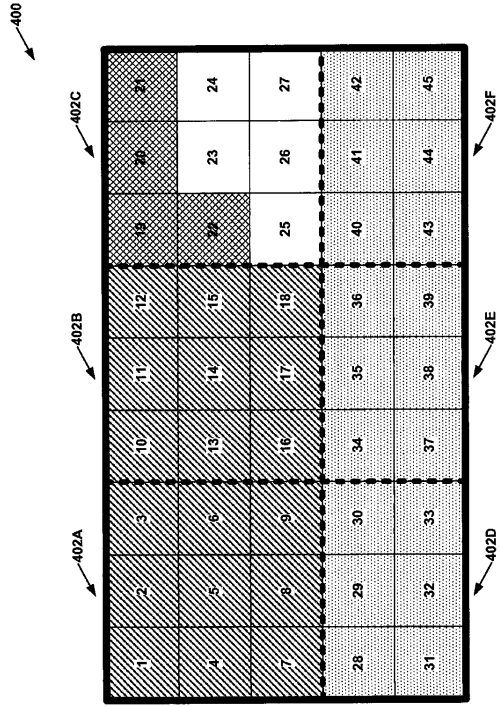


FIG. 11

 フロントページの続き

- (74)代理人 100153051
弁理士 河野 直樹
- (74)代理人 100140176
弁理士 砂川 克
- (74)代理人 100158805
弁理士 井関 守三
- (74)代理人 100179062
弁理士 井上 正
- (74)代理人 100124394
弁理士 佐藤 立志
- (74)代理人 100112807
弁理士 岡田 貴志
- (74)代理人 100111073
弁理士 堀内 美保子
- (72)発明者 ワン、イエ - クイ
アメリカ合衆国、カリフォルニア州 9 2 1 2 1、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
- (72)発明者 コバン、ムハンメド・ゼイド
アメリカ合衆国、カリフォルニア州 9 2 1 2 1、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

審査官 堀井 啓明

- (56)参考文献 Chih-Wei Hsu, et al., Wavefront Parallel Processing with Tiles, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting: Torino, IT, 14-22 July, 2011, 米国, JCTVC, 2 0 1 1 年 7 月 1 5 日, JCTVC-F063, P.1-P.8, URL, <http://phenix.it-sudparis.eu/jct/index.php>
- Muhammed Coban, et al., Unification of picture partitioning schemes, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7th Meeting: Geneva, CH, 21-30 November, 2011, 米国, JCTVC, 2 0 1 1 年 1 1 月 2 3 日, JCTVC-G315, P.1-P.11, URL, <http://phenix.it-sudparis.eu/jct/index.php>
- Kiran Misra, et al., Harmonization of Entry Points for Tiles and Wavefront Processing, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7th Meeting: Geneva, CH, 21-30 November, 2011, 米国, JCTVC, 2 0 1 1 年 1 1 月 2 4 日, JCTVC-G722, P.1-P.4, URL, <http://phenix.it-sudparis.eu/jct/index.php>
- Gordon Clare, et al., Wavefront Parallel Processing for HEVC Encoding and Decoding, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting: Torino, IT, 14-22 July, 2011, 米国, JCTVC, 2 0 1 1 年 7 月 1 6 日, JCTVC-F274, P.1-P.16, URL, <http://phenix.it-sudparis.eu/jct/index.php>
- Gordon Clare, et al., Wavefront and Cabac Flush: Different Degrees of Parallelism Without Transcoding, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting: Torino, IT, 14-22 July, 2011, 米国, JCTVC, 2 0 1 1 年 7 月 1 5 日, JCTVC-F275, P.1-P.11, URL, <http://phenix.it-sudparis.eu/jct/index.php>
- Felix Henry, et al., Wavefront Parallel Processing, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 5th Meeting: Geneva, CH, 16-23 March, 2011, 米国, JCTVC, 2 0 1 1 年 3 月 1 9 日, JCTVC-E196, P.1-P.9, URL, <http://phenix.it-sudparis.eu/jct/index.php>

(58)調査した分野(Int.Cl. , D B 名)

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