

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6246919号
(P6246919)

(45) 発行日 平成29年12月13日(2017.12.13)

(24) 登録日 平成29年11月24日(2017.11.24)

(51) Int.Cl.	F 1
HO4N 19/119	(2014.01)
HO4N 19/597	(2014.01)
HO4N 19/593	(2014.01)
HO4N 19/176	(2014.01)
HO4N 19/147	(2014.01)

HO 4 N 19/119
HO 4 N 19/597
HO 4 N 19/593
HO 4 N 19/176
HO 4 N 19/147

請求項の数 25 (全 70 頁) 最終頁に続く

(21) 出願番号	特願2016-522805 (P2016-522805)
(86) (22) 出願日	平成26年10月15日 (2014.10.15)
(65) 公表番号	特表2016-537860 (P2016-537860A)
(43) 公表日	平成28年12月1日 (2016.12.1)
(86) 國際出願番号	PCT/US2014/060732
(87) 國際公開番号	W02015/057861
(87) 國際公開日	平成27年4月23日 (2015.4.23)
審査請求日	平成29年2月24日 (2017.2.24)
(31) 優先権主張番号	61/891,203
(32) 優先日	平成25年10月15日 (2013.10.15)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	14/513,839
(32) 優先日	平成26年10月14日 (2014.10.14)
(33) 優先権主張国	米国(US)

早期審査対象出願

(73) 特許権者	595020643 クアアルコム・インコーポレイテッド QUALCOMM INCORPORATED アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
(74) 代理人	100108855 弁理士 蔵田 昌俊
(74) 代理人	100109830 弁理士 福原 淑弘
(74) 代理人	100158805 弁理士 井関 守三
(74) 代理人	100112807 弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】深度イントラコーディングのためのウェッジレットパターン拡張

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを復号する方法であって、

第1のサイズのブロックの1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成すること、

前記パーティションパターンリストへのインデックスを受信することと、

前記受信された前記パーティションパターンリストへのインデックスに基づいて、前記第1のサイズの前記ブロックのパーティションパターンを決定することと、

前記第1のサイズの前記ブロックの前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きく、前記深度ブロックは、3次元(3D)ビデオコーディングのフレームワークにおける深度マップのブロックである、

前記第1のサイズの前記ブロックの前記決定されたパーティションパターンを使用して前記第1のサイズの前記ブロックをイントラ予測復号することなく、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測復号することと

を備える方法。

【請求項 2】

前記第1のサイズのブロックの前記パーティションパターンは、サイズ16×16のブ

10

20

ロックのパーティションパターンを備え、前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することは、サイズ 16×16 の前記プロックの前記決定されたパーティションパターンに基づいて、サイズ 32×32 の前記深度プロックに関する前記パーティションパターンを決定することを備える、請求項1に記載の方法。

【請求項3】

前記深度プロック内のサブプロックを識別する情報を受信することをさらに備え、前記サブプロックのサイズが前記第1のサイズであり、

前記第1のサイズのプロックの前記パーティションパターンを決定することは、前記サブプロックに関する前記パーティションパターンを決定することを備え、

前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することは、前記サブプロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することを備える、請求項1に記載の方法。 10

【請求項4】

前記サブプロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することは、前記サブプロックに関する前記パーティションパターンの線形線を前記深度プロックの境界まで拡張することを備え、得られる線形線が、前記深度プロックに関する前記パーティションパターンを画定する、請求項3に記載の方法。

【請求項5】

前記第1のサイズの前記プロックの前記決定されたパーティションパターンは、前記第1のサイズの前記プロックの複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度プロックに関する前記パーティションパターンは、2つのパーティションに前記深度プロックを区分する前記深度プロックに関する線ベースのパーティションパターンを備える、請求項1に記載の方法。 20

【請求項6】

前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することは、前記第2のサイズの前記深度プロックに関する前記パーティションパターンを示す情報を受信することなく、前記第2のサイズの前記深度プロックに関する前記パーティションパターンを決定することを備える、請求項1に記載の方法。 30

【請求項7】

ビデオデータを符号化する方法であって、

第1のサイズのプロックの1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと、

前記パーティションパターンリストに基づいて、前記第1のサイズの前記プロックのパーティションパターンを決定することと、

前記第1のサイズの前記プロックの前記決定されたパーティションパターンに基づいて、第2のサイズの深度プロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きく、前記深度プロックは、3次元(3D)ビデオコーディングのフレームワークにおける深度マップのプロックである 40

、
前記第1のサイズの前記プロックの前記決定されたパーティションパターンを使用して前記第1のサイズの前記プロックをイントラ予測符号化することなく、前記深度プロックに関する前記決定されたパーティションパターンに基づいて、前記深度プロックをイントラ予測符号化することと、

前記第1のサイズの前記プロックの前記決定されたパーティションパターンを識別する前記構成されたパーティションパターンリストへのインデックスをシグナリングすることと
を備え、ビデオデコーダは、前記深度プロックをイントラ予測復号するために前記インデックスを使用する、方法。 50

【請求項 8】

前記第1のサイズのブロックの前記パーティションパターンは、サイズ 16×16 のブロックのパーティションパターンを備え、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、サイズ 16×16 の前記ブロックの前記決定されたパーティションパターンに基づいて、サイズ 32×32 の前記深度ブロックに関する前記パーティションパターンを決定することを備える、請求項7に記載の方法。

【請求項 9】

前記第1のサイズの前記ブロックの前記パーティションパターンを決定することは、前記構成されたパーティションパターンリストにおいて識別されたパーティションパターンから、前記第1のサイズの前記ブロックの前記パーティションパターンを決定することを備える、請求項7に記載の方法。 10

【請求項 10】

前記深度ブロック内のサブブロックを識別することをさらに備え、前記サブブロックのサイズが前記第1のサイズであり、

前記第1のサイズの前記ブロックの前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記パーティションパターンを決定することを備え、

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記決定されたパーティションに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを備える、請求項7に記載の方法。 20

【請求項 11】

前記識別されたサブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記パーティションパターンの線形線を前記深度ブロックの境界まで拡張することを備え、得られる線形線が、前記深度ブロックに関する前記パーティションパターンを画定する、請求項10に記載の方法。

【請求項 12】

前記第1のサイズの前記ブロックの前記パーティションパターンは、前記第1のサイズの前記ブロックの複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、請求項7に記載の方法。 30

【請求項 13】

ビデオコーディングのためのデバイスであって、

第1のサイズのブロックの1つまたは複数のパーティションパターンを記憶するビデオデータメモリと、

1つまたは複数のプロセッサを備えるビデオコーダと
を備え、前記ビデオコーダは、

前記第1のサイズの前記ブロックの前記1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと、

前記パーティションパターンリストに基づいて、前記記憶された1つまたは複数のパーティションパターンから、前記第1のサイズの前記ブロックのパーティションパターンを決定することと、ここにおいて、前記パーティションパターンリスト中の前記パーティションパターンは、前記パーティションパターンリスト中のインデックスによって識別される、

前記第1のサイズの前記ブロックの前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きく、前記深度ブロックは、3次元(3D)ビデオコーディングのフレームワークにおける深度マップのブロックである、

前記第1のサイズの前記ブロックの前記決定されたパーティションパターンを使用して前記第1のサイズの前記ブロックをイントラ予測コーディングすることなく、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測コーディングすることと
を行うように構成される、デバイス。

【請求項14】

前記第1のサイズのブロックの前記パーティションパターンは、サイズ 16×16 のブロックのパーティションパターンを備え、前記第2のサイズの前記深度ブロックは、サイズ 32×32 の深度ブロックを備え、前記ビデオコーダは、サイズ 16×16 の前記ブロックの前記決定されたパーティションパターンに基づいて、サイズ 32×32 の前記深度ブロックに関する前記パーティションパターンを決定するように構成される、請求項13に記載のデバイス。10

【請求項15】

前記ビデオコーダはビデオデコーダを備え、
前記ビデオデコーダは、
前記パーティションパターンリストへの前記インデックスを受信することを行うように構成され、20

前記第1のサイズの前記ブロックの前記パーティションパターンを決定するために、前記ビデオデコーダは、前記受信された前記パーティションパターンリストへのインデックスに基づいて、前記第1のサイズの前記ブロックの前記パーティションパターンを決定するように構成され、20

前記深度ブロックをイントラ予測コーディングするために、前記ビデオデコーダは、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測復号するように構成される、請求項13に記載のデバイス。

【請求項16】

前記ビデオコーダはビデオデコーダを備え、
前記ビデオデコーダは、前記深度ブロック内のサブブロックを識別する情報を受信するように構成され、前記サブブロックのサイズが前記第1のサイズであり、30

前記第1のサイズの前記ブロックの前記パーティションパターンを決定するために、前記ビデオデコーダは、前記サブブロックに関する前記パーティションパターンを決定するように構成され、30

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記サブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、請求項13に記載のデバイス。

【請求項17】

前記ビデオコーダはビデオデコーダを備え、
前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを示す情報を受信することなく、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、請求項13に記載のデバイス。40

【請求項18】

前記ビデオコーダはビデオエンコーダを備え、
前記深度ブロックをイントラ予測コーディングするために、前記ビデオエンコーダは、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測符号化するように構成される、請求項13に記載のデバイス。

【請求項19】

前記ビデオエンコーダは、前記第1のサイズの前記ブロックの前記決定されたパーティションパターンを識別する前記構成されたパーティションパターンリストへの前記インデ50

ックスをシグナリングするように構成され、ビデオデコーダが、前記深度ブロックをイントロ予測復号するために前記インデックスを使用する、請求項 18 に記載のデバイス。

【請求項 20】

前記ビデオコーダはビデオエンコーダを備え、

前記ビデオエンコーダは、前記深度ブロック内のサブブロックを識別するように構成され、前記サブブロックのサイズが前記第 1 のサイズであり、

前記第 1 のサイズの前記ブロックの前記パーティションパターンを決定するために、前記ビデオエンコーダは、前記識別されたサブブロックに関する前記パーティションパターンを決定するように構成され、

前記第 2 のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオエンコーダは、前記識別されたサブブロックに関する前記決定されたパーティションに基づいて、前記第 2 のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、請求項 13 に記載のデバイス。 10

【請求項 21】

前記第 1 のサイズの前記ブロックの前記パーティションパターンは、前記第 1 のサイズの前記ブロックの複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2 つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、請求項 13 に記載のデバイス。

【請求項 22】

集積回路 (I C)、マイクロコントローラ、ワイヤレスデバイス、デスクトップコンピュータ、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、電話ハンドセット、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲーム機、またはビデオストリーミングデバイスのうちの少なくとも 1 つを備える、請求項 13 に記載のデバイス。 20

【請求項 23】

ビデオコーディングのためのデバイスの 1 つまたは複数のプロセッサによって実行されたときに、前記 1 つまたは複数のプロセッサに、

第 1 のサイズのブロックの 1 つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと、 30

前記パーティションパターンリストに基づいて、前記第 1 のサイズの前記ブロックのパーティションパターンを決定することと、ここにおいて、前記パーティションパターンリスト中の前記パーティションパターンは、前記パーティションパターンリスト中のインデックスによって識別される、

前記第 1 のサイズの前記ブロックの前記決定されたパーティションパターンに基づいて、第 2 のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第 2 のサイズが前記第 1 のサイズよりも大きく、前記深度ブロックは、3 次元 (3 D) ビデオコーディングのフレームワークにおける深度マップのブロックである、

前記第 1 のサイズの前記ブロックの前記決定されたパーティションパターンを使用して前記第 1 のサイズの前記ブロックをイントロ予測コーディングすることなく、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントロ予測コーディングすることと 40

を行わせる命令を記憶した非一時的コンピュータ可読記憶媒体。

【請求項 24】

前記第 1 のサイズの前記ブロックの前記パーティションパターンは、サイズ 16×16 のブロックに関するパーティションパターンを備え、前記 1 つまたは複数のプロセッサに、前記第 2 のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを行わせる前記命令は、前記 1 つまたは複数のプロセッサに、サイズ 16×16 の前記ブロックに関する前記決定されたパーティションパターンに基づいて、サイズ 32×3 50

2 の前記深度ブロックに関する前記パーティションパターンを決定することを行わせる命令を備える、請求項 2 3 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 2 5】

前記第 1 のサイズのブロックの前記パーティションパターンは、前記第 1 のサイズの前記ブロックの複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2 つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、請求項 2 3 に記載の非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

10

【0 0 0 1】

[0001] 本出願は、その内容全体が参照によって本明細書に組み込まれている、2 0 1 3 年 1 0 月 1 5 日に出願された米国仮出願第 6 1 / 8 9 1 , 2 0 3 号の利益を主張する。

【0 0 0 2】

[0002] 本開示は、ビデオコーディングに関する。

【背景技術】

【0 0 0 3】

[0003] デジタルビデオ能力は、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、ワイアレスブロードキャストシステム、携帯情報端末（P D A）、ラップトップコンピュータもしく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、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲームデバイス、ビデオゲーム機、セルラー電話機もしくは衛星無線電話機、いわゆる「スマートフォン」、ビデオ会議デバイス、ビデオストリーミングデバイスなどを含む広範囲のデバイスに組み込まれ得る。デジタルビデオデバイスは、M P E G - 2 、M P E G - 4 、I T U - T H . 2 6 3 、I T U - T H . 2 6 4 / M P E G - 4 , P a r t 1 0 , アドバンストビデオコーディング（A V C）、現在開発中の高効率ビデオコーディング（H E V C）規格によって定義された規格、およびそのような規格の拡張に記載されているもののような、ビデオ圧縮技法を実装する。これらのビデオデバイスは、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し、受信し、符号化し、復号し、および／または記憶することができる。

20

【0 0 0 4】

[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または除去するために、空間的（ピクチャ内）予測および／または時間的（ピクチャ間）予測を実行する。ブロックベースのビデオコーディングの場合、ビデオスライス（すなわち、ピクチャまたはピクチャの一部分）は、ツリーブロック、コーディングユニット（C U）および／またはコーディングノードと呼ばれることがあるビデオブロックに区分され得る。ピクチャのイントラコーディングされた（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中の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フォーマンス（P または B）スライスの中のビデオブロックは、同じピクチャの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、また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。

30

【0 0 0 5】

[0005] 空間的予測または時間的予測は、コーディングされるべきブロックの予測ブロックをもたらす。残差データは、コーディングされるべき元のブロックと予測ブロックとの間のピクセル差分を表す。インターフォーマンスされたブロックは、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、およびコーディングされた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。イントラコーディングされたブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは空間領域から変換領域に変換されて、残差変換係数が得られ得、その残差変換係数は、次いで量子化され得る。最初に 2 次元アレイで構

40

50

成される量子化変換係数は、変換係数の1次元ベクトルを生成するために走査されてよく、エントロピーコーディングが、さらなる圧縮を達成するために適用されてよい。

【発明の概要】

【0006】

[0006]本開示の技法は一般に、ビデオデータのイントラ予測されたブロックに関するパーティションパターンを、より小さいサイズのブロックに関連するパーティションパターンに基づいて導出することに関する。いくつかの例では、イントラ予測されたブロックは、深度マップのブロックであり得、パーティションパターンは、サブブロックを二等分する線形線によって分割された2つのパーティションにサブブロックを区分する線ベースのパーティションパターンであり得る。

10

【0007】

[0007]一例では、本開示は、ビデオデータを復号する方法について説明し、本方法は、第1のサイズのブロックに関するパーティションパターンを決定することと、第1のサイズのブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、第2のサイズが第1のサイズよりも大きい、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測復号することとを備える。

【0008】

[0008]一例では、本開示は、ビデオデータを符号化する方法について説明し、本方法は、第1のサイズのブロックに関するパーティションパターンを決定することと、第1のサイズのブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、第2のサイズが第1のサイズよりも大きい、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測符号化することとを備える。

20

【0009】

[0009]一例では、本開示は、ビデオコーディングのためのデバイスについて説明し、本デバイスは、第1のサイズのブロックに関する1つまたは複数のパーティションパターンを記憶するビデオデータメモリと、1つまたは複数のプロセッサを備えるビデオコーダとを備える。ビデオコーダは、記憶された1つまたは複数のパーティションパターンから、第1のサイズのブロックに関するパーティションパターンを決定することと、第1のサイズのブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、第2のサイズが第1のサイズよりも大きい、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測コーディングすることを行なうように構成され得る。

30

【0010】

[0010]一例では、本開示は、ビデオコーディングのためのデバイスの1つまたは複数のプロセッサによって実行されたときに、1つまたは複数のプロセッサに、第1のサイズのブロックに関するパーティションパターンを決定することと、第1のサイズのブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、第2のサイズが第1のサイズよりも大きい、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測コーディングすることを行わせ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。

40

【0011】

[0011]本開示の1つまたは複数の例の詳細は、添付図面および下記の説明に記載される。他の特徴、目的、および利点は、その説明および図面、ならびに特許請求の範囲から明らかになろう。

【図面の簡単な説明】

【0012】

50

【図1】[0012]深度コーディングのための本開示の技法を利用し得る例示的なビデオ符号化および復号システム10を示すプロック図。

【図2】[0013]方向性イントラ予測モードに関連する予測方向を概して示す図。

【図3A】[0014]深度モデリングモード(DMM)の例を示す概念図。

【図3B】深度モデリングモード(DMM)の例を示す概念図。

【図4A】[0015]パーティションパターンの例を示す概念図。

【図4B】パーティションパターンの例を示す概念図。

【図4C】パーティションパターンの例を示す概念図。

【図4D】パーティションパターンの例を示す概念図。

【図4E】パーティションパターンの例を示す概念図。

【図4F】パーティションパターンの例を示す概念図。

【図5A】[0016]1/2サンプル精度のパーティションパターンサンプルのマッピングを示す概念図。

【図5B】1/2サンプル精度のパーティションパターンサンプルのマッピングを示す概念図。

【図6】[0017]深度コーディングのための技法を実装し得るビデオエンコーダの一例を示すプロック図。

【図7】[0018]深度コーディングのための技法を実装し得るビデオデコーダの一例を示すプロック図。

【図8】[0019]ビデオデータを復号する例示的な方法を示すフローチャート。

【図9】[0020]ビデオデータを符号化する例示的な方法を示すフローチャート。

【図10】[0021]領域境界チェーンコーディングモードを示す概念図。

【発明を実施するための形態】

【0013】

[0022]たとえば3次元(3D)-HEVC(高効率ビデオコーディング)規格による3Dビデオコーディングでは、(テクスチャビューのテクスチャコンポーネントとも呼ばれる)キャプチャされたビデオデータが深度ビューの対応する深度マップに関連付けられる、マルチビュービデオ+深度フォーマットを使用して、3Dビデオデータが表される。たとえば、テクスチャビューのテクスチャコンポーネントは、実際のビデオコンテンツを表し、テクスチャコンポーネントに対応する深度コンポーネントは、テクスチャコンポーネント中のピクセルの相対深度を表す深度マップを提供する。ビデオエンコーダは、テクスチャコンポーネントと深度マップとを符号化し、テクスチャコンポーネントと深度マップとを3Dビデオビットストリーム中に多重化するように構成される。ビデオデコーダは、3Dビデオビットストリームを受信し、3Dビデオデータを再構成するためにテクスチャコンポーネントと深度マップとを復号する。

【0014】

[0023]ビデオエンコーダのビデオ符号化能力とビデオデコーダの復号能力とを活用するために、ルーマサンプルが深度値を表すグレースケールビデオとして深度マップが形成される。このようにして、深度値を符号化および復号するための何らかの追加の特殊な符号化および復号の技法を利用する必要なく、ビデオエンコーダおよびビデオデコーダの既存の能力を使用して、深度値が符号化され復号され得る。

【0015】

[0024]たとえば、ビデオのピクチャをコーディングするために、ビデオエンコーダは、インター予測符号化および/またはイントラ予測符号化を利用するように構成され得、ビデオデコーダは、インター予測復号および/またはイントラ予測復号を利用するように構成され得る。ビデオエンコーダおよびビデオデコーダは、同様にそれぞれ、深度マップをコーディングするためのインター予測およびイントラ予測の符号化および復号の技法を使用することができます。

【0016】

[0025]場合によっては、深度マップは、シャープエッジと定数領域(constant area)

10

20

30

40

50

とを含み、シャープエッジは、エッジの一方の側のルーマ値とエッジの他方の側のルーマ値との間に比較的大きい差があるときに発生する。深度マップサンプルのそのような異なる統計値（たとえば、ルーマ値）に起因して、2次元（2D）ビデオコーデックに基づいて深度マップのために設計された異なるコーディング方式があり得る。たとえば、2次元ビデオコーディングの場合、深度マップがないことがある。一方、深度マップを含む3Dビデオコーディングの場合、深度マップを符号化および復号するために追加のビデオコーディング技法が有用であり得る。

【0017】

[0026]たとえば、3D - HEVC規格は、HEVC規格において定義されるような2Dビデオコーディングに関する概念を3Dビデオコーディングに拡張する。10 3D - HEVC規格は、イントラ予測符号化および復号のためにHEVC規格において定義されるイントラ予測モードを使用する。さらに、3D - HEVC規格は、深度マップのスライスのイントラ予測ユニットを符号化または復号する（すなわち、深度スライスの予測ユニット（P U）をイントラ予測する）ために、HEVCイントラ予測モードとともに深度モデリングモード（DMM）を導入した。

【0018】

[0027]DMMでは、深度マップのブロックが2つの領域に区分され、各領域が定数値によって表される。ビデオエンコーダは、深度マップのブロックをイントラ予測するためにビデオエンコーダが使用する領域の各々に関する予測値を決定する。ビデオエンコーダは、領域の各々に関する予測値をビデオデコーダにシグナリングすることもでき、またはビデオデコーダは、ビデオエンコーダからの明示的なシグナリングなしで予測値を決定するように構成され得る。いずれの場合も、ビデオデコーダは、深度マップのブロックをイントラ予測するために予測値を利用することができる。20

【0019】

[0028]（深度ブロックと呼ばれる）深度マップのブロックを区分する方法のうちの1つは、ウェッジレット区分と呼ばれる。ウェッジレット区分では、ビデオエンコーダは、深度ブロックを二等分して2つの領域を生成する線形線を決定する。このようにして、ウェッジレット区分は、線ベースの区分と考えられることがあり、いくつかの例では、非長方形パーティションを形成し得る（ただし、長方形パーティションを形成する可能性もある）。線形線は、深度ブロックの1つの側にある点から開始し、深度ブロックの反対側または直交側にある点で終了することがある。一例として、一方位では、線形線は、深度ブロックの左側にある点から開始し、深度ブロックの上側にある点で終了することがある。別の方位では、線形線は、深度ブロックの上側にある点から開始し、深度ブロックの下側にある点で終了することがある。30

【0020】

[0029]ウェッジレットパターンは、深度マップのブロックが二等分線形線により2つの領域に区分され得る1つの方法を指し、ブロックのために存在し得るウェッジレットパターンの数は、ブロックサイズの関数であり得る。たとえば、所与の解像度（たとえば、1ピクセル、1/2ピクセル、または1/4ピクセル）で、より小さいサイズのブロック（たとえば、4×4ブロック）の場合、より大きいサイズのブロック（たとえば、64×64）と比較して、ブロックの側に沿ってより少ない点がある。したがって、より大きいサイズのブロックと比較して、より小さいサイズのブロックの各側に沿ってより少ない開始点および終了点があり、より少ないウェッジレットパターンが生じる。40

【0021】

[0030]DMMモードのうちの1つが利用される例では、ビデオエンコーダは、深度ブロックに関するウェッジレットパターンを決定し、決定されたウェッジレットパターンに基づいてウェッジレットパターンをイントラ予測することができる。ビデオデコーダは、ブロックを復号するために逆のプロセスを実行するように構成され得るので、ビデオデコーダは、ビデオエンコーダが決定したウェッジレットパターンと同じウェッジレットパターンを決定することができる。たとえば、いくつかの例では、ビデオエンコーダおよびビデ50

オデコーダはそれぞれ、ウェッジレットパターンのリストを記憶することができる。ビデオエンコーダは、ビデオピットストリームにおいて、決定されたウェッジレットパターンを識別するウェッジレットパターンのリストへのインデックスをシグナリングすることができる。次いでビデオデコーダは、ウェッジレットパターンのリストのシグナリングされたインデックスに基づいて、ビデオエンコーダと同じウェッジレットパターンを決定することができる。別の例として、ビデオエンコーダは、深度ブロックに関するウェッジレットパターンを、対応するテクスチャブロックのビデオコンテンツ特性から決定することができる。ビデオデコーダは、ビデオデコーダが決定するウェッジレットパターンおよびビデオエンコーダが決定するウェッジレットパターンが同じウェッジレットパターンになるように、ウェッジレットパターンを決定するためのビデオエンコーダと同じ技法を実施するように構成され得る。

10

【0022】

[0031]ウェッジレットパターンを決定する上記の例では、ビデオエンコーダおよびビデオデコーダは、すべてのブロックサイズに関するすべてのウェッジレットパターンについての情報を記憶するように構成され得る。しかしながら、より大きいブロックサイズの場合、比較的多数のウェッジレットパターンを記憶するために必要なメモリの量は、望ましくないほど大きいことがある。たとえば、3D - H E V C 規格のいくつかのドラフトは、 64×64 サイズの深度ブロックに関するウェッジレットパターンをまったく持たないことを提案した。言い換れば、 64×64 サイズの深度ブロックに関するウェッジレットパターンを記憶するための大きいサイズのメモリを持つ不利益は、 64×64 サイズの深度ブロックをイントラ予測するためにウェッジレットパターンを使用する利益を圧倒すると考えられた。

20

【0023】

[0032]本開示で説明する技法では、ビデオコーダ（たとえば、ビデオエンコーダまたはビデオデコーダ）は、パーティションパターン（たとえば、ウェッジレットパターン）を、第1のサイズのブロックのパーティションパターンから決定し（たとえば、第1のサイズのブロックに関するパーティションパターンを決定し）、第1のサイズのブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することができ、第1のサイズが第2のサイズよりも小さい。ビデオコーダは、決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックをイントラ予測コーディングする（たとえば、符号化または復号する）ことができる。いくつかの例では、ビデオコーダは、第2のサイズの深度ブロックサブブロック（たとえば、深度ブロック内のブロック）のパーティションパターンからのパーティションパターンを利用することができる。一例として、第2のサイズの深度ブロックは $N \times N$ ブロックであり得、サブブロックは $M \times M$ ブロックであり得、 M が N よりも小さく、 $N \times N$ ブロック内にある。

30

【0024】

[0033]いくつかの例では、特定のブロックサイズに関して記憶されるためにビデオコーダが必要とするパーティションパターンの数は、特定のブロックサイズの深度ブロックに関するパーティションパターンが、より小さいサイズのブロックに関するパーティションパターンから決定される場合には（すなわち、それに基づいて）減らされ得る。たとえば、ビデオコーダは、より小さいサイズのブロックに関するパーティションパターンを記憶し、そして後に、より小さいサイズのブロックの記憶されたパーティションパターンから、より大きいサイズのブロックに関するパーティションパターンを決定することができる。

40

【0025】

[0034]たとえば、以前にビデオコーダが第1のサイズの深度ブロックに関する Y 個のパーティションパターンと、第2のサイズの深度ブロックに関する X 個のパーティションパターンとを記憶しており、第1のサイズが第2のサイズよりも小さいと仮定する。この例では、ビデオコーダは、第2のサイズのブロックに関するパーティションパターンを決定

50

するためにY個のパーティションパターンを使用することができるので、ビデオコーダは、第2のサイズの深度ロックに関するX個のパーティションパターンよりも少ないパーティションパターンを記憶する必要があり得る。

【0026】

[0035]第1のサイズのロックに関するパーティションパターンからは決定できない第2のサイズのロックに関連するパーティションパターンがいくつあることも考えられ得る。したがって、いくつかの例では、ビデオコーダは依然として、第2のサイズの深度ロックに関するいくつかの（たとえば、少なくとも1つの）パーティションパターンを記憶し得るが、記憶される必要のあるパーティションパターンの数は、従来の技法と比較して減らされ得る。いくつかの例では、ビデオコーダは、第2の、より大きいサイズのロックに関するパーティションパターンをまったく記憶せず、代わりに、第1の、より小さいサイズのロックに関するパーティションパターンから、第2の、より大きいサイズのロックに関するパーティションパターンを決定することがある。10

【0027】

[0036]図1は、深度コーディングのための本開示の技法を利用し得る例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示すように、システム10は、宛先デバイス14によって後で復号されるべき符号化ビデオデータを提供するソースデバイス12を含む。特に、ソースデバイス12は、コンピュータ可読媒体16を介してビデオデータを宛先デバイス14に提供する。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック（すなわち、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、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲーム機、ビデオストリーミングデバイスなどを含む、広範囲にわたるデバイスのいずれかを備え得る。場合によっては、ソースデバイス12および宛先デバイス14は、ワイヤレス通信に対する機能を備え得る。20

【0028】

[0037]宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化ビデオデータを受信することができる。コンピュータ可読媒体16は、ソースデバイス12から宛先デバイス14に符号化ビデオデータを移動することができる、任意のタイプの媒体またはデバイスを備え得る。一例では、コンピュータ可読媒体16は、ソースデバイス12が符号化ビデオデータを宛先デバイス14にリアルタイムで直接送信するのを可能にするための通信媒体を備え得る。30

【0029】

[0038]符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信標準規格に従って変調され、宛先デバイス14に送信され得る。通信媒体は、無線周波（RF）スペクトルまたは1つもしくは複数の物理伝送線路など、任意のワイヤレスまたはワイヤード通信媒体を備え得る。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、またはインターネットなどのグローバルネットワークなどのパケットベースのネットワークの一部を形成し得る。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス12から宛先デバイス14への通信を容易にするために有用であり得る任意の他の機器を含み得る。40

【0030】

[0039]いくつかの例では、符号化データは、出力インターフェース22からストレージデバイスへ出力され得る。同様に、符号化データは、ストレージデバイスから入力インターフェースによってアクセスされ得る。ストレージデバイスは、ハードドライブ、Blu-rayディスク、DVD、CD-ROM、フラッシュメモリ、揮発性もしくは不揮発性のメモリ、または符号化ビデオデータを記憶するための任意の他の適切なデジタル記憶媒体など、様々な分散されたまたはローカルにアクセスされるデータ記憶媒体のいずれかを含み得る。さらなる例では、ストレージデバイスは、ソースデバイス12によって生成された符号化ビデオを記憶することができるファイルサーバまたは別の中間ストレージデ50

バイスに対応し得る。

【0031】

[0040]宛先デバイス14は、ストリーミングまたはダウンロードを介して、ストレージデバイスから記憶されたビデオデータにアクセスすることができる。ファイルサーバは、符号化ビデオデータを記憶でき、符号化ビデオデータを宛先デバイス14に送信できる、任意のタイプのサーバとすることができます。例示的なファイルサーバは、(たとえば、ウェブサイト用の)ウェブサーバ、FTPサーバ、ネットワークアタッチドストレージ(NAS)デバイス、またはローカルディスクドライブを含む。宛先デバイス14は、インターネット接続を含む、任意の標準データ接続を介して、符号化ビデオデータにアクセスすることができる。これは、ファイルサーバ上に記憶された符号化ビデオデータにアクセスするのに適した、ワイヤレスチャネル(たとえば、Wi-Fi(登録商標)接続)、ワイヤード接続(たとえば、DSL、ケーブルモ뎀など)、またはその両方の組合せを含み得る。ストレージデバイス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、またはその組合せとすることができます。10

【0032】

[0041]本開示の技法は、ワイヤレス応用またはワイヤレス設定に必ずしも制限されない。本技法は、無線テレビジョンプロードキャスト、ケーブルテレビジョン送信、衛星テレビジョン送信、HTTP上の動的適応ストリーミング(DASH:dynamic adaptive streaming over HTTP)などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されたデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の応用例など、様々なマルチメディア応用のいずれかをサポートするビデオコーディングに適用され得る。いくつかの例では、システム10は、ビデオストリーミング、ビデオ再生、ビデオプロードキャスト、および/またはビデオ電話などの応用をサポートするために一方向また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。20

【0033】

[0042]図1の例では、ソースデバイス12は、ビデオソース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。宛先デバイス14は、入力インターフェース28と、ビデオデコーダ30と、ディスプレイデバイス32とを含む。本開示によれば、ソースデバイス12のビデオエンコーダ20は、マルチビューコーディングにおける動きベクトル予測のための技法を適用するように構成され得る。他の例では、ソースデバイスおよび宛先デバイスは他の構成要素または構成を含み得る。たとえば、ソースデバイス12は、外部カメラなどの外部のビデオソース18からビデオデータを受信し得る。同様に、宛先デバイス14は、集積ディスプレイデバイスを含むのではなく、外部のディスプレイデバイスとインターフェースしてもよい。30

【0034】

[0043]図1の図示のシステム10は一例にすぎない。本開示による技法は、任意のデジタルビデオ符号化および/または復号デバイスによって実行され得る。概して、本開示の技法はビデオ符号化デバイスおよびビデオ復号デバイスによって実行されるが、本技法は、一般に「コーデック」と呼ばれるビデオエンコーダ/デコーダによって実行されてもよい。その上、本開示の技法は、ビデオプリプロセッサによって実行されてもよい。ソースデバイス12および宛先デバイス14は、宛先デバイス14に送信するためのコーディングされたビデオデータをソースデバイス12が生成するコーディングデバイスの例にすぎない。いくつかの例では、デバイス12、14は、デバイス12、14の各々がビデオ符号化構成要素とビデオ復号構成要素とを含むように、実質的に対称の形で動作することができる。したがって、システム10は、たとえば、ビデオストリーミング、ビデオ再生、ビデオプロードキャスト、またはビデオ電話のためのビデオデバイス12、14の間の一方向または双方向のビデオ送信をサポートし得る。40

【0035】

[0044]ソースデバイス12のビデオソース18は、ビデオカメラなどのビデオキャプチャデバイス、前にキャプチャされたビデオを含んでいるビデオアーカイブ、および/また50

はビデオコンテンツプロバイダからビデオを受信するためのビデオフィードインターフェースを含み得る。さらなる代替案として、ビデオソース 18 は、ソースビデオとしてコンピュータグラフィックスベースのデータまたは、生ビデオと、アーカイブされたビデオと、コンピュータ生成ビデオとの組合せを生成することができる。場合によっては、ビデオソース 18 がビデオカメラである場合、ソースデバイス 12 および宛先デバイス 14 は、いわゆるカメラ付き電話またはビデオ付き電話を形成し得る。しかしながら、上記で述べたように、本開示で説明する技法は、一般にビデオコーディングに適用可能であり、ワイアレスおよび／またはワイヤードの応用例に適用され得る。各場合において、キャプチャされたビデオ、前にキャプチャされたビデオ、またはコンピュータ生成ビデオは、ビデオエンコーダ 20 によって符号化され得る。次いで、符号化ビデオ情報は、出力インターフェース 22 によってコンピュータ可読媒体 16 に出力され得る。10

【 0 0 3 6 】

[0045]コンピュータ可読媒体 16 は、ワイアレスブロードキャストまたはワイヤードネットワーク送信などの一時媒体、またはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、B l u - r a y (登録商標) ディスク、または他のコンピュータ可読媒体などの記憶媒体(すなわち、非一時的記憶媒体)を含み得る。いくつかの例では、ネットワークサーバ(図示せず)は、たとえば、ネットワーク送信を介して、ソースデバイス 12 から符号化ビデオデータを受信し、宛先デバイス 14 に符号化ビデオデータを提供することができる。同様に、ディスクスタンピング設備など、媒体製造設備のコンピューティングデバイスは、ソースデバイス 12 から符号化ビデオデータを受信し、その符号化ビデオデータを含んでいるディスクを生成することができる。したがって、様々な例では、コンピュータ可読媒体 16 は、様々な形態の 1 つまたは複数のコンピュータ可読媒体を含むと理解され得る。20

【 0 0 3 7 】

[0046]本開示では、概して、ビデオエンコーダ 20 が、ある情報をビデオデコーダ 30 などの別のデバイスに「シグナリング」することに言及することがある。ただし、ビデオエンコーダ 20 は、いくつかのシンタックス要素をビデオデータの様々な符号化部分に関連付けることによって情報をシグナリングし得ることを理解されたい。すなわち、ビデオエンコーダ 20 は、ビデオデータの様々な符号化部分のヘッダにいくつかのシンタックス要素を格納することによって、データを「シグナリング」することができる。場合によっては、そのようなシンタックス要素は、ビデオデコーダ 30 によって受信され、復号されるより前に、符号化され、記憶され(たとえば、コンピュータ可読媒体 16 に記憶され)得る。したがって、「シグナリング」という用語は全般に、圧縮されたビデオデータを復号するためのシンタックスまたは他のデータの通信を、そのような通信がリアルタイムで発生するか、またはほぼリアルタイムで発生するか、またはある期間にわたって発生するかにかかわらず指すことがあり、ある期間にわたる通信は、シンタックス要素を符号化の時点で媒体に記憶し、次いで、シンタックス要素がこの媒体に記憶された後の任意の時点で復号デバイスによって取り出され得るときに、発生し得る。30

【 0 0 3 8 】

[0047]宛先デバイス 14 の入力インターフェース 28 はコンピュータ可読媒体 16 から情報を受信する。コンピュータ可読媒体 16 の情報は、ビデオエンコーダ 20 によって定義され、またビデオデコーダ 30 によって使用される、ロックおよび他のコーディングされたユニット、たとえば、G O P の特性および／または処理を記述するシンタックス要素を含む、シンタックス情報を含み得る。ディスプレイデバイス 32 は、復号されたビデオデータをユーザに表示し、陰極線管(C R T)、液晶ディスプレイ(L C D)、プラズマディスプレイ、有機発光ダイオード(O L E D)ディスプレイ、または別のタイプのディスプレイデバイスなどの様々なディスプレイデバイスのうちのいずれかを備え得る。40

【 0 0 3 9 】

[0048]図 1 には示されないが、いくつかの態様では、ビデオエンコーダ 20 およびビデオデコーダ 30 は、それぞれオーディオエンコーダおよびオーディオデコーダと統合され50

てよく、共通のデータストリームまたは別個のデータストリーム中のオーディオとビデオの両方の符号化を処理するための、適切な MUX - DEMUX ユニット、または他のハードウェアおよびソフトウェアを含み得る。適用可能な場合、MUX - DEMUX ユニットは、ITU-H.223 マルチプレクサプロトコル、またはユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠し得る。

【0040】

[0049] ビデオエンコーダ 20 およびビデオデコーダ 30 はそれぞれ、適用可能なとき、1つまた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回路、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組合せのような、種々の好適なエンコーダまたはデコーダ回路のいずれかとして実装され得る。ビデオエンコーダ 20 およびビデオデコーダ 30 の各々は1つまたは複数のエンコーダまたはデコーダ中に含まれ得、そのいずれもが合成ビデオエンコーダ / デコーダ(コードック)の一部として組み込まれ得る。ビデオエンコーダ 20 および / またはビデオデコーダ 30 を含むデバイスは、集積回路、マイクロプロセッサ、および / または携帯電話のようなワイヤレス通信デバイスを備え得る。

【0041】

[0050] ビデオコーディング規格の一例としては、Joint Video Team(JVT)として知られる共同パートナーシップの成果として、ISO/IEC Moving Picture Experts Group(MPEG)とともにITU-T Video Coding Experts Group(VCEG)によって策定されたITU-T H.264/MPEG-4(AVC)規格がある。別のビデオコーディング規格は、そのスケーラブルビデオコーディング(SVC)およびマルチビュービデオコーディング(MVC)拡張を含む、H.264 規格は、ITU-T Study GroupによるITU-T勧告H.264、Advanced Video Coding for generic audiovisual servicesに記載されている。Joint Video Team(JVT)はH.264/MPEG-4 AVCへの拡張に取り組み続けている。MVCの最新のジョイントドラフトは、「Advanced video coding for generic audiovisual services」、ITU-T勧告H.264、2010年3月に記載されている。

【0042】

[0051] いくつかの例では、ビデオエンコーダ 20 およびビデオデコーダ 30 は、高効率ビデオコーディング(HEVC)規格およびHEVC規格の拡張に従って動作し得、HEVCテストモデル(HM)に準拠し得る。HEVCが、ITU-T Video Coding Experts Group(VCEG)とISO/IEC Motion Picture Experts Group(MPEG)とのJoint Collaboration Team on Video Coding(JCT-VC)によって開発された。HEVCの最近のドラフトは、http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Genewa/wg11/JCTVC-L1003-v14.zipから入手可能である。HEVC規格化の取組みは、HEVCテストモデル(HM)と呼ばれるビデオコーディングデバイスの発展的モデルに基づいていた。HMは、たとえばITU-T H.264 / AVCに従う既存のデバイスに対する、ビデオコーディングデバイスの複数の追加能力を仮定する。たとえば、H.264は9個のイントラ予測符号化モードを提供するが、HMは35個ものイントラ予測符号化モードを提供し得る。

【0043】

[0052] 概して、HMの作業モデルは、ビデオピクチャ(または「フレーム」)が、ルーマサンプルとクロマサンプルの両方を含む一連のツリープロックまたは最大コーディングユニット(LCU)に分割され得ることを記述する。ビットストリーム内のシンタックス

10

20

30

40

50

データが、ピクセルの数に関して最大のコーディングユニットであるLCUのサイズを定義し得る。スライスは、コーディング順序での、いくつかの連続するツリーブロックを含む。ピクチャは、1つまたは複数のスライスに区分され得る。各ツリーブロックは、4分木に従ってコーディングユニット(CU)に分割され得る。一般に、4分木データ構造は、1CUあたり1つのノードを含み、ルートノードがツリーブロックに対応する。CUが4つのサブCUに分割される場合には、CUに対応するノードは、4つのリーフノードを含み、リーフノードの各々がサブCUのうちの1つに対応する。

【0044】

[0053] 4分木データ構造の各ノードは、対応するCUのシンタックスデータを提供し得る。たとえば、4分木中のノードは、そのノードに対応するCUがサブCUに分割されるのかどうかを示す分割フラグを含むことができる。CUのシンタックス要素は、再帰的に定義され得、CUがサブCUに分割されるかどうかに依存することができる。CUがさらに分割されない場合、そのCUはリーフCUと呼ばれる。本開示では、元のリーフCUの明示的な分割が存在しない場合でも、リーフCUの4つのサブCUもリーフCUと呼ばれる。たとえば、 16×16 サイズのCUがさらに分割されない場合、4つの 8×8 サブCUも、 16×16 CUが分割されなくても、リーフCUと呼ばれる。

【0045】

[0054] CUは、CUがサイズの特異性を有しないことを別にすれば、H.264規格のマクロブロックと同様の目的を有する。たとえば、ツリーブロックは、(サブCUとも呼ばれる)4つの子ノードに分割されてよく、各子ノードは、今度は親ノードとなり、別の4つの子ノードに分割されてよい。4分木のリーフノードと呼ばれる、最後の分割されていない子ノードは、リーフCUとも呼ばれるコーディングノードを備える。コーディングされたビットストリームに関連するシンタックスデータは、最大CU深度と呼ばれる、ツリーブロックが分割され得る最大回数を定義することができ、コーディングノードの最小サイズを定義することもできる。それに応じて、ビットストリームは最小コーディングユニット(SCU)を定義することもできる。本開示は、「ブロック」という用語を、HEVCのコンテキストにおいてCU、PU、またはTUのうちのいずれか、または他の規格のコンテキストにおいて類似のデータ構造(たとえば、H.264/AVCのマクロブロックおよびそのサブブロック)を指すために使用する。

【0046】

[0055] CUは、コーディングノードと、コーディングノードに関連付けられた予測ユニット(PU)および変換ユニット(TU)とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状において正方形でなければならない。CUのサイズは、 8×8 ピクセルから、 64×64 ピクセル以上の最大値を有するツリーブロックのサイズまでに及び得る。各CUは、1つまたは複数のPUと1つまたは複数のTUとを包含し得る。CUに関連付けられたシンタックスデータは、たとえば、1つまたは複数のPUへのCUの区分を記述することができる。区分モードは、CUがスキップモード符号化もしくは直接モード符号化されるのか、イントラ予測モード符号化されるのか、またはインター予測モード符号化されるのかの間で異なる可能性がある。PUは、形状において非正方形に区分され得る。CUに関連付けられたシンタックスデータは、たとえば、4分木に従う1つまたは複数のTUへのCUの区分を記述することもできる。TUは、形状において正方形または非正方形(たとえば、長方形)とすることができます。

【0047】

[0056] HEVC規格は、CUごとに異なり得る、TUに従う変換を可能にする。TUは、通常、区分されたLCUについて定義される所与のCU内のPUのサイズに基づくサイズを与えられるが、必ずそうなっているとは限らない。TUは、通常、PU以下のサイズである。いくつかの例では、CUに対応する残差サンプルは、「残差4分木」(RQT)として知られている4分木構造を使用して、より小さいユニットに副分割され得る。RQTのリーフノードは、変換ユニット(TU)と呼ばれる場合がある。TUに関連付けられたピクセル差分値は、変換係数を生成するために変換され得、変換係数は量子化され得る

10

20

30

40

50

。

【0048】

[0057] リーフ C U は、1つまたは複数の予測ユニット (P U) を含むことができる。一般に、P U は、対応する C U のすべてまたは一部分に対応する空間的エリアを表し、P U の参照サンプルを取り出すためのデータを含むことができる。さらに、P U は、予測に関するデータを含む。たとえば、P U がイントラモード符号化 (たとえば、イントラ予測符号化) されるとき、P U のデータは、残差 4 分木 (R Q T) に含まれ得、この R Q T は、P U に対応する T U のイントラ予測モードを記述するデータを含むことができる。別の例として、P U がインター モード符号化 (たとえば、インター予測符号化) されるとき、P U は、P U の1つまたは複数の動きベクトルを定義するデータを含み得る。P U の動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度 (たとえば、1 / 4 ピクセル精度または 1 / 8 ピクセル精度) 、動きベクトルが指す参照ピクチャ、および / または動きベクトル用の参照ピクチャリスト (たとえば、RefPicList0 または RefPicList1) を記述することができる。10

【0049】

[0058] 1つまたは複数の P U を有するリーフ C U は、1つまたは複数の変換ユニット (T U) をも含むことができる。変換ユニットは、上述したように、(T U 4 分木構造とも呼ばれる) R Q T を使用して指定され得る。たとえば、分割フラグは、リーフ C U が4つの変換ユニットに分割されるのかどうかを示すことができる。その場合に、各変換ユニットは、さらなるサブ T U にさらに分割され得る。T U がさらに分割されないとき、その T U はリーフ T U と呼ばれることがある。一般に、イントラコーディングの場合、リーフ C U に属するすべてのリーフ T U は同じイントラ予測モードを共有する。すなわち、同じイントラ予測モードが、概して、リーフ C U のすべての T U の予測値を計算するために適用される。イントラコーディングの場合、ビデオエンコーダ 20 は、イントラ予測モードを使用して各リーフ T U の残差値を、T U に対応する C U の一部分と元のブロックとの間の差分として計算し得る。T U は、必ずしも P U のサイズに制限されるとは限らない。したがって、T U は、P U よりも大きくまたは小さくなり得る。イントラコーディング (たとえば、イントラ予測コーディング) の場合、P U は、同じ C U の対応するリーフ T U とコロケートされ得る。いくつかの例では、リーフ T U の最大サイズは、対応するリーフ C U のサイズに対応することができる。2030

【0050】

[0059] さらに、リーフ C U の T U は、残差 4 分木 (R Q T) と呼ばれるそれぞれの 4 分木データ構造にも関連付けられ得る。すなわち、リーフ C U は、そのリーフ C U が T U にどのように区分されるのかを示す 4 分木を含むことができる。T U 4 分木のルートノードは、一般に、リーフ C U に対応し、C U 4 分木のルートノードは、一般に、ツリープロック (または L C U) に対応する。分割されない R Q T の T U は、リーフ T U と呼ばれる。一般に、本開示は、別段に記載されていない限り、C U および T U という用語を、それぞれ、リーフ C U およびリーフ T U を指すために使用する。

【0051】

[0060] ビデオシーケンスは、一般に一連のピクチャを含む。本明細書で説明する「ピクチャ」および「フレーム」という用語は互換的に使用され得る。すなわち、ビデオデータを含んでいるピクチャは、ビデオフレームまたは単に「フレーム」と呼ばれることがある。ピクチャグループ (G O P) は、概して、ビデオピクチャのうちの一連の1つまたは複数を備える。G O P は、G O P 中に含まれるいくつかのピクチャを記述するシンタックスデータを、G O P のヘッダ中、ピクチャのうちの1つもしくは複数のヘッダ中、または他の場所に含み得る。ピクチャの各スライスは、それぞれのスライスのための符号化モードを記述するスライスシンタックスデータを含み得る。ビデオエンコーダ 20 は、一般に、ビデオデータを符号化するために、個々のビデオスライス内のビデオブロックに作用する。ビデオブロックは C U 内のコーディングノードに対応する場合がある。ビデオブロック4050

は、固定サイズまたは可変サイズを有することができ、指定されたコーディング規格に従ってサイズが異なり得る。

【0052】

[0061]例として、HMは、様々なPUサイズでの予測をサポートする。特定のCUのサイズが $2N \times 2N$ であると仮定すると、HMは、 $2N \times 2N$ または $N \times N$ のPUサイズでのイントラ予測と、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称PUサイズでのインター予測とをサポートする。HMは、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ のPUサイズでのインター予測のための非対称区分をもサポートする。非対称区分では、CUの一方の方向は、区分されず、他方の方向は、25%と75%とに区分される。CUのうちで25%パーティションに対応する部分は、「n」とそれに続く「Up(上)」、「Down(下)」、「Left(左)」、または「Right(右)」の表示とによって示される。したがって、たとえば、「 $2N \times nU$ 」は、上部で $2N \times 0.5N$ PU、および下部で $2N \times 1.5N$ PUに水平に区分される $2N \times 2N$ CUを指す。10

【0053】

[0062]本開示では、「 $N \times N (NxN)$ 」および「 $N \times N (N by N)$ 」は、垂直寸法および水平寸法の観点からビデオブロックのピクセル寸法、たとえば、 16×16 (16×16)ピクセルまたは 16×16 ($16 by 16$)ピクセルを指すために互換的に使用され得る。一般に、 16×16 ブロックは、垂直方向に16ピクセルを有し($y = 16$)、水平方向に16ピクセルを有する($x = 16$)。同様に、 $N \times N$ ブロックは、一般に、垂直方向にNピクセル、水平方向にNピクセルを有し、Nが非負の整数値を表す。ブロック中のピクセルは、行および列に配列され得る。さらに、ブロックは、必ずしも、水平方向において垂直方向と同じ数のピクセルを有するとは限らない。たとえば、ブロックは $N \times M$ ピクセルを備え得、この場合、Mは必ずしもNに等しいとは限らない。20

【0054】

[0063]CUのPUを使用したイントラ予測コーディングまたはインター予測コーディングに続いて、ビデオエンコーダ20は、CUのTUに関する残差データを計算することができる。PUは、(ピクセル領域とも呼ばれる)空間領域において予測ピクセルデータを生成する方法またはモードを記述するシンタックスデータを備えることができ、TUは、残差ビデオデータに対する変換、たとえば離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または概念的に同様の変換の適用に続く変換領域での係数を備えることができる。残差データは、符号化されていないピクチャのピクセルとPUに対応する予測値との間のピクセル差分に対応することができる。ビデオエンコーダ20は、CUに関する残差データを含むTUを形成し、次いで、CUに関する変換係数を生成するためにTUを変換することができる。30

【0055】

[0064]変換係数を生成するためのすべての変換に続いて、ビデオエンコーダ20は、変換係数の量子化を実行することができる。量子化は、一般に、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、変換係数が量子化され、さらなる圧縮を実現するプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。たとえば、nビット値は、量子化中にmビット値に切り捨てられ得、ここで、nはmよりも大きい。40

【0056】

[0065]量子化の後に、ビデオエンコーダ20は、変換係数を走査して、量子化変換係数を含む2次元行列から1次元ベクトルを生成し得る。走査は、アレイの前部により高いエネルギー(したがって、より低い周波数)係数を配置し、アレイの後部により低いエネルギー(したがって、より高い周波数)係数を配置するように設計され得る。

【0057】

[0066]いくつかの例では、ビデオエンコーダ20は、エントロピー符号化され得るシリアル化ベクトルを生成するために、量子化変換係数を走査するために所定の走査順序を利50

用し得る。他の例では、ビデオエンコーダ20は、適応走査を実行することができる。量子化変換係数を走査して1次元ベクトルを形成した後に、ビデオエンコーダ20は、たとえば、コンテキスト適応型可変長コーディング(CAVLC:context-adaptive variable length coding)、コンテキスト適応型バイナリ算術コーディング(CABAC:context-adaptive binary arithmetic coding)、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ディング(SBAC:syntax-based context-adaptive binary arithmetic coding)、確率間隔区分エントロピー(PIVE:Probability Interval Partitioning Entropy)コーディング、または別のエントロピー符号化方法に従って、1次元ベクトルをエントロピー符号化し得る。ビデオエンコーダ20は、ビデオデータを復号する際のビデオデコーダ30による使用のために、符号化ビデオデータに関連付けられたシンタックス要素をエントロピー符号化することもできる。

【0058】

[0067]ビデオエンコーダ20は、さらに、ブロックベースのシンタックスデータ、ピクチャベースのシンタックスデータ、およびGOPベースのシンタックスデータなどのシンタックスデータを、たとえば、ピクチャヘッダ、ブロックヘッダ、スライスヘッダ、またはGOPヘッダ内でビデオデコーダ30に送り得る。GOPシンタックスデータは、それぞれのGOP中のピクチャの数を記述し得、ピクチャシンタックスデータは、対応するピクチャを符号化するために使用される符号化／予測モードを示し得る。

【0059】

[0068]ビデオデコーダ30は、ビデオデータを復号し、ピクチャを再構成するために、ビデオエンコーダ20が符号化目的で利用したのとは逆の手順を一般に実行するように構成され得る。たとえば、ビデオデコーダ30は、シグナリングされたビットストリームからシンタックス要素とビデオデータとを受信することができ、ピクチャを再構成するためにビデオデータをイントラ予測復号および／またはインター予測復号するために、逆の動作を実行することができる。

【0060】

[0069]上記の説明は、ビデオエンコーダ20およびビデオデコーダ30がHEVC規格に基づいてビデオデータを符号化および復号することができる例示的な方法を提供している。本開示で説明する技法では、ビデオエンコーダ20およびビデオデコーダ30は、3次元(3D)ビデオ符号化および復号のために構成され得る。たとえば、ビデオエンコーダ20およびビデオデコーダ30は、HEVCビデオコーディング規格を活用する開発中のビデオコーディング規格を使用する3Dビデオ符号化および復号のために構成され得る。ただし、本開示で説明する技法は、そのように限定されず、他の3Dビデオ符号化および復号技法に拡張され得る。

【0061】

[0070]JCT-3Vでは、マルチビュー拡張(MV-HEVC)および3Dビデオ拡張(3D-HEVC)と呼ばれる、開発中の2つのHEVC拡張がある。3D-HEVCのための最新の参照ソフトウェア3D-HTMバージョン8.0が、2014年10月13日時点では以下のリンクからダウンロードされ得る。[3D-HTMバージョン8.0]:
https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-8.0/。最新のワーキングドラフト(文書番号:E1001(JCT3V-E1001))は、以下から入手できる。
http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip(ただし、このリンクは無効になる可能性がある)。JCT3V-E1001文書は、Sullivanらによる「3D-HEVC Draft Text 1」と題するものであり、同じく2014年10月13日時点ではhttp://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361から入手可能である。

【0062】

10

20

30

40

50

[0071] 一般に、3Dビデオコーディング技法を使用してコーディングされたビデオデータは、3次元効果を生成するためにレンダリングされ、表示され得る。たとえば、(たとえば、わずかに異なる水平位置を有する2つのカメラパースペクティブに対応する)異なるビューの2つの画像は、一方の画像が閲覧者の左眼によって見られ、他方の画像が閲覧者の右眼によって見られるように、実質的に同時に表示され得る。

【0063】

[0072] この3D効果は、たとえば、立体視(stereoscopic)ディスプレイまたは自動立体視(autostereoscopic)ディスプレイを使用して達成され得る。立体視ディスプレイは、2つの画像を相応にフィルタ処理するアイウェア(eyewear)とともに使用され得る。
たとえば、パッシブ眼鏡は、正しい眼が正しい画像を閲覧することを保証するために偏光レンズまたは異なる色のレンズを使用して画像をフィルタ処理し得る。アクティブ眼鏡は、別の例として、立体視ディスプレイと協調して交互のレンズを迅速に閉じ得、それにより、左眼画像を表示することと右眼画像を表示することとを交互に行い得る。自動立体視ディスプレイは、眼鏡が必要とされないような方法で2つの画像を表示する。たとえば、自動立体視ディスプレイは、各画像が閲覧者の適切な眼に投影されるように構成されたミラーまたはプリズムを含み得る。

10

【0064】

[0073] 3Dビデオコーディングでは、複数のビューが存在し、各ビューは、テクスチャピクチャおよび深度ピクチャ、または単にテクスチャコンポーネントおよび深度マップと呼ばれる複数のピクチャを含む。各テクスチャコンポーネントは、1つの深度マップに対応し得る。テクスチャコンポーネントは、画像コンテンツを含むことができ、対応する深度マップは、テクスチャ中のピクセルの相対深度を示す。実質的に同時に表示されるべき異なるビューのテクスチャコンポーネントは、同様の画像コンテンツを含むが、異なるビューのテクスチャ中のオブジェクト間に水平視差がある。テクスチャおよび深度マップについて以下により詳細に説明する。

20

【0065】

[0074] 3D-HEVCでは、アクセスユニットが、実質的に同時に表示されるべきテクスチャピクチャとそれらの対応する深度ピクチャとを含む。各ビュー中のテクスチャピクチャおよび深度ピクチャは、どのビューにピクチャが属するかを識別するための一意のビュー識別子(ビューid)またはビュー順序インデックスを有する。しかしながら、同じビューの深度ピクチャおよびテクスチャピクチャは、異なるレイヤ識別子(レイヤid)を有することがある。

30

【0066】

[0075] 本開示の技法は、テクスチャコンポーネントと深度マップとをコーディングすることによって3Dビデオデータをコーディングすることに関する。概して、「テクスチャ」という用語は、画像のルミナンス(輝度または「ルーマ」)値と画像のクロミナンス(色または「クロマ」)値とを表すために使用される。いくつかの例では、テクスチャ画像(すなわち、テクスチャピクチャ)は、1セットのルミナンスデータと、青色相(Cb)および赤色相(Cr)のための2セットのクロミナンスデータとを含む場合がある。4:2:2または4:2:0などの特定のクロマフォーマットでは、クロマデータは、ルーマデータに対してダウンサンプリングされる。言い換えれば、クロミナンスピクセルの空間解像度は、対応するルミナンスピクセルの空間解像度よりも低い(たとえば、ルミナンス解像度の1/2または1/4)ことがある。

40

【0067】

[0076] 深度データは、一般に、対応するテクスチャデータについての深度値を記述する。たとえば、深度画像(たとえば、深度ピクチャ)は、対応するテクスチャピクチャの対応するテクスチャデータについての深度をそれぞれ記述する深度ピクセルのセットを含む場合がある。深度データは、対応するテクスチャデータについての水平視差を決定するために使用され得る。したがって、テクスチャデータと深度データとを受信するデバイスは、一方のビュー(たとえば、左眼ビュー)のための第1のテクスチャピクチャを表示し、

50

深度値に基づいて決定された水平視差値だけ第1のピクチャのピクセル値をオフセットすることによって、他方のビュー（たとえば、右眼ビュー）のための第2のテクスチャピクチャを生成するように第1のテクスチャピクチャを変更するために深度データを使用することができる。一般に、水平視差（または単に「視差」）は、右ビュー中の対応するピクセルに対する第1のビュー中のピクセルの水平空間オフセットを表し、2つのピクセルは、2つのビュー中で表される同じオブジェクトの同じ部分に対応する。

【0068】

[0077]さらに他の例では、画像プレーンに直交するz次元におけるピクセルについて深度データが定義され得、その結果、画像について定義されたゼロ視差プレーンに対して、所与のピクセルに関連付けられた深度が定義される。そのような深度は、ピクセルを表示するための水平視差を作成するために使用され得、その結果、ピクセルは、ゼロ視差プレーンに対するピクセルのz次元深度値に応じて、左眼と右眼で異なるように表示される。ゼロ視差プレーンはビデオシーケンスの異なる部分に対して変化する場合があり、ゼロ視差プレーンに対する深度の量も変化する場合がある。ゼロ視差プレーン上に位置するピクセルは、左眼と右眼とに対して同様に定義され得る。ゼロ視差プレーンの前に位置するピクセルは、ピクセルが画像プレーンに直交するz方向の画像から出てくるように見える知覚を作成するように、（たとえば、水平視差を用いて）左眼と右眼とに対して異なるロケーションに表示され得る。ゼロ視差プレーンの後に位置するピクセルは、深度をわずかに知覚する程度に、わずかなぼかしとともに表示されてよく、または（たとえば、ゼロ視差プレーンの前に位置するピクセルの水平視差とは反対の水平視差を用いて）左眼と右眼とに対して異なるロケーションに表示され得る。他の多くの技法も、画像の深度データを伝達または定義するために使用され得る。10

【0069】

[0078]2次元ビデオデータは、概して、その各々が特定の時間インスタンスに対応する、個別ピクチャのシーケンスとしてコーディングされる。すなわち、各ピクチャは、シーケンス中の他の画像の再生時間に対する関連する再生時間を有する。これらのピクチャはテクスチャピクチャまたはテクスチャ画像と考えられ得る。深度ベースの3Dビデオコーディングでは、シーケンス中の各テクスチャピクチャは（深度マップとも呼ばれる）深度ピクチャにも対応し得る。すなわち、テクスチャピクチャに対応する深度マップは、対応するテクスチャピクチャのための深度データを表す。マルチビュービデオデータは、様々な異なるビューのためのデータを含んでよく、各ビューは、テクスチャピクチャと、対応する深度ピクチャとのそれぞれのシーケンスを含み得る。20

【0070】

[0079]上述したように、ピクチャは特定の時間インスタンスに対応し得る。ビデオデータは、アクセスユニットのシーケンスを使用して表され得、各アクセスユニットは、特定の時間インスタンスに対応するすべてのデータを含む。したがって、たとえば、マルチビュービデオデータ+深度の場合、共通時間インスタンスについての各ビューからのテクスチャピクチャ+テクスチャピクチャの各々についての深度マップはすべて、特定のアクセスユニット内に含まれ得る。アクセスユニットは、テクスチャピクチャに対応するテクスチャコンポーネントのためのデータと、深度マップに対応する深度コンポーネントのためのデータとを含み得る。30

【0071】

[0080]このようにして、3Dビデオデータは、キャプチャまたは生成されたビュー（テクスチャコンポーネント）が対応する深度マップに関連する、マルチビュービデオ+深度フォーマットを使用して表され得る。その上、3Dビデオコーディングでは、テクスチャコンポーネントおよび深度マップはコーディングされ、3Dビデオビットストリーム中に多重化され得る。深度マップはグレースケール画像としてコーディングされ得、深度マップの「ルーマ」サンプル（すなわち、ピクセル）は深度値を表す。一般に、深度データのブロック（深度マップのサンプルのブロック）は深度ブロックと呼ばれることがある。深度値は、深度サンプルに関連するルーマ値を指すことがある。40

【0072】

[0081] いずれの場合も、イントラコーディングおよびインターフォーマンス方法（たとえば、イントラ予測符号化および復号ならびにインター予測符号化および復号の技法）が深度マップコーディングに適用され得る。たとえば、上述したように、深度マップは、対応するテクスチャピクチャについての深度値を示すが、ビデオエンコーダ20およびビデオデコーダ30は、深度マップがグレースケール画像として形成されるので、ビデオコーディング技法を使用して深度マップを符号化および復号することができ、深度マップのルーマサンプルは、対応するテクスチャピクチャ中の対応するピクセルの相対深度を示す。

【0073】

[0082] 深度マップは、通常、シャープエッジと定数領域とを含み、深度マップ中のエッジは、一般に、対応するテクスチャピクチャの対応するテクスチャデータとの強い相関を提示する。テクスチャと対応する深度との間の異なる統計値および相関により、異なるコーディング方式が、2Dビデオコーデックに基づく深度マップのために設計されており、設計され続ける。たとえば、ベースHEVC規格において利用可能なものと比べて、深度マップをビデオコーディングするための、テクスチャと対応する深度との間の異なる統計値と相関とを活用する追加のビデオコーディング方式があり得る。10

【0074】

[0083] 一例として、現在のHEVC規格では、各予測ユニット(PU)のルーマ成分のためのイントラ予測技法は、(2から34までインデックス付けされた)33個の角度予測モードと、(1によりインデックス付けされた)DCモードと、(0によりインデックス付けされた)平面モードとを利用することができます。図2は、方向性イントラ予測モードに関連する予測方向を概して示している。たとえば、上述のように、HEVC規格は、平面モード(モード0)と、DCモード(モード1)と、33個の方向性予測モード(モード2~34)とを含む、35個のイントラ予測モードを含み得る。平面モードの場合、いわゆる「平面」関数を使用して予測が実行される。(たとえば、DC予測値を生成するための)DCモードの場合、ブロック内のピクセル値の平均化に基づいて予測が実行され得る。方向性予測モードの場合、(そのモードによって示される)特定の方向に沿った隣接ブロックの再構成されたピクセルに基づいて予測が実行される。概して、図2に示されている矢印の末端は、値がそこから取り出される隣接ピクセルのうちの相対的な1つを表し、矢印のヘッドは、予測ブロックを形成するために取り出された値が伝搬される方向を表す。20

【0075】

[0084] 3D-HEVC規格は、イントラ予測モードの、HEVC規格と同じ定義を使用する。さらに、3D-HEVCは、深度マップの深度スライスの深度ブロック(たとえば、予測ユニット)をイントラ予測符号化または復号するために、HEVCイントラ予測モードとともに深度モデリングモード(DMM)を導入した。DMMは、深度マップのインターフォーマンス方法(符号化または復号)のための深度マップにおけるシャープエッジの表現に、より適している。30

【0076】

[0085] 3D-HEVCワーキングドラフトのいくつかの先行バージョンは、4つのDMMモード、すなわち、モード1(明示的ウェッジレットシグナリング)と、モード2(イントラ予測ウェッジレット区分)と、モード3(コンポーネント間ウェッジレット区分)と、モード4(コンポーネント間輪郭区分)とを提供している。すべての4つのモードにおいて、ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、DMMパターンによって指定された2つの領域に深度ブロックを区分することができ、各領域が定数値によって表される。DMMパターンは、明示的にシグナリングされる(モード1)か、空間的に隣接するブロックによって予測される(モード2)か、またはコロケートテクスチャブロックを使用して予測される(モード3およびモード4)かのいずれかであり得る。40

【0077】

[0086] 3 D - H E V C ワーキングドラフトのいくつかのバージョンは、DMMモード2を除去し、DMMモード1、3および4を残した。3 D - H E V C ワーキングドラフトのいくつかのバージョンは、DMMモード3も除去し、DMMモード1および4を残した。

【 0 0 7 8 】

[0087] 言い換れば、DMMに新しいイントラ予測モードがある。これらのモードにおいて、ビデオエンコーダ20およびビデオデコーダ30は、(パーティションパターンと呼ばれる)DMMパターンによって指定された2つの領域に深度ブロックを区分するように構成され得、各領域が定数値によって表される。たとえば、ビデオエンコーダ20は、深度ブロックをイントラ予測符号化するために異なるパーティションパターン(パーティションパターンの例は上述している)を使用し、どのパーティションパターンが(たとえば、圧縮およびビデオ品質の点で)最適なコーディングをもたらしたかを決定するように構成され得る。その場合に、ビデオエンコーダ20は、決定されたパーティションパターンを使用して深度ブロックをイントラ予測符号化することができる。ビデオデコーダ30は、深度ブロックをイントラ予測復号するために、ビデオエンコーダ20とは逆のプロセスを実行するので、ビデオデコーダ30は、深度ブロックをイントラ予測復号するために、ビデオエンコーダ20が決定したのと同じパーティションパターンを決定するように構成され得る。10

【 0 0 7 9 】

[0088] ビデオエンコーダ20は、深度モデリングモードを示す情報をビデオデコーダ30にシグナリングすることができ、ビデオデコーダ30は、深度モデリングモードを示すシグナリングされた情報から、パーティションパターンを決定することができる。たとえば、深度モデリングモードが1(DMMモード1)であることを示す情報をビデオエンコーダ20がシグナリングした場合、ビデオデコーダ30は、深度ブロックに関するパーティションパターンを識別するビデオエンコーダ20からの情報のビットストリームをパスするように構成され得る。言い換れば、DMMモード1が使用されるべきであるとビデオエンコーダ20が決定した場合、ビデオエンコーダ20は、DMMモードが使用されることを示す情報を明示的にシグナリングし、深度ブロックに関するパーティションパターンを識別するためにビデオデコーダ30が使用する情報をシグナリングすることができる。このようにして、深度ブロックをイントラ予測コーディングするためにビデオエンコーダ20およびビデオデコーダ30が使用するパーティションパターン(たとえば、DMMパターン)は同じとなる。20

【 0 0 8 0 】

[0089] DMMモード3および4の場合、ビデオエンコーダ20は、DMMモードがモード3または4であることを示す情報をシグナリングし得るが、深度ブロックに関するパーティションパターンを識別する情報をシグナリングしないことがある。むしろ、ビデオデコーダ30は、対応するテクスチャピクチャ中のコロケートテクスチャブロックから、深度ブロックに関するパーティションパターンを決定するように構成され得る。ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、DMMモード3の場合に対応するテクスチャピクチャ中のコロケートテクスチャブロックからパーティションパターン(たとえば、DMMパターン)を決定するための同じプロセスを実施するように構成され得る。30

【 0 0 8 1 】

[0090] ウェッジレット区分と輪郭区分とを含む、DMMにおいて定義されている2つのタイプの区分モードがある。図3Aおよび図3Bは、深度モデリングモード(DMM)の例を示す概念図である。図3Aは、8×8ブロックに関するウェッジレットパターンの一例を示し、図3Bは、8×8ブロックに関する輪郭パターンの例上に示す。

【 0 0 8 2 】

[0091] 図3Aは、たとえば、ウェッジレット区分を使用して区分された深度ブロック150

10を示し、図3Bは、別の例として、輪郭区分を使用して区分された深度ブロック130を示す。3D-HEVCは、深度スライスのイントラ予測ユニットをコーディングするために、イントラ予測モードとともに、ブロックを区分するための深度モデリングモード(DMM)のための技法を含む。HTMバージョン3.1は、場合によっては深度マップ中のよりシャープなエッジをより良く表し得る、深度マップのイントラコーディングのためのDMM方法を適用する。

【0083】

[0092]ウェッジレット区分と輪郭区分とを含む、DMMにおいて定義されている2つの区分モデルがある。この場合も、図3Aはウェッジレット区分の一例を示しており、図3Bは輪郭区分の一例を示している。深度ブロック110および130内の各個々の正方形は、それぞれ、深度ブロック110および130のそれぞれの個々のピクセルを表す。正方形内の数値は、対応するピクセルが領域112(図3Aの例における値「0」)に属するのか、領域114(図3Aの例における値「1」)に属するのかを表す。また、図3Aにおいて、ピクセルが領域112(白い正方形)に属するのか、領域114(灰色の影つき正方形)に属するのかを示すために陰影が使用される。

【0084】

[0093]各パターン(すなわち、ウェッジレットと輪郭の両方)は、対応するサンプル(すなわち、ピクセル)が領域 P_1 に属するのか P_2 に属するのか(P_1 は図3A中の領域112と図3B中の領域132とに対応し、 P_2 は図3A中の領域114と図3B中の領域134A、134Bとに対応する)を標示するサイズ $u_B \times v_B$ 2進数字のアレイによって画定され得、 u_B および v_B はそれぞれ、現在のPUの水平サイズおよび垂直サイズを表す。図3Aおよび図3Bの例では、PUは、それぞれブロック110および130に対応する。ビデオエンコーダ20およびビデオデコーダ30などのビデオコーダは、コーディングの開始(たとえば、符号化の開始または復号の開始)時に、ウェッジレットパターンを初期化し得る。

【0085】

[0094]図3Aの例に示すように、ウェッジレット区分の場合、深度ブロック110は、(X_s, Y_s)に位置する開始点118と(X_e, Y_e)に位置する終了点120とをもつ直線116によって2つの領域、すなわち、領域112と領域114とに区分される。図3Aの例では、開始点118は点(8, 0)として定義され得、終了点120は点(0, 8)として定義され得る。

【0086】

[0095]図3Bの例に示すように、輪郭区分の場合、深度ブロック130などの深度ブロックは2つの不規則形状領域に区分され得る。図3Bの例では、深度ブロック130は領域132と領域134A、134Bとに区分される。領域134A中のピクセルは領域134B中のピクセルに直接隣接しないが、領域134Aおよび134Bは、深度ブロック130のPUを予測する目的で1つの単一の領域を形成するように画定される。輪郭区分は、ウェッジレット区分よりもフレキシブルであるが、シグナリングすることが相対的により困難であり得る。DMMモード4では、3D-HEVCの場合、輪郭区分パターンは、コロケートテクスチャブロックの再構成されたルーマサンプルを使用して暗黙的に導出される。

【0087】

[0096]このようにして、ビデオエンコーダ20およびビデオデコーダ30などのビデオコーダは、深度ブロック110のピクセルが(領域「P₁」と呼ばれることがある)領域112に属するのか、(領域「P₂」と呼ばれることがある)領域114に属するのかを決定するために、開始点118と終了点120とによって画定された線116を使用し得る。同様に、ビデオコーダは、深度ブロック130のピクセルが(領域「P₁」と呼ばれることがある)領域132に属するのか、(領域「P₂」と呼ばれることがある)領域134に属するのかを決定するために、図5Bの線136、138を使用し得る。領域「P₁」および「P₂」は、DMMに従って区分された異なる領域のためのデフォルト命名規

10

20

30

40

50

則であり、したがって、深度ブロック 110 の領域 P_1 は、深度ブロック 130 の領域 P_1 と同じ領域と考えられるべきでない。

【0088】

[0097] 上記のように、DMM の各々は、DMM がウェッジレット区分を使用するのかまたは輪郭区分を使用するのか、およびパターンが明示的にシグナリングされるのかまたは暗黙的に決定されるのかによって定義され得る。DMM プロセスは、(図 2 に示す) HEVCにおいて指定されたイントラ予測モードの代替として組み込まれ得る。DMM が適用されるのかまたは従来のイントラ予測が適用されるのかを指定するために、PUごとに 1 ビットフラグがシグナリングされ得る。

【0089】

[0098] 図 3 A および図 3 B に示すように、 $N \times N$ パーティションパターンは、 $N \times N$ 2 進ブロックを示す。本開示では、 $N \times N$ 2 進ブロックの位置 (i, j) における値は、位置 (i, j) におけるパーティションパターンのパーティション値と呼ばれ、 $i, j = 0, 1, \dots, N - 1$ である。 $N \times N$ パーティションパターンの位置ごとに、2 進数字値は、現在の位置のパーティション (0 または 1) を示す。

【0090】

[0099] たとえば、図 3 A は、線形線 (たとえば、直線 116) が深度ブロック 110 を二等分するウェッジレットパターンの一例を示している。ただし、多くの異なるウェッジレットパターンがあり得る。たとえば、図 3 A に示すような、(0, 8) から開始して (8, 0) において終了する線形線ではなく、別の例では、(1, 8) から開始して (8, 1) において終了する線形線も可能である。ウェッジレットパターンのさらに多くのそのような例があり得る。

【0091】

[0100] 一般に、ウェッジレットパターンの数は、ブロックサイズの関数であり得る。たとえば、より大きいサイズの深度ブロックは、より小さいサイズの深度ブロックよりも多くの開始点と終了点とを含み、これは、より大きいサイズの深度ブロックの場合にはより小さいサイズの深度ブロックの場合よりも多くの可能なウェッジレットパターンがあることを意味する。

【0092】

[0101] 初期化中、ビデオコーダ (たとえば、ビデオエンコーダ 20 およびビデオデコーダ 30) は、すべての利用可能なウェッジレットパターンを生成することができ、ウェッジレットパターンリストを構成する。この目的で、開始点位置および終了点位置のすべての可能な組合せのウェッジレットパターンが生成され、ビデオコーダは、コーディングプロセスの前にブロックサイズごとに開始点位置と終了点位置とをルックアップテーブルに記憶する。深度ブロックのイントラ予測にどのウェッジレットパターンを使用すべきかをビデオエンコーダ 20 が識別する例では、ビデオエンコーダ 20 は、ウェッジレットパターンのルックアップテーブルへのインデックスをシグナリングすることができ、このインデックスは、深度ブロックをイントラ予測符号化するためにビデオエンコーダ 20 が使用したウェッジレットパターンを識別する。ビデオデコーダ 30 は、初期化中にビデオデコーダ 30 が構成したウェッジレットパターンのルックアップテーブルへのインデックスを受信する。次いでビデオデコーダ 30 は、インデックスによって識別されたウェッジレットパターンを決定し、深度ブロックのイントラ予測復号にそのウェッジレットパターンを使用することができる。このようにして、ビデオエンコーダ 20 およびビデオデコーダ 30 は、それぞれ、イントラ予測符号化およびイントラ予測復号に同じウェッジレットパターンを利用することができる。

【0093】

[0102] 開始点位置および終了点位置の可能な組合せは、開始点位置と終了点位置とをつなぐパーティション境界線の方位に応じて、6 つのカテゴリーに分類され得る。たとえば、深度ブロックを二等分する線形線が、最上行から左列、最下行、または右列に延び得る。深度ブロックを二等分する線形線が、左列から最下行または右列に延び得る (最上行へ

10

20

30

40

50

の延びは、先行事例すでにカバーされている）。深度ブロックを二等分する線形線が、最下行から右列に延び得る（他は、先行事例すでにカバーされている）。この方法では、深度ブロックを二等分する線形線のための 6 つのカテゴリーがある。これらの 6 つのカテゴリーを以下の表 1 に記載する。

【0094】

【表 1】

表1:開始点と終了点とをつなぐパーティション境界線の方位

開始点位置	終了点位置	方位
最上行	左列	0
右列	最上行	1
最下行	右列	2
左列	最下行	3
最上行	最下行	4
右列	左列	5

10

【0095】

20

[0103] $N \times N$ ウェッジレットパーティションパターンを生成するにあたり、開始点 (x_S, y_S) および終了点 (x_E, y_E) 位置を仮定すると、ビデオエンコーダ 20 およびビデオデコーダ 30 は最初に、すべてのサンプルが 0 に初期化された一時的 $K \times K$ パーティションパターンを生成し、ここで、 $1/2$ サンプル ($1/2$ ピクセル) 精度の場合に K は $2N$ に等しく、他の場合に K は N に等しい。ビデオエンコーダ 20 およびビデオデコーダ 30 は、(x_S, y_S) と (x_E, y_E) とをつなぐパーティション境界線を形成するサンプルを 1 に設定し、一時的パーティションパターンを 2 つの部分（たとえば、図 4 に示すような部分 A および部分 B）として分割することができる。

【0096】

30

[0104] 図 4 A ~ 図 4 F は、一時的パーティションパターンの例を示す概念図である。たとえば、図 4 A ~ 図 4 F は、パーティション境界線（すなわち、深度ブロックを二等分する線形線）によって分割される 2 つの部分を示している。

【0097】

40

[0105] ビデオエンコーダ 20 およびビデオデコーダ 30 が一時的パーティションパターンを 2 つの部分として分割した後、ビデオエンコーダ 20 およびビデオデコーダ 30 は、パーティション 1 となる 2 つの部分のうちの 1 つを選択する。ビデオエンコーダ 20 およびビデオデコーダ 30 は、ウェッジレットパターン境界線の方位に基づいて、部分のうちのどちらがパーティション 1 となるかを選択するように構成され得る。ビデオエンコーダ 20 およびビデオデコーダ 30 は、同じ方法で部分のうちのどちらがパーティション 1 となるかを選択するように構成されるので、ビデオエンコーダ 20 およびビデオデコーダ 30 は、パーティション 1 となる同じ部分を選択する。また、他方の部分（すなわち、パーティション 1 ではないパーティション）はデフォルトによってパーティション 0 となるので、ビデオエンコーダ 20 およびビデオデコーダ 30 は、パーティション 0 となる同じ部分を選択する。

【0098】

50

[0106] 上述のように、表 1 によって示されるように、ウェッジレットパターンの 2 つのパーティションに深度ブロックを二等分する線形線の 6 つの方位があり得る。図 4 A ~ 図 4 F は、そのような方位の例を示している。図 4 A ~ 図 4 F はそれぞれ、それぞれの方位の一例を示しており、他の例もあり得ることを理解されたい。たとえば、図 4 B は、方位 1 の場合であり、上記の表 1 は、方位 1 が右列から最上行までであることを示している。

図4Bでは、線形線は(8, 5)から開始し、(4, 0)において終了する。方位1の別の例では、線形線は(8, 7)から開始し、(1, 0)において終了し得る。

【0099】

[0107]下記の表2は、ビデオエンコーダ20およびビデオデコーダ30がパーティション境界の方位に基づいて、どちらの部分がパーティション1となるかを選択する方法を示している。たとえば、ビデオエンコーダ20およびビデオデコーダ30は最初に、パーティション境界の方位(すなわち、深度ブロックを二等分する線形線の方位)を決定することができる。次いで、ビデオエンコーダ20およびビデオデコーダ30は、二等分線形線によって作成された2つのパーティションのうちのどちらのパーティションが1と識別されるべきか、またどちらのパーティションが0と識別されるべきかを決定することができる。10

【0100】

[0108]一例として、表2に示すように、方位パーティション境界が0である場合、ビデオエンコーダ20およびビデオデコーダ30は、1により識別されるパーティションとして部分Aを選択する。図4Aは、Aと標示されたパーティションが1と識別されることになり、Bと標示されたパーティションが0と識別されることになる一例を示している。ただし、図4Aは方位パーティション境界1の一例であり、方位パーティション境界1の他の例がある。ビデオエンコーダ20およびビデオデコーダ30は、表2に記載の基準に基づいて方位1~5の場合に、どちらのパーティションが1と識別され、どちらのパーティションが0と識別されるかを決定するための同様の機能を実行することができる。20

【0101】

【表2】

表2:パーティション1のサンプルの選択

パーティション境界の方位	1として選択されるサンプルの部分
0	A
1	B
2	B
3	A
4	xS+xE<K ? A : B
5	yS+yE<K ? A : B

【0102】

[0109]図4A~図4Fに示す例では、線形線は深度ブロック内のピクセル(たとえば、サンプル)から開始し、深度ブロック内のピクセル(たとえば、サンプル)において終了する。そのような例では、線形線は、フルサンプル精度を有すると考えられ得る。しかしながら、本開示で説明する技法は、そのように限定されない。たとえば、技法は、1/2サンプル精度または場合によっては1/4サンプル精度に拡張され得る。40

【0103】

[0110]1/2サンプル精度の場合、ビデオエンコーダ20およびビデオデコーダ30は、bPatternと呼ばれるN×Nパーティションパターンを、一時的K×K(K=2N)パーティションパターンbTempPatternのダウンサンプリングされたパーティションとして生成することができる。この例では、bPattern[i][j]=bTempPattern[m][n]であり、i, j=0, 1, ..., N-1、そしてm, n=0, 1, ..., 2N-1である。本開示では、一時的K×Kパーティションパターンは、開始位置(xS, yS)と終了位置(xE, yE)とを有するこのN×Nパーティションパターンの対応するウェッジレットパターンと呼ばれ、この対応するウェッジレ50

ットパターンは、開始位置 ($2 \times x S, 2 \times y S$) と終了位置 ($2 \times x E, 2 \times y E$) を有するサイズ $2 N \times 2 N$ を有する。1 / 2 サンプル精度のために、 $2 \times x S, 2 \times y S$ 、 $2 \times x E$ および $2 \times y E$ は整数であるが、 $x S, y S, x E$ 、および $y E$ は分数であり得る。 (i, j) と (m, n) との間のマッピングは、パーティション境界線の方位に依存する。技法は、1 / 4 サンプル精度の場合も同様に拡張され得る。

【0104】

[0111]以下では、1 / 2 サンプル精度の場合のパーティションパターンを生成する例について説明する。例示のために、例は、方位 0 および方位 1 に関して説明される。

【0105】

[0112]図 5 A および図 5 B は、1 / 2 サンプル精度のパーティションパターンサンプルのマッピングを示す概念図である。たとえば、図 5 A および図 5 B は、(たとえば、1 / 2 サンプル精度の) $2 N \times 2 N$ パーティションパターンサンプルから $N \times N$ パーティションパターンサンプルへのマッピングを示している。図 5 A は、開始点が最上行にあり、終了点が左列にある方位 0 の例を示している。図 5 B は、開始点が右列にあり、終了点が最上行にある方位 1 の例を示している。10

【0106】

[0113]図 5 A および図 5 B では、影つきブロックは、ダウンサンプリングされた $N \times N$ パーティションパターンのサンプルを示している。たとえば、図 5 A および図 5 B におけるブロックは、サイズが $2 N \times 2 N$ であり、図 5 A および図 5 B では、影つきサンプルは、 $2 N \times 2 N$ ブロックにおいて 1 つおきのサンプルを示し、結果的に $N \times N$ ブロックとなる。20

【0107】

[0114]いくつかの例では、方位 0 を有するウェッジレットパターンを生成するために、ビデオエンコーダ 20 およびビデオデコーダ 30 は、方位 0 を有するウェッジレットパターンのすべての可能な開始点と終了点とをカバーするように、開始点 ($(x S, y S)$) を $(0, 0)$ から $(2 N - 1, 0)$ にループし、終了点 ($(x E, y E)$) を $(0, 0)$ から $(0, 2 N - 1)$ にループすることができる。ビデオエンコーダ 20 およびビデオデコーダ 30 はまた、方位 0 の場合に $m = 2 i$ および $n = 2 j$ として、 (i, j) を (m, n) にマッピングすることができる。方位 1 の場合、ビデオエンコーダ 20 およびビデオデコーダ 30 は、方位 1 を有するウェッジレットパターンのすべての可能な開始点と終了点とをカバーするように、開始点 ($(x S, y S)$) を $(2 N - 1, 0)$ から $(2 N - 1, 2 N - 1)$ にループし、終了点 ($(x E, y E)$) を $(2 N - 1, 0)$ から $(0, 0)$ にループすることができる。ビデオエンコーダ 20 およびビデオデコーダ 30 はまた、方位 1 の場合に $m = 2 i + 1$ および $n = 2 j$ として、 (i, j) を (m, n) にマッピングすることができる。30

【0108】

[0115]他の方位の場合、ビデオエンコーダ 20 およびビデオデコーダ 30 は同様に、方位に基づいて開始点と終了点とをループすることができる。すべての方位のマッピングの式は、 $m = 2 i + \text{オフセット } X$ および $n = 2 j + \text{オフセット } Y$ と一般化され得、オフセット X 、オフセット Y は、表 3 に指定された移動値である。ビデオエンコーダ 20 およびビデオデコーダ 30 は、方位の各々のマッピングのために、式 = $m = 2 i + \text{オフセット } X$ および $n = 2 j + \text{オフセット } Y$ を実施することができ、オフセット X およびオフセット Y は表 3 によって定義されている。40

【0109】

【表3】

表3:パーティション境界線の方位に基づくオフセットXおよびオフセットYの設定

方位	オフセットX	オフセットY
0	0	0
1	1	0
2	1	1
3	0	1
4	$xS+xE < K ? 0 : 1$	0
5	0	$yS+yE < K ? 0 : 1$

10

【0110】

[0116] 上述のように、初期化中、ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、すべてのウェッジレットパターンの開始点および終了点（すなわち、方位ごとに2つのパーティションに深度ブロックを二等分する異なる可能な線形線の開始点および終了点）を含むウェッジレットパターンリストを構成することができる。場合によっては、2つのウェッジレットパターンが同じであり得る。ただし、ビデオエンコーダ20およびビデオデコーダ30は、ウェッジレットパターンリストが唯一の固有のパターンを含むように、重複するウェッジレットパターンをウェッジレットパターンリスト初期化プロセス中に除去することができる。

20

【0111】

[0117] ウェッジレットパターンリストを生成する際、ウェッジレットパターンを生成するためには、使用される開始位置および終了位置の解像度は、ブロックサイズに依存する。32×32ブロックの場合、可能な開始位置および終了位置は、2サンプルの精度を有するロケーションに制限される。16×16ブロックの場合、フルサンプル精度が使用され、4×4および8×8ブロックの場合、1/2サンプル精度が使用される。したがって、異なるブロックサイズで可能なウェッジレットパターンは異なり得る。一般に、可能なウェッジレットパターンの数は、ブロックのサイズに正比例する（すなわち、ブロックサイズが大きくなるほど、ウェッジレットパターンは多くなり、ブロックサイズが小さくなるほど、ウェッジレットパターンは少なくなる）。

30

【0112】

[0118] ウェッジレットパターンリスト初期化プロセス中に、重複するウェッジレットパターンをウェッジレットリストに追加するのを回避するために、ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、新たに生成されたウェッジレットパターンがリスト中の現在のウェッジレットパターンのいずれかと同じパターンを表していないときのみ、新たに生成されたウェッジレットパターンをウェッジレットリストの終わりに追加することができる。サイズN×Nの2つのウェッジレットパターン（すなわち、パターンA[i][j]、パターンB[i][j]、i, j = 0, 1, ..., N - 1）を比較するとき、[0, N - 1]の範囲におけるiおよびjのすべての可能な組合せにおいて、パターンA[i][j]が常にパターンB[i][j]に等しいか、またはパターンA[i][j]がパターンB[i][j]に等しいことが決してない場合、パターンAはパターンBと同じであると考えられる。

40

【0113】

[0119] 異なるブロックサイズと開始点位置および終了点位置の異なる精度とに起因して、以下の表4に記載のように、異なる数および異なるパターンのウェッジレットパターンが、異なるブロックサイズに利用可能である。より大きいブロックサイズの場合、ウェッジレットパターンの総数は格段に大きいことがあり、パターン生成の記憶要件および複雑性が高まり得る。

50

【0114】

【表4】

表4:DMMにおける利用可能なウェッジレットパターンの数

ブロック サイズ	利用可能な ウェッジレットパターンの数
4x4	86
8x8	782
16x16	1394
32x32	1503
64x64	なし

10

【0115】

[0120] ウェッジレットリスト初期化の詳細なプロセスが、ワーキングドラフト JCT3 V - E1001 の H . 8 . 4 . 4 . 2 . 1 2 、 H . 8 . 4 . 4 . 2 . 1 2 . 1 および H . 8 . 4 . 4 . 2 . 1 2 . 2 に指定されている。上述のように、JCT3V - E1001 文書は、Sullivan による「3D - HEVC Draft Text 1」と題するものであり、2014年10月13日時点では http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361 から入手可能である。

20

【0116】

[0121] ブロックサイズ依存型ウェッジレットパターン設計に関するいくつかの問題があり得る。たとえば、大きいブロックサイズの場合、利用可能なウェッジレットパターンの総数は、非常に大きいことがある。表4に示すように、16×16のブロックサイズの場合、ウェッジレットパターンの総数は1394であり、32×32のブロックサイズの場合、ウェッジレットパターンの総数は1503である。多数のウェッジレットパターンは、大規模メモリ要件をもたらし、これは望ましくないことがある。言い換えれば、ビデオエンコーダ20およびビデオデコーダ30は、ウェッジレットパターンのすべてを記憶するために大きいサイズのメモリを必要とし得る。また、ウェッジレットパターンリストを構成するために、特により大きいブロックサイズの場合には、比較的多数のウェッジレットパターンがあるので、ビデオエンコーダ20およびビデオデコーダ30は、ウェッジレットパターンを取り出すためにメモリユニットに対する望ましくない数のメモリ呼出しを必要とし得る。

30

【0117】

[0122] この理由（すなわち、多数のパターン）により、ウェッジレットパターンは64×64PUの場合にはサポートされない。たとえば、既存のブロックサイズ依存型ウェッジレットパターン設計では、深度ブロックのイントラ予測にウェッジレットパターンを使用することに関連する利益が、64×64サイズの深度ブロックに関するウェッジレットパターンの数の多さに圧倒され得る。したがって、64×64サイズのブロックに関するウェッジレットパターンに基づくイントラ予測に関連する利益がある場合でも、既存の技法は、64×64サイズのブロックに関するウェッジレットパターンに基づくそのようなイントラ予測をサポートしないことがある。

40

【0118】

[0123] 既存のブロックサイズ依存型ウェッジレットパターン設計に伴う問題の別の例として、3D - HEVC では、ウェッジレットパターンを生成するプロセスが、特にブロックがより大きいサイズを有するときに複雑であり得る。さらに、3D - HEVC は、ブロックサイズごとに、1セットのウェッジレットパターンが記憶されることを必要とし得る。しかしながら、64×64サイズのブロックの場合、ウェッジレットパターンを使用す

50

るイントラ予測はサポートされないので、現在のウェッジレットパターン設計では、 64×64 サイズのブロックに関してウェッジレットパターンが記憶される必要はない。

【 0 1 1 9 】

[0124]既存のブロックサイズ依存型ウェッジレットパターン設計に伴う問題のまた別の例として、ウェッジレットパターンリスト初期化中に、ビデオエンコーダ 20 およびビデオデコーダ 30 は、重複するウェッジレットパターンを追加するのを回避するために、2つのウェッジレットパターン間でかなりの数の比較計算を実行することを求められ得る。そのような比較計算は、計算の複雑性を高め、ビデオエンコーダ 20 またはビデオデコーダ 30 にとって望ましくない。10

【 0 1 2 0 】

[0125]本開示で説明する技法は、深度イントラコーディングのための簡略化された深度パーティションパターン生成に関する。以下の例では、深度イントラコーディングのためのそのような簡略化された深度パーティションパターン生成のための技法について説明する。以下の例示的な技法は、ビデオエンコーダ 20 およびビデオデコーダ 30 によって実行され得る。たとえば、ビデオエンコーダ 20 および / またはビデオデコーダ 30 は、深度データをイントラコーディング(たとえば、それぞれイントラ予測符号化またはイントラ予測復号)することができる。また、例示的な技法が別個に実行されたり、または 1 つもしくは複数の技法が組合せで実行されることがある。

【 0 1 2 1 】

[0126]さらに、簡潔のために、本開示で説明する技法は、ビデオコーダに関して説明され得る。ビデオコーダの例としては、ビデオデータの符号化について説明するときのビデオエンコーダ 20 、およびビデオデータの復号について説明するときのビデオデコーダ 30 がある。本開示は、「コーディングする」または「コーディング」という用語を使用することもある。この文脈では、コーディングするという用語は、符号化するまたは復号するを総称的に指し得、コーディングという用語は、符号化または復号を総称的に指し得る。たとえば、ビデオコーダはコーディングすることができ、これは、ビデオエンコーダ 20 が符号化できること、またはビデオデコーダ 30 が復号できることを意味する。20

【 0 1 2 2 】

[0127]上述のように、既存のブロックサイズ依存型ウェッジレットパターン設計に伴う問題のうちの 1 つは、より大きいサイズのブロックの場合、ウェッジレットパターンの数があまりにも大きくなってしまって、望ましくない量のメモリを必要とし、計算性の複雑性を高めることである。本開示で説明する技法では、より大きいサイズの深度ブロックに関するウェッジレットパターンに依拠するのではなく、より大きいサイズの深度ブロックに関するウェッジレットパターンが、より小さいサイズのブロックに関するウェッジレットパターンから決定され得る。このようにして、ビデオエンコーダ 20 およびビデオデコーダ 30 は、より小さいサイズのブロックに関するウェッジレットパターン(たとえば、パーティションパターン)を記憶し、より大きいサイズのブロックに関するウェッジレットパターンをより少なく記憶するか、またはまったく記憶しないことがあり、その理由は、ビデオエンコーダ 20 およびビデオデコーダ 30 が、より小さいサイズのブロックに関するウェッジレットパターンのうちのウェッジレットパターンから、より大きいサイズの深度ブロックに関するウェッジレットパターンを決定し得ることにある。40

【 0 1 2 3 】

[0128]一例として、ビデオエンコーダ 20 およびビデオデコーダ 30 は、第 1 のブロックサイズのブロックに関するパーティションパターンリスト(たとえば、ウェッジレットパターンリスト)を構成するように構成され得る。パーティションパターンリストは、第 1 のブロックサイズのブロックに関する 1 つまたは複数のパーティションパターンを含むことができる。第 1 のブロックサイズよりも大きい第 2 のブロックサイズの深度ブロックのイントラ予測符号化の場合、ビデオエンコーダ 20 は、第 1 のブロックサイズのブロ50

ックに関するパーティションパターンから決定されたパーティションパターン（たとえば、ウェッジレットパターン）を使用して、第2のサイズの深度ブロックをイントラ予測符号化することができる。言い換えれば、ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンを使用して、第2のサイズの深度ブロックをイントラ予測符号化することができる。深度モデリングモード（DMM）1の場合、ビデオエンコーダ20は、第1のブロックサイズの1つまたは複数のパーティションパターン関連ブロックを含むパーティションパターンリストへのインデックスをシグナリングすることができる。

【0124】

[0129] 第2のブロックサイズの深度ブロックのイントラ予測復号の場合、ビデオデコーダ30は、第1のブロックサイズのブロックに関する1つまたは複数のパーティションパターンを含むパーティションパターンリストへのインデックスを受信することができ、インデックスから第1のサイズのブロックに関するパーティションパターンを決定することができる。ここでも、パーティションパターンは、復号される深度ブロックの第2のブロックサイズよりも小さい第1のブロックサイズのブロックに関するものである。次いでビデオデコーダ30は、決定されたパーティションパターンから、第2のブロックサイズの深度ブロックに関するパーティションパターンを決定することができる。ビデオデコーダ30は、決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックをイントラ予測復号することができる。

【0125】

[0130] 上述のように、より大きいサイズのブロックに関するパーティションパターン（たとえば、ウェッジレットパターン）を記憶することは、より大きいサイズのブロックに利用可能なパーティションパターンの数が比較的大きいので、望ましくないことがある。より小さいサイズのブロックに関するパーティションパターンに基づいて、より大きいサイズのブロックに関するパーティションパターンを決定することによって、より大きいサイズのブロックに関して記憶される必要のあるパーティションパターンの数は低減され得る。たとえば、前の例では、ビデオエンコーダ20およびビデオデコーダ30は、第1のブロックサイズのブロックに関する1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成し、第2のより大きいサイズの深度ブロックに関するパーティションパターンを決定するために、パーティションパターンリストからパーティションパターンのうちの1つを使用することができる。したがって、ビデオエンコーダ20およびビデオデコーダ30が、第2のサイズのパーティションパターンに関するパーティションパターンリストを構成した場合、より小さいブロックサイズに関するパーティションパターンからパーティションパターンの一部が決定され得るので、パーティションパターンリスト中のパーティションパターンの数は低減されることになる。

【0126】

[0131] いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、いくつかのブロックサイズに関するパーティションリストパターンを構成しないこともある。そのような例では、ビデオエンコーダ20およびビデオデコーダ30は、ビデオエンコーダ20およびビデオデコーダ30がパーティションリストパターンを構成したブロックサイズの深度ブロックに関するパーティションパターンを決定するために、より小さいサイズのブロックに関するパーティションパターンに依拠し得る。

【0127】

[0132] また、より大きいブロックサイズの場合は、パーティションパターンリストにより少ないパーティションパターンがあること、またはパーティションパターンがまったくないことがあるので、パーティションパターンを記憶するために必要とされるメモリの量の減少があり得る。その上、より大きいブロックサイズの場合は、パーティションパターンリストにより少ないパーティションパターンがあること、またはパーティションパターンがまったくないことがあるので、パーティションパターンを生成する計算の複雑性および重複がないようにするためにパーティションパターンを比較する複雑性も低減され得る

10

20

30

40

50

。

【 0 1 2 8 】

[0133] 上述のように、より小さいサイズのブロックに関するパーティションパターンに基づいて、より大きいサイズの深度ブロックに関するパーティションパターンを決定することによって、本開示で説明する技法は、メモリ要件を低減し、複雑性を低減することができる。いくつかの例では、より小さいサイズのブロックは、より大きいサイズの深度ブロック内のブロックであり得る。しかしながら、本開示で説明する技法は、そのように限定されない。いくつかの例では、より小さいサイズのブロックは必ずしも、イントラ予測符号化または復号されている実際のブロックである必要はない。そうではなく、より小さいサイズのブロックは、より大きいサイズのブロックに関するパーティションパターンを決定するためにパーティションパターンが使用される概念的ブロックであり得る。10

【 0 1 2 9 】

[0134] 一例として、イントラ予測符号化または復号される深度ブロックが 32×32 サイズの深度ブロックであると仮定する。この例では、エンコーダ 20 は、サイズ 16×16 のブロックに関するパーティションパターンリストへのインデックスをシグナリングすることができる。ビデオデコーダ 30 は、サイズ 16×16 のブロックに関するパーティションパターンリストへのインデックスを受信し、サイズ 16×16 のブロックに関するパーティションパターンからパーティションパターンを決定することができる。この例では、ビデオエンコーダ 20 とビデオデコーダ 30 の両方は、サイズ 16×16 のブロックに関する決定されたパーティションパターンから、 32×32 サイズの深度ブロックに関するパーティションパターンを決定するように構成され得る。一例として、ビデオエンコーダ 20 およびビデオデコーダ 30 は、 32×32 サイズの深度ブロックに関するパーティションパターンを決定するために、サイズ 16×16 のブロックに関する決定されたパーティションパターンをアップサンプリングすることができる。20

【 0 1 3 0 】

[0135] より小さいサイズのブロック（そのパーティションパターンリストを、ビデオエンコーダ 20 およびビデオデコーダ 30 がより大きいサイズのブロックに関するパーティションパターンを決定するために使用する）は、より大きいサイズのブロックの一部である必要はないが、本開示で説明する技法は、より小さいサイズのブロックがより大きいサイズのブロック内のブロックである例とともに説明される。たとえば、深度コーディング（たとえば、深度ブロックのイントラ予測コーディング）の一例では、パーティションパターンは、ブロックサイズとは無関係である。30

【 0 1 3 1 】

[0136] 本開示で説明する技法において、ビデオエンコーダ 20 およびビデオデコーダ 30 が、より大きいサイズのブロックに関するパーティションパターンを決定するための中間ステップとして、より小さいサイズのブロックに関するパーティションパターンを決定し得ることを理解されたい。たとえば、より小さいサイズのブロックは、より小さいサイズのブロックのパーティションパターンに基づいてイントラ予測符号化または復号されないことがある。そうではなく、より小さいサイズのブロックは、より大きいサイズのブロックのイントラ予測の一部としてイントラ予測符号化または復号され得る。40

【 0 1 3 2 】

[0137] 一例として、現在の深度ブロック（たとえば、現在の深度 P U）は $N \times N$ のサイズを有すると仮定する。この例では、現在の深度ブロックは、整数個の $M \times M$ ($M < N$) ブロックを含む。一例として、現在の深度ブロック（たとえば、イントラ予測符号化または復号されるべき深度ブロック）が 32×32 サイズのブロックである場合、 32×32 サイズのブロック内に 4 個の 16×16 サイズのブロック、または 32×32 サイズのブロック内に 64 個の 4×4 サイズのブロックがある。

【 0 1 3 3 】

[0138] いくつかの例では、ビデオエンコーダ 20 は、1 つの特定の $M \times M$ ブロック（たとえば、 M は 4 に等しい）と、線ベースのパーティションパターン（たとえば、2 つのパ50

ーティションに $M \times M$ ブロックを二等分する線形線を画定するウェッジレットパターン)とを識別する情報をシグナリングすることができる。ビデオデコーダ30は、 $M \times M$ ブロックと線ベースのパーティションパターンとを識別する情報を受信することができる。特定の $M \times M$ ブロックの線ベースのパーティションパターンに基づいて、ビデオエンコーダ20およびビデオデコーダ30は、全体的 $N \times N$ PUに関する線ベースのパーティションを決定する(たとえば、導出する)ことができる。言い換えれば、 $M \times M$ ブロックに関するパーティションパターンに基づいて、ビデオエンコーダ20およびビデオデコーダ30は、 $N \times N$ 深度ブロック全体に関するパーティションパターンを決定することができ、ここで、 $N \times N$ 深度ブロックは $M \times M$ ブロックを含む。

【0134】

10

[0139]一例では、線ベースのパーティションパターンは、現在の3D-HEVCの場合のように、特定の $M \times M$ ブロックに適用されるDMM1のウェッジレットパターンであり得る。たとえば、ビデオエンコーダ20は、深度モデリングモード(DMM)が1であることを示す情報をシグナリングし、 $N \times N$ ブロック内の $M \times M$ ブロックを識別する情報をシグナリングし、 $M \times M$ ブロックに関するパーティションパターンを識別するために使用される情報をシグナリングすることができる。(たとえば、サイズ $M \times M$ のブロックに関するパーティションパターンリストへのインデックス値)。ビデオデコーダ30は、DMMが1であることを示す情報を受信し、 $M \times M$ ブロックを識別する受信された情報に基づいて $N \times N$ ブロック内の $M \times M$ ブロックを決定し、(たとえば、サイズ $M \times M$ のブロックに関するパーティションパターンリストへのインデックス値に基づいて) $M \times M$ ブロックに関するパーティションパターンを決定することができる。次いでビデオデコーダ30は、 $M \times M$ ブロックに関する決定されたパーティションパターンに基づいて、全体的 $N \times N$ ブロックに関するパーティションパターンを決定することができる。

20

【0135】

[0140]本明細書で説明する「全体的」ブロックは、ブロックに含まれ得るサブブロックを含む、ブロック全体を含み得る。したがって、4個のサブブロックを有するブロックの場合、全体的ブロックは、4個のサブブロックのすべてを含むブロック全体を指し得る。

【0136】

[0141]たとえば、ビデオエンコーダ20は、イントラ深度符号化(イントラ予測符号化)のためにブロック(たとえば、深度PUである $N \times N$ ブロック)のサブブロックに関する線ベースのパーティションパターン(たとえば、2つのパーティションに $M \times M$ ブロックを二等分する線形線を画定するウェッジレットパターン)を決定することができ、ここにおいて、サブブロックがブロックよりも小さい。いくつかの例では、ビデオエンコーダ20は、ブロックのサブブロックに関する線ベースのパーティションパターンを、決定される線ベースのパーティションパターンをブロックのサイズに基づかせることなく(すなわち、ブロックサイズとは無関係に)決定することができる。ビデオエンコーダ20は、サブブロックに関する線ベースのパーティションパターンに基づいて、ブロックに関する線ベースのパーティションパターンを決定することができる。ビデオエンコーダ20は、ブロックに関する決定された線ベースのパーティションパターンに基づいて、ブロックをイントラ符号化する(イントラ予測符号化する)ことができる。ビデオエンコーダ20は、サブブロックに関する線ベースのパーティションパターンを示す情報をシグナリングする(たとえば、サブブロックに関する線ベースのパーティションパターンを示す情報を符号化する)ことができる。いくつかの例では、ビデオエンコーダ20は、ビデオエンコーダ20が線ベースのパーティションパターンを決定した、ブロック内のサブブロックを識別するために使用される情報をシグナリングすることができるが、これは、あらゆる例において必要とされるとは限らない。

30

【0137】

[0142]ビデオエンコーダ20は、ブロックに関する線ベースのパーティションパターンを示す情報をシグナリングしないことがある。たとえば、ビデオエンコーダ20は、ブロックに関する線ベースのパーティションパターンを示す情報を符号化するのを回避するこ

40

50

と、または符号化しないことがある。むしろ、ビデオエンコーダ20は、サブブロックに関する線ベースのパーティションパターンを示す情報をシグナリングすることができる。

【0138】

[0143]ビデオデコーダ30は、イントラ深度復号のために（たとえば、イントラ予測復号のために）ブロックのサブブロックに関する線ベースのパーティションパターンを示す情報を受信する（たとえば、サブブロックに関する線ベースのパーティションパターンを示す情報を復号する）ことができる。ビデオデコーダ30は、サブブロックに関する線ベースのパーティションパターンに基づいて、ブロックに関する線ベースのパーティションパターンを決定することができる。ビデオデコーダ30は、ブロックに関する決定された線ベースのパーティションパターンに基づいて、ブロックをイントラ復号する（イントラ予測復号する）ことができる。いくつかの例では、ビデオデコーダ30は、ビデオデコーダ30によって線ベースのパーティションパターンが受信された、ブロック内のサブブロックを識別するために使用される情報を受信することができるが、これは、あらゆる例において必要とされるとは限らない。ビデオデコーダ30は、ブロックに関する線ベースのパーティションパターンを示す情報を受信することなく、ブロックに関する線ベースのパーティションパターンを決定することができる。むしろ、ビデオデコーダ30は、サブブロックから線ベースのパーティションパターンを決定することができる。また、ビデオデコーダ30は、ブロックに関する線ベースのパーティションパターンを、決定される線ベースのパーティションパターンをブロックのサイズに基づかせることなく（すなわち、ブロックサイズとは無関係に）決定することができる。10

【0139】

[0144]M × Mサブブロックは必ずしも、より大きいN × Nブロック内に入るブロックである必要はないことを理解されたい。そうではなく、M × Mサブブロックは、N × Nブロックに関する線ベースのパーティションパターンを決定する目的で線ベースのパーティションパターンが使用される概念的ブロックであり得る。言い換えれば、上記の例では、ビデオエンコーダ20は、第1のサイズ（たとえば、32 × 32）の深度ブロックに関するパーティションパターン（たとえば、ウェッジレットパターンのような線ベースのパーティションパターン）を、第2のより小さいサイズ（たとえば、16 × 16）のブロックに関するパーティションパターンに基づいて決定することができる。ビデオエンコーダ20は、決定されたパーティションパターンに基づいて深度ブロックをイントラ予測符号化することができる。ビデオデコーダ30は同様に、第1のサイズ（たとえば、32 × 32）の深度ブロックに関するパーティションパターンを、第2のより小さいサイズ（たとえば、16 × 16）のブロックに関するパーティションパターンに基づいて決定することができる。ここでも、このより小さいサイズのブロックは、ピクチャ中または深度ブロック内の実際のブロックである必要はなく、代わりに、より大きいブロックに関するパーティションパターンを決定するためにパーティションパターンが使用される概念的ブロックである。20

【0140】

[0145]いくつかの例では、ビデオエンコーダ20は、第2のサイズのブロックに関するパーティションパターンを識別する情報をシグナリングし、ビデオデコーダ30は、シグナリングされた情報に基づいて第2のサイズのブロックに関するパーティションパターンを決定する。たとえば、ビデオエンコーダ20は、第2のサイズのパーティションパターンに関するパーティションパターンリストへのインデックスをシグナリングすることができ、インデックスによって識別されるパーティションパターンは、ビデオエンコーダ20が第1のサイズの深度ブロックに関するパーティションパターンを決定するために利用したパーティションパターンである。ビデオデコーダ30は、第2のサイズのパーティションパターンに関するパーティションパターンリストへのインデックスを受信し、第2のサイズのブロックに関するパーティションパターンに基づいて、第1のサイズの深度ブロックに関するパーティションパターンを決定することができる。30

【0141】

50

50

50

[0146]上記の例では、より小さいサイズのブロックは概念的ブロックであり、必ずしも、深度ブロックを含むピクチャのブロック、または深度ブロック内のブロックであるとは限らない。ただし、いくつかの例では、より小さいブロックは、より大きいサイズのブロック内のブロックであり得る。より小さいサイズのブロックがより大きいサイズのブロック内のブロックである（たとえば、 16×16 ブロックが、 32×32 ブロック内の 4 個の 16×16 ブロックのうちの 1 つである）例では、ビデオエンコーダ 20 は、より大きいサイズのブロック中のより小さいサイズのブロックのロケーションを識別する情報をシグナリングすることができる。次いでビデオデコーダ 30 は、より小さいサイズのブロックに関するパーティションパターンおよびより大きいサイズのブロック内により小さいサイズのブロックの位置に基づいて、より大きいサイズのブロックに関するパーティションパターンを決定することができる。たとえば、ビデオデコーダ 30 は、より小さいサイズのブロックを二等分する線形線を、より大きいサイズのブロックのエッジに線形線がぶつかるまで、外側に拡張することができる。より大きいサイズのブロックの得られる二等分は、より小さいサイズのブロックのパーティションパターンから決定された、より大きいサイズのブロックに関するパーティションパターンであり得る。

【 0 1 4 2 】

[0147]言い換えれば、ビデオデコーダ 30 は、サブブロックに関するパーティションパターンの線形線を深度ブロックの境界まで拡張することができる。得られる線形線は、深度ブロックをイントラ予測復号するために使用される、深度ブロックに関するパーティションパターンであり得る。ビデオエンコーダ 20 は同様に、サブブロックに関するパーティションパターンの線形線を深度ブロックの境界まで拡張することができ、得られる線形線は、深度ブロックをイントラ予測符号化するために使用される、深度ブロックに関するパーティションであり得る。

【 0 1 4 3 】

[0148]いくつかの例では、ビデオエンコーダ 20 は、より小さいサイズのブロックの位置を識別する情報をシグナリングする必要はないことがある。そのような例では、ビデオデコーダ 30 は、ビデオデコーダ 30 がより大きいサイズの深度ブロックに関するパーティションパターンを決定するためにパーティションパターンを拡張する際の起点となるブロックとして、所定のより小さいサイズのブロックを使用するように事前構成され得る。

【 0 1 4 4 】

[0149]たとえば、ビデオエンコーダ 20 は、 i および j が 0 から $N / M - 1$ まで（両端の値を含む）の範囲内にある相対水平および垂直インデックス (i, j) とともに、ビデオデコーダ 30 が受信する特定の $M \times M$ ブロック（すなわち、サブブロック）のインデックスをシグナリングすることができる。この例では、 $M \times M$ ブロックの左上位置は ($M \times i, N \times j$) である。このようにして、ビデオデコーダ 30 は、 $M \times M$ ブロックの位置を決定し、さらに、以下でより詳細に説明するように、線形線を拡張することによって全体的 $N \times N$ ブロックに関するパーティションパターンを決定することができる。

【 0 1 4 5 】

[0150]ビデオエンコーダ 20 およびビデオデコーダ 30 は、それぞれ、バイパスモードまたはコンテキストモデリングのいずれかにより、 $M \times M$ ブロックの位置 (i, j) を符号化または復号することができる。言い換えれば、ビデオエンコーダ 20 は、線ベースのパーティションパターンが決定されたサブブロックを識別するためにバイパスモードまたはコンテキストモデリングを使用して位置 (i, j) を符号化することができ、ビデオデコーダ 30 は、線ベースのパーティションパターンが受信されたサブブロックを識別するためにバイパスモードまたはコンテキストモデリングを使用して位置 (i, j) を復号することができる。

【 0 1 4 6 】

[0151]代替または追加として、ビデオエンコーダ 20 およびビデオデコーダ 30 は、現在の P U の左境界 $M \times M$ ブロックまたは下境界 $M \times M$ ブロックのいずれかにおいて常に開始するように特定の $M \times M$ ブロックを設定するように事前構成され得る。この場合、ビデ

10

20

30

40

50

エンコーダ20は、フラグをシグナリングすることができ、たった1つのオフセットがエンコーダによってさらにシグナリングされる。たとえば、この例では、ビデオエンコーダ20は、線ベースのパーティションパターンが決定されたサブロックを識別するためのフラグと1つのオフセットとをシグナリングすることができ、ビデオデコーダ30は、線ベースのパーティションパターンが受信されたサブロックを識別するためのフラグと1つのオフセットとを受信することができる。

【0147】

[0152]場合によっては、1つの $N \times N$ ブロック内の $M \times M$ ユニット中の水平および垂直インデックスをシグナリングする代わりに、ビデオエンコーダ20は、4分木構造によって $M \times M$ ブロックを識別することができ、各レベル表現が、ゼロまたはより多くの「0」と1つの「1」を有し、「1」に達すると終了し、より低いレベルに進む。この例では、ビデオエンコーダ20は、サブロックを識別するための情報をシグナリングする必要がないことがあり、ビデオデコーダ30は、サブロックを識別するための情報を受信する必要がないことがある。たとえば、ビデオデコーダ30は同様に、 $N \times N$ 深度ブロックに関するパーティションパターンを決定するためにビデオデコーダ30がパーティションパターンを決定した $M \times M$ ブロックを決定するために、4分木構造を使用することができる。

【0148】

[0153]いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、特定の $M \times M$ ブロックの線ベースのパーティションパターンをより大きい $N \times N$ ブロックに拡張することを、利用可能な場合に $N \times N$ ブロック内のすべての $M \times M$ ブロックのパーティションパターンが全体的PU（すなわち、全体的深度ブロック）に関する線ベースのパーティションを一緒に形成する方法で、行うことができる。たとえば、特定の $M \times M$ ブロックの右上隅に、 $N \times N$ ブロック内の別の $M \times M$ ブロックの左下隅がある。いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、特定の $M \times M$ ブロックを二等分する線形線（たとえば、パーティション線）を、特定の $M \times M$ ブロックの右上隅に連結された他方のブロックを通じて拡張することができる。同様に、特定の $M \times M$ ブロックの左下隅に、 $N \times N$ ブロック内のまた別の $M \times M$ ブロックの右上隅がある。いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、特定の $M \times M$ ブロックを二等分する線形線（たとえば、パーティション線）を、特定の $M \times M$ ブロックの左下隅に連結された他方のブロックを通じて拡張することができる。このようにして、ビデオエンコーダ20およびビデオデコーダ30は、特定の $M \times M$ ブロックの線ベースのパーティションを全体的PUに拡張することができる。

【0149】

[0154]上述のように、ビデオエンコーダ20およびビデオデコーダ30は、より小さいサイズのブロックを二等分する線形線を、より大きいサイズの深度ブロックに関するパーティションパターンを決定するためにより大きいサイズのブロックを線形線が二等分するように、拡張するように構成され得る。いくつかの例では、特定の $M \times M$ の線ベースのパーティションパターンを現在のPUに（すなわち、全般的なより大きいサイズの深度ブロックに）拡張するために、ビデオエンコーダ20およびビデオデコーダ30は、最初に、パーティション境界線関数 $y = a \times x + b$ を、 $M \times M$ パーティションパターンの($M \times i$, $M \times j$)と開始点位置および終了点位置とによって導出することができ、 a および b がそれぞれ、パーティション境界線の傾きおよび切片を表す。言い換えれば、ビデオエンコーダ20およびビデオデコーダ30は、傾き-線の式に基づいて線形線の線-式を決定することができる。パーティション境界線関数により、ビデオエンコーダ20およびビデオデコーダ30は、 $N \times N$ 2進ブロックである（ b Pattern[x][y] = ($y - a \times x$) < b ? 1 : 0として導出することができ、ここで $x, y = 0, 1, \dots, N - 1$ である。一例では、さらに、 a および b は整数に丸められる。

【0150】

10

20

30

40

50

[0155] D M M モード 3 または 4 が可能であるとき、特定の $M \times M$ ブロックが、ビデオエンコーダ 2 0 によってシグナリングされ、ビデオデコーダ 3 0 によって受信され得る。さらに、 $M \times M$ ブロック内で、ビデオエンコーダ 2 0 は線ベースのパーティションパターン（たとえば、ウェッジレットパターン）を、現在の 3 D - H E V C の場合のように、ビデオエンコーダ 2 0 が $M \times M$ の P U（すなわち、 $M \times M$ ブロック）に関して D M M モード 3 でウェッジレットパターンをシグナリングする方法と同様の方法でシグナリングする。この場合、ビデオエンコーダ 2 0 は、 $M \times M$ の P U（すなわち、 $M \times M$ ブロック）に対するウェッジレットサブセットインデックスをシグナリングすることができる。

【 0 1 5 1 】

[0156] 上述のように、本開示で説明する技法により、より大きいサイズのブロックに必要とされるパーティションパターンの数の減少があり得る。たとえば、本開示で説明する技法では、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックのみがパーティションを必要とし得る（すなわち、ビデオエンコーダ 2 0 およびビデオデコーダ 3 0 は、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックのみを含むパーティションリストを構成する必要があり得る）。したがって、ビデオエンコーダ 2 0 は、単に、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックに関するパーティションパターンをシグナリングし、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックを導出または拡張する必要があり得る。同様に、ビデオデコーダ 3 0 は、単に、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックに関するパーティションパターンを受信し、 N / M （または $2 \times N / M$ ）個までの $M \times M$ ブロックを導出または拡張する必要があり得る。10

【 0 1 5 2 】

[0157] さらに、上述のように、本開示で説明する技法は、記憶される必要のあるパーティションパターンの数を減らし、パーティションパターンを使用してイントラ予測を実施する複雑性を低減することができる。したがって、技法は、 64×64 サイズの深度ブロックに関する線ベースの区分（たとえば、ウェッジレット区分）をサポートすることに伴う問題を克服することができる。言い換えれば、現在の 3 D - H E V C の場合のように、 32×32 までの線ベースのパーティションのみをサポートする代わりに、本開示で説明する技法を使用して、ウェッジレットパターンは、 64×64 であり得る $N \times N$ の P U に拡張される。20

【 0 1 5 3 】

[0158] $N \times N$ サイズの深度ブロック（すなわち、 32×32 以下のサイズの深度ブロックに限定されない）に関する線ベースのパーティションパターンを使用するための技法について説明することに加えて、本開示は、線ベースの区分（たとえば、ウェッジレット区分）のためにイントラ予測符号化および復号を実施する効率性を高めるための技法について説明する。たとえば、ビデオエンコーダ 2 0 において、D M M 1 において $N \times N$ 深度 P U（すなわち、深度ブロック）のウェッジレット探索プロセスを加速させるために、元の深度ピクチャにおける現在の深度 P U（すなわち、深度ブロック）の左上、右上、左下および右下のサンプルが同じ値を有するとき、ビデオエンコーダ 2 0 は、ウェッジレットパターン探索プロセスを省略することができ、ビデオエンコーダ 2 0 は、現在の P U（すなわち、現在の深度ブロック）のために D M M 1 を選択することのないように構成され得る。30

【 0 1 5 4 】

[0159] 別の例として、代替または追加として、ビデオエンコーダ 2 0 は、複数の隣接する現在の深度 P U の値をチェックすることができる。ビデオエンコーダ 2 0 は、隣接する現在の深度 P U の値に基づいて、探索プロセスが省略され、D M M 1 であるかどうかを決定することができ、決定に基づいて現在の P U のために D M M 1 を選択することができる。代替または追加として、ビデオエンコーダ 2 0 は、元の深度ピクチャにおける現在の深度 P U の左上、右上、左下および右下のサンプルが同じ値を有するとき、ウェッジレットパターン探索を省略し、1 サンプルパーティションパターン（たとえば、左上サンプルは、すべての他のサンプルの異なる部分に属する）を選択することができる。40

【0155】

[0160]いくつかの例では、DMM3モードでは、コロケートテクスチャルーマイントラモードによって指定される $N \times N$ ウェッジレットサブセットが空である場合、ビデオエンコーダ20は、現在のPUのためにDMM3モードを省略することができる。コロケートテクスチャルーマイントラモードが空であることは、コロケートテクスチャブロックのルーマ成分がイントラ予測符号化または復号されないことを意味する。たとえば、コロケートテクスチャブロックのルーマ成分がイントラ予測コーディングされる場合、利用可能なルーマイントラモードはない。代替または追加として、ビデオデコーダ30は、コロケートテクスチャルーマイントラモードによって指定される $N \times N$ ウェッジレットサブセットが空である場合に、ビデオデコーダ30によって復号されるDMMモードインデックスが決して現在のPUのためのDMM3モードではあり得ないように、制約され得る。10

【0156】

[0161]上述のように、DMM3の場合、ビデオエンコーダ20およびビデオデコーダ30は、コロケートテクスチャブロックに基づいて、深度ブロックに関するパーティションパターンを決定することができる。しかしながら、本開示で説明する技法は、そのように限定されない。いくつかの例では、DMM3モードにおいて、ビデオエンコーダ20およびビデオデコーダ30は、現在の $N \times N$ のPUとコロケートされたルーマブロックによってウェッジレットパーティションパターンを導出しないことがある。そうではなく、ビデオエンコーダ20およびビデオデコーダ30は、現在の $M \times M$ ブロックとコロケートされたルーマブロックによって線ベースのパーティションパターンを決定し得る。20

【0157】

[0162] $N \times N$ のPUの開始／終了点位置の1／2サンプル精度を使用するウェッジレットパターンの場合、ビデオエンコーダ20およびビデオデコーダ30は、対応するウェッジレットパターンとともに $2N \times 2N$ のPUにおいて決定位置(2m, 2n)が属するパーティションに基づいて、位置(m, n)がどのパーティションに属するかを決定することができる。ビデオエンコーダ20およびビデオデコーダ30は、(オフセットX, オフセットY)のオフセットベクトルにより、 $2N \times 2N$ ブロックにおいて決定位置(2m, 2n)を移動させることなく、位置(m, n)がどのパーティションに属するかを決定することができ、ここにおいて、オフセットXまたはオフセットYが0または1に等しく、現在の3D-HEVCの場合のように様々な条件に依存する。30

【0158】

[0163]図6は、深度コーディングのための技法を実装し得るビデオエンコーダ20の一例を示すブロック図である。ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターフォームコーディングとを実行し得る。イントラコーディング(すなわち、イントラ予測コーディング)は、所与のビデオフレームまたはピクチャ内のビデオの空間的冗長性を低減または除去するために空間的予測に依拠する。インターフォームコーディング(すなわち、インターフォーム予測コーディング)は、ビデオシーケンスの隣接するフレームまたはピクチャ中のビデオの時間的冗長性を低減または除去するために時間的予測に依拠する。イントラモード(Iモード)は、いくつかの空間ベースのコーディングモードのいずれかを指し得る。単一方向予測(Pモード)または双予測(Bモード)などのインターモードは、いくつかの時間ベースのコーディングモードのいずれかを指し得る。40

【0159】

[0164]上述のように、ビデオエンコーダ20は、マルチビュービデオコーディングを実行するように適応され得る。いくつかの事例では、ビデオエンコーダ20は、時間インスタンス中の各ビューがビデオデコーダ30などのデコーダによって処理され得るように、マルチビューHEVCをコーディングするように構成され得る。HEVC-3Dの場合、各ビューに対するテクスチャマップ(すなわち、ルーマ値およびクロマ値)を符号化することに加えて、ビデオエンコーダ20はさらに、各ビューに対する深度マップを符号化し得る。

【0160】

[0165] いずれの場合も、図6に示すように、ビデオエンコーダ20は、符号化されるべきビデオピクチャ内の現在のビデオブロックを受信する。図6の例では、ビデオエンコーダ20は、ビデオメモリデータ39と、モード選択ユニット40と、(復号済みピクチャバッファ(DPB)とも呼ばれる)参照ピクチャメモリ64と、加算器50と、変換処理ユニット52と、量子化ユニット54と、エントロピー符号化ユニット56とを含む。そして、モード選択ユニット40は、動き補償ユニット44と、動き推定ユニット42と、イントラ予測ユニット46と、パーティションユニット48とを含む。ビデオブロック再構成のために、ビデオエンコーダ20はまた、逆量子化ユニット58と、逆変換ユニット60と、加算器62とを含む。再構成されたビデオからプロックキネスアーティファクトを除去するためにブロック境界をフィルタ処理するためのデブロッキングフィルタ(図6には図示せず)も含まれ得る。所望される場合、デブロッキングフィルタは、通常、加算器62の出力をフィルタ処理することになる。デブロッキングフィルタに加えて、追加のフィルタ(ループ内またはループ後)も使用され得る。そのようなフィルタは簡潔のために示されていないが、所望される場合、(ループ内フィルタとして)加算器50の出力をフィルタ処理することができる。
10

【0161】

[0166] 図6に示すように、ビデオデータメモリ39は、ビデオピクチャ内の現在のビデオブロックを符号化するために使用されるビデオデータを受信する。ビデオデータメモリ39は、(たとえば、ビデオデータを記憶するように構成された)ビデオエンコーダ20の構成要素によって符号化されるべきビデオデータを記憶するか、またはビデオピクチャを符号化するために使用されるべきビデオデータを記憶することができる。たとえば、ビデオデータメモリ39は、異なるサイズのブロックに関するパーティションパターン(たとえば、ウェッジレットパーティションパターン)を記憶することができる。ビデオデータメモリ39は、ビデオエンコーダ20によって実施される初期化プロセスの一部としてパーティションパターンを記憶することができる。
20

【0162】

[0167] いくつかの例では、ビデオデータメモリ39に記憶されるビデオデータは、たとえば、ビデオソース18から取得される場合がある。(復号ピクチャバッファ(DPB)とも呼ばれる)参照ピクチャメモリ64は、(たとえば、イントラコーディングモードまたはインターフォーマンスモードで)ビデオエンコーダ20によってビデオデータを符号化する際に使用するための参考ビデオデータを記憶する。ビデオデータメモリ39および参照ピクチャメモリ64は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗RAM(RRAM(登録商標))、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ39および参照ピクチャメモリ64は、同じメモリデバイスまたは別個のメモリデバイスによって設けられ得る。様々な例では、ビデオデータメモリ39は、ビデオエンコーダ20の他の構成要素とともにオンチップであつてよく、またはそれらの構成要素に対してオフチップであつてもよい。
30

【0163】

[0168] 符号化プロセス中に、ビデオエンコーダ20は、コーディングされるべきビデオフレームまたはスライスを受信する。フレームまたはスライスは複数のビデオブロックに分割され得る。動き推定ユニット42および動き補償ユニット44は、時間的予測を行うために、1つまたは複数の参考フレーム中の1つまたは複数のブロックに対して、受信されたビデオブロックのイントラ予測コーディングを実行する。イントラ予測ユニット46は代替的に、空間的予測を行うために、コーディングされるべきブロックと同じフレームまたはスライス中の1つまたは複数の隣接ブロックに対して、受信されたビデオブロックのイントラ予測コーディング(たとえば、イントラ予測符号化)を実行し得る。ビデオエンコーダ20は、たとえば、ビデオデータの各ブロックに適切なコーディングモードを選択するために、複数のコーディングパスを実行することができる。
40

【0164】

[0169]さらに、パーティションユニット48は、以前のコーディングパスにおける以前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分することができる。たとえば、パーティションユニット48は、最初に、フレームまたはスライスをLCUに区分し、レートひずみ分析（たとえば、レートひずみ最適化）に基づいて、LCUの各々をサブCUに区分することができる。モード選択ユニット40は、サブCUへのLCUの区分を示す4分木データ構造をさらに生成することができる。4分木のリーフノードCUは、1つまたは複数のPUと1つまたは複数のTUとを含み得る。

【0165】

[0170]モード選択ユニット40は、たとえば、誤差結果に基づいてコーディングモードのうちの1つ、すなわち、イントラまたはインターを選択し、得られたイントラコーディングまたはインターコーディングされたブロックを、残差ブロックデータを生成するために加算器50に与え、参照フレームとして使用するための符号化されたブロックを再構成するために加算器62に与え得る。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、パーティション情報、および他のそのようなシンタックス情報など、シンタックス要素をエントロピーコーディングユニット56に与え得る。10

【0166】

[0171]動き推定ユニット42および動き補償ユニット44は、高度に統合され得るが、概念的な目的のために別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、たとえば、現在のフレーム（または他のコーディングされたユニット）内でコーディングされている現在のブロックに対する参照フレーム（または他のコーディングされたユニット）内の予測ブロックに対する現在のビデオフレームまたはピクチャ内のビデオブロックのPUの変位を示すことができる。予測ブロックは、絶対差分和（SAD : sum of absolute difference）、2乗差分和（SSD : sum of square difference）、または他の差分メトリックによって決定され得るピクセル差分に関して、コーディングされるべきブロックにぴったり一致することがわかるブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ64に記憶された参照ピクチャのサブ整数ピクセル位置の値を計算し得る。たとえば、ビデオエンコーダ20は、参照ピクチャの1/4ピクセル位置、1/8ピクセル位置、または他の分数のピクセル位置の値を補間することができる。したがって、動き推定ユニット42は、フルピクセル位置および分数ピクセル位置に対する動き探索を実行し、分数ピクセル精度を有する動きベクトルを出力することができる。20

【0167】

[0172]動き推定ユニット42は、PUの位置を参照ピクチャの予測ブロックの位置と比較することによって、インターコーディングされたスライス中のビデオブロックのPUに関する動きベクトルを計算する。参照ピクチャは、その各々が、参照ピクチャメモリ64に記憶された1つまたは複数の参照ピクチャを識別する、第1の参照ピクチャリスト（リスト0）または第2の参照ピクチャリスト（リスト1）から選択され得る。動き推定ユニット42は、計算された動きベクトルをエントロピー符号化ユニット56と動き補償ユニット44とに送る。30

【0168】

[0173]動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定された動きベクトルに基づいて予測ブロックを取り出し、またはこれを生成することを含むことができる。やはり、動き推定ユニット42および動き補償ユニット44は、いくつかの例で、機能的に統合され得る。現在のビデオブロックのPUに関する動きベクトルを受信すると、動き補償ユニット44は、参照ピクチャリストのうちの1つにおいて動きベクトルが指す予測ブロックを位置特定することができる。加算器50は、以下で説明するように、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42は、ルーマ成分に対して動き推定を実4050

行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方に関して、ルーマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40は、ビデオスライスのビデオブロックを復号する際のビデオデコーダ30による使用のために、ビデオブロックとビデオスライスとに関連付けられたシンタックス要素を生成することもできる。

【0169】

[0174] イントラ予測ユニット46は、上述のように、動き推定ユニット42と動き補償ユニット44とによって実行されるインター予測の代替として、現在のブロックをイントラ予測し得る。特に、イントラ予測ユニット46は、現在のブロックを符号化するために使用すべきイントラ予測モードを決定することができる。いくつかの例では、イントラ予測ユニット46は、たとえば別々の符号化バス中に、様々なイントラ予測モードを使用して現在のブロックを符号化することができ、イントラ予測ユニット46（または、いくつかの例ではモード選択ユニット40）は、テストされたモードから使用すべき適切なイントラ予測モードを選択することができる。

10

【0170】

[0175] たとえば、イントラ予測ユニット46は、様々なテストされたイントラ予測モードのためのレートひずみ分析を使用してレートひずみ値を計算し、テストされたモードの間で最良のレートひずみ特性を有するイントラ予測モードを選択し得る。レートひずみ分析は、概して、符号化ブロックと、符号化ブロックを生成するために符号化された元の符号化されていないブロックとの間のひずみ（または誤差）の量、ならびに符号化ブロックを生成するために使用されるビットレート（すなわち、ビット数）を決定する。イントラ予測ユニット46は、どのイントラ予測モードがブロックについて最良のレートひずみ値を呈するかを決定するために、様々な符号化されたブロックのひずみおよびレートから比を計算することができる。

20

【0171】

[0176] さらに、イントラ予測ユニット46は、深度マップの深度ブロックをコーディングするように構成され得る。たとえば、イントラ予測ユニット46は、深度スライスのイントラ予測されたPUをコーディングするために、（たとえば、上記で図2に関して説明したように）ベース（2D）HEVC規格からのイントラ予測モードと、（たとえば、上記で図3Aおよび図3Bに関して説明したように）深度モデリングモード（DMM）と、（たとえば、以下で図10に関して説明するように）領域境界チェーンコーディングとを使用する。

30

【0172】

[0177] いずれの場合も、ビデオエンコーダ20は、コーディングされている元のビデオブロックから、モード選択ユニット40からの予測データを減算することによって残差ビデオブロックを形成する。加算器50は、この減算演算を実行する1つまたは複数の構成要素を表す。変換処理ユニット52は、離散コサイン変換（DCT）または概念的に同様の変換などの変換を残差ブロックに適用して、残差変換係数値を備えるビデオブロックを生成する。変換処理ユニット52は、DCTと概念的に同様である他の変換を実行することができる。ウェーブレット変換、整数変換、サブバンド変換、または他のタイプ変換も使用され得る。

40

【0173】

[0178] いずれの場合も、変換処理ユニット52は、残差ブロックに変換を適用して、残差変換係数のブロックを生成する。変換は、ピクセル値領域からの残差情報を、周波数領域のような変換領域に変換することができる。変換処理ユニット52は、得られた変換係数を量子化ユニット54に送り得る。量子化ユニット54は、ビットレートをさらに低減するために、変換係数を量子化する。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって、変更され得る。いくつかの例では、量子化ユニット54は次いで、量子化変換係数を含む行列の走査を実行することができる。代替的に、エントロピー符号化ユニット56が走査を実行することができる。

50

【0174】

[0179]量子化に続いて、エントロピー符号化ユニット56は、量子化変換係数をエントロピーコーディングする。たとえば、エントロピー符号化ユニット56は、コンテキスト適応型可変長コーディング(CAVLC)、コンテキスト適応型バイナリ算術コーディング(CABAC)、シンタックスペースコンテキスト適応型バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピーコーディング技法を実行することができる。コンテキストベースのエントロピーコーディングの場合、コンテキストは、隣接ブロックに基づき得る。エントロピー符号化ユニット56によるエントロピーコーディングに続いて、符号化されたビットストリームは、別のデバイス(たとえば、ビデオデコーダ30)に送信され、または後の送信もしくは取出のためにアーカイブされ得る。10

【0175】

[0180]逆量子化ユニット58および逆変換ユニット60は、たとえば参照ブロックとして後で使用するために、ピクセル領域内で残差ブロックを再構成するために、それぞれ逆量子化と逆変換とを適用する。動き補償ユニット44は、参照ピクチャメモリ64のフレームのうちの1つの予測ブロックに残差ブロックを加算することによって参照ブロックを計算し得る。動き補償ユニット44はまた、動き推定において使用するためのサブ整数ピクセル値を計算するために、再構成された残差ブロックに1つまたは複数の補間フィルタを適用し得る。加算器62は、参照ピクチャメモリ64に記憶するための再構成されたビデオブロックを生成するために、動き補償ユニット44によって生成された動き補償予測ブロックに再構成された残差ブロックを加算する。再構成されたビデオブロックは、動き推定ユニット42および動き補償ユニット44によって、後続のビデオフレーム中のブロックをインターフォーマンスするためのインターフォーマンスするための参照ブロックとして使用され得る。20

【0176】

[0181]この意味で、ビデオエンコーダ20は、本開示で説明する1つまたは複数の例示的技法を実装するように構成され得る。たとえば、イントラ予測ユニット46は、本開示で説明する深度ブロックのための例示的なイントラ予測符号化技法を実装するように構成され得る。いくつかの例では、イントラ予測ユニット46は、他のプロセッサとの組合せで、本開示で説明する技法を実装するように構成され得る。30

【0177】

[0182]いくつかの例では、ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンからパーティションパターンを決定する(たとえば、サイズ 16×16 のブロックに関するパーティションパターンを決定する)ように構成され得る。ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンからの決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションを決定する(たとえば、サイズ 16×16 のブロックに関するパーティションパターンに基づいて、サイズ 32×32 の深度ブロックに関するパーティションを決定する)ことができ、ここで、第2のサイズが第1のサイズよりも大きい。ビデオエンコーダ20は、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測符号化することができる。40

【0178】

[0183]この例では、サイズ 16×16 のブロック(たとえば、第1のサイズのブロック)は、深度ブロックを含むピクチャ内の実際のブロックではないことがある。そうではなく、サイズ 16×16 のブロックは概念的ブロックであり、サイズ 16×16 のブロックに使用されるパーティションパターンが代わりにサイズ 32×32 のブロック(たとえば、第2のサイズの深度ブロック)に使用される。ただし、いくつかの例では、サイズ 16×16 のブロックは、ピクチャの実際のブロック、またはサイズ 32×32 のブロック内のブロックであり得る。

【0179】

[0184]ビデオエンコーダ20は、イントラ深度符号化のために深度ブロック(たとえば50

、現在のPUであるN×Nブロック)のサブブロック(たとえば、M×Mブロック)に関するパーティションパターン(たとえば、線ベースのパーティションパターン)を決定するように構成され得る。ビデオエンコーダ20は、サブブロックに関するパーティションパターンに基づいて、全体的深度ブロックに関するパーティションパターン(たとえば、線ベースのパーティションパターン)を決定することができる。ビデオエンコーダ20は、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ符号化することができる。

【0180】

[0185]ビデオエンコーダ20は、サブブロックに関するパーティションパターンを示す情報を符号化することがある。ビデオエンコーダ20はまた、深度ブロックに関するパーティションパターンを示す情報の符号化を回避する(たとえば、符号化しない)ことがある。ビデオエンコーダ20はまた、深度ブロックに関するパーティションパターンを深度ブロックのサイズに基づかせることなく(たとえば、ブロックサイズとは無関係に)、深度ブロックに関するパーティションパターンを決定することができる。ビデオエンコーダ20はまた、いくつかの例では、パーティションパターンが決定されたサブブロックを識別するための情報(たとえば、インデックスまたはフラグおよび1つのオフセット)を(たとえば、バイパスモードおよび/またはコンテキストモデリングを使用して)符号化し、シグナリングすることができる。

【0181】

[0186]図7は、深度コーディングのための技法を実装し得るビデオデコーダ30の一例を示すブロック図である。図7の例では、ビデオデコーダ30は、ビデオデータメモリ69と、エントロピー復号ユニット70と、動き補償ユニット72、動きベクトル予測ユニット73、およびイントラ予測ユニット74を含む予測処理ユニット71とを含む。ビデオデコーダ30はまた、逆量子化ユニット76と、逆変換処理ユニット78と、参照フレームメモリ82と、加算器80とを含む。ビデオデコーダ30は、いくつかの例では、ビデオエンコーダ20(図6)に関して説明した符号化パスとは概して逆の復号パスを実行することができる。動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルに基づいて予測データを生成することができ、イントラ予測ユニット74は、エントロピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて予測データを生成することができる。

【0182】

[0187]図7の例では、ビデオデータメモリ69は、符号化ビデオを受信する。ビデオデータメモリ69は、ビデオデコーダ30の構成要素によって復号されるために、符号化ビデオビットストリームなど、(たとえば、ビデオデータを記憶するように構成された)ビデオデータを記憶し得る。ビデオデータメモリ68はまた、ビデオデータを復号してピクチャを再構成するためにビデオデコーダ30の構成要素によって使用されるビデオデータを記憶することができる。たとえば、ビデオデータメモリ69は、異なるサイズのブロックに関するパーティションパターン(たとえば、ウェッジレットパーティションパターン)を記憶することができる。ビデオデータメモリ69は、ビデオエンコーダ30によって実施される初期化プロセスの一部としてパーティションパターンを記憶することができる。

【0183】

[0188]ビデオデータメモリ69に記憶されたビデオデータは、カメラなどのローカルビデオソースから、ビデオデータのワイヤードもしくはワイヤレスのネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって取得され得る。ビデオデータメモリ69は、符号化ビデオビットストリームからの符号化ビデオデータを記憶するコーディング済みピクチャバッファ(CPB)を形成し得る。

【0184】

[0189]参照ピクチャメモリ82は、(たとえば、イントラコーディングモードまたはインターコーディングモードで)ビデオデコーダ30によってビデオデータを復号する際に

10

20

30

40

50

使用するための参照ビデオデータを記憶する復号済みピクチャバッファ(D P B)の一例である。ビデオデータメモリ 6 9 および参照ピクチャメモリ 8 2 は、同期 D R A M (S D R A M)、磁気抵抗 R A M (M R A M)、抵抗 R A M (R R A M)、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(D R A M)など、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ 6 9 および参照ピクチャメモリ 8 2 は、同じメモリデバイスまたは別個のメモリデバイスによって設けられ得る。様々な例では、ビデオデータメモリ 6 9 は、ビデオデコーダ 3 0 の他の構成要素とともにオンチップであってよく、またはそれらの構成要素に対してオフチップであってもよい。

【 0 1 8 5 】

10

[0190]復号プロセス中に、ビデオデコーダ 3 0 はビデオエンコーダ 2 0 から、符号化ビデオスライスのビデオブロックと関連付けられるシンタックス要素とを表す符号化ビデオビットストリームを受信する。ビデオデコーダ 3 0 のエントロピー復号ユニット 7 0 は、量子化された係数と、動きベクトルまたはイントラ予測モードインジケータと、他のシンタックス要素とを生成するために、ビットストリームをエントロピー復号する。エントロピー復号ユニット 7 0 は、動きベクトルと他のシンタックス要素とを動き補償ユニット 7 2 に転送する。エントロピー復号ユニット 7 0 は、イントラ予測ユニット 7 4 にイントラ予測復号のための情報を転送する。ビデオデコーダ 3 0 は、ビデオスライスレベルおよび/またはビデオブロックレベルでシンタックス要素を受信することができる。

【 0 1 8 6 】

20

[0191]上述のように、ビデオデコーダ 3 0 は、マルチビュービデオコーディングを実行するように適応され得る。いくつかの例では、ビデオデコーダ 3 0 は、マルチビュー H E V C を復号するように構成され得る。H E V C - 3 D の場合、各ビューに対するテクスチャ値(すなわち、ルーマ値およびクロマ値)を復号することに加えて、ビデオデコーダ 3 0 はさらに、各ビューに対する深度マップを復号し得る。

【 0 1 8 7 】

[0192]いずれの場合も、ビデオスライスがイントラコーディングされた(I)スライスとしてコーディングされるとき、イントラ予測ユニット 7 4 は、シグナリングされたイントラ予測モードと、現在のフレームまたはピクチャの以前復号されたブロックからのデータとに基づいて、現在のビデオスライスのビデオブロックのための予測データを生成し得る。

30

【 0 1 8 8 】

[0193]イントラ予測ユニット 7 4 はまた、深度データをイントラコーディングし得る。たとえば、イントラ予測ユニット 7 4 は、深度スライスのイントラ予測された P U をコーディングするために、(たとえば、上記で図 2 に関して説明したように)ベース(2 D) H E V C 規格からのイントラ予測モードと、(たとえば、上記で図 3 A および図 3 B に関して説明したように)深度モデリングモード(D M M)と、(たとえば、以下で図 1 0 に関する説明するように)領域境界チェーンコーディングとを使用する。

【 0 1 8 9 】

40

[0194]ビデオフレームがインターフォーマンスされた(すなわち、B (双方向予測された)、P (前のフレームから予測された) または G P B (一般化された P または B スライス))スライスとしてコーディングされるとき、動き補償ユニット 7 2 は、エントロピー復号ユニット 7 0 から受信された動きベクトルと他のシンタックス要素とに基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリストのうちの 1 つ内の参照ピクチャのうちの 1 つから生成され得る。ビデオデコーダ 3 0 は、参照フレームメモリ 8 2 に記憶された参照ピクチャに基づいて、デフォルト構成技法を使用して、参照フレームリスト、すなわち、リスト 0 とリスト 1 とを構成し得る。

【 0 1 9 0 】

[0195]動き補償ユニット 7 2 は、動きベクトルと他のシンタックス要素とをパースする

50

ことによって現在のビデオスライスのビデオブロックのための予測情報を決定し、復号されている現在のビデオブロックのための予測ブロックを生成するために、その予測情報を使用する。たとえば、動き補償ユニット72は、ビデオスライスのビデオブロックをコーディングするために使用される予測モード(たとえば、イントラまたはインター予測)と、インター予測スライスタイプ(たとえば、Bスライス、Pスライス、またはGPBスライス)と、スライスに関する参照ピクチャリストのうちの1つまたは複数に関する構成情報と、スライスの各インター符号化ビデオブロックに関する動きベクトルと、スライスの各インターコーディングされたビデオブロックに関するインター予測ステータスと、現在のビデオスライス中のビデオブロックを復号するための他の情報とを決定するために、受信されたシンタックス要素のうちのいくつかを使用する。

10

【0191】

[0196]動き補償ユニット72はまた、補間フィルタに基づいて、補間を実行することができる。動き補償ユニット72は、参照ブロックのサブ整数ピクセルのための補間値を計算するために、ビデオブロックの符号化中にビデオエンコーダ20によって使用された補間フィルタを使用し得る。この場合に、動き補償ユニット72は、受信されたシンタックス要素からビデオエンコーダ20によって使用された補間フィルタを決定し、予測ブロックを生成するために補間フィルタを使用し得る。

【0192】

[0197]逆量子化ユニット76は、ビットストリーム中で提供され、エントロピー復号ユニット80によって復号された、量子化変換係数を逆量子化(inverse quantize)、すなわち、逆量子化(de-quantize)する。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するための、ビデオスライス中の各ビデオブロックに関してビデオデコーダ30によって計算される量子化パラメータ Q_{PY} の使用を含み得る。

20

【0193】

[0198]逆変換処理ユニット78は、ピクセル領域において残差ブロックを生成するために、逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換プロセスを変換係数に適用する。動き補償ユニット72またはイントラ予測ユニット74が、動きベクトルまたは他のシンタックス要素に基づいて現在のビデオブロック(たとえば、テクスチャブロックまたは深度ブロック)のための予測ブロックを生成した後、ビデオデコーダ30は、逆変換ユニット78からの残差ブロックを、動き補償ユニット82またはイントラ予測ユニット74によって生成された対応する予測ブロックと加算することによって、復号されたビデオブロックを形成する。加算器90は、この加算演算を実行する1つまたは複数の構成要素を表す。

30

【0194】

[0199]所望される場合、ブロッキネスアーティファクトを除去するために復号されたブロックをフィルタ処理するためのデブロッキングフィルタも適用され得る。ピクセル推移を平滑化し、または他の形でビデオ品質を改善するために、他のループフィルタ(コーディングループ内またはコーディングループの後のいずれであれ)も使用され得る。その後、所与のフレームまたはピクチャ中の復号されたビデオブロックは、参照ピクチャメモリ82に記憶され、参照ピクチャメモリ82は、後続の動き補償に使用される参照ピクチャを記憶する。参照ピクチャメモリ82はまた、復号されたビデオを図1のディスプレイデバイス32などのディスプレイデバイス上に後で表示できるように記憶する。

40

【0195】

[0200]この意味で、ビデオデコーダ30は、本開示で説明する1つまたは複数の例示的技法を実装するように構成され得る。たとえば、イントラ予測ユニット72は、本開示で説明する深度ブロックのための例示的なイントラ予測復号技法を実装するように構成され得る。いくつかの例では、イントラ予測ユニット72は、単独で、または他のプロセッサとの組合せで、本開示で説明する技法を実装するように構成され得る。

【0196】

50

[0201]いくつかの例では、ビデオデコーダ30は、第1のサイズのブロックに関するパーティションパターンからパーティションパターンを決定する（たとえば、サイズ 16×16 のブロックに関するパーティションパターンを決定する）ように構成され得る。ビデオデコーダ30は、第1のサイズのブロックに関するパーティションパターンからの決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションを決定する（たとえば、サイズ 16×16 のブロックの決定されたパーティションパターンに基づいて、サイズ 32×32 の深度ブロックに関するパーティションを決定する）ことができ、ここで、第2のサイズが第1のサイズよりも大きい。ビデオデコーダ30は、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測復号することができる。

10

【0197】

[0202]この例では、サイズ 16×16 のブロック（たとえば、第1のサイズのブロック）は、深度ブロックを含むピクチャ内の実際のブロックではないことがある。そうではなく、サイズ 16×16 のブロックは概念的ブロックであり、サイズ 16×16 のブロックに使用されるパーティションパターンが代わりにサイズ 32×32 のブロック（たとえば、第2のサイズの深度ブロック）に使用される。ただし、いくつかの例では、サイズ 16×16 のブロックは、ピクチャの実際のブロック、またはサイズ 32×32 のブロック内のブロックであり得る。

【0198】

[0203]たとえば、ビデオデコーダ30は、イントラ深度復号のために深度ブロック（たとえば、現在のPUである $N \times N$ ブロック）のサブブロック（たとえば、 $M \times M$ ブロック）に関するパーティションパターンを決定する（たとえば、パーティションパターンを示す情報を受信する）ように構成され得る。パーティションパターンは、線ベースのパーティションパターンであり得る。ビデオデコーダ30は、サブブロックに関するパーティションパターンに基づいて、全体的深度ブロックに関するパーティションパターン（たとえば、線ベースのパーティションパターン）を決定することができる。ビデオデコーダ30は、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ復号することができる。

20

【0199】

[0204]ビデオデコーダ30は、サブブロックに関するパーティションパターンを示す情報を復号することがある。ビデオデコーダ30はまた、深度ブロックに関するパーティションパターンを示す情報の復号を回避する（たとえば、復号しない）ことがある。言い換えるれば、ビデオデコーダ30は、深度ブロックに関するパーティションパターンを示す情報を受信することなく、深度ブロックに関するパーティションパターンを決定することができる。

30

【0200】

[0205]ビデオデコーダ30はまた、深度ブロックに関するパーティションパターンを深度ブロックのサイズに基づかせることなく（たとえば、ブロックサイズとは無関係に）、深度ブロックに関するパーティションパターンを決定することができる。ビデオデコーダ30はまた、いくつかの例では、パーティションパターンが決定されたサブブロックを識別するための受信された情報（たとえば、インデックスまたはフラグおよび1つのオフセット）を（たとえば、バイパスモードおよび/またはコンテキストモデリングを使用して）復号することができる。

40

【0201】

[0206]上記で説明した技法は、その両方が一般にビデオコーダと呼ばれることがある、ビデオエンコーダ20（図1および図6）および/またはビデオデコーダ30（図1および図7）によって実行され得る。さらに、ビデオコーディングは、概して、適用可能な場合、ビデオ符号化および/またはビデオ復号を指すことがある。

【0202】

[0207]図8は、ビデオデータを復号する例示的な方法を示すフローチャートである。図

50

示のように、ビデオデコーダ30は、第1のサイズのブロックに関するパーティションパターンを構成し、記憶することができる(800)。たとえば、ビデオデコーダ30は、初期化の一部として第1のサイズのブロックに関連する(たとえば、サイズ 16×16 のブロックに関する)1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成し、第1のサイズのブロックに関するパーティションパターンリストをビデオメモリデータ69に記憶することができる。

【0203】

[0208]ビデオデコーダ30は、第1のサイズのブロックに関するパーティションパターンを決定することができる(802)。たとえば、ビデオデコーダ30は、第1のサイズのブロックに関するパーティションパターンリストへのインデックスを受信することができる。ビデオデコーダ30は、パーティションパターンリストへのインデックスによって参照されるパーティションパターンを識別することができる。ビデオデコーダ30は、識別されたパーティションパターンに基づいて、第1のサイズのブロックに関するパーティションパターンからパーティションパターンを決定することができる(すなわち、ビデオデコーダ30は、パーティションパターンリストへの受信されたインデックスに基づいて、第1のサイズのブロックに関するパーティションパターンからパーティションパターンを決定することができる)。

10

【0204】

[0209]ビデオデコーダ30は、第1のサイズ(たとえば、 16×16)のブロックに関するパーティションパターンからの決定されたパーティションパターンに基づいて、第2のサイズ(たとえば、 32×32)の深度ブロックに関するパーティションパターンを決定することができる(804)。この例では、第2のサイズは、第1のサイズよりも大きい。

20

【0205】

[0210]第1のサイズのブロックは必ずしも、ピクチャ内の実際のブロックである必要はない。むしろ、ビデオデコーダ30が第2のより大きいサイズの深度ブロックに関するパーティションパターンを決定するために使用するのは、第1のサイズのブロックのパーティションパターンである。ただし、いくつかの例では、第1のサイズのブロックは、深度ブロック内などのピクチャ内の実際のブロックであり得る。

【0206】

30

[0211]いくつかの例では、ビデオデコーダ30は、深度ブロック内のサブブロックを識別する情報を受信することができ、サブブロックのサイズが第1のサイズである。そのような例では、ビデオデコーダ30は、サブブロックに関するパーティションパターンを決定することができる。ビデオデコーダ30はまた、サブブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することができる。また、これらの例示的な技法では、ビデオデコーダ30は、第2のサイズの深度ブロックに関するパーティションパターンを、サブブロックに関するパーティションパターンの線形線を深度ブロックの境界まで拡張することによって決定することができる。得られる線形線は、深度ブロックに関するパーティションパターンを備える。

40

【0207】

[0212]ビデオデコーダ30は、深度ブロックに関する決定されたパーティションパターンに基づいて、深度ブロックをイントラ予測復号することができる(806)。たとえば、ビデオデコーダ30は、深度ブロックをイントラ予測復号するために、DMMに依拠する3D-HAVECにおいて説明されるイントラ予測復号技法を実施することができる。いくつかの例では、第1のサイズのブロックに関するパーティションパターンからのパーティションパターンは、第1のサイズのブロックに関する複数の線ベースのパーティションパターンからの線ベースのパーティションパターン(たとえば、第1のサイズのブロックに関する複数のウェッジレットパターンからのウェッジレットパターン)を備える。また、深度ブロックに関するパーティションパターンは、2つのパーティションに深度ブロッ

50

クを区分する深度ブロックに関する線ベースのパーティションパターンを備える。いくつかの例では、ビデオデコーダ30は、第2のサイズの深度ブロックに関するパーティションパターンを示す情報を受信することなく、第2のサイズの深度ブロックに関するパーティションパターンを決定することができる。

【0208】

[0213]図9は、ビデオデータを符号化する例示的な方法を示すフローチャートである。図示のように、ビデオエンコーダ20は、第1のサイズのブロックに関連するパーティションパターンを構成し、記憶することができる(900)。たとえば、ビデオエンコーダ20は、初期化の一部として第1のサイズのブロックに関連する(たとえば、サイズ 16×16 のブロックに関する)パーティションパターンリストを構成し、第1のサイズのブロックに関するパーティションパターンリストをビデオメモリデータ39に記憶することができる。
10

【0209】

[0214]ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンを決定することができる(902)。たとえば、ビデオエンコーダ20は、第2のより大きいサイズの深度ブロックに関するパーティションパターンを決定するために使用され得る適切なパーティションパターンを選択するために、複数の符号化パスを実施することができる。次いでビデオエンコーダ20は、パーティションパターンリスト中でパーティションパターンを識別することができる。たとえば、ビデオエンコーダ20は、構成されたパーティションパターンリスト中で識別されたパーティションパターンから、第1のサイズのブロックに関するパーティションパターンからのパーティションパターンを決定することができる。ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンからの決定するパーティションパターンを識別する構成されたパーティションパターンリストへのインデックスをシグナリングすることができ、ビデオデコーダ30が、深度ブロックをイントラ予測復号するためにそのインデックスを使用する。
20

【0210】

[0215]ビデオエンコーダ20は、第1のサイズのブロックに関するパーティションパターンからの決定されたパーティションパターン(たとえば、サイズ 16×16 のブロックに関するパーティションパターン)に基づいて、第2のサイズ(たとえば、 32×32)の深度ブロックに関するパーティションパターンを決定することができる(904)。この例では、第2のサイズは、第1のサイズよりも大きい。
30

【0211】

[0216]第1のサイズのブロックは必ずしも、ピクチャ内の実際のブロックである必要はない。むしろ、ビデオエンコーダ20が第2のより大きいサイズの深度ブロックに関するパーティションパターンを決定するために使用するのは、第1のサイズのブロックのパーティションパターンである。ただし、いくつかの例では、第1のサイズのブロックは、深度ブロック内などのピクチャ内の実際のブロックであり得る。

【0212】

[0217]いくつかの例では、ビデオエンコーダ20は、深度ブロック内のサブブロックを識別することができ、サブブロックのサイズが第1のサイズである。そのような例では、ビデオエンコーダ20は、識別されたサブブロックに関するパーティションパターンを決定することができる。ビデオエンコーダ20はまた、識別されたサブブロックに関する決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することができる。また、これらの例示的な技法では、ビデオエンコーダ20は、第2のサイズの深度ブロックに関するパーティションパターンを、サブブロックに関するパーティションパターンの線形線を深度ブロックの境界まで拡張することによって決定することができる。得られる線形線は、深度ブロックに関するパーティションパターンを画定する。
40

【0213】

[0218]ビデオエンコーダ20は、深度ブロックに関する決定されたパーティションパタ
50

ーンに基づいて、深度ブロックをイントラ予測符号化することができる(906)。たとえば、ビデオエンコーダ20は、深度ブロックをイントラ予測符号化するために、DMMに依拠する3D-HEVCにおいて説明されるイントラ予測符号化技法を実施することができる。いくつかの例では、第1のサイズのブロックに関するパーティションパターンからのパーティションパターンは、第1のサイズのブロックに関する複数の線ベースのパーティションパターンからの線ベースのパーティションパターン(たとえば、第1のサイズのブロックに関する複数のウェッジレットパターンからのウェッジレットパターン)を備える。また、深度ブロックに関するパーティションパターンは、2つのパーティションに深度ブロックを区分する深度ブロックに関する線ベースのパーティションパターンを備える。いくつかの例では、ビデオエンコーダ20は、第2のサイズの深度ブロックに関するパーティションパターンを識別する情報をシグナリングしないことがある。

【0214】

[0219]以下では、本開示で説明する例示的な技法を実施するいくつかの例について説明する。以下では、ポールド体およびイタリック体の文字を包含する二重角括弧([])は、現在の技法に対する削除を示し、下線は、現在の技法に対する追加を示す。以下の例示的な技法が、別個に、または組合せで適用され得る。

【0215】

[0220]深度モードパラメータシンタックスの場合

【0216】

【表5】

10

20

30

40

記述子
depth_mode_parameters(x0 , y0 , log2CbSize) {
if((log2CbSize == 3 && PartMode[xC][yC] == PART_2Nx2N) (log2CbSize> 3 && log2CbSize< 6))
depth_intra_mode_set_indication_flag
depth_intra_mode[x0][y0]
if(DepthIntraMode[x0][y0] == INTRA_DEP_DMM_WFULL DepthIntraMode[x0][y0] == INTRA_DEP_SDC_DMM_WFULL){
wedge_full_tab_idx[x0][y0]
<u>wedge_sub_col[x0][y0]</u>
<u>wedge_sub_row[x0][y0]</u>
}
else if(DepthIntraMode[x0][y0] == INTRA_DEP_DMM_WPREDTEX){
wedge_predtex_tab_idx[x0][y0]
<u>wedge_sub_col[x0][y0]</u>
<u>wedge_sub_row[x0][y0]</u>
} else if(DepthIntraMode[x0][y0] == INTRA_DEP_CHAIN) {
edge_start_left_flag[x0][y0]
edge_start_position[x0][y0]
num_edge_codes_minus1[x0][y0]
for(k = 0; k <= num_edge_codes_minus1; k++)
edge_code[k]
}
...
}

【0217】

[0221]深度モードパラメータイントラセマンティクスの場合：

 w e d g e _ f u l l _ t a b _ i d x [x 0] [y 0] は、 D e p t h I n t r a M o

50

`d e [x 0] [y 0]` が `I N T R A _ D E P _ D M M _ W F U L L` に等しいときの、対応する 4×4 パターンリストにおけるウェッジレットパターンのインデックスを指定する。`w e d g e _ s u b _ c o l [x 0] [y 0]` および `w e d g e _ s u b _ r o w [x 0] [y 0]` は、ウェッジレットパターンがシグナリングされる 4×4 ブロックの位置を指定する。

`w e d g e _ p r e d t e x _ t a b _ i d x [x 0] [y 0]` は、`D e p t h I n t r a M o d e [x 0] [y 0]` が `I N T R A _ D E P _ D M M _ W P R E D T E X` に等しいときの、現在の予測ユニット内の 4×4 ブロックのウェッジレットパターンのインデックスを指定する。

...

10

[0222] 復号プロセスの場合

`H . 8 . 4 . 4 . 2 . 7` イントラ予測モード `I N T R A _ D M M _ W F U L L` の指定 このプロセスへの入力は以下の通りである。

- 現在のピクチャの左上サンプルに対する現在のブロックの左上サンプルを指定するサンプルロケーション (`x T b , y T b`)
- $x = -1$ 、 $y = -1 \dots n T b S \times 2 - 1$ 、および $x = 0 \dots n T b S \times 2 - 1$ 、 $y = -1$ である、隣接サンプル `p [x] [y]`
- 変換ブロックサイズを指定する変数 `n T b S`

このプロセスの出力は以下の通りである。

- $x, y = 0 \dots n T b S - 1$ である、予測サンプル `p r e d S a m p l e s [x] [y]`

20

$x, y = 0 \dots n T b S - 1$ である予測サンプル `p r e d S a m p l e s [x] [y]` の値は、次の順序のステップによって指定されるように導出される。

1. 下位条項 `H . 8 . 4 . 4 . 2 . 1 2 . 3` に指定されているウェッジレットパターン拡張プロセスは、`L o g 2 (n T b S)`、`w e d g e _ f u l l _ t a b _ i d x [x T b] [y T b]`、`w e d g e _ s u b _ c o l [x T b] [y T b]` および `w e d g e _ s u b _ r o w [x T b] [y T b]` を入力として引き起こされ、出力は、`w e d g e P a t t e r n` に割り当てられる。[[2進パーティションパターンを指定する、 $x, y = 0 \dots n T b S - 1$ である、変数 `w e d g e P a t t e r n [x] [y]` は、以下のように導出される。]

30

【0218】

【数1】

wedgePattern

```
= WedgePatternTable[ Log2( n T b S ) // wedge_full_tab_idx[ x T b ] // y T b ] //  
(H-39)]
```

【0219】

40

2. 下位条項 `H . 8 . 4 . 4 . 2 . 1 1` に指定されている深度パーティション値導出および割当てプロセスは、隣接サンプル `p [x] [y]`、2進パターン `w e d g e P a t t e r n [x T b] [y T b]`、変換サイズ `n T b S`、`d m m _ d c _ f l a g [x T b] [y T b]` に等しく設定された `d c O f f s e t A v a i l F l a g`、0 に等しく設定された `i n t r a C h a i n F l a g`、ならびに DC オフセットの `D c O f f s e t [x T b] [y T b] [0]` および `D c O f f s e t [x T b] [y T b] [1]` を入力として引き起こされ、出力は、`p r e d S a m p l e s [x] [y]` に割り当てられる。

3. [[$x, y = 0 \dots n T b S - 1$ (両端値を含む) として、以下が適用される。]

- `W e d g e I d x [x T b + x] [y T b + y]` は、`w e d g e _ f u l l _ t a b _ i d x [x T b] [y T b]` に等しく設定される。]]

50

H . 8 . 4 . 4 . 2 . 8 イントラ予測モード I N T R A _ D M M _ W P R E D T E X の指定

このプロセスへの入力は以下の通りである。

- 現在のピクチャの左上サンプルに対する現在のブロックの左上サンプルを指定するサンプルロケーション (x Tb, y Tb)
- $x = -1, y = -1 \dots nTbS \times 2 - 1$ 、および $x = 0 \dots nTbS \times 2 - 1, y = -1$ である、隣接サンプル p [x] [y]
- 変換ブロックサイズを指定する変数 nTbS

このプロセスの出力は以下の通りである。

- $x, y = 0 \dots nTbS - 1$ である、予測サンプル predSamples [x] [y]

$x, y = 0 \dots nTbS - 1$ である予測サンプル predSamples [x] [y] の値は、以下の通りである。

- textureIntraPredMode に応じて、変数 wedgeIdx が、以下で指定されるように導出される。

【0220】

【数2】

candWedgeIndList =

TextureModePredWedgeIndTable[[[*Log2(nTbS)*]]2][TextureIntraPredMode
[xTb][yTb]] (H-40)

20

【0221】

【数3】

wedgeIdx = candWedgeIndList[wedge_predtex_idx[xTb][yTb]]

(H-41)

30

【0222】

- 下位条項 H . 8 . 4 . 4 . 2 . 1 2 . 3 に指定されているウェッジレットパターン拡張プロセスは、*Log2(nTbS)*、wedgeIdx、wedge_sub_col [xTb][yTb] および wedge_sub_row [xTb][yTb] を入力として引き起こされ、出力は、wedgePattern に割り当てられる。[[2進パーティションパターンを指定する、 $x, y = 0 \dots nTbS - 1$ である、変数 wedgePattern [x] [y]]] は、以下のように導出される。

【0223】

【数4】

40

wedgePattern = WedgePatternTable[*Log2(nTbS)*][wedgeIdx]
(H-42)]

【0224】

- 下位条項 H . 8 . 4 . 4 . 2 . 1 1 に指定されている深度パーティション値導出および割当てプロセスは、隣接サンプル p [x] [y]、2進パターン wedgePattern [x] [y]、変換サイズ nTbS、dmm_dc_flag [xTb][yTb] に等しく設定された dcOffsetAvailableFlag、0 に等しく設定された intra

50

C h a i n F l a g、ならびにDCオフセットのD c O f f s e t [x T b] [y T b] [0]およびD c O f f s e t [x T b] [y T b] [1]を入力として引き起こされ、出力は、p r e d S a m p l e s [x] [y]に割り当てられる。

H . 8 . 4 . 4 . 2 . 1 2 . 3 ウエッジレットパターン拡張プロセス

このプロセスへの入力は以下の通りである。

- 2進パーティションパターンサイズを指定する変数l o g 2 B l k S i z e
- 4×4 ウエッジレットインデックスを指定する変数s u b I d x
- 左上 4×4 ブロックの水平位置を指定する変数x S u b
- 左上 4×4 ブロックの垂直位置を指定する変数y S u b

このプロセスの出力は以下の通りである。

- サイズ($1 << \log 2 B l k S i z e$) \times ($1 << \log 2 B l k S i z e$)の2進アレイw e d g e P a t t e r n [x] [y]

4×4 ウエッジレットパターンのパーティション線開始位置を指定する変数x S およびy S は、次のように導出される。

【0 2 2 5】

【数5】

- xS = W e d g e S t a r t P o s [2] [sub I d x] [0];
- yS = W e d g e S t a r t P o s [2] [sub I d x] [1];

10

20

【0 2 2 6】

4×4 ウエッジレットパターンのパーティション線終了位置を指定する変数x E およびy E は、次のように導出される。

【0 2 2 7】

【数6】

- xE = W e d g e E n d P o s [2] [sub I d x] [0];
- yE = W e d g e E n d P o s [2] [sub I d x] [1];

30

【0 2 2 8】

l o g 2 B l k S i z eが2に等しい場合、以下が適用される。

- W e d g e P a t t e r n T a b l e [2] [s u b I d x]がw e d g e P a t t e r nに割り当てられる。

そうでない場合、以下が適用される。

- x S ! = x Eまたはy S ! = y Eの場合、以下が適用される。

【0 2 2 9】

【数7】

- a = (y E - y S);
- b = (y E + (y S u b << 3)) \times (x E - x S) - a \times (x E + (x S u b << 3));
- x, y = 0 .. blkSize - 1として
- w e d g e P a t t e r n [x] [y] = = ((y \times (x E - x S) - a \times x) << 1) < b ? 1 : 0

40

【0 2 3 0】

50

- そうでない場合、以下が適用される。

【0 2 3 1】

【数8】

$x, y = 0..blkSize - 1$ として

【0 2 3 2】

- $ySub == 0$ の場合、以下が適用される。

【0 2 3 3】

【数9】

10

- wedgePattern[x][y] = $x < (xSub << 2) ? 1 : 0$

【0 2 3 4】

- そうでない場合、以下が適用される。

【0 2 3 5】

【数10】

20

wedgePattern[x][y] = $y < (ySub << 2) ? 1 : 0$

【0 2 3 6】

H . 8 . 4 . 4 . 3 深度値再構成プロセス

このプロセスへの入力は以下の通りである。

- 現在のピクチャの左上ルーマサンプルに対する現在のブロックの左上ルーマサンプルを指定するルーマロケーション (xTb , yTb)
- 変換ブロックサイズを指定する変数 $nTbS$
- $x, y = 0 .. nTbS - 1$ である、予測サンプル $\text{predSamples}[x][y]$
- イントラ予測モード predModeIntra

このプロセスの出力は以下の通りである。

- $x, y = 0 .. nTbS - 1$ である、再構成された深度値サンプル $\text{resSamples}[x][y]$

predModeIntra に応じて、2進セグメント化パターンを指定する、 $x, y = 0 .. nTbS - 1$ である、アレイ $\text{wedgePattern}[x][y]$ は、次のように導出される。

- predModeIntra が INTRA_DMM_WFULL に等しい場合、下位条項 H . 8 . 4 . 4 . 2 . 12 . 3 に指定されているウェッジレットパターン拡張プロセスは、 $\text{Log2}(nTbS)$ 、 $\text{wedge_full_table_idx}[xTb][yTb]$ 、 $\text{wedge_sub_col}[xTb][yTb]$ および $\text{wedge_sub_row}[xTb][yTb]$ を入力として引き起こされ、出力は、 wedgePattern に割り当てられる。[[以下が適用される。]]

【0 2 3 7】

40

【数11】

wedgePattern

= WedgePatternTable[Log2(nTbS) // wedge_full_tab_idx[xTb][yTb]]

【0238】

- そうでない(predModeIntraがINTRA_DCに等しい場合、以下が適用される)場合、以下が適用される。
- $x, y = 0 \dots nTbS - 1$ として、wedgePattern[x][y]が0に等しく設定される。
dlt_flag[nuh_layer_id]に応じて、再構成された深度値サンプルresSamples[x][y]が、以下で指定されるように導出される。
- dlt_flag[nuh_layer_id]が0に等しい場合、以下が適用される。
- $x, y = 0 \dots nTbS - 1$ として、再構成された深度値サンプルresSamples[x][y]が、以下で指定されるように導出される。

【0239】

【数12】

20

$$\text{resSamples}[x][y] = \text{predSamples}[x][y] + \\ \text{SdcResidual}[xTb][yTb][\text{wedgePattern}[x][y]] \quad (\text{H-59})$$

【0240】

- そうでない(dlt_flag[nuh_layer_id]が1に等しい)場合、以下が適用される。
- 変数dcPred[0]およびdcPred[1]が、以下で指定されるように導出される。

30

【0241】

【数13】

- predModeIntraがINTRA_DCに等しい場合、以下が適用される

$$\text{dcPred}[0] = \text{predSamples}[nTbS - 1][nTbS - 1] \quad (\text{H-60})$$

【0242】

- そうでなく、predModeIntraがINTRA_PLANARに等しい場合、以下が適用される。

40

【0243】

【数14】

$$\text{dcPred}[0] = (\text{predSamples}[0][0] + \text{predSamples}[0][nTbS - 1] + \\ \text{predSamples}[nTbS - 1][0] \\ + \text{predSamples}[nTbS - 1][nTbS - 1] + 2) \gg 2 \quad (\text{H-61})$$

【0244】

50

- そうでない（predModeIntraがINTRA_DMM_WFULLに等しい）場合、以下が適用される。

【0245】

【数15】

$$\text{dcPred[wedgePattern[0][0]]} = \text{predSamples[0][0]} \quad (\text{H-62})$$

【0246】

【数16】

$$\text{dcPred[wedgePattern[nTbS-1][0]]} = \text{predSamples[nTbS-1][0]} \quad (\text{H-63})$$

【0247】

【数17】

$$\text{dcPred[wedgePattern[0][nTbS-1]]} = \text{predSamples[0][nTbS-1]} \quad (\text{H-64})$$

【0248】

【数18】

$$\text{dcPred[wedgePattern[nTbS-1][nTbS-1]]} = \text{predSamples[nTbS-1][nTbS-1]} \quad (\text{H-65})$$

【0249】

- $x, y = 0 \dots nTbS - 1$ として、再構成された深度値サンプルresSamplees[x][y]が、以下で指定されるように導出される。

【0250】

【数19】

$$\text{dltIdxPred} = \text{DepthValue2Idx[dcPred[wedgePattern[x][y]]]} \quad (\text{H-66})$$

【0251】

【数20】

$$\text{dltIdxResi} = \text{SdcResidual[xTb][yTb][wedgePattern[x][y]]} \quad (\text{H-67})$$

【0252】

【数21】

```
resSamples[ x ][ y ] = predSamples[ x ][ y ] +
Idx2DepthValue[ dltIdxPred + dltIdxResi ] -
dcPred[ wedgePattern[ x ][ y ] ]      (H-68)
```

【0253】

[0223]第2の例示的な技法では、第1の例示的な技法に類似するが、現在の3D-HEVCの復号プロセスに対する以下の変更も含まれる。この第2の例示的な技法は、 $N \times N$ ウェッジレットパターンにより $N \times N$ のPUにおいて位置(m, n)がどのパーティションに属するかを、対応するウェッジレットパターンにより $2N \times 2N$ のPUにおいて決定位置($2m, 2n$)がどのパーティションに属するかと、 $2N \times 2N$ のブロックにおける決定位置($2m, 2n$)の移動、すなわち、($2m, 2n$)に加えられる(オフセットX, オフセットY)のオフセットベクトルとに従って導出するために説明され、ここにおいて、オフセットXまたはオフセットYが0または1に等しく、現在の3D-HEVCの場合のような様々な条件への依存が取り除かれる。いくつかの例では、この変更は、例1に関して上記で説明した技法とともに行われ得る。他の例では、この変更は、第1の例示的な技法において説明した技法とは無関係に行われることもある。

H . 8 . 4 . 4 . 2 . 1 2 . 1 ウェッジレットパターン生成プロセス

...

4 . x , y = 0 . . patternSize - 1 である、2進パーティションパターンwedgePattern[x][y]が、以下で指定されるように導出される。

- resShiftが1に等しい場合、以下が適用される。

- [[wedgeOriに応じて、変数xOFFおよびyOFFが表H-8に指定されているように設定される。

【0254】

【表6】

10

20

30

表H-8 xOFF, yOFFの指定

wedgeOri	(xS + xE) < curSize	xOff	yOff
0		0	0
1		1	0
2		1	1
3		0	1
4	0	1	0
4	1	0	0
5	0	0	1
5	1	0	0

40

【0255】

-]]

50

- $x, y = 0 \dots \text{patternSize} - 1$ として、以下が適用される。
- 【0256】
【数22】

```
wedgePattern[x][y] =  
curPattern[(x << 1) + [xOff]][(y << 1) [+yOff]] (H-58)
```

- 【0257】
- そうでない (ressShift が 1 に等しくない) 場合、wedgePattern 10
は curPattern に等しく設定される。

【0258】
[0224] 第 1 の例示的な技法に類似するが、(第 1 の例示的な技法に対する、下線および
ボールド体とともに示される) 以下の変更はまた、 $N \times N$ ($N > 4$) ウェッジレットパターン
のための提案される方法における 1 サンプルパーティション (パーティションパターン
における左上サンプルは、すべての他のサンプルの異なるパーティションに属する) を
カバーするようにさらに含まれ、そのような 1 サンプルパーティションは、第 1 の例示的
な技法を使用して生成されることが可能ではないことがある。

H . 8 . 4 . 4 . 2 . 1 2 . 3 ウェッジレットパターン拡張プロセス
このプロセスへの入力は以下の通りである。 20

- 2 進パーティションパターンサイズを指定する変数 log2BlockSize
- 4×4 ウェッジレットインデックスを指定する変数 subIdx
- 左上 4×4 ブロックの水平位置を指定する変数 xSub
- 左上 4×4 ブロックの垂直位置を指定する変数 ySub

このプロセスの出力は以下の通りである。

- サイズ ($1 << \log2BlockSize \times 1 << \log2BlockSize$) の $2^{2 \log2BlockSize}$ の 4×4 ウェッジレットパターンのパーティション線開始位置を指定する変数 xS および yS
は、次のように導出される。

【0259】
【数23】 30

- xS = WedgeStartPos[2][subIdx][0];
- yS = WedgeStartPos[2][subIdx][1];

【0260】
 4×4 ウェッジレットパターンのパーティション線終了位置を指定する変数 xE および yE
は、次のように導出される。

【0261】
【数24】 40

- xE = WedgeEndPos[2][subIdx][0];
- yE = WedgeEndPos[2][subIdx][1];

【0262】
log2BlockSize が 2 に等しい場合、以下が適用される。
- WedgePatternTable[2][subIdx] が wedgePat

e r n に割り当てられる。

そうでない場合、以下が適用される。

- x S ! = x E または y S ! = y E の場合、以下が適用される。

【0 2 6 3】

【数 2 5】

- a = (yE - yS);
- b = (yE + (ySub << 3)) × (xE - xS) - a × (xE + (xSub << 3));
- x, y = 0..blkSize - 1 として
- wedgePattern[x][y] == ((y × (xE - xS) - a × x) << 1) < b ? 1 : 0

10

【0 2 6 4】

- そうでない場合、以下が適用される。

【0 2 6 5】

【数 2 6】

- x, y = 0..blkSize - 1 として

20

【0 2 6 6】

- x S u b == 0 & & y S u b == 0 の場合、以下が適用される。

【0 2 6 7】

【数 2 7】

- wedgePattern[x][y] = (!x) && (!y) ? 1 : 0

【0 2 6 8】

- そうでなく、y S u b == 0 の場合、以下が適用される。

【0 2 6 9】

【数 2 8】

- wedgePattern[x][y] = x < (xSub << 2) ? 1 : 0

30

【0 2 7 0】

- そうでない場合、以下が適用される。

【0 2 7 1】

【数 2 9】

wedgePattern[x][y] = y < (ySub << 2) ? 1 : 0

40

【0 2 7 2】

[0225]本開示の技法について概して 3 D - H E V C について説明したが、本技法はこのように限定されない。上記で説明した技法はまた、他の現在の規格またはまだ開発されていない将来の規格に適用可能であり得る。たとえば、深度コーディングのための技法はまた、H E V C のマルチビュー拡張（たとえば、いわゆる M V - H E V C ）、H E V C へのスケーラブル拡張、または深度コンポーネントを有する他の現在もしくは将来の規格に適用可能であり得る。

【0 2 7 3】

50

[0226]以下では、3D - HEVCにおけるイントラ予測モードに関するいくつかのさらなる技法について説明する。本開示は、本開示で説明する技法を理解するのをさらに助けるために提供される。ただし、いくつかの技法は、本開示に必要とされず、本開示で説明する技法とともに利用され得るか、または本開示で説明する技法の別の例であり得る。

【0274】

[0227]領域境界チェーンコーディングモードの場合、3D - HEVCでは、領域境界チェーンコーディングモードは、深度スライスのイントラ予測ユニットをコーディングするためにHEVCイントラ予測モードおよびDMMモードとともに導入される。簡潔のために、「領域境界チェーンコーディングモード」は、本明細書で説明する本文、表および図において簡潔のために「チェーンコーディング」によって示される。

10

【0275】

[0228]ビデオエンコーダ20は、チェーンの開始位置、チェーンコードの数、およびチェーンコードごとの、および方向インデックスにより、PUのチェーンコーディングをシグナリングし得る。チェーンは、サンプルとその8連結性サンプルのうちの1つとの間の連結である。図10に示すように、8つの異なるタイプのチェーンがあり、各々は、0から7にわたる方向インデックスを割り当てられる。

【0276】

[0229]図10は、領域境界チェーンコーディングモードを示す概念図である。たとえば、図10の上部分は、チェーンコーディングにおいて定義される8個の可能なタイプのチェーンを示している。図10の下部分は、チェーンコーディングにおける1つの深度PUパーティションパターンとコーディングされたチェーンとを示している。

20

【0277】

[0230]チェーンコーディングプロセスの一例が図10に示されている。図10に示す任意のパーティションパターンをシグナリングするために、ビデオエンコーダ20は、パーティションパターンを識別することができ、ビットストリームにおける以下の情報を符号化する。ビデオエンコーダ20は、チェーンが上境界から開始することをシグナリングするために、1ビット「0」を符号化し得る。ビデオエンコーダ20は、上境界における開始位置「3」をシグナリングするために、3ビット「011」を符号化し得る。ビデオエンコーダ20は、チェーンの総数が7であることをシグナリングするために、4ビット「0110」を符号化し得る。ビデオエンコーダ30は、一連の連結チェーンインデックス「3、3、3、7、1、1、1」を符号化することができ、各チェーンインデックスが、ルックアップテーブルを使用してコードワードに変換される。

30

【0278】

[0231]3D - HEVCは、(たとえば、DMMに関して上記で説明した、コロケートテクスチャコンポーネントに基づく区分ではなく)パーティション境界の明示的シグナリングを可能にする領域境界チェーンコーディングモードを含む。上述のように、本開示では、「領域境界チェーンコーディングモード」を「チェーンコーディング」と呼ぶことがある。

【0279】

[0232]概して、チェーンは、サンプルとその8連結性サンプルのうちの1つとの間の連結である。図10のブロック160によって示されているように、8つの異なるチェーン方向タイプがあり、各々は、0から7にわたる方向インデックスを割り当てられる。チェーン方向タイプは、ビデオコーダが深度ブロックのパーティションを決定するのを助け得る。

40

【0280】

[0233]たとえば、図10の例は、区分構造を示すチェーン166によって分離された第1のパーティション162と第2のパーティション164とを含む。(ビデオエンコーダ20などの)ビデオエンコーダは、PUのためのチェーン166を決定し、それを符号化されたビットストリーム中でシグナリングし得、(ビデオデコーダ30などの)ビデオデコーダは、符号化ビットストリームからのチェーン166を表すデータをペースし得る。

50

【0281】

[0234]概して、チェーン166は、開始位置と、チェーン中のリンクの数（たとえば、チェーンコードの数）の指示と、チェーンコードごとの方向インデックスとを含む。チェーン166のための他のタイプのシグナリングも使用され得る。一例では、図10の例に示された任意のパーティションパターンをシグナリングするために、ビデオエンコーダ20は、チェーン166が上境界から開始することを示すために1ビット（たとえば、0）を符号化し得る。ビデオエンコーダ20は、チェーン166が上境界の第3の深度サンプルの後に開始することを示すために3ビット（たとえば、011）を符号化し得る。ビデオエンコーダ20は、チェーン166中に合計7つのリンクがあることを示すために4ビット（たとえば、0110）を符号化し得る。ビデオエンコーダ20はまた、（たとえば、ブロック160に従って）各チェーンリンクの方向を示すために一連の連結チェーンインデックス（たとえば、3, 3, 3, 7, 1, 1, 1）を符号化し得る。いくつかの例では、ビデオエンコーダ20は、ルックアップテーブルを使用して各インデックスをコードワードに変換し得る。ビデオデコーダ30などのビデオデコーダは、ブロックの区分パターンを決定するために、上記で説明したシグナリングをパースし得る。ビデオデコーダ30は、次いで、各パーティションのための深度値を復号し得る。

【0282】

[0235]本開示で説明する技法は、簡略深度コーディング（SDC）とともに機能することができるが、技法はそのように限定されず、SDCが利用されない例において機能し得る。以下では、簡略深度コーディング（SDC）モードについて説明する。SDCモードは、深度スライスのイントラPUをコーディングするために、HEVCイントラ予測モード、DMMモード、およびチェーンコーディングモードとともに導入されている。現在の3D-HEVCでは、SDCは、 $2N \times 2N$ PUパーティションサイズのためにのみ適用される。量子化変換係数をコーディングする代わりに、SDCモードは、以下の2つのタイプの情報を用いて深度ブロックを表す。（1）DMMモード1（2つのパーティション）と平面（1つのパーティション）とを含む、現在の深度ブロックのパーティションのタイプ、および（2）パーティションごとに、（ピクセル領域における）残差値がビットストリームにおいてシグナリングされる。

【0283】

[0236]それぞれ平面およびDMMモード1のパーティショントイプに対応するSDCモード1とSDCモード2とを含む2つのサブモードが、SDCにおいて定義される。簡略残差コーディングがSDCにおいて使用される。簡略残差コーディングでは、PUのパーティションごとに1つのDC残差値がシグナリングされ、変換または量子化は適用されない。

【0284】

[0237]例に応じて、本明細書で説明した方法のうちのいずれかのいくつかの動作またはイベントは、異なる順序で実行されてよく、互いに追加、統合、または完全に除外され得る（たとえば、すべての説明された動作またはイベントが、本方法の実施のために必要であるとは限らない）ことを理解されたい。その上、いくつかの例では、動作またはイベントは、連続してではなく、同時に、たとえば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて実行され得る。加えて、本開示の特定の態様が、明確さの目的のため、単一のモジュールまたはユニットによって実行されるものとして説明されているが、本開示の技法は、ビデオコーダに関連付けられているユニットまたはモジュールの組合せによって実行され得ることを理解されたい。

【0285】

[0238]技法の様々な態様の特定の組合せについて上記で説明したが、これらの組合せは、本開示で説明した技法の例を示すために与えられたものにすぎない。したがって、本開示の技法は、これらの例示的な組合せに限定されるべきでなく、本開示で説明した技法の様々な態様の任意の想起可能な組合せを包含し得る。

【0286】

10

20

30

40

50

[0239] 1つまたは複数の例では、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、また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、コンピュータ可読媒体上の1つもしくは複数の命令またはコード上に記憶され、あるいはこれを介して伝送され、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の媒体に対応するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従う、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容易にする任意の媒体を含む通信媒体を含み得る。

【0287】

[0240] このようにして、コンピュータ可読媒体は、一般に、(1)非一時的である有形のコンピュータ可読記憶媒体、または(2)信号または搬送波のような通信媒体に対応し得る。データ記憶媒体は、本開示で説明する技法の実装のための命令、コードおよび／またはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしく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記憶媒体とパッケージング材料とを含み得る。

10

【0288】

[0241] 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、RAM、ROM、EEPROM(登録商標)、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージ、もしくは他の磁気ストレージデバイス、フラッシュメモリ、または、命令またはデータ構造の形態の所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得る。また、任意の接続が、コンピュータ可読媒体と適切に呼ばれる。たとえば、命令が、ウェブサイト、サーバ、または他の遠隔ソースから、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)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、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。

20

【0289】

[0242] ただし、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的媒体を含まないが、代わりに、非一時的な有形記憶媒体を対象とするなどを理解されたい。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(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)、およびbulk-rayディスク(disc)を含み、ディスク(disk)は、通常、データを磁気的に再生し、ディスク(disc)は、データをレーザー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の中に含まれるべきである。

30

【0290】

[0243] 命令は、1つもしく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(ASSIC)、フィールドプログラマブルゲートアレイ(FPGA)、または他の同等の集積回路もしくはディスクリート論理回路などの1つまたは複数のプロセッサによって実行され得る。したがって、「プロセッサ」という用語は、本明細書で使用されるときに、前述の構造のいずれかまたは本明細書で説明した技法の実装に適切な任意の他の構造を指すことができる。加えて、いくつかの態様では、本明細書で説明した機能は、符号化および復号のために構成されるか、または複合コーデックに組み込まれる、専用のハードウェアモジュールおよび／またはソフトウェアモジュール内で提供され得る。また、本技法は、1つまたは複数の回路または論理素子において完全に実装され得る。

40

【0291】

[0244] 本開示の技法は、ワイヤレスハンドセット、集積回路(IC)またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置で実装され得る。様々な構成要素、モジュール、またはユニットは、開示されている技術を実行するよう

50

構成されたデバイスの機能的態様を強調するように本開示において説明されているが、異なるハードウェアユニットによる実現を必ずしも必要としない。そうではなく、上記で説明したように、様々なユニットは、コーデックハードウェアユニット中で組み合わせられるか、または上記で説明した1つまたは複数のプロセッサを含む、適切なソフトウェアおよび／またはファームウェアとともに相互動作可能なハードウェアユニットの集合によって提供され得る。

【0292】

[0245]本開示の様々な態様が説明されてきた。これらおよび他の態様は以下の特許請求の範囲内に入る。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

10

[C 1]

ビデオデータを復号する方法であって、

第1のサイズのブロックに関連するパーティションパターンを決定することと、

前記第1のサイズの前記ブロックに関連する前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きい、

前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測復号することと備える方法。

[C 2]

20

前記第1のサイズのブロックに関連する前記パーティションパターンは、サイズ16×16のブロックに関連するパーティションパターンを備え、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、サイズ16×16の前記ブロックに関連する前記決定されたパーティションパターンに基づいて、サイズ32×32の前記深度ブロックに関する前記パーティションパターンを決定することを備える、C 1に記載の方法。

[C 3]

前記第1のサイズの前記ブロックに関連する1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと、

前記パーティションパターンリストへのインデックスを受信することとさらに備え、

30

前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定することは、前記パーティションパターンリストへの前記受信されたインデックスに基づいて、前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定することを備える、C 1に記載の方法。

[C 4]

前記深度ブロック内のサブブロックを識別する情報を受信することをさらに備え、前記サブブロックのサイズが前記第1のサイズであり、

前記第1のサイズのブロックに関連する前記パーティションパターンを決定することは、前記サブブロックに関する前記パーティションパターンを決定することを備え、

40

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記サブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを備える、C 1に記載の方法。

[C 5]

前記サブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記サブブロックに関する前記パーティションパターンの線形線を前記深度ブロックの境界まで拡張することを備え、得られる線形線が、前記深度ブロックに関する前記パーティションパターンを画定する、C 4に記載の方法。

50

[C 6]

前記第1のサイズの前記ブロックに関する前記決定されたパーティションパターンは、前記第1のサイズの前記ブロックに関する複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、C 1に記載の方法。

[C 7]

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを示す情報を受信することなく、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを備える、C 1に記載の方法。

10

[C 8]

ビデオデータを符号化する方法であって、
第1のサイズのブロックに関するパーティションパターンを決定することと、
前記第1のサイズの前記ブロックに関する前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きい、

前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測符号化することと
を備える方法。

20

[C 9]

前記第1のサイズのブロックに関する前記パーティションパターンは、サイズ16×16のブロックに関するパーティションパターンを備え、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、サイズ16×16の前記ブロックに関する前記決定されたパーティションパターンに基づいて、サイズ32×32の前記深度ブロックに関する前記パーティションパターンを決定することを備える、C 8に記載の方法。

[C 10]

前記第1のサイズの前記ブロックに関する1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと
をさらに備え、

30

前記第1のサイズの前記ブロックに関する前記パーティションパターンを決定することは、前記構成されたパーティションパターンリストにおいて識別されたパーティションパターンから、前記第1のサイズの前記ブロックに関する前記パーティションパターンを決定することを備える、C 8に記載の方法。

[C 11]

前記第1のサイズの前記ブロックに関する前記決定されたパーティションパターンを識別する前記構成されたパーティションパターンリストへのインデックスをシグナリングすることをさらに備え、ビデオデコーダが、前記深度ブロックをイントラ予測復号するために前記インデックスを使用する、C 10に記載の方法。

40

[C 12]

前記深度ブロック内のサブブロックを識別することをさらに備え、前記サブブロックのサイズが前記第1のサイズであり、

前記第1のサイズの前記ブロックに関する前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記パーティションパターンを決定することを備え、

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記決定されたパーティションに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを備える、C 8に記載の方法。

50

[C 1 3]

前記識別されたサブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することは、前記識別されたサブブロックに関する前記パーティションパターンの線形線を前記深度ブロックの境界まで拡張することを備え、得られる線形線が、前記深度ブロックに関する前記パーティションパターンを画定する、C 1 2に記載の方法。

[C 1 4]

前記第1のサイズの前記ブロックに関連する前記パーティションパターンは、前記第1のサイズの前記ブロックに関連する複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、C 8に記載の方法。

10

[C 1 5]

ビデオコーディングのためのデバイスであって、第1のサイズのブロックに関連する1つまたは複数のパーティションパターンを記憶するビデオデータメモリと、

1つまたは複数のプロセッサを備えるビデオコーダとを備え、前記ビデオコーダは、前記記憶された1つまたは複数のパーティションパターンから、前記第1のサイズの前記ブロックに関連するパーティションパターンを決定することと、

20

前記第1のサイズの前記ブロック関連する前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きい、

前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測コーディングすることとを行うように構成される、デバイス。

[C 1 6]

前記第1のサイズのブロックに関連する前記パーティションパターンは、サイズ16×16のブロック関連するパーティションパターンを備え、前記第2のサイズの前記深度ブロックは、サイズ32×32の深度ブロックを備え、前記ビデオコーダは、サイズ16×16の前記ブロックに関連する前記決定されたパーティションパターンに基づいて、サイズ32×32の前記深度ブロックに関する前記パーティションパターンを決定するよう構成される、C 1 5に記載のデバイス。

30

[C 1 7]

前記ビデオコーダはビデオデコーダを備え、

前記ビデオデコーダは、

前記第1のサイズの前記ブロックに関連する1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成することと、

前記パーティションパターンリストへのインデックスを受信することとを行うように構成され、

40

前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記パーティションパターンリストへの前記受信されたインデックスに基づいて、前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するように構成され、

前記深度ブロックをイントラ予測コーディングするために、前記ビデオデコーダは、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測復号するように構成される、C 1 5に記載のデバイス。

[C 1 8]

前記ビデオコーダはビデオデコーダを備え、

前記ビデオデコーダは、前記深度ブロック内のサブブロックを識別する情報を受信するように構成され、前記サブブロックのサイズが前記第1のサイズであり、

50

前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記サブブロックに関する前記パーティションパターンを決定するように構成され、

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記サブブロックに関する前記決定されたパーティションパターンに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、C 15に記載のデバイス。

[C 19]

前記ビデオコーダはビデオデコーダを備え、

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオデコーダは、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを示す情報を受信することなく、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、C 15に記載のデバイス。

10

[C 20]

前記ビデオコーダはビデオエンコーダを備え、

前記ビデオエンコーダは、前記第1のサイズの前記ブロックに関連する1つまたは複数のパーティションパターンを含むパーティションパターンリストを構成するように構成され、

20

前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するために、前記ビデオエンコーダは、前記構成されたパーティションパターンリストにおいて識別されたパーティションパターンから、前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するように構成され、

前記深度ブロックをイントラ予測コーディングするために、前記ビデオエンコーダは、前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測符号化するように構成される、C 15に記載のデバイス。

[C 21]

前記ビデオエンコーダは、前記第1のサイズの前記ブロックに関連する前記決定されたパーティションパターンを識別する前記構成されたパーティションパターンリストへのインデックスをシグナリングするように構成され、ビデオデコーダが、前記深度ブロックをイントラ予測復号するために前記インデックスを使用する、C 20に記載のデバイス。

30

[C 22]

前記ビデオコーダはビデオエンコーダを備え、

前記ビデオエンコーダは、前記深度ブロック内のサブブロックを識別するように構成され、前記サブブロックのサイズが前記第1のサイズであり、

前記第1のサイズの前記ブロックに関連する前記パーティションパターンを決定するために、前記ビデオエンコーダは、前記識別されたサブブロックに関する前記パーティションパターンを決定するように構成され、

前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するために、前記ビデオエンコーダは、前記識別されたサブブロックに関する前記決定されたパーティションに基づいて、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定するように構成される、C 15に記載のデバイス。

40

[C 23]

前記第1のサイズの前記ブロックに関連する前記パーティションパターンは、前記第1のサイズの前記ブロックに関連する複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、C 15に記載のデバイス。

[C 24]

集積回路(I C)、

50

マイクロコントローラ、および
前記ビデオコーダを備えるワイヤレスデバイス
のうちの1つを備える、C 15に記載のデバイス。

[C 25]

ビデオコーディングのためのデバイスの1つまたは複数のプロセッサによって実行されたときに、前記1つまたは複数のプロセッサに、

第1のサイズのブロックに関するパーティションパターンを決定することと、

前記第1のサイズの前記ブロックに関する前記決定されたパーティションパターンに基づいて、第2のサイズの深度ブロックに関するパーティションパターンを決定することと、ここにおいて、前記第2のサイズが前記第1のサイズよりも大きい、

前記深度ブロックに関する前記決定されたパーティションパターンに基づいて、前記深度ブロックをイントラ予測コーディングすることと

を行わせる命令を記憶したコンピュータ可読記憶媒体。

[C 26]

前記第1のサイズの前記ブロックに関する前記パーティションパターンは、サイズ16×16のブロックに関するパーティションパターンを備え、前記1つまたは複数のプロセッサに、前記第2のサイズの前記深度ブロックに関する前記パーティションパターンを決定することを行わせる前記命令は、前記1つまたは複数のプロセッサに、サイズ16×16の前記ブロックに関する前記決定されたパーティションパターンに基づいて、サイズ32×32の前記深度ブロックに関する前記パーティションパターンを決定することを行わせる命令を備える、C 25に記載のコンピュータ可読記憶媒体。

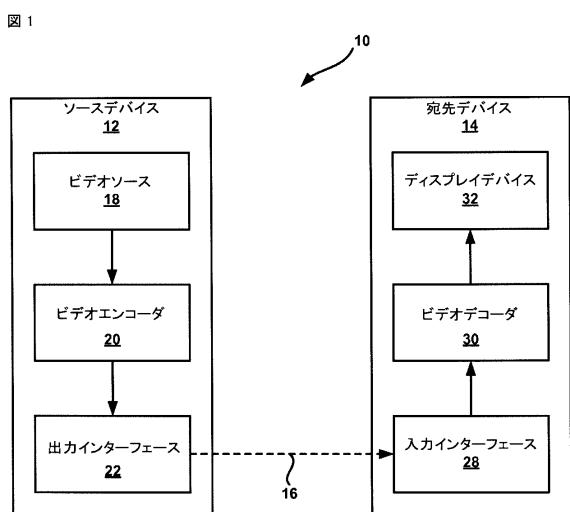
10

[C 27]

前記第1のサイズのブロックに前記関連する前記パーティションパターンは、前記第1のサイズの前記ブロックに関する複数の線ベースのパーティションパターンからの線ベースのパーティションパターンを備え、前記深度ブロックに関する前記パーティションパターンは、2つのパーティションに前記深度ブロックを区分する前記深度ブロックに関する線ベースのパーティションパターンを備える、C 25に記載のコンピュータ可読記憶媒体。

20

〔 1 〕



【 2 】

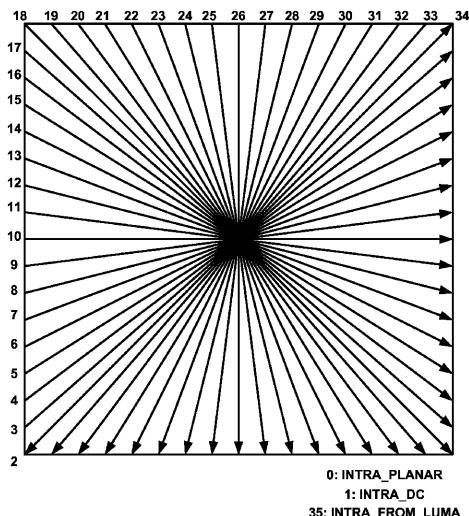


FIG. 1

FIG. 2

【図3A】

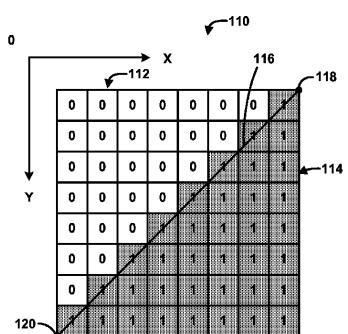


FIG. 3A

【図4A】

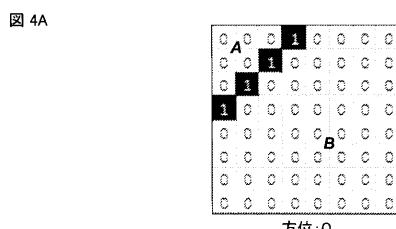


FIG. 4A

【図3B】

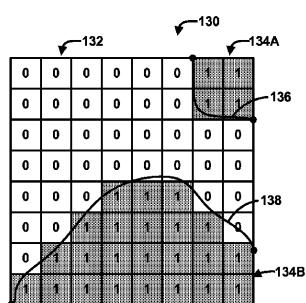


FIG. 3B

図 4 B]

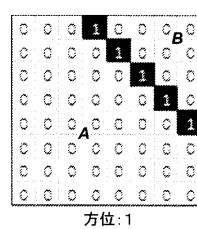
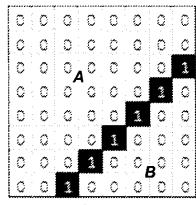


FIG. 4B

【図4C】

図4C

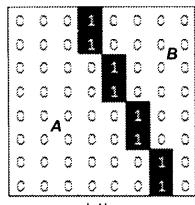


方位:2

FIG. 4C

【図4E】

図4E

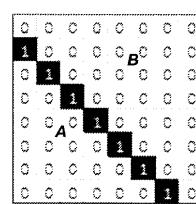


方位:4

FIG. 4E

【図4D】

図4D

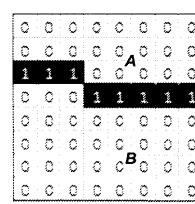


方位:3

FIG. 4D

【図4F】

図4F



方位:5

FIG. 4F

【図5A】

図5A

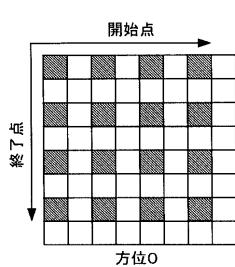


FIG. 5A

【図5B】

図5B

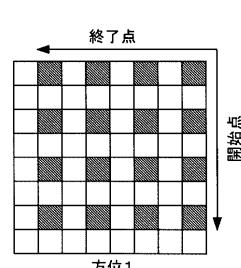


FIG. 5B

【図6】

図6

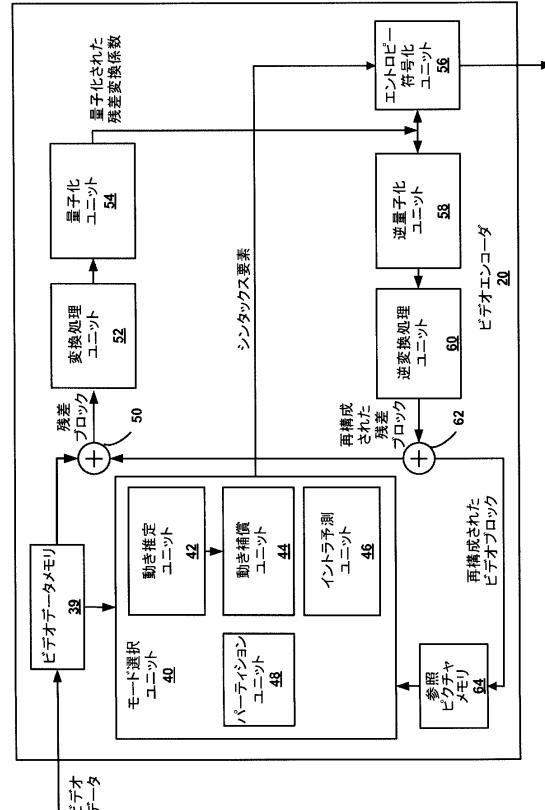
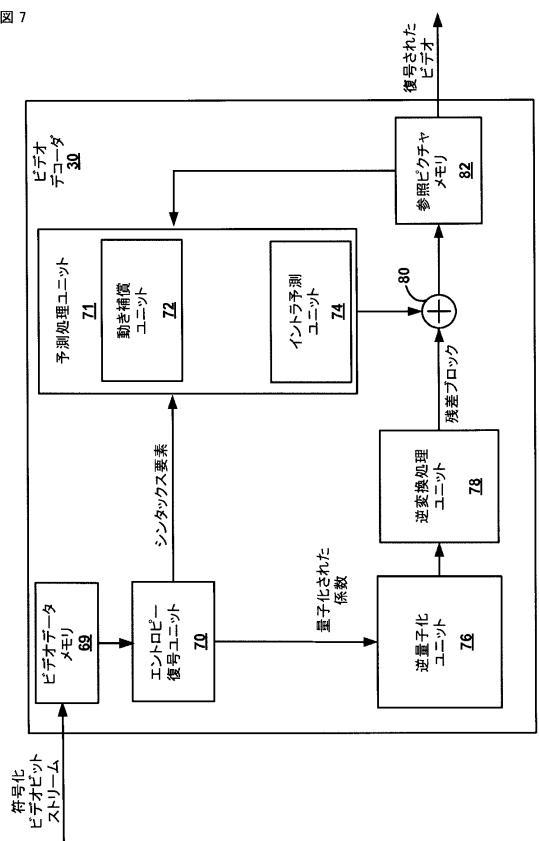


FIG. 6

【図7】

図7



【図8】

図8

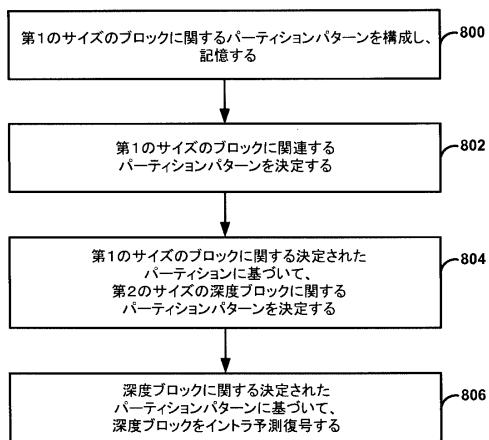


FIG. 8

【図9】

図9

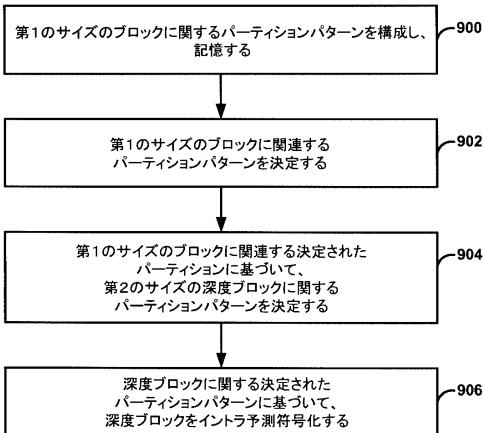


FIG. 9

【図10】

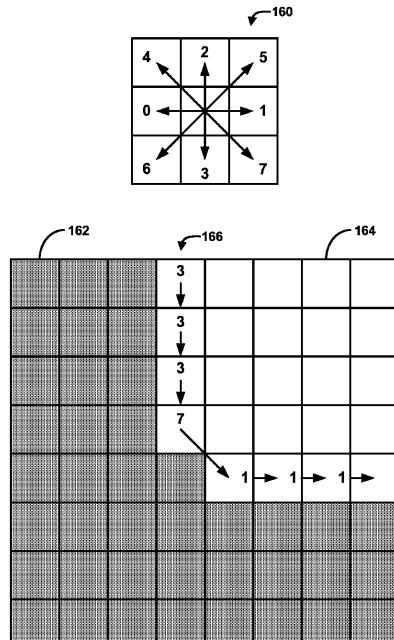


FIG. 10

フロントページの続き

(51)Int.Cl. F I
H 04N 19/14 (2014.01) H 04N 19/14
H 04N 19/70 (2014.01) H 04N 19/70

- (72)発明者 ジャオ、シン
アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
- (72)発明者 ジャン、リ
アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
- (72)発明者 チェン、イン
アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
- (72)発明者 カークゼウィックズ、マルタ
アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

審査官 赤穂 州一郎

- (56)参考文献 国際公開第2013/068566 (WO, A1)
特開2014-200021 (JP, A)
特表2013-533714 (JP, A)
Li Zhang et al., 3D-HEVC Test Model 5, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 5th Meeting: Vienna, AT, 27 Jul. -2 Aug. 2013, SW, 2013年 9月14日, JCT3V-E1005, 23 - 33頁
佐藤雅治、河村圭、内藤整, 3D - HEVCにおけるデブス符号化のデコーダ計算量削減法, 電子情報通信学会2013年総合大会, 日本, 一般社団法人電子情報通信学会, 2013年 3月 19日, 47頁
Xin Zhao et al., CE5 related: Wedgelet pattern extension from 4x4 block, Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 6th Meeting: Geneva, CH, 25 Oct. - 1 Nov. 2013, 2013年10月18日, JCT3V-F0134

- (58)調査した分野(Int.Cl., DB名)
H 04N 19 / 00 - 19 / 98