

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6541682号

(P6541682)

(45) 発行日 令和1年7月10日 (2019.7.10)

(24) 登録日 令和1年6月21日 (2019.6.21)

(51) Int. Cl. F I
HO 4 N 19/186 (2014.01) HO 4 N 19/186
HO 4 N 19/50 (2014.01) HO 4 N 19/50
HO 4 N 19/70 (2014.01) HO 4 N 19/70
HO 4 N 19/93 (2014.01) HO 4 N 19/93

請求項の数 17 (全 51 頁)

(21) 出願番号	特願2016-558209 (P2016-558209)	(73) 特許権者	507364838
(86) (22) 出願日	平成27年3月25日 (2015.3.25)		クアルコム、インコーポレイテッド
(65) 公表番号	特表2017-513369 (P2017-513369A)		アメリカ合衆国 カリフォルニア 921
(43) 公表日	平成29年5月25日 (2017.5.25)		21 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2015/022468		イブ 5775
(87) 国際公開番号	W02015/148652	(74) 代理人	100108453
(87) 国際公開日	平成27年10月1日 (2015.10.1)		弁理士 村山 靖彦
審査請求日	平成30年3月9日 (2018.3.9)	(74) 代理人	100163522
(31) 優先権主張番号	61/970,257		弁理士 黒田 晋平
(32) 優先日	平成26年3月25日 (2014.3.25)	(72) 発明者	マルタ・カルチェヴィチ
(33) 優先権主張国	米国 (US)		アメリカ合衆国・カリフォルニア・921
(31) 優先権主張番号	61/981,105		21-1714・サン・ディエゴ・モアハ
(32) 優先日	平成26年4月17日 (2014.4.17)		ウス・ドライブ・5775
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ビデオコーディングのためのランレングスコードを用いるパレット予測子シグナリング

(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、
 ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信するステップ
 と、

ランレングス復号技法を使用して前記符号化バイナリ予測ベクトルを復号するステップ
 であって、

前記符号化バイナリ予測ベクトルを復号するステップが、
前記ランレングス復号技法および予約済みランレングス値Lを使用して前記符号化バ
イナリ予測ベクトルを復号するステップを備え、

前記予約済みランレングス値Lが前記符号化バイナリ予測ベクトルの末尾位置を示す
、ステップと、

前記符号化バイナリ予測ベクトルに基づいてビデオデータの前記現在ブロックのための
 パレットを生成するステップであって、前記符号化バイナリ予測ベクトルが、前に使用さ
 れたパレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して
 再使用されるか否かを示すエントリを備える、ステップと、

前記パレットを使用してビデオデータの前記現在ブロックを復号するステップと
 を備える方法。

【請求項2】

前記符号化バイナリ予測ベクトルを復号するステップが、

10

20

前記ランレングス復号技法および前記予約済みランレングス値Lに加えて、最大パレットサイズを使用して前記符号化バイナリ予測ベクトルを復号するステップを備え、

ここにおいて、前に使用された前記パレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して再使用されることを示す、前記符号化バイナリ予測ベクトルの中のエントリの総数が前記最大パレットサイズに等しい場合、前記予約済みランレングス値Lが使用されない、

請求項1に記載の方法。

【請求項3】

前記符号化バイナリ予測ベクトルの中の最後のランレングスが0のランレングスを示さない場合、前記予約済みランレングス値Lが使用されない、

請求項1に記載の方法。

【請求項4】

前記符号化バイナリ予測ベクトルに基づいてビデオデータの前記現在ブロックのための前記パレットを生成するステップが、

前記パレットに対して再使用されるものとして前記符号化バイナリ予測ベクトルによって示されている、前に使用された前記パレットエントリを、バッファから前記パレットの中にコピーするステップと、

前記パレットに対してコピーされた、前に使用された前記パレットエントリのが最大パレットサイズよりも少ない場合、追加のパレットエントリを受信するステップと

を備える、請求項1に記載の方法。

【請求項5】

前記ランレングス復号技法が、0のランレングスをコーディングすることを備える、請求項1に記載の方法。

【請求項6】

ビデオデータを符号化する方法であって、

ビデオデータの現在ブロックのためのパレットを生成するステップと、

ビデオデータの前記現在ブロックのための前記パレットに関するバイナリ予測ベクトルを生成するステップであって、前記バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して再使用されるか否かを示すエントリを備える、ステップと、

ランレングス符号化技法を使用して前記バイナリ予測ベクトルを符号化するステップであって、

前記バイナリ予測ベクトルを符号化するステップが、

前記ランレングス符号化技法および予約済みランレングス値Lを使用して前記バイナリ予測ベクトルを符号化するステップを備え、

前記予約済みランレングス値Lが前記バイナリ予測ベクトルの末尾位置を示す、ステップと、

前記パレットを使用してビデオデータの前記現在ブロックを符号化するステップとを備える方法。

【請求項7】

前記予約済みランレングス値Lが1である、請求項1または6に記載の方法。

【請求項8】

前記バイナリ予測ベクトルを符号化するステップが、

前記ランレングス符号化技法および前記予約済みランレングス値Lに加えて、最大パレットサイズを使用して前記バイナリ予測ベクトルを符号化するステップを備え、

ここにおいて、前に使用された前記パレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して再使用されることを示す、前記バイナリ予測ベクトルの中のエントリの総数が前記最大パレットサイズに等しい場合、前記予約済みランレングス値Lが使用されない、

請求項6に記載の方法。

【請求項 9】

前記バイナリ予測ベクトルの中の最後のランレングスが0のランレングスを示さない場合、前記予約済みランレングス値Lが使用されない、
請求項6に記載の方法。

【請求項 10】

ビデオデータの、前に符号化された1つまたは複数のブロックに対して前に使用された前記パレットエントリをバッファに記憶するステップ
をさらに備える、請求項4または6に記載の方法。

【請求項 11】

前記バッファに記憶されている、前に使用された前記パレットエントリと全く同じエントリを除去するステップ
をさらに備える、請求項10に記載の方法。

10

【請求項 12】

ビデオデータの、前に符号化された1つまたは複数の前記ブロックに対して前に使用された前記パレットエントリが、ビデオデータの前記現在ブロックの上の画素のラインに対して前に使用されたパレットエントリ、およびビデオデータの前記現在ブロックの左の画素のラインに対して前に使用されたパレットエントリを備える、請求項10に記載の方法。

【請求項 13】

前記ランレングス符号化技法が、0のランレングスをコーディングすることを備える、
請求項6に記載の方法。

20

【請求項 14】

ビデオデータを復号するように構成された装置であって、
ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信するための手段と、
ランレングス復号技法を使用して前記符号化バイナリ予測ベクトルを復号するための手段であって、

前記符号化バイナリ予測ベクトルを復号するための手段が、
前記ランレングス復号技法および予約済みランレングス値Lを使用して前記符号化バイナリ予測ベクトルを復号するための手段を備え、

前記予約済みランレングス値Lが前記符号化バイナリ予測ベクトルの末尾位置を示す、手段と、

30

前記符号化バイナリ予測ベクトルに基づいてビデオデータの前記現在ブロックのためのパレットを生成するための手段であって、前記符号化バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して再使用されるか否かを示すエントリを備える、手段と、

前記パレットを使用してビデオデータの前記現在ブロックを復号するための手段と
を備える装置。

【請求項 15】

ビデオデータを符号化するように構成された装置であって、
ビデオデータの現在ブロックのためのパレットを生成するための手段と、
ビデオデータの前記現在ブロックのための前記パレットに関するバイナリ予測ベクトルを生成するための手段であって、前記バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの前記現在ブロックのための前記パレットに対して再使用されるか否かを示すエントリを備える、手段と、

40

ランレングス符号化技法を使用して前記バイナリ予測ベクトルを符号化するための手段であって、

前記バイナリ予測ベクトルを符号化するための手段が、
前記ランレングス符号化技法および予約済みランレングス値Lを使用して前記バイナリ予測ベクトルを符号化するための手段を備え、

前記予約済みランレングス値Lが前記バイナリ予測ベクトルの末尾位置を示す、手段

50

と、

前記パレットを使用してビデオデータの前記現在ブロックを符号化するための手段とを備える装置。

【請求項 16】

実行されたとき、ビデオデータを復号するように構成されたデバイスの1つまたは複数のプロセッサに、請求項1乃至5、請求項7、および請求項10乃至12の何れか1項に記載の方法を実施させる命令を記憶するコンピュータ可読記憶媒体。

【請求項 17】

実行されたとき、ビデオデータを符号化するように構成されたデバイスの1つまたは複数のプロセッサに、請求項6乃至13の何れか1項に記載の方法を実施させる命令を記憶するコンピュータ可読記憶媒体。

10

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、2014年3月25日に出願された米国仮出願第61/970,257号、2014年4月17日に出願された米国仮出願第61/981,105号、2014年5月23日に出願された米国仮出願第62/002,668号、2014年6月20日に出願された米国仮出願第62/015,327号、2014年6月27日に出願された米国仮出願第62/018,461号、および2014年8月24日に出願された米国仮出願第62/041,119号の利益を主張するものであり、それらの各々の内容全体が参照により本明細書に組み込まれる。

20

【0002】

本開示は、ビデオ符号化および復号に関する。

【背景技術】

【0003】

デジタルビデオ機能は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップまた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、ビデオゲームコンソール、セルラーまたは衛星無線電話、いわゆる「スマートフォン」、ビデオ遠隔会議デバイス、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、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、パート10、アドバンスドビデオコーディング(AVC:Advanced Video Coding)、現在開発中の高効率ビデオコーディング(HEVC:High Efficiency Video Coding)規格、およびそのような規格の拡張によって定義された規格に記載されるものなどのビデオ圧縮技法を実装する。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、受信、符号化、復号、および/または記憶し得る。

30

【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(ピクチャ内)予測および/または時間(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、ビデオフレーム、またはビデオフレームの一部分)は、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ド化(I)スライスの中のビデオブロックは、同じピクチャにおける隣接ブロックの中の参照サンプルに対する空間予測を使用して符号化される。ピクチャのインターコード化(PまたはB)スライスの中のビデオブロックは、同じピクチャにおける隣接ブロックの中の参照サンプルに対する空間予測、または他の参照ピクチャの中の参照サンプルに対する時間予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

40

【0005】

空間予測または時間予測が、コーディングされるべきブロックに対する予測ブロックを

50

もたらす。残差データが、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。インターコード化ブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルに従って符号化され、残差データは、コード化ブロックと予測ブロックとの間の差分を示す。イントラコード化ブロックは、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、残差データはピクセル領域から変換領域に変換されてよく、結果として残差係数をもたらす。残差係数は次いで量子化され得る。最初に2次元アレイに配置される量子化係数は、係数の1次元ベクトルを生成するために走査され得、なお一層の圧縮を達成するためにエントロピーコーディングが適用され得る。

【先行技術文献】

10

【特許文献】

【0006】

【特許文献1】米国仮出願第62/002,741号

【非特許文献】

【0007】

【非特許文献1】「ITU-T H.265(V1)」、<http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885&lang=en>

【非特許文献2】ITU-T H.265、SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of Audiovisual Services-Coding of Moving Video、「High Efficiency Video Coding」、2013年4月

20

【非特許文献3】「ITU-T H.265(V2)」、<http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12296&lang=en>

【非特許文献4】X. GuoおよびA. Saxena、「RCE4: Summary report of HEVC Range Extension Core Experiments 4 (RCE4) on palette coding for screen content」、JCTVC-P0035、サンノゼ、米国、2014年1月9～17日

【非特許文献5】X. Guo、Y. Lu、およびS. Li、「RCE4: Test 1. Major-color-based screen content coding」、JCTVC-P0108、サンノゼ、米国、2014年1月9～17日

【非特許文献6】L. Guo、W. Pu、M. Karczewicz、J. Sole、R. Joshi、およびF. Zou、「RCE4: Results of Test 2 on Palette Mode for Screen Content Coding」、JCTVC-P0198、サンノゼ、米国、2014年1月9～17日

30

【非特許文献7】C. Gisquet、G. Laroche、およびP. Onno、「AhG10: Palette predictor stuffing」、JCTVC-Q0063

【非特許文献8】JCTVC-Q0094、http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0094-v1.zip

【非特許文献9】C. Gisquet、G. Laroche、およびP. Onno、「AhG10: Palette predictor stuffing」、JCTVC-Q0063

【発明の概要】

【課題を解決するための手段】

【0008】

本開示は、ビデオ符号化および復号技法に関する。詳細には、本開示は、パレットベースコーディングモードを用いてビデオデータを符号化および復号するための技法を説明する。パレットベースコーディングモードでは、ビデオデータのブロックに関する画素値が、ビデオデータのブロックに関連付けられたカラー値のパレットに対してコーディングされ得る。カラー値のパレットは、ビデオエンコーダによって決定され得、特定のブロックにとって最も共通であるカラー値を含み得る。ビデオエンコーダは、カラー値のパレットへのインデックスをビデオデータのブロックの中の各画素に割り当て得、そのようなインデックスをビデオデコーダにシグナリングし得る。ビデオデコーダは、次いで、ブロックの中の特定の画素に対してどのカラー値が使用されるべきかを決定するために、パレットへのインデックスを使用し得る。

40

【0009】

50

パレットの中のインデックスをシグナリングすることに加えて、ビデオエンコーダはまた、パレット自体を符号化ビデオビットストリームの中で送信し得る。パレットを送信するための技法は、パレット値を明示的にシグナリングすること、ならびに前にコーディングされた1つまたは複数のブロックからのパレットエントリから現在ブロック用のパレットエントリを予測することを含み得る。本開示は、ビデオデータの、前にコーディングされたブロックからのどのパレットエントリがビデオデータの現在ブロックに対して再使用され得るのかを示す、1つまたは複数のシンタックス要素(たとえば、バイナリ予測ベクトル)をシグナリングするための技法を説明する。本開示は、バイナリ予測ベクトルを符号化および復号するための技法をさらに説明する。

【0010】

10

本開示の一例では、ビデオデータを復号する方法は、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信することと、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成することと、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、パレットを使用してビデオデータの現在ブロックを復号することとを備える。

【0011】

本開示の別の例では、ビデオデータを符号化する方法は、ビデオデータの現在ブロック用のパレットを生成することと、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化することと、パレットを使用してビデオデータの現在ブロックを符号化することとを備える。

20

【0012】

本開示の別の例では、ビデオデータを復号するように構成された装置は、ビデオデータを記憶するように構成されたメモリ、ならびに、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信することと、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成することと、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、パレットを使用してビデオデータの現在ブロックを復号することとを行うように構成された、メモリと通信しているビデオデコーダを備える。

30

【0013】

本開示の別の例では、ビデオデータを符号化するように構成された装置は、ビデオデータを記憶するように構成されたメモリ、ならびに、ビデオデータの現在ブロック用のパレットを生成することと、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化することと、パレットを使用してビデオデータの現在ブロックを符号化することとを行うように構成された、メモリと通信しているビデオエンコーダを備える。

40

【0014】

本開示の別の例では、ビデオデータを復号するように構成された装置は、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信するための手段と、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号するための手段と、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成するための手段と、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの

50

現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備える手段と、パレットを使用してビデオデータの現在ブロックを復号するための手段とを備える。

【0015】

本開示の別の例では、ビデオデータを符号化するように構成された装置は、ビデオデータの現在ブロック用のパレットを生成するための手段と、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成するための手段であって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備える手段と、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化するための手段と、パレットを使用してビデオデータの現在ブロックを符号化するための手段とを備える。

10

【0016】

別の例では、本開示は、実行されたとき、ビデオデータを復号するように構成されたデバイスの1つまたは複数のプロセッサに、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信することと、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成することであって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、パレットを使用してビデオデータの現在ブロックを復号することとを行わせる命令を記憶するコンピュータ可読記憶媒体を説明する。

20

【0017】

別の例では、本開示は、実行されたとき、ビデオデータを符号化するように構成されたデバイスの1つまたは複数のプロセッサに、ビデオデータの現在ブロック用のパレットを生成することと、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成することであって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化することと、パレットを使用してビデオデータの現在ブロックを符号化することとを行わせる命令を記憶するコンピュータ可読記憶媒体を説明する。

【0018】

1つまたは複数の例の詳細が、以下の添付の図面および説明において述べられる。他の特徴、目的、および利点は、説明および図面から、ならびに特許請求の範囲から明らかになるであろう。

30

【図面の簡単な説明】

【0019】

【図1】本開示で説明する技法を実施し得る例示的なビデオコーディングシステムを示すブロック図である。

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

【図3】本開示で説明する技法を実施し得る例示的なビデオデコーダを示すブロック図である。

40

【図4】図2のビデオエンコーダの例示的なパレットベース符号化ユニットを示すブロック図である。

【図5】本開示の技法による例示的なパレット予測技法を示す概念図である。

【図6】本開示の技法による例示的なバイナリ予測ベクトル符号化技法を示す概念図である。

【図7】図3のビデオエンコーダの例示的なパレットベース復号ユニットを示すブロック図である。

【図8】本開示の例示的な符号化方法を示すフローチャートである。

【図9】本開示の例示的な復号方法を示すフローチャートである。

【図10】本開示の技法に一致するパレットベースビデオコーディングのためのパレット

50

エントリを決定することの一例を示す概念図である。

【発明を実施するための形態】

【0020】

本開示は、ビデオコーディングおよび圧縮のための技法を含む。詳細には、本開示は、ビデオデータのパレットベースコーディングのための技法を説明する。従来のビデオコーディングでは、画像は、連続階調かつ空間的に滑らかであるものと想定される。これらの想定に基づいて、ブロックベースの変換、フィルタ処理などの様々なツールが開発されており、そのようなツールは、自然コンテンツビデオにとって良好な性能を示してきた。

【0021】

しかしながら、リモートデスクトップ、共同作業、およびワイヤレスディスプレイのような用途では、コンピュータ生成されたスクリーンコンテンツ(たとえば、テキストまたはコンピュータグラフィックスなどの)が、圧縮されるべき主要なコンテンツであり得る。このタイプのコンテンツは、離散階調を有するとともに鋭い線および高コントラストのオブジェクト境界を特徴とする傾向がある。連続階調および滑らかさという想定は、もはやスクリーンコンテンツにとって当てはまらない場合があり、したがって、従来のビデオコーディング技法は、スクリーンコンテンツを含むビデオデータを圧縮するのに効率的な方法でない場合がある。

【0022】

スクリーン生成されたコンテンツのコーディングにとって、パレットベースコーディング技法が特に適当であり得る。たとえば、ビデオデータの特定のエリアが比較的少数のカラーを有すると想定する。ビデオコード(たとえば、ビデオエンコードまたはビデオデコード)は、特定のエリア(たとえば、所与のブロック)のビデオデータを表すためのカラーのテーブルとして、いわゆる「パレット」を形成し得る。各画素は、画素のカラーを表すパレットの中のエントリに関連付けられ得る。たとえば、ビデオコードは、画素値をパレットの中の適切な値に関連付けるインデックスをコーディングし得る。

【0023】

ビデオデータのパレットベースコーディングのための技法は、インター予測コーディングまたはイントラ予測コーディングなどの、1つまたは複数の他のコーディング技法とともに使用され得る。たとえば、以下でより詳細に説明されるように、エンコードもしくはデコード、または組み合わされたエンコードデコード(コーデック)が、インター予測コーディングおよびイントラ予測コーディング、ならびにパレットベースコーディングを実行するように構成され得る。

【0024】

いくつかの例では、パレットベースコーディング技法は、1つまたは複数のビデオコーディング規格とともに使用するために構成され得る。いくつかの例示的な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)拡張およびマルチビュービデオコーディング(MVC)拡張を含む、ITU-T H.261、ISO/IEC MPEG-1ビジュアル、ITU-T H.262またはISO/IEC MPEG-2ビジュアル、ITU-T H.263、ISO/IEC MPEG-4ビジュアルおよびITU-T H.264(ISO/IEC MPEG-4 AVCとも呼ばれる)を含む。別の例では、パレットベースコーディング技法は、高効率ビデオコーディング(HEVC)とともに使用するために構成され得る。HEVCは、ITU-Tビデオコーディングエキスパートグループ(VCEG)およびISO/IECモーションピクチャエキスパートグループ(MPEG)のビデオコーディング共同研究部会(JCT-VC)によって開発された新しいビデオコーディング規格である。

【0025】

最近、ITU-Tビデオコーディングエキスパートグループ(VCEG)およびISO/IECモーションピクチャエキスパートグループ(MPEG)のビデオコーディング共同研究部会(JCT-VC)によって、HEVCの設計が確定された。これ以降、HEVCバージョン1と呼ばれる最新のHEVC仕様は、「ITU-T H.265(V1)」に記載されており、2015年3月24日現在、<http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885&lang=en>から入手可能である。ITU-T H.265、SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of Audiovisual Servic

10

20

30

40

50

es-Coding of Moving Video、「High Efficiency Video Coding」、2013年4月という文書も、HEVC規格を記載している。これ以降、RExtと呼ばれる範囲拡張の最近の仕様は、「ITU-T H.265(V2)」に記載されており、2015年3月24日現在、<http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12296&lang=en>から入手可能である。

【0026】

例示的なパレットベースコーディング技法を使用するとき、ビデオエンコーダは、ブロック用のパレットを決定すること(たとえば、パレットを明示的にコーディングすること、パレットを予測すること、またはそれらの組合せ)と、各画素の値を表すためのエントリをパレットの中に配置することと、画素値をパレットに関連付ける画素に関するインデックス値とともにブロックを符号化することによって、ビデオデータのブロックを符号化し得る。ビデオデコーダは、ブロック用のパレット、ならびにブロックの画素に関するインデックス値を、符号化ビットストリームから取得し得る。ビデオデコーダは、ブロックの画素値を再構成するために、画素のインデックス値をパレットのエントリに関連付け得る。上記の例は、パレットベースコーディングの一般的な説明を提供することを意図する。

【0027】

インデックス値をパレットの中にシグナリングすることに加えて、ビデオエンコーダはまた、パレット自体を符号化ビデオビットストリームの中で送信し得る。パレットを送信するための技法は、パレット値を明示的にシグナリングすること、ならびに前にコーディングされた1つまたは複数のブロックからのパレットエントリから現在ブロック用のパレットエントリを予測することを含み得る。パレット予測技法は、一続きのシンタックス要素(たとえば、バイナリ予測ベクトル)をシグナリングすることを含み得、その場合、バイナリ予測ベクトルの中の各シンタックス要素の値は、ビデオデータの、前にコーディングされたブロックからのパレットエントリがビデオデータの現在ブロックに対して再使用され得るか否かを示す。大きいパレットサイズの場合、かつ/またはパレット予測プロセスにおいてできるだけ使用するために多数のパレットエントリが記憶される場合、そのようなバイナリ予測ベクトルは長くなることがある。大きいバイナリ予測ベクトルは、許容できないビットレートレベルをもたらす場合がある。これらの欠点に鑑みて、本開示は、パレット予測プロセスにおいてバイナリ予測ベクトルを符号化および復号するための技法を説明する。

【0028】

図1は、本開示の技法を利用し得る例示的なビデオコーディングシステム10を示すブロック図である。本明細書で使用する「ビデオコーダ」という用語は、総称的に、ビデオエンコーダとビデオデコーダの両方を指す。本開示では、「ビデオコーディング」または「コーディング」という用語は、総称的に、ビデオ符号化またはビデオ復号のいずれかを指す場合がある。ビデオコーディングシステム10のビデオエンコーダ20およびビデオデコーダ30は、本開示で説明する様々な例によるパレットベースビデオコーディングのための技法を実行するように構成され得るデバイスの例を表す。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、高効率ビデオコーディング(HEVC)規格におけるようなCUまたはPUなどのビデオデータの様々なブロックを、パレットベースコーディングまたは非パレットベースコーディングのいずれかを使用して選択的にコーディングするように構成され得る。非パレットベースコーディングモードとは、HEVCバージョン1によって規定される様々なコーディングモードなどの、様々なインター予測、時間コーディングモード、またはイントラ予測、空間コーディングモードを指すことがある。

【0029】

図1に示すように、ビデオコーディングシステム10は、ソースデバイス12および宛先デバイス14を含む。ソースデバイス12は、符号化ビデオデータを生成する。したがって、ソースデバイス12は、ビデオ符号化デバイスまたはビデオ符号化装置と呼ばれることがある。宛先デバイス14は、ソースデバイス12によって生成された符号化ビデオデータを復号し得る。したがって、宛先デバイス14は、ビデオ復号デバイスまたはビデオ復号装置と呼ば

れることがある。ソースデバイス12および宛先デバイス14は、ビデオコーディングデバイスまたはビデオコーディング装置の例であり得る。ソースデバイス12、宛先デバイス14、またはその両方の様々な実装形態は、1つまたは複数のプロセッサと、1つまたは複数のプロセッサに結合されたメモリとを含み得る。メモリは、本明細書で説明するように、限定はしないが、RAM、ROM、EEPROM、フラッシュメモリ、または所望のプログラムコードを命令またはデータ構造の形で記憶するために使用され得るとともに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含み得る。

【0030】

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、モバイルコンピューティングデバイス、ノートブック(たとえば、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、車載コンピュータなどを含む、広範囲のデバイスを備え得る。

【0031】

宛先デバイス14は、符号化ビデオデータをソースデバイス12からチャンネル16を介して受信し得る。チャンネル16は、符号化ビデオデータをソースデバイス12から宛先デバイス14に移動させることが可能な1つまたは複数の媒体またはデバイスを備え得る。一例では、チャンネル16は、ソースデバイス12がリアルタイムで符号化ビデオデータを直接宛先デバイス14に送信することを可能にする、1つまたは複数の通信媒体を備え得る。この例では、ソースデバイス12は、ワイヤレス通信プロトコルなどの通信規格に従って符号化ビデオデータを変調し得、変調されたビデオデータを宛先デバイス14へ送信し得る。1つまたは複数の通信媒体は、無線周波数(RF)スペクトルまたは1つまたは複数の物理伝送線路などの、ワイヤレスおよび/または有線の通信媒体を含み得る。1つまたは複数の通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはグローバルネットワーク(たとえば、インターネット)などの、パケットベースネットワークの一部を形成し得る。1つまたは複数の通信媒体は、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他の機器を含み得る。

【0032】

別の例では、チャンネル16は、ソースデバイス12によって生成された符号化ビデオデータを記憶する記憶媒体を含み得る。この例では、宛先デバイス14は、たとえば、ディスクアクセスまたはカードアクセスを介して記憶媒体にアクセスし得る。記憶媒体は、ブルーレイディスク、DVD、CD-ROM、フラッシュメモリ、または符号化ビデオデータを記憶するための他の適当なデジタル記憶媒体などの、ローカルにアクセスされる様々なデータ記憶媒体を含み得る。

【0033】

さらなる例では、チャンネル16は、ソースデバイス12によって生成された符号化ビデオデータを記憶するファイルサーバまたは別の中間記憶デバイスを含み得る。この例では、宛先デバイス14は、ファイルサーバまたは他の中間記憶デバイスにおいて記憶された符号化ビデオデータに、ストリーミングまたはダウンロードを介してアクセスし得る。ファイルサーバは、符号化ビデオデータを記憶するとともに符号化ビデオデータを宛先デバイス14へ送信することが可能なタイプのサーバであり得る。例示的なファイルサーバは、ウェブサーバ(たとえば、ウェブサイト用の)、ファイル転送プロトコル(FTP)サーバ、ネットワーク接続ストレージ(NAS)デバイス、HTTPストリーミングサーバ、およびローカルディスクドライブを含む。

【0034】

宛先デバイス14は、インターネット接続などの標準的なデータ接続を通じて符号化ビデオデータにアクセスし得る。データ接続の例示的なタイプは、ファイルサーバに記憶されている符号化ビデオデータにアクセスするために適当である、ワイヤレスチャンネル(たとえば、Wi-Fi接続)、有線接続(たとえば、DSL、ケーブルモデムなど)、または両方の組合せを含み得る。ファイルサーバからの符号化ビデオデータの送信は、ストリーミング送信

、ダウンロード送信、または両方の組合せであってよい。

【 0 0 3 5 】

ビデオコーディングにおけるパレットベースコーディングモードのための本開示の技法は、ワイヤレスの用途またはセッティングに限定されない。技法は、オーバージエアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、たとえば、インターネットを介したストリーミングビデオ送信、データ記憶媒体に記憶するためのビデオデータの符号化、データ記憶媒体に記憶されたビデオデータの復号、または他の用途などの様々なマルチメディア用途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、ビデオコーディングシステム10は、ビデオストリーミング、ビデオ再生、ビデオ放送、および/またはビデオテレフォニーなどの用途をサポートするために、片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。

10

【 0 0 3 6 】

図1に示すビデオコーディングシステム10は一例にすぎず、本開示の技法は、符号化デバイスと復号デバイスとの間に必ずしもデータ通信を含まないビデオコーディングのセッティング(たとえば、ビデオ符号化またはビデオ復号)に適用され得る。他の例では、データは、ローカルメモリから、ネットワークを介してストリーミングされて、または類似のやり方で取り出される。ビデオ符号化デバイスがデータを符号化するとともにメモリに記憶してよく、および/またはビデオ復号デバイスがメモリからデータを取り出すとともに復号してもよい。多くの例では、互いに通信しないが、単にデータをメモリへ符号化し、かつ/またはメモリからデータを取り出すとともに復号するデバイスによって、符号化および復号が実行される。

20

【 0 0 3 7 】

図1の例では、ソースデバイス12は、ビデオソース18、ビデオエンコーダ20、および出力インターフェース22を含む。いくつかの例では、出力インターフェース22は、変調器/復調器(モデム)および/または送信機を含み得る。ビデオソース18は、ビデオキャプチャデバイス、たとえば、ビデオカメラ、前にキャプチャされたビデオデータを含むビデオアーカイブ、ビデオデータをビデオコンテンツプロバイダから受信するためのビデオフィードインターフェース、および/もしくはビデオデータを生成するためのコンピュータグラフィックスシステム、またはビデオデータのそのようなソースの組合せを含み得る。

【 0 0 3 8 】

30

ビデオエンコーダ20は、ビデオソース18からのビデオデータを符号化し得る。いくつかの例では、ソースデバイス12は、符号化ビデオデータを宛先デバイス14へ出力インターフェース22を介して直接送信する。他の例では、復号および/または再生のために宛先デバイス14によって後でアクセスできるように、符号化ビデオデータはまた、記憶媒体またはファイルサーバへ記憶され得る。

【 0 0 3 9 】

図1の例では、宛先デバイス14は、入力インターフェース28、ビデオデコーダ30、およびディスプレイデバイス32を含む。いくつかの例では、入力インターフェース28は、受信機および/またはモデムを含む。入力インターフェース28は、符号化ビデオデータをチャネル16を介して受信し得る。ディスプレイデバイス32は、宛先デバイス14と一体化されてよく、または宛先デバイス14の外部にあってもよい。概して、ディスプレイデバイス32は復号ビデオデータを表示する。ディスプレイデバイス32は、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)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を備え得る。

40

【 0 0 4 0 】

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、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)、個別論理、ハードウェア、またはそれらの任意の組合せなどの、様々な適当な回路のいずれかとして実装され得る。技法が部分的にソフトウェアで実施される場合、本開示の技法を実行するために、デバイスは、ソフトウェアのため

50

の命令を適当な非一時的コンピュータ可読記憶媒体に記憶し得、1つまたは複数のプロセッサを使用するハードウェアにおいて命令を実行し得る。前述のもののいずれか(ハードウェア、ソフトウェア、ハードウェアとソフトウェアの組合せなどを含む)は、1つまたは複数のプロセッサであると思なされてよい。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に含まれてよく、それらのいずれかが、組み合わされたエンコーダ/デコーダ(コーデック)の一部としてそれぞれのデバイスの中で一体化されてよい。

【0041】

本開示は、概して、ある種の情報をビデオデコーダ30などの別のデバイスへ「シグナリングする」または「送信する」ビデオエンコーダ20に言及することがある。「シグナリングすること」または「送信すること」という用語は、概して、シンタックス要素、および/または圧縮ビデオデータを復号するために使用される他のデータの通信を指すことがある。そのような通信は、リアルタイムで、またはほぼリアルタイムで発生し得る。代替的に、そのような通信は、符号化の時点において符号化ビットストリームの中のシンタックス要素をコンピュータ可読記憶媒体に記憶し、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において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ような、時間の範囲にわたって発生することもある。

【0042】

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、上述のHEVC規格などの、HEVCバージョン1において記載されているビデオ圧縮規格に従って動作する。ベースのHEVC規格に加えて、HEVC向けのスケーラブルビデオコーディング拡張、マルチビュービデオコーディング拡張、および3Dコーディング拡張を制作するための取組みが進行中である。加えて、たとえば、本開示で説明するようなパレットベースコーディングモードは、HEVC規格向けの拡張を提供し得る。いくつかの例では、パレットベースコーディングのための本開示で説明する技法は、他のビデオコーディング規格または将来の規格に従って動作するように構成されたエンコーダおよびデコーダに適用され得る。したがって、HEVCコーデックにおけるコーディングユニット(CU)または予測ユニット(PU)のコーディングのためのパレットベースコーディングモードの適用例は、例として説明される。

【0043】

様々な例では、本開示で説明する技法は、パレットベースコーディングにおいて、パレットを決定すること、予測すること、および/またはシグナリングすることの様々な組合せのための技法を含み得る。以下でより詳細に説明されるように、技法は、パレット予測子を決定すること、バイナリ予測ベクトルを生成すること、およびバイナリ予測ベクトルを符号化/復号することの任意の組合せを含み得る。例示および明快のためにいくつかの例が個々に説明されるが、本開示は、本明細書で説明するパレットベースコーディングのための技法の任意の組合せを意図する。

【0044】

図4~図9を参照しながら以下でより詳細に説明するように、ビデオエンコーダ20は、ビデオデータの現在ブロック用のパレットを生成することと、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成することとであって、バイナリ予測ベクトルが、前に使用されたパレットエントリ(すなわち、前にコーディングされたブロック用のパレットの中のエントリ)がビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化することと、パレットを使用してビデオデータの現在ブロックを符号化することとを行うように構成され得る。同様に、ビデオデコーダ30は、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信することと、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成することとであって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、パレットを使用

してビデオデータの現在ブロックを復号することを行うように構成され得る。

【0045】

HEVCおよび他のビデオコーディング規格では、ビデオシーケンスは、通常、一連のピクチャを含む。ピクチャは、「フレーム」と呼ばれることもある。ピクチャは、 S_L 、 S_{Cb} 、および S_{Cr} と示される3つのサンプルアレイを含み得る。 S_L は、ルーマサンプルの2次元アレイ(すなわち、ブロック)である。 S_{Cb} は、Cbクロミナンスサンプルの2次元アレイである。 S_{Cr} は、Crクロミナンスサンプルの2次元アレイである。クロミナンスサンプルは、本明細書で「クロマ」サンプルと呼ばれることもある。他の事例では、ピクチャはモノクロであってよく、ルーマサンプルのアレイだけを含んでよい。

【0046】

ピクチャの符号化表現を生成するために、ビデオエンコーダ20が、コーディングツリーユニット(CTU)のセットを生成し得る。CTUの各々は、ルーマサンプルのコーディングツリーブロック、クロマサンプルの2つの対応するコーディングツリーブロック、およびコーディングツリーブロックのサンプルをコーディングするために使用されるシンタックス構造であり得る。コーディングツリーブロックは、サンプルの $N \times N$ ブロックであり得る。CTUは、「ツリーブロック」または「最大コーディングユニット」(LCU)と呼ばれることもある。HEVCのCTUは、H.264/AVCなどの他の規格のマクロブロックと概して類似であり得る。しかしながら、CTUは、必ずしも特定のサイズに限定されず、1つまたは複数のコーディングユニット(CU)を含んでよい。スライスは、ラスタ走査において連続的に順序付けられた整数個のCTUを含み得る。

【0047】

コード化CTUを生成するために、ビデオエンコーダ20がCTUのコーディングツリーブロック上で4分木区分を再帰的に実行してコーディングツリーブロックをコーディングブロックに分割してよく、したがって、「コーディングツリーユニット」という名前である。コーディングブロックは、サンプルの $N \times N$ ブロックである。CUは、ルーマサンプルアレイ、Cbサンプルアレイ、およびCrサンプルアレイを有するピクチャの、ルーマサンプルのコーディングブロック、およびクロマサンプルの2つの対応するコーディングブロック、ならびにコーディングブロックのサンプルをコーディングするために使用されるシンタックス構造であり得る。ビデオエンコーダ20は、CUのコーディングブロックを1つまたは複数の予測ブロックに区分し得る。予測ブロックは、同じ予測が適用されるサンプルの、長方形(すなわち、正方形または非正方形)のブロックであってよい。CUの予測ユニット(PU)は、ピクチャの、ルーマサンプルの予測ブロック、クロマサンプルの2つの対応する予測ブロック、および予測ブロックサンプルを予測するために使用されるシンタックス構造であり得る。ビデオエンコーダ20は、CUの各PUのルーマ予測ブロック、Cb予測ブロック、およびCr予測ブロックに対して、予測ルーマブロック、予測Cbブロック、および予測Crブロックを生成し得る。

【0048】

ビデオエンコーダ20は、PUに関する予測ブロックを生成するために、イントラ予測またはインター予測を使用し得る。ビデオエンコーダ20がPUの予測ブロックを生成するためにイントラ予測を使用する場合、ビデオエンコーダ20は、PUと関連したピクチャの復号サンプルに基づいて、PUの予測ブロックを生成し得る。

【0049】

ビデオエンコーダ20がPUの予測ブロックを生成するためにインター予測を使用する場合、ビデオエンコーダ20は、PUと関連したピクチャ以外の1つまたは複数のピクチャの復号サンプルに基づいて、PUの予測ブロックを生成し得る。ビデオエンコーダ20は、PUの予測ブロックを生成するために、単予測または双予測を使用し得る。ビデオエンコーダ20がPUに関する予測ブロックを生成するために単予測を使用するとき、PUは単一の動きベクトル(MV)を有し得る。ビデオエンコーダ20がPUに関する予測ブロックを生成するために双予測を使用するとき、PUは2つのMVを有し得る。

【0050】

10

20

30

40

50

ビデオエンコーダ20がCUの1つまたは複数のPUに対して1つまたは複数の予測ブロック(たとえば、予測ルーマブロック、予測Cbブロック、および予測Crブロック)を生成した後、ビデオエンコーダ20は、CUに関する残差ブロックを生成し得る。残差ブロックの中の各サンプルは、CUの予測ブロックのうちの1つの中のサンプルとCUのコーディングブロックの中の対応するサンプルとの間の差分を示し得る。たとえば、ビデオエンコーダ20は、CUに関するルーマ残差ブロックを生成し得る。CUのルーマ残差ブロックの中の各サンプルは、CUの予測ルーマブロックのうちの1つの中のルーマサンプルとCUの元のルーマコーディングブロックの中の対応するサンプルとの間の差分を示す。加えて、ビデオエンコーダ20は、CUに関するCb残差ブロックを生成し得る。CUのCb残差ブロックの中の各サンプルは、CUの予測Cbブロックのうちの1つの中のCbサンプルとCUの元のCbコーディングブロックの中の対応するサンプルとの間の差分を示し得る。ビデオエンコーダ20はまた、CUに関するCr残差ブロックを生成し得る。CUのCr残差ブロックの中の各サンプルは、CUの予測Crブロックのうちの1つの中のCrサンプルとCUの元のCrコーディングブロックの中の対応するサンプルとの間の差分を示し得る。

10

【0051】

さらに、ビデオエンコーダ20は、4分木区画を使用して、CUの残差ブロック(たとえば、ルーマ残差ブロック、Cb残差ブロック、およびCr残差ブロック)を1つまたは複数の変換ブロック(たとえば、ルーマ変換ブロック、Cb変換ブロック、およびCr変換ブロック)に分解し得る。変換ブロックは、同じ変換が適用されるサンプルの、長方形のブロックであってよい。CUの変換ユニット(TU)は、ルーマサンプルの変換ブロック、クロマサンプルの2つの対応する変換ブロック、および変換ブロックサンプルを変換するために使用されるシンタックス構造であり得る。したがって、CUの各TUは、ルーマ変換ブロック、Cb変換ブロック、およびCr変換ブロックに関連付けられ得る。TUに関連付けられたルーマ変換ブロックは、CUのルーマ残差ブロックのサブブロックであり得る。Cb変換ブロックは、CUのCb残差ブロックのサブブロックであり得る。Cr変換ブロックは、CUのCr残差ブロックのサブブロックであり得る。

20

【0052】

ビデオエンコーダ20は、1つまたは複数の変換をTUの変換ブロックに適用して、TUに関する係数ブロックを生成し得る。係数ブロックは、変換係数の2次元アレイであり得る。変換係数はスカラー量であり得る。たとえば、ビデオエンコーダ20は、1つまたは複数の変換をTUのルーマ変換ブロックに適用して、TUに関するルーマ係数ブロックを生成し得る。ビデオエンコーダ20は、1つまたは複数の変換をTUのCb変換ブロックに適用して、TUに関するCb係数ブロックを生成し得る。ビデオエンコーダ20は、1つまたは複数の変換をTUのCr変換ブロックに適用して、TUに関するCr係数ブロックを生成し得る。

30

【0053】

係数ブロック(たとえば、ルーマ係数ブロック、Cb係数ブロック、またはCr係数ブロック)を生成した後、ビデオエンコーダ20は、係数ブロックを量子化し得る。量子化とは、概して、変換係数が量子化されて、場合によっては、変換係数を表すために使用されるデータの量を低減し、さらなる圧縮をもたらすプロセスを指す。ビデオエンコーダ20が係数ブロックを量子化した後、ビデオエンコーダ20は、量子化変換係数を示すシンタックス要素をエントロピー符号化し得る。たとえば、ビデオエンコーダ20は、量子化変換係数を示すシンタックス要素に対してコンテキスト適応型バイナリ算術コーディング(CABAC)を実行し得る。ビデオエンコーダ20は、エントロピー符号化されたシンタックス要素をビットストリームの中に出力し得る。

40

【0054】

CABAC符号化をシンタックス要素に適用するために、ビデオエンコーダは、シンタックス要素を二値化して「ピン」と呼ばれる1つまたは複数のビットの系列を形成し得る。加えて、ビデオエンコーダは、コーディングコンテキストを識別し得る。コーディングコンテキストは、特定の値を有するピンをコーディングすることの確率を識別し得る。たとえば、コーディングコンテキストは、値0のピンをコーディングすることとしての確率0.7、

50

および値1のピンをコーディングすることとしての確率0.3を示し得る。コーディングコンテキストを識別した後、ビデオエンコーダ20は、区間を下位部分区間および上位部分区間に分割し得る。部分区間のうちの一方は値0に関連付けられてよく、他方の部分区間は値1に関連付けられてよい。部分区間の幅は、関連付けられた値に関して、識別されたコーディングコンテキストによって示される確率に比例し得る。シンタックス要素のピンは、下位部分区間に関連付けられた値を有する場合、符号化される値は、下位部分区間の下の方の境界に等しくあり得る。シンタックス要素の同じピンが上位部分区間に関連付けられた値を有する場合、符号化される値は、上位部分区間の下の方の境界に等しくあり得る。シンタックス要素の次のピンを符号化するために、ビデオエンコーダは、符号化されるビットの値に関連付けられた部分区間である区間を用いて、これらのステップを繰り返してよい。ビデオエンコーダ20が次のピンのためのこれらのステップを繰り返すとき、ビデオエンコーダは、識別されたコーディングコンテキストによって示される確率および符号化されたピンの実際の値に基づいて、修正された確率を使用してよい。

【0055】

ビデオデコーダ30がCABAC復号をシンタックス要素に対して実行するとき、ビデオデコーダは、コーディングコンテキストを識別し得る。ビデオデコーダは、次いで、区間を下位部分区間および上位部分区間に分割し得る。部分区間のうちの一方は値0に関連付けられてよく、他方の部分区間は値1に関連付けられてよい。部分区間の幅は、関連付けられた値に関して、識別されたコーディングコンテキストによって示される確率に比例し得る。符号化された値が下位部分区間内にある場合、ビデオデコーダは、下位部分区間に関連付けられた値を有するピンを復号し得る。符号化された値が上位部分区間内にある場合、ビデオデコーダは、上位部分区間に関連付けられた値を有するピンを復号し得る。シンタックス要素の次のピンを復号するために、ビデオデコーダは、符号化された値を含む部分区間である区間を用いて、これらのステップを繰り返してよい。ビデオデコーダが次のピンのためのこれらのステップを繰り返すとき、ビデオデコーダは、識別されたコーディングコンテキストによって示される確率および復号されたピンに基づいて、修正された確率を使用してよい。ビデオデコーダは、次いで、ピンを逆二値化してシンタックス要素を復元し得る。

【0056】

通常のCABAC符号化をすべてのシンタックス要素に対して実行するのではなく、ビデオエンコーダは、バイパスCABACコーディングを使用して一部のシンタックス要素(たとえば、ピン)を符号化し得る。バイパスCABACコーディングをピンに対して実行することは、通常のCABACコーディングをピンに対して実行するよりも、計算量的にコストがかからない場合がある。さらに、バイパスCABACコーディングを実行することは、より高度な並列化およびスループットを可能にし得る。バイパスCABACコーディングを使用して符号化されたピンは、「バイパスピン」と呼ばれることがある。バイパスピンと一緒にグループ化することは、ビデオエンコーダおよびビデオデコーダのスループットを増大させ得る。バイパスCABACコーディングエンジンは、いくつかのピンを単一のサイクルの中でコーディングすることができる場合があるが、通常のCABACコーディングエンジンは、単一のピンしか1つのサイクルの中でコーディングできない場合がある。バイパスCABACコーディングエンジンは、コンテキストを選択せず両方のシンボル(0および1)に対して1/2の確率を前提とし得るので、バイパスCABACコーディングエンジンはより簡単であり得る。したがって、バイパスCABACコーディングでは、区間はちょうど半分に分割される。

【0057】

ビデオエンコーダ20は、エントロピー符号化されたシンタックス要素を含むビットストリームを出力し得る。ビットストリームは、コード化ピクチャの表現および関連したデータを形成するビットのシーケンスを含み得る。言い換えれば、ビットストリームは、ビデオデータのコード化表現を備え得る。ビットストリームは、ネットワークアブストラクションレイヤ(NAL)ユニットのシーケンスを備え得る。NALユニットの各々は、NALユニットヘッダを含み、ローバイトシーケンスペイロード(RBSP)をカプセル化する。NALユニット

10

20

30

40

50

ヘッダは、NALユニットタイプコードを示すシンタックス要素を含み得る。NALユニットのNALユニットヘッダによって規定されるNALユニットタイプコードは、NALユニットのタイプを示す。RBSPは、NALユニット内にカプセル化された整数個のバイトを含むシンタックス構造であり得る。いくつかの事例では、RBSPは、0個のビットを含む。

【0058】

異なるタイプのNALユニットが、異なるタイプのRBSPをカプセル化してよい。たとえば、第1のタイプのNALユニットがピクチャパラメータセット(PPS)用のRBSPをカプセル化してよく、第2のタイプのNALユニットがコード化スライス用のRBSPをカプセル化してよく、第3のタイプのNALユニットがSEI用のRBSPをカプセル化してよく、以下同様である。(パラメータセット用およびSEIメッセージ用のRBSPではなく)ビデオコーディングデータ用のRBSPをカプセル化するNALユニットは、ビデオコーディングレイヤ(VCL)NALユニットと呼ばれることがある。

【0059】

ビデオデコーダ30は、ビデオエンコーダ20によって生成されたビットストリームを受信し得る。加えて、ビデオデコーダ30は、シンタックス要素をビットストリームから復号するために、ビットストリームを構文解析し得る。ビデオデコーダ30は、ビットストリームから復号されたシンタックス要素に少なくとも部分的に基づいて、ビデオデータのピクチャを再構成し得る。ビデオデータを再構成するためのプロセスは、概して、ビデオエンコーダ20によって実行されるプロセスと相反であり得る。たとえば、ビデオデコーダ30は、PUのMVを使用して、現在CUのPUに関する予測ブロックを決定し得る。加えて、ビデオデコーダ30は、現在CUのTUと関連した変換係数ブロックを逆量子化し得る。ビデオデコーダ30は、変換係数ブロックに対して逆変換を実行して、現在CUのTUと関連した変換ブロックを再構成し得る。ビデオデコーダ30は、現在CUのPUに関する予測ブロックのサンプルを、現在CUのTUの変換ブロックの対応するサンプルに加算することによって、現在CUのコーディングブロックを再構成し得る。ピクチャのCUごとにコーディングブロックを再構成することによって、ビデオデコーダ30はピクチャを再構成し得る。

【0060】

上述されたように、本明細書で説明する技法は、ビデオデータのパレットベースコーディングに適用され得る。パレットベースコーディングは、概して、以下の技法のうちの1つまたは複数を含み得る。パレットベースビデオコーディングを用いて、ビデオコーダ(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、特定のエリア(たとえば、所与のブロック)のビデオデータを表すためのカラーのテーブルとして、いわゆる「パレット」を形成し得る。各画素は、画素のカラーを表すパレットの中のエントリに関連付けられ得る。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、画素値をパレットの中の適切な値に関連付けるインデックスをコーディングし得る。

【0061】

HEVCフレームワークに関して、一例として、パレットベースコーディング技法は、コーディングユニット(CU)モードとして使用されるように構成され得る。他の例では、パレットベースコーディング技法は、HEVCのフレームワークにおけるPUモードとして使用されるように構成され得る。したがって、CUモードのコンテキストにおいて説明される、以下の開示されるプロセスのすべては、付加的または代替的に、PUに適用される。しかしながら、そのような技法は、独立に機能するように、または他の既存の、もしくはこれから開発されるべきシステム/規格の一部として機能するように適用され得るので、これらのHEVCベースの例は、本明細書で説明するパレットベースコーディング技法の制約または限定と見なされるべきでない。これらの場合には、パレットコーディングのためのユニットは、正方形ブロック、長方形ブロック、または非矩形形状の領域でさえあり得る。

【0062】

X. GuoおよびA. Saxena、「RCE4: Summary report of HEVC Range Extension Core Experiments 4 (RCE4) on palette coding for screen content」、JCTVC-P0035、サンノゼ、米国、2014年1月9~17日という文書がパレットベースモードの2つのテスト結果を記載

10

20

30

40

50

し、それらは、特にスクリーンコンテンツにとって、ボンテガード(Bjontegaard)ひずみレート(BDレート)の著しい低減を達成すると報告された。2つの方法が、以下に手短に要約される。

【 0 0 6 3 】

たとえば、X. Guo、Y. Lu、およびS. Li、「RCE4: Test 1. Major-color-based screen content coding」、JCTVC-P0108、サンノゼ、米国、2014年1月9～17日という文書において説明されるような1つの例示的な方法では、画素を分類するためにヒストグラムベースのアルゴリズムが使用される。詳細には、ヒストグラムにおいて最も重要なN個のピーク値が、コーディングのためのメジャーカラー(major color)として選択される。メジャーカラーに近接する画素値は、メジャーカラーに量子化される。いかなるメジャーカラーセットにも属さない他の画素は、同様にコーディングの前に量子化されるエスケープピクセル(escape pixel)である。可逆コーディングの場合、量子化は使用されない。

10

【 0 0 6 4 】

分類を使用することによって、コーディングユニット(CU)の画素が、カラーインデックスに変換され得る。その後、メジャーカラー番号および値がコーディングされる。次いで、カラーインデックスは、次のようにコーディングされる。

- ・画素ラインごとに、コーディングモードを示すためのフラグがシグナリングされる。3つのモード、すなわち、水平モード、垂直モード、および標準モードがある。

- ・モードが水平モードである場合、ライン全体が同じカラーインデックスを共有する。この場合、カラーインデックスが送信される。

20

- ・モードが垂直モードである場合、ライン全体が上のラインと同じである。この場合、何も送信されない。現在のラインは、上のラインのカラーインデックスをコピーする。

- ・モードが標準モードである場合、それが左の画素および上の画素のうちの1つと同じであるかどうかを示すために、画素位置ごとにフラグがシグナリングされる。そうでない場合、インデックス自体が送信される。

加えて、画素がエスケープピクセルである場合、画素値が送信される。

【 0 0 6 5 】

たとえば、L. Guo、W. Pu、M. Karczewicz、J. Sole、R. Joshi、およびF. Zou、「RCE 4: Results of Test 2 on Palette Mode for Screen Content Coding」、JCTVC-P0198、サンノゼ、米国、2014年1月9～17日という文書において説明されるような別の例示的な方法では、パレットベースコーディングモードがCUモードとして含まれる。第2の方法の符号化プロセスは、以下のことを含み得る。

30

- ・パレットの送信: 左のCU(現在コーディングされているCUの左側に隣接するCU)のパレットに基づいて現在パレットを符号化するために、エントリ単位の予測方式が使用される。その後、パレットの予測されないエントリが送信される。

- ・画素値の送信: CUの中の画素が以下の3つのモードを使用してラスト走査順序で符号化される。

- ・「ランモード」: パレットインデックスが最初にシグナリングされ、「palette_run」(M)が後続する。後続するM個のパレットインデックスは、最初にシグナリングされたパレットインデックスと同じである。

40

- ・「上方コピーモード」: 後続するN個のパレットインデックスが、それぞれ、それらの上方に隣接するものと同じであることを示すために、値「copy_run」(N)が送信される。

- ・「画素モード」: 予測フラグが最初に送信される。1に等しいフラグ値は、再構成された上の隣接画素を予測子として使用する予測残差が送信されることを示す。このフラグの値が0である場合、画素値は、予測なしに送信される。

【 0 0 6 6 】

パレットは、パレットコード化ブロック(たとえば、CU)にとって比較的重要なビットの部分を作り上げることができる。したがって、ビデオコードは、前にコーディングされた

50

パレットの1つまたは複数のエントリに基づいて、パレットの1つまたは複数のエントリを予測し得る(たとえば、「パレットの送信」に関して上述したように)。

【0067】

いくつかの例では、ビデオコードは、パレットエントリを予測するとき、パレット予測子リストを生成し得る。たとえば、C. Gisquet、G. Laroche、およびP. Onno、「AhG10: Palette predictor stuffing」、JCTVC-Q0063という文書が、パレット予測子を決定するための1つの例示的なプロセスを開示する。いくつかの例では、ビデオコードは、現在コーディングされているブロック用のパレットの中の1つまたは複数のエントリを予測するためにパレット予測子リストの中の各アイテムが使用されるのか(それとも、使用されないのか)を示すために、ブールベクトルを使用し得る。

10

【0068】

いくつかの例では、パレット予測子リストの中のアイテムのすべては、前にコーディングされたパレット(たとえば、前にコーディングされたブロックを用いてコーディングされたパレット)から導出される。しかしながら、そのようなパレットは、現在CUから空間的に遠く離れている場合があり、そのことがパレット相関を比較的弱くさせることがある。概して、パレット予測子テーブルを拡張することが役に立つ場合がある(たとえば、より正確な予測子を提供し得、そのことが効率の向上をもたらし得る)。しかしながら、比較的大きいパレット予測子テーブルを決定および使用することは、比較的長いブールベクトルをもたらす。

【0069】

20

パレットコーディングの一例では、ビデオエンコーダ20は、ビデオフレームの特定の領域のためにパレットベースコーディングモードが使用されるか否かを示すフラグ「PLT_Mode_flag」などのシンタックス要素を生成し得る。たとえば、PLT_Mode_flagは、スライスレベル、CUレベル、PUレベル、またはビデオフレームの任意の他のレベルにおいて生成され得る。たとえば、ビデオエンコーダ20は、PLT_Mode_flagをCUレベルにおいて生成し得、PLT_Mode_flagを符号化ビデオビットストリームの中でシグナリングし得る。ビデオデコーダ30は、次いで、符号化ビデオビットストリームを復号すると、PLT_Mode_flagを構文解析し得る。この例では、このPLT_Mode_flagの1に等しい値は、パレットモードを使用して現在CUが符号化されていることを規定する。この場合、ビデオデコーダ30は、CUを復号するためにパレットベースコーディングモードを適用し得る。いくつかの例では、シンタックス要素が、CU用の複数の異なるパレットモードのうちの1つを示し得る。

30

【0070】

このPLT_Mode_flagの0に等しい値は、パレットモード以外のモードを使用して現在CUが符号化されていることを規定する。たとえば、様々なインター予測モード、イントラ予測モード、または他のコーディングモードのいずれかが使用され得る。PLT_Mode_flagの値が0であるとき、それぞれのCUを符号化するためにどの特定のモードが使用されているのかをシグナリングするためにさらなる情報が送信されてよく、その場合、そのような特定のモードは、通常、HEVCコーディングモードであり得る。PLT_Mode_flagの使用が例として説明される。しかしながら、他の例では、CU(または、他の例ではPU)のためにパレットベースコーディングモードが使用されるべきかどうかを示すために、または複数のモードのうちのどれが使用されるべきかを示すために、マルチビットコードなどの他のシンタックス要素が使用され得る。

40

【0071】

PLT_Mode_flagまたは他のシンタックス要素はまた、より高位のレベルにおいて送信されてよい。たとえば、PLT_Mode_flagは、スライスレベルにおいて送信されてよい。この場合、フラグの1に等しい値は、スライスの中のCUのすべてがパレットモードを使用して符号化されることを暗示する(そのことは、たとえば、パレットモードまたは他のモードのためのモード情報がCUレベルにおいて送信される必要がないことを意味する)。同様に、このフラグは、ピクチャパラメータセット(PPS)レベル、シーケンスパラメータセット(PPS)レベル、またはビデオパラメータセット(VPS)レベルにおいてシグナリングされ得る

50

。また、特定のピクチャ、スライスなどに対してパレットモードが有効化されているのか、それとも無効化されているのかを規定するフラグがこれらのレベルのうちの1つにおいて送信されてよく、一方、PLT_Mode_flagは、CUごとにパレットベースコーディングモードが使用されているのかどうかを示す。この場合、スライスレベル、PPSレベル、SPSレベル、またはVPSレベルにおいて送信されたフラグまたは他のシンタックス要素が、パレットコーディングモードが無効化されていることを示す場合、いくつかの例では、CUごとにPLT_Mode_flagをシグナリングする必要がない場合がある。あるいは、スライスレベル、PPSレベル、SPSレベル、またはVPSレベルにおいて送信されたフラグまたは他のシンタックス要素が、パレットコーディングモードが有効化されていることを示す場合、パレットベースコーディングモードが使用されているかどうかを示すために、CUごとにPLT_Mode_flagがさらにシグナリングされてよい。やはり、上述のように、CUのパレットベースコーディングを示すためにこれらの技法を適用することは、付加的または代替的に、PUのパレットベースコーディングを示すために使用され得る。

【 0 0 7 2 】

PLT_Mode_flagなどのフラグは、同様に、または代替的に、条件付きで送信または推定され得る。PLT_Mode_flagを送信し、またはそのフラグを推定するための条件は、例として、CUのサイズ、フレームタイプ、色空間、色成分、フレームサイズ、フレームレート、スケーラブルビデオコーディングにおけるレイヤid、またはマルチビューコーディングにおけるビューidのうちの1つまたは複数であり得る。

【 0 0 7 3 】

パレットの生成および送信のための技法が、次に説明される。ビデオエンコーダ20は、ビデオフレームの特定のレベル(たとえば、CU)を符号化するためにビデオエンコーダ20によって使用されたパレットを、構成および/または再構成するためにビデオデコーダ30によって使用され得る、1つまたは複数のシンタックス要素および値を、生成およびシグナリングするように構成され得る。いくつかの例では、ビデオエンコーダ20は、CUごとにパレットを示し得、または別のやり方でシグナリングし得る。他の例では、ビデオエンコーダ20は、いくつかのCUの間で共有され得るパレットを示し得、または別のやり方でシグナリングし得る。

【 0 0 7 4 】

たとえば、含まれる画素値の数に換算したパレットのサイズは固定値であってよく、またはビデオエンコーダ20によって符号化ビデオビットストリームの中でシグナリングされ得る。ビデオデコーダ30は、パレットサイズの表示を、符号化ビデオビットストリームから受信および復号し得る。シグナリングは異なる成分に対して別個であってよく、または単一のサイズがすべての成分に対してシグナリングされてもよい。異なる成分は、たとえば、ルーマ成分およびクロマ成分であり得る。シグナリングは、単項コードまたは(たとえば、パレットサイズの最大限において切り取られた)短縮単項コードを使用することができる。指数ゴロムコードまたはライスゴロムコードも使用され得る。いくつかの例では、サイズのシグナリングは次の方法で行われ得、すなわち、パレットの中のエントリをシグナリングした後、「ストップ」フラグがシグナリングされる。このフラグの1に等しい値は、現在のエントリがパレットの中の最後のエントリであることを規定し、このフラグの0に等しい値は、パレットの中にさらに多くのエントリがあることを規定する。すでに構成されたパレットがパレットサイズの最大限に達している場合、「ストップ」フラグはエンコーダによって送信されなくてよい。いくつかの例では、パレットのサイズはまた、「フラグPLT_Mode_flagの送信」について上記で説明したものと同一方法で、副次的情報に基づいて条件付きで送信または推定され得る。

【 0 0 7 5 】

パレットは、CUの中の色成分ごとに別個に送信され得る。たとえば、このCUのY成分用のパレット、このCUのU成分用の別のパレット、およびこのCUのV成分用のさらに別のパレットがあってもよい。Yパレットの場合、エントリはこのCUの中の代表的なY値で(おそらく)あり得る。同じことがU成分およびV成分に適用される。パレットがCUの中の色成分のすべ

10

20

30

40

50

てに対して送信され得ることも可能である。この例では、パレットの中の i 番目のエントリは、トリプル(Y_i 、 U_i 、 V_i)である。この場合、パレットは成分の各々に関する値を含む。

【0076】

パレットの予測は、上述された「パレットの送信」の代替手法である。いくつかの例では、パレット予測技法は、パレットシグナリング技法と一緒に使用され得る。すなわち、ビデオエンコーダ20は、パレットエントリの総数の一部分を予測するためにビデオデコーダ30によって使用され得るシンタックス要素をシグナリングするように構成され得る。加えて、ビデオエンコーダ20は、パレットエントリの別の部分を明示的にシグナリングするように構成され得る。

10

【0077】

パレット予測手法の一例では、CUごとに、1つのフラグ「pred_palette_flag」が送信される。このフラグの1に等しい値は、現在CU用のパレットが過去のデータから予測され、したがって、パレットが送信される必要がないことを規定する。このフラグの0に等しい値は、現在CUのパレットが送信される必要があることを意味する。フラグは、異なる色成分に対して別個であってよく(たとえば、YUVビデオにおけるCUに対して3つのフラグが送信される必要があるように)、または単一のフラグがすべての色成分に対してシグナリングされてもよい。たとえば、単一のフラグが、成分のすべてに対してパレットが送信されるかどうか、または成分のすべてに対してパレットが予測されるかどうかを示し得る。

【0078】

20

いくつかの例では、予測は、以下の方式で実行され得る。予測フラグ値が1に等しい場合、現在CUに対して、ビデオエンコーダ20は、すでに符号化された隣接CUのうちの1つまたは複数のパレットをコピーする。すでに符号化された隣接CUのパレットは、送信または予測されていることがある。たとえば、コピーされた隣接CUは、左の隣接CUであり得る。左のCUのパレットが利用可能でない場合(左のCUがパレットモードを使用して符号化されていないか、または現在CUがピクチャの最初の列にある場合のように)、パレットのコピーは、現在CUの上のCUからであってよい。コピーされるパレットはまた、いくつかの隣接CUのパレットの組合せであり得る。たとえば、複数の隣接CUの1つまたは組合せのパレットに基づいてパレットを生成するために、1つまたは複数の公式、関数、規則などが適用されてよい。

30

【0079】

現在CUがパレットをそこからコピーする候補CUを示すために、ビデオエンコーダ20によって候補リストが構成され得るとともにインデックスが送信されることも可能である。ビデオデコーダ30は、同じ候補リストを構成し得、次いで、インデックスを使用して、現在CUとともに使用するための対応するCUのパレットを選択し得る。たとえば、候補リストは、スライス内またはピクチャ内でコーディングされるべき現在CUに対して上の1つのCU、および左側の1つのCUを含んでよい。この例では、フラグまたは他のシンタックス要素が、候補選択肢を示すためにシグナリングされてよい。たとえば、送信される0に等しいフラグは、コピーが左のCUからであることを意味し、送信される1に等しいフラグは、コピーが上のCUからであることを意味する。ビデオデコーダ30は、対応する隣接CUからコピーされるべきパレットを選択し、現在CUを復号する際に使用するためにそれをコピーする。予測はまた、現在CUの因果的隣接物(causal neighbor)における最多のサンプル値を使用して導出され得る。

40

【0080】

パレットの予測はまた、エントリ単位であり得る。パレットの中のエントリごとに、ビデオエンコーダ20は、フラグを生成およびシグナリングする。所与のエントリに対するフラグの1に等しい値は、予測される値(たとえば、左のCUのような選択された候補CUからの対応するエントリ)がこのエントリの値として使用されることを規定する。フラグの0に等しい値は、このエントリが予測されず、その値がビデオエンコーダ20からビデオデコーダ30へ送信されること、たとえば、後でビデオデコーダ30によって復号できるようにビデオ

50

エンコーダ20によって符号化されたビットストリームの中でシグナリングされることを規定する。

【0081】

「pred_palette_flag」の値、現在CUのパレットを予測するためにそのパレットが使用される候補CU、または候補を構成するための規則はまた、「フラグPLT_Mode_flagの送信」について上記で説明したものと同一方法で、副次的情報に基づいて条件付きで送信または推定され得る。

【0082】

次に、ビデオエンコーダ20は、それぞれのどのパレットエントリがCUの中の各画素に関連付けられているのかを示すマップを、生成およびシグナリングし得る。マップの中のi番目のエントリは、CUの中のi番目の位置に対応する。i番目のエントリの1に等しい値は、CUの中のこのi番目のロケーションにおける画素値がパレットの中の値のうちの1つであり、ビデオデコーダ30が画素値を再構成できるようにパレットインデックスがさらに送信されることを規定する(パレットの中にただ1つのエントリしかない場合、パレットインデックスの送信はスキップされてよい)。i番目のエントリの0に等しい値は、CUの中のi番目の位置における画素値がパレットの中になく、したがって、画素値がビデオデコーダ30へ明示的に送信されることを規定する。

【0083】

CUの中のある位置における画素値がパレットの中の値である場合、CUの中で隣接する位置が同じ画素値を有する確率が高いことが観察される。そのため、ある位置に関するパレットインデックス(jと呼び、それは画素値sに対応する)を符号化した後、ビデオエンコーダ20は、異なる画素値に走査が到達する前の、CUの中での同じ画素値sとして連続した値の数を示すためのシンタックス要素「ラン」を送信し得る。たとえば、すぐ次のものがsと異なる値を有する場合、ラン=0が送信される。次のものがsであるがその後のものがsでない場合、ラン=1である。

【0084】

ランが送信されない場合(たとえば、暗黙的ラン導出)、ランの値は、たとえば、4、8、16などの定数であってよく、またはランの値も副次的情報に依存し得る。たとえば、ランの値はブロックサイズに依存し得、たとえば、ランは、現在ブロックの幅か、現在ブロックの高さか、現在ブロックの半分の幅(または、半分の高さ)か、ブロックの幅および高さの分数か、ブロックの高さ/幅の複数に等しい。ランの値はまた、QP、フレームタイプ、色成分、カラーフォーマット(たとえば、444、422、420)および/または色空間(たとえば、YUV、RGB)に依存し得る。ランの値はまた、走査方向に依存し得る。他の例では、ランの値は、他のタイプの副次的情報に依存し得る。ランの値はまた、高レベルシンタックス(たとえば、PPS、SPS)を使用してシグナリングされてよい。

【0085】

いくつかの例では、マップが送信される必要がない場合がある。ランは、いくつかのロケーションにおいてのみ開始し得る。たとえば、ランは、各行の先頭またはN行ごとの先頭においてのみ開始し得る。開始ロケーションは、異なる走査方向に対して異なってよい。たとえば、垂直走査が使用される場合、ランは、列の先頭またはN列ごとの先頭においてのみ開始し得る。開始ロケーションは、副次的情報に依存し得る。たとえば、開始ロケーションは、各行もしくは各列の中間点、または各行/列の $1/n$ 、 $2/n$ 、 $\dots (n-1)/n$ (すなわち、分数)であってよい。開始ロケーションはまた、QP、フレームタイプ、色成分、カラーフォーマット(たとえば、444、422、420)および/または色空間(たとえば、YUV、RGB)に依存し得る。他の例では、ランの開始位置は、他のタイプの副次的情報に依存し得る。開始位置はまた、高レベルシンタックス(たとえば、PPS、SPSなど)を使用してシグナリングされ得る。

【0086】

暗黙的開始位置導出および暗黙的ラン導出が組み合わされることも可能である。たとえば、ランは、2つの隣接する開始位置の間の距離に等しい。開始点がすべての行の先頭(す

10

20

30

40

50

なわち、1番目の位置)である場合、ランの長さは行である。

【0087】

走査方向は垂直または水平であってよい。走査方向を示すために、CUごとにフラグが送信されることが可能である。フラグは成分ごとに別個に送信されてよく、または単一のフラグが送信されてよく、示された走査方向がすべての色成分に適用される。45度または135度のような、他の走査方向が使用されることも可能である。走査順序は固定であってよく、または「フラグPLT_Mode_flagの送信」について上記で説明したものと同一方法で、副次的情報に依存し得る。

【0088】

上で、パレットをどのように送信するのが説明されている。上記で説明した例の代案は、パレットをオンザフライで構成することである。この場合、CUの先頭において、パレットの中にエントリがなく、ビデオエンコーダ20がCUの中の位置に対して画素の新しい値をシグナリングするとき、これらの値がパレットの中に含まれる。すなわち、CUの中の位置に対して画素値が生成および送信されるとき、ビデオエンコーダ20は画素値をパレットに追加する。次いで、ビデオエンコーダ20に画素値を送信させる代わりに、同じ値を有するCUの中の後の位置が、たとえば、インデックス値を用いてパレットの中の画素値を参照し得る。同様に、ビデオデコーダ30は、CUの中の位置に対する新しい画素値(たとえば、エンコーダによってシグナリングされる)を受信すると、ビデオデコーダ30によって構成されるパレットの中に画素値を含める。パレットに追加された画素値をCUの中の後の位置が有するとき、ビデオデコーダ30は、たとえば、CUの中の画素値の再構成のために、パレットの中の対応する画素値を識別するインデックス値などの情報を受信し得る。

【0089】

最大パレットサイズに到達した場合、たとえば、パレットが動的にオンザフライで構成されるとき、エンコーダおよびデコーダは、パレットのエントリを除去するための同じメカニズムを共有する。1つの方法は、パレットの中の最も古いエントリを除去することである(FIFO待ち行列)。別の方法は、パレットの中の最も使用されないエントリを除去することである。別の方法は、両方の方法(パレットの中での時間および使用頻度)を重み付けして、置き換えられるべきエントリを決定することである。一例として、ある画素値エントリがパレットから除去され、パレットの中の後の位置においてその画素値が再び発生する場合、エンコーダは、エントリをパレットの中に含める代わりに、その画素値を送信し得る。追加または代替として、そのような画素値は、除去された後、たとえば、エンコーダおよびデコーダがCUの中の位置を走査するとき、パレットの中に再び入れられ得ることが可能である。

【0090】

本開示はまた、初期パレットシグナリングをパレットのオンザフライ導出と組み合わせることを考慮する。一例では、初期パレットは、画素のコーディングとともに更新されることになる。たとえば、初期パレットを送信すると、ビデオエンコーダ20は、値を初期パレットに追加し得、またはCUの中のさらなるロケーションの画素値が走査されるとき、初期パレットの中の値を変更し得る。同様に、初期パレットを受信すると、ビデオデコーダ30は、値を初期パレットに追加し得、またはCUの中のさらなるロケーションの画素値が走査されるとき、初期パレットの中の値を変更し得る。同様に、現在CUが、パレット全体の送信を使用するのか、オンザフライのパレット生成を使用するのか、それとも初期パレットの送信とオンザフライ導出による初期パレットの更新との組合せを使用するのかを、エンコーダはシグナリングすることができる。いくつかの例では、初期パレットは最大パレットサイズの完全なパレットであり得、その場合、初期パレットの中の値は変更されてよく、または初期パレットはサイズが低減されたパレットであり得、その場合、値が初期パレットに追加され、場合によっては、初期パレットの値が変更される。

【0091】

上で、画素値を識別することによってマップをどのように送信するのが説明された。上述された方法と一緒に、ラインをコピーすることをシグナリングすることによって、マ

10

20

30

40

50

ップの送信が行われ得る。一例では、エントリに対する画素値が上方の(または、走査が垂直である場合、左側の列の中の)ラインのエントリの画素値に等しくなるように、ラインをコピーすることがビデオエンコーダ20によってシグナリングされる。次いで、ラインからコピーされるエントリの「ラン」がシグナリングされ得る。同様に、コピー元のラインが示され得、この目的のために上方のいくつかのラインがバッファリングされ得る。たとえば、前の4行が記憶され、どの行がコピーされるのかが短縮単項コードまたは他のコードを用いてシグナリングされ得、次いで、その行の何個のエントリがコピーされるのか、すなわち、ランがシグナリングされ得る。したがって、いくつかの例では、エントリに対する画素値は、現在の行のすぐ上の行、または現在の行の上方の2つ以上の行の中のエントリの画素値に等しいものとして、シグナリングされ得る。

10

【0092】

ランがシグナリングされない場合、ランの値は、定数/固定であってよく、または上述された方法を使用して、副次的情報に依存し得る(デコーダによって導出されてよい)。

【0093】

マップが送信される必要がないことも可能である。たとえば、ランは、いくつかの位置においてのみ開始し得る。開始位置は、固定であってよく、または副次的情報に依存し得る(デコーダによって導出されてよい)、そのため、開始位置のシグナリングはスキップされてよい。代わりに、上述された1つまたは複数の技法が適用されてもよい。暗黙的開始位置導出および暗黙的ラン導出はまた、上述されたものと同じ方法を使用して組み合わせられてよい。

20

【0094】

マップ送信の両方の方法が使用され、次いで、フラグが、画素がパレットから取得されるのか、それとも前のラインから取得されるのかを示し得、次いで、インデックスが、パレットの中のエントリまたは行を示し、最後に「ラン」を示す。

【0095】

図2は、本開示のパレットコーディング技法を実施し得る例示的なビデオエンコーダ20を示すブロック図である。図2は説明のために提供され、広く例示されるとともに本開示で説明されるような技法の限定と見なされるべきでない。説明のために、本開示は、HEVCコーディングのコンテキストにおけるビデオエンコーダ20を説明する。しかしながら、本開示の技法は、他のコーディング規格または方法に適用可能であり得る。

30

【0096】

ビデオエンコーダ20は、本開示で説明する様々な例によるパレットベースビデオコーディングのための技法を実行するように構成され得るデバイスの一例を表す。たとえば、ビデオエンコーダ20は、HEVCコーディングにおけるCUまたはPUなどのビデオデータの様々なブロックを、パレットベースコーディングまたは非パレットベースコーディングのいずれかを使用して選択的にコーディングするように構成され得る。

【0097】

図2の例では、ビデオエンコーダ20は、予測処理ユニット100、残差生成ユニット102、変換処理ユニット104、量子化ユニット106、逆量子化ユニット108、逆変換処理ユニット110、再構成ユニット112、フィルタユニット114、復号ピクチャバッファ(DPB)116、ビデオメモリ119、およびエントリピー符号化ユニット118を含む。予測処理ユニット100は、インター予測処理ユニット120およびイントラ予測処理ユニット126を含む。インター予測処理ユニット120は、動き推定ユニットおよび動き補償ユニット(図示せず)を含む。ビデオエンコーダ20はまた、本開示で説明するパレットベースコーディング技法の様々な態様を実行するように構成されたパレットベース符号化ユニット122を含む。他の例では、ビデオエンコーダ20は、より多数の、より少数の、または異なる機能の構成要素を含んでよい。

40

【0098】

図4～図9を参照しながら以下でより詳細に説明するように、ビデオエンコーダ20のパレットベース符号化ユニット122は、ビデオデータの現在ブロック用のパレットを生成する

50

ことと、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成することであって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化することと、パレットを使用してビデオデータの現在ブロックを符号化することとを行うように構成され得る。

【 0 0 9 9 】

ビデオエンコーダ20は、ビデオデータを受信し得る。ビデオエンコーダ20は、ビデオデータのピクチャのスライスの中の各CTUを符号化し得る。CTUの各々は、ピクチャの、等しいサイズのルーマコーディングツリーブロック(CTB)、および対応するCTBに関連付けられ得る。CTUを符号化することの一部として、予測処理ユニット100は、4分木区分を実行して、CTUのCTBを次第に小さくなるブロックに分割し得る。より小さいブロックは、CUのコーディングブロックであり得る。たとえば、予測処理ユニット100は、CTUと関連したCTBを4つの等しいサイズのサブブロックに区分し得、サブブロックのうちの1つまたは複数を4つの等しいサイズのサブサブブロックに区分し得、以下同様である。

【 0 1 0 0 】

図2に示すように、ビデオメモリ119は、ビデオフレーム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。ビデオメモリ119は、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し得る(たとえば、ビデオデータを記憶するように構成され得る)。ビデオメモリ119に記憶されるビデオデータは、たとえば、図1のビデオソース18から取得され得る。DPB116は、ビデオデータをビデオエンコーダ20によって(たとえば、イントラ予測コーディングモードまたはインター予測コーディングモードとも呼ばれる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)符号化するために使用するために、参照ビデオデータを記憶するDPBの一例である。ビデオメモリ119およびDPB116は、同期DRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、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などの、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。ビデオメモリ119およびDPB116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備えられてよい。様々な例では、ビデオメモリ119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 0 1 】

ビデオエンコーダ20は、CTUのCUを符号化して、CUの符号化表現(すなわち、コード化CU)を生成し得る。CUを符号化することの一部として、予測処理ユニット100は、CUの1つまたは複数のPUの中でCUに関連付けられたコーディングブロックを区分し得る。したがって、各PUは、ルーマ予測ブロックおよび対応するクロマ予測ブロックに関連付けられ得る。ビデオエンコーダ20およびビデオデコーダ30は、様々なサイズを有するPUをサポートし得る。先に示したように、CUのサイズは、CUのルーマコーディングブロックのサイズを指すことがあり、PUのサイズは、PUのルーマ予測ブロックのサイズを指すことがある。特定のCUのサイズが $2N \times 2N$ であると仮定すると、ビデオエンコーダ20およびビデオデコーダ30は、イントラ予測に対して $2N \times 2N$ または $N \times N$ としてのPUサイズ、およびインター予測に対して $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 、または類似のものとしての対称のPUサイズをサポートし得る。ビデオエンコーダ20およびビデオデコーダ30はまた、インター予測に対して $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ としてのPUサイズ向けの非対称区分をサポートし得る。

【 0 1 0 2 】

インター予測処理ユニット120は、インター予測をCUの各PUに対して実行することによって、PUに関する予測データを生成し得る。PUに関する予測データは、PUの予測ブロックおよびPUに関する動き情報を含み得る。インター予測ユニット121は、PUがIスライスの中にあるのか、Pスライスの中にあるのか、それともBスライスの中にあるのかに応じて、CUのPUのための異なる動作を実行し得る。Iスライスでは、すべてのPUがイントラ予測され

る。したがって、PUがIスライスの中にある場合、インター予測ユニット121は、インター予測をPUに対して実行しない。したがって、Iモードで符号化されるブロックの場合、予測されるブロックは、同じフレーム内で前に符号化された隣接ブロックからの空間予測を使用して形成される。

【0103】

PUがPスライスの中にある場合、インター予測処理ユニット120の動き推定ユニットは、PUに関する参照領域を求めて参照ピクチャのリスト(たとえば、「RefPicList0」)の中の参照ピクチャを検索し得る。PUに関する参照領域は、PUのサンプルブロックに最も密に対応するサンプルブロックを含む、参照ピクチャ内の領域であり得る。動き推定ユニットは、PUに関する参照領域を含む参照ピクチャのRefPicList0の中での位置を示す参照インデックスを生成し得る。加えて、動き推定ユニットは、PUのコーディングブロックと参照領域に関連付けられた参照ロケーションとの間の空間変位を示すMVを生成し得る。たとえば、MVは、現在の復号ピクチャの中の座標から参照ピクチャの中の座標までのオフセットを提供する2次元ベクトルであってよい。動き推定ユニットは、PUの動き情報として参照インデックスおよびMVを出力し得る。インター予測処理ユニット120の動き補償ユニットは、PUの動きベクトルによって示された参照ロケーションにおける実際のサンプルまたは補間されたサンプルに基づいて、PUの予測ブロックを生成し得る。

10

【0104】

PUがBスライスの中にある場合、動き推定ユニットは、PUに対して単予測または双予測を実行し得る。PUに対して単予測を実行するために、動き推定ユニットは、PUに関する参照領域を求めてRefPicList0または第2の参照ピクチャリスト(「RefPicList1」)の参照ピクチャを検索し得る。動き推定ユニットは、PUの動き情報として、参照領域を含む参照ピクチャのRefPicList0またはRefPicList1の中での位置を示す参照インデックス、PUの予測ブロックと参照領域に関連付けられた参照ロケーションとの間の空間変位を示すMV、および参照ピクチャがRefPicList0の中にあるのか、それともRefPicList1の中にあるのかを示す1つまたは複数の予測方向インジケータを出力し得る。インター予測処理ユニット120の動き補償ユニットは、PUの動きベクトルによって示された参照領域における実際のサンプルまたは補間されたサンプルに少なくとも部分的に基づいて、PUの予測ブロックを生成し得る。

20

【0105】

PUに対して双方向インター予測を実行するために、動き推定ユニットは、PUに関する参照領域を求めてRefPicList0の中の参照ピクチャを検索し得、また、PUに関する別の参照領域を求めてRefPicList1の中の参照ピクチャを検索し得る。動き推定ユニットは、参照領域を含む参照ピクチャのRefPicList0およびRefPicList1の中での位置を示す参照ピクチャインデックスを生成し得る。加えて、動き推定ユニットは、参照領域に関連付けられた参照ロケーションとPUのサンプルブロックとの間の空間変位を示すMVを生成し得る。PUの動き情報は、PUの参照インデックスおよびMVを含み得る。動き補償ユニットは、PUの動きベクトルによって示された参照領域における実際のサンプルまたは補間されたサンプルに少なくとも部分的に基づいて、PUの予測ブロックを生成し得る。

30

【0106】

イントラ予測処理ユニット126は、イントラ予測をPUに対して実行することによって、PUに関する予測データを生成し得る。PUに関する予測データは、PUに関する予測ブロックおよび様々なシンタックス要素を含み得る。イントラ予測処理ユニット126は、Iスライス、Pスライス、およびBスライスの中のPUに対して、イントラ予測を実行し得る。

40

【0107】

イントラ予測をPUに対して実行するために、イントラ予測処理ユニット126は、複数のイントラ予測モードを使用して、PUに関する予測データの複数のセットを生成し得る。イントラ予測処理ユニット126は、隣接PUのサンプルブロックからのサンプルを使用して、PUに関する予測ブロックを生成し得る。PU、CU、およびCTUに対して左から右、上から下への符号化順序を仮定すると、隣接PUは、PUの上、右上、左上、または左であってよい。イ

50

ントラ予測処理ユニット126は、様々な数のイントラ予測モード、たとえば、33方向性イントラ予測モードを使用し得る。いくつかの例では、イントラ予測モードの数は、PUと関連した領域のサイズに依存し得る。

【0108】

予測処理ユニット100は、PUに対してインター予測処理ユニット120によって生成される予測データ、またはPUに対してイントラ予測処理ユニット126によって生成される予測データの中から、CUのPUに関する予測データを選択し得る。いくつかの例では、予測処理ユニット100は、予測データのセットのレート/ひずみメトリックに基づいて、CUのPUに関する予測データを選択する。選択される予測データの予測ブロックは、選択予測ブロックと本明細書で呼ばれることがある。

10

【0109】

残差生成ユニット102は、CUのコーディングブロック(たとえば、ルーマコーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)およびCUのPUの選択予測ブロック(たとえば、予測ルーマブロック、予測Cbブロック、および予測Crブロック)に基づいて、CUの残差ブロック(たとえば、ルーマ残差ブロック、Cb残差ブロック、およびCr残差ブロック)を生成し得る。たとえば、残差生成ユニット102は、残差ブロックの中の各サンプルがCUのコーディングブロックの中のサンプルとCUのPUの対応する選択予測ブロックの中の対応するサンプルとの間の差分に等しい値を有するような、CUの残差ブロックを生成し得る。

【0110】

20

変換処理ユニット104は、4分木区分を実行して、CUに関連付けられた残差ブロックをCUのTUに関連付けられた変換ブロックに区分し得る。したがって、TUは、ルーマ変換ブロックおよび2つのクロマ変換ブロックに対応し得る。CUのTUのルーマ変換ブロックおよびクロマ変換ブロックのサイズおよび位置は、CUのPUの予測ブロックのサイズおよび位置に基づいてもよく、基づかなくてもよい。「残差4分木」(RQT)と呼ばれる4分木構造が、領域の各々に対応するノードを含み得る。CUのTUは、RQTのリーフノードに対応し得る。

【0111】

変換処理ユニット104は、1つまたは複数の変換をTUの変換ブロックに適用することによって、CUのTUごとに変換係数ブロックを生成し得る。変換処理ユニット104は、TUに関連付けられたブロックを変換するために、様々な変換を適用し得る。たとえば、変換処理ユニット104は、離散コサイン変換(DCT)、方向変換、または概念的に類似の変換を、変換ブロックに適用し得る。いくつかの例では、変換処理ユニット104は、変換ブロックに変換を適用しない。そのような例では、変換ブロックは変換係数ブロックとして扱われてよい。

30

【0112】

量子化ユニット106は、係数ブロックの中の変換係数を量子化し得る。量子化プロセスは、変換係数の一部または全部に関連するビット深度を低減し得る。たとえば、nビットの変換係数は、量子化の間にmビットの変換係数に切り捨てられてよく、ただし、nはmよりも大きい。量子化ユニット106は、CUに関連した量子化パラメータ(QP)値に基づいて、CUのTUに関連付けられた係数ブロックを量子化し得る。ビデオエンコーダ20は、CUに関連したQP値を調整することによって、CUに関連付けられた係数ブロックに適用される量子化の程度を調整し得る。量子化が情報の損失をもたらすことがあり、したがって、量子化変換係数の精度は元の精度よりも低い場合がある。

40

【0113】

逆量子化ユニット108および逆変換処理ユニット110は、それぞれ、逆量子化および逆変換を係数ブロックに適用して、係数ブロックから残差ブロックを再構成し得る。再構成ユニット112は、予測処理ユニット100によって生成された1つまたは複数の予測ブロックからの対応するサンプルに、再構成された残差ブロックを加算して、TUに関連付けられ再構成された変換ブロックを生成し得る。このようにしてCUのTUごとに変換ブロックを再構成することによって、ビデオエンコーダ20は、CUのコーディングブロックを再構成し得る。

50

【0114】

フィルタユニット114は、1つまたは複数のデブロッキング動作を実行して、CUに関連したコーディングブロックにおけるブロッキングアーティファクトを低減し得る。フィルタユニット114が1つまたは複数のデブロッキング動作を再構成されたコーディングブロックに対して実行した後、復号ピクチャバッファ116は、再構成されたコーディングブロックを記憶し得る。インター予測処理ユニット120は、インター予測を他のピクチャのPUに対して実行するために、再構成されたコーディングブロックを含む参照ピクチャを使用し得る。加えて、イントラ予測処理ユニット126は、CUと同じピクチャの中の他のPUに対してイントラ予測を実行するために、復号ピクチャバッファ116の中の再構成されたコーディングブロックを使用し得る。

10

【0115】

エントロピー符号化ユニット118は、ビデオエンコーダ20の他の機能構成要素からデータを受信し得る。たとえば、エントロピー符号化ユニット118は、係数ブロックを量子化ユニット106から受信し得、シンタックス要素を予測処理ユニット100から受信し得る。エントロピー符号化ユニット118は、エントロピー符号化されたデータを生成するために、1つまたは複数のエントロピー符号化動作をデータに対して実行し得る。たとえば、エントロピー符号化ユニット118は、コンテキスト適応型可変長コーディング(CAVLC)動作、CABAC動作、可変長-可変長(V2V)コーディング動作、シンタックス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)動作、確率間隔区分エントロピー(PIPE)コーディング動作、指数ゴロム符号化動作、または別のタイプのエントロピー符号化動作を、データに対して実行し得る。ビデオエンコーダ20は、エントロピー符号化ユニット118によって生成された、エントロピー符号化されたデータを含むビットストリームを出力し得る。たとえば、ビットストリームは、CUに関するRQTを表すデータを含み得る。

20

【0116】

図3は、本開示の技法を実施するように構成されている例示的なビデオデコーダ30を示すブロック図である。図3は説明のために提供され、広く例示されるとともに本開示で説明されるような技法の限定でない。説明のために、本開示は、HEVCコーディングのコンテキストにおけるビデオデコーダ30を説明する。しかしながら、本開示の技法は、他のコーディング規格または方法に適用可能であり得る。

【0117】

30

図3の例では、ビデオデコーダ30は、エントロピー復号ユニット150、予測処理ユニット152、逆量子化ユニット154、逆変換処理ユニット156、再構成ユニット158、フィルタユニット160、ビデオメモリ163、および復号ピクチャバッファ(DPB)162を含む。予測処理ユニット152は、動き補償ユニット164およびイントラ予測処理ユニット166を含む。ビデオデコーダ30はまた、本開示で説明するパレットベースコーディング技法の様々な態様を実行するように構成されたパレットベース復号ユニット165を含む。他の例では、ビデオデコーダ30は、より多数の、より少数の、または異なる機能の構成要素を含んでよい。

【0118】

図4～図9を参照しながら以下でより詳細に説明するように、ビデオデコーダ30のパレットベース復号ユニット165は、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトル(たとえば、ランレングス符号化を用いて符号化されたバイナリ予測ベクトル)を受信することと、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号することと、バイナリ予測ベクトルに基づいてビデオデータの現在ブロック用のパレットを生成することとあって、バイナリ予測ベクトルが、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備えることと、パレットを使用してビデオデータの現在ブロックを復号することとを行うように構成され得る。

40

【0119】

コード化ピクチャバッファ(CPB)、たとえば、ビデオメモリ163は、ビットストリームの符号化ビデオデータ(たとえば、NALユニット)を受信および記憶し得る。エントロピー復

50

号ユニット150は、符号化ビデオデータ(たとえば、NALユニット)をCPBから受信するとともにNALユニットを構文解析して、シンタックス要素を復号し得る。エントロピー復号ユニット150は、NALユニットの中のエントロピー符号化されたシンタックス要素をエントロピー復号し得る。予測処理ユニット152、逆量子化ユニット154、逆変換処理ユニット156、再構成ユニット158、およびフィルタユニット160は、ビットストリームから抽出されたシンタックス要素に基づいて、復号ビデオデータを生成し得る。

【0120】

図3に示すように、ビデオメモリ163は、ビデオフレーム内の現在のビデオブロックを復号するために使用されるビデオデータを受信する。ビデオメモリ163は、ビデオデコーダ30の構成要素によって復号されるべきビデオデータを記憶し得る(たとえば、ビデオデータを記憶するように構成される)。ビデオメモリ163に記憶されるビデオデータは、たとえば、ビデオエンコーダ20によって生成された符号化ビデオビットストリームから取得され得る。DPB162は、ビデオデータをビデオデコーダ30によって(たとえば、イントラ予測コーディングモードまたはインター予測コーディングモードとも呼ばれる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)復号する際に使用するために、参照ビデオデータを記憶するDPBの一例である。ビデオメモリ163およびDPB162は、同期DRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、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などの、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。ビデオメモリ163およびDPB162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備えられてよい。様々な例では、ビデオメモリ163は、ビデオデコーダ30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よい。

【0121】

ビットストリームのNALユニットは、コード化スライスNALユニットを含み得る。ビットストリームを復号することの一部として、エントロピー復号ユニット150は、シンタックス要素をコード化スライスNALユニットから抽出し、エントロピー復号し得る。コード化スライスの各々は、スライスヘッダおよびスライスデータを含み得る。スライスヘッダは、スライスに関係しているシンタックス要素を含み得る。スライスヘッダの中のシンタックス要素は、スライスを含むピクチャに関連付けられたPPSを識別するシンタックス要素を含み得る。

【0122】

シンタックス要素をビットストリームから復号することに加えて、ビデオデコーダ30は、区分されていないCUに対して再構成動作を実行し得る。区分されていないCUに対して再構成動作を実行するために、ビデオデコーダ30は、CUの各TUに対して再構成動作を実行し得る。CUのTUごとに再構成動作を実行することによって、ビデオデコーダ30は、CUの残差ブロックを再構成し得る。

【0123】

CUのTUに対して再構成動作を実行することの一部として、逆量子化ユニット154は、TUに関連付けられた係数ブロックを逆量子化(inverse quantize)、すなわち、逆量子化(de-quantize)し得る。逆量子化ユニット154は、逆量子化ユニット154が適用するべき量子化の程度と、同様に逆量子化の程度とを決定するために、TUのCUに関連したQP値を使用し得る。すなわち、圧縮比、すなわち、元のシーケンスおよび圧縮されたシーケンスを表すために使用されるビット数の比が、変換係数を量子化するとき使用されるQPの値を調整することによって制御され得る。圧縮比はまた、採用されるエントロピーコーディングの方法に依存し得る。

【0124】

逆量子化ユニット154が係数ブロックを逆量子化した後、逆変換処理ユニット156は、TUに関連付けられた残差ブロックを生成するために、1つまたは複数の逆変換を係数ブロックに適用し得る。たとえば、逆変換処理ユニット156は、逆DCT、逆整数変換、逆カルーネンレーベ変換(KLT)、逆回転変換、逆方向変換、または別の逆変換を係数ブロックに適用

し得る。

【0125】

PUがイントラ予測を使用して符号化されている場合、イントラ予測処理ユニット166は、イントラ予測を実行して、PUに関する予測ブロックを生成し得る。イントラ予測処理ユニット166は、イントラ予測モードを使用して、空間的に隣接するPUの予測ブロックに基づいて、PUに関する予測ルーマブロック、予測Cbブロック、および予測Crブロックを生成し得る。イントラ予測処理ユニット166は、ビットストリームから復号された1つまたは複数のシンタックス要素に基づいて、PUに対するイントラ予測モードを決定し得る。

【0126】

予測処理ユニット152は、ビットストリームから抽出されたシンタックス要素に基づいて、第1の参照ピクチャリスト(RefPicList0)および第2の参照ピクチャリスト(RefPicList1)を構成し得る。さらに、PUがインター予測を使用して符号化されている場合、エンтроピー復号ユニット150は、PUに関する動き情報を抽出し得る。動き補償ユニット164は、PUの動き情報に基づいて、PUに関する1つまたは複数の参照領域を決定し得る。動き補償ユニット164は、PUに関する1つまたは複数の参照ブロックにおけるサンプルブロックに基づいて、PUに関する予測ブロック(たとえば、予測ルーマブロック、予測Cbブロック、および予測Crブロック)を生成し得る。

【0127】

再構成ユニット158は、CUのTUに関連付けられた変換ブロック(たとえば、ルーマ変換ブロック、Cb変換ブロック、およびCr変換ブロック)、およびCUのPUの予測ブロック(たとえば、予測ルーマブロック、予測Cbブロック、および予測Crブロック)、すなわち、適用可能なとき、イントラ予測データまたはインター予測データのいずれかを使用して、CUのコーディングブロック(たとえば、ルーマコーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)を再構成し得る。たとえば、再構成ユニット158は、予測ブロック(たとえば、予測ルーマブロック、予測Cbブロック、および予測Crブロック)の対応するサンプルに、変換ブロック(たとえば、ルーマ変換ブロック、Cb変換ブロック、およびCr変換ブロック)のサンプルを加算して、CUのコーディングブロック(たとえば、ルーマコーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)を再構成し得る。

【0128】

フィルタユニット160は、デブロック動作を実行して、CUのコーディングブロック(たとえば、ルーマコーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)に関連したブロックングアーティファクトを低減し得る。ビデオデコーダ30は、CUのコーディングブロック(たとえば、ルーマコーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)を復号ピクチャバッファ162に記憶し得る。復号ピクチャバッファ162は、その後の動き補償、イントラ予測、および図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での提示のために、参照ピクチャを提供し得る。たとえば、ビデオデコーダ30は、復号ピクチャバッファ162の中のブロック(たとえば、ルーマブロック、Cbブロック、およびCrブロック)に基づいて、他のCUのPUに対してイントラ予測動作またはインター予測動作を実行し得る。

【0129】

図4は、ビデオエンコーダ20のパレットベース符号化ユニット122をより詳細に示すブロック図である。パレットベース符号化ユニット122は、バイナリ予測ベクトルを符号化するための本開示の例示的な技法のうちの1つまたは複数を実行するように構成され得る。

【0130】

上述されたように、パレットベース符号化ユニット122は、ビデオデータのブロック(たとえば、CUまたはPU)を、パレットベース符号化モードを用いて符号化するように構成され得る。パレットベース符号化モードでは、パレットは、インデックスによって番号付けされるとともに画素値を示すために使用され得る色成分値(たとえば、RGB、YUVなど)または強度を表すエントリを含み得る。パレット生成ユニット203は、ビデオデータの現在ブ

10

20

30

40

50

ロックに関する画素値212を受信し、ビデオデータの現在ブロックに関するカラー値のパレットを生成するように構成され得る。パレット生成ユニット203は、上述のヒストグラムベースの技法を含む、ビデオデータの現在ブロック用のパレットを生成するための任意の技法を使用してよい。パレット生成ユニット203は、任意のサイズのパレットを生成するように構成され得る。一例では、パレット生成ユニット203は、32個のパレットエントリを生成するように構成され得、その場合、各パレットエントリは、画素のY、Cr、およびCb成分に関する画素値を含む。前の例では、各パレットエントリがサンプル(画素)のすべての色成分に関する値を規定することが仮定される。しかしながら、本明細書で説明される概念は、色成分ごとに別個のパレットを使用することに適用可能である。

【0131】

パレットがパレット生成ユニット203によって生成されると、マップユニット204は、ビデオデータの現在ブロックに関して、ビデオデータの現在ブロックの中の特定の画素が、パレット生成ユニット203によって生成されたパレットの中のエントリによって表され得るか否かを示すマップを生成し得る。マップユニット204は、各画素がパレットからのエントリをどのように使用するか(または、使用しないこと)を示すシンタックス要素を含むマップ214を生成し得る。ビデオデータの現在ブロックの中の画素に関する値が、パレットの中に見出されない、したがって、パレットの中へのインデックスによって表されない場合、マップユニット204は、その特定の画素に関する画素値を明示的に送信し得る。いくつかの例では、マップユニット204は、明示的な画素値をパレットの中のエントリのうちの1つから予測し得る。いくつかの他の例では、マップユニット204は、画素を量子化し得、量子化された値を送信し得る。

【0132】

ブロックの中の画素の各々に対して使用されるカラー値を示すシンタックス要素をシグナリングすることに加えて、パレットベース符号化ユニット122はまた、ビデオデータの現在ブロックに対して使用されるべきパレットをシグナリングするように構成され得る。本開示の技法によれば、パレットベース符号化ユニット122は、ビデオデータの特定のブロック用のパレットの値を示すためにシグナリングされるデータの量を低減するために、パレット予測技法を採用するように構成され得る。

【0133】

パレット予測の一例として、2014年6月20日現在、http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0094-v1.zipから入手可能であるJCTVC-Q0094に記載されるように、パレットは、予測子パレットからコピーされるエントリを含んでよい。予測子パレットは、パレットモードを使用する、前にコーディングされたブロックからの、または他の再構成サンプルからのパレットエントリを含み得る。図4に示すように、パレットベース符号化ユニット122は、予測子パレットバッファ210を含み得る。予測子パレットバッファ210は、前に符号化されたブロックからの、前に使用されたいくつかのパレットエントリを記憶するように構成され得る。一例として、予測子パレットバッファ210は、所定のサイズの先入れ先出し(FIFO)バッファとして構成され得る。予測子パレットバッファ210は、任意のサイズであってよい。一例では、予測子パレットバッファ210は、64個までの、前に使用されたパレットエントリを含む。

【0134】

いくつかの例では、パレットベース符号化ユニット122は、予測子パレットバッファ210の中のすべてのパレットエントリが一意となるように、予測子パレットバッファ210の中のエントリを取り除くように構成され得る。すなわち、予測子パレットバッファ210に追加されるべき新しいパレットエントリごとに、パレットベース符号化ユニット122は、最初に、予測子パレットバッファ210にすでに記憶されている他の同一のエントリがないことを検査するように構成され得る。同一のエントリがない場合、新しいパレットエントリが予測子パレットバッファ210に追加される。新しいエントリが既存のエントリと全く同じものである場合、新しいパレットエントリが予測子パレットバッファ210に追加され、全く同じエントリは予測子パレットバッファ210から除去される。

【 0 1 3 5 】

予測子パレットバッファ210の中のパレットエントリがビデオデータの現在ブロック用のパレットの中のエントリのうちの1つに関してコピー(または、再使用)されているかどうかを示すために(たとえば、フラグ=1によって示される)、パレットベース符号化ユニット122は、パレット生成ユニット203によって生成されたビデオデータの現在ブロック用のパレットの中のエントリごとにバイナリフラグを生成およびシグナリングするように構成された、バイナリ予測ベクトル生成ユニット206を含み得る。すなわち、バイナリ予測ベクトルの中の1としての値を有するフラグは、予測子パレットバッファ210の中の対応するエントリが現在ブロック用のパレットに対して再使用されることを示し、バイナリ予測ベクトルの中の0としての値を有するフラグは、予測子パレットバッファ210の中の対応するエントリが現在ブロック用のパレットに対して再使用されないことを示す。さらに、パレットベース符号化ユニット122は、予測子パレットバッファ210の中のエントリからコピーされ得ない現在パレットに関するいくつかの値を明示的にシグナリングするように構成され得る。新しいエントリのが、同様にシグナリングされ得る。

10

【 0 1 3 6 】

2014年3月25日に出願された米国仮出願第61/970,257号、2014年4月17日に出願された米国仮出願第61/981,105号、および2014年5月23日に出願された米国仮出願第62/002,668号では、2分木ベースのシグナリング方法および末尾位置ベースのシグナリング方法が、パレットバイナリ予測ベクトルのコーディングのために提案された。2014年5月23日に出願された米国仮出願第62/002,741号では、グループベースのシグナリング方法が提案された。本開示は、バイナリ予測ベクトルを生成、符号化、および復号するための追加の技法を提案する。

20

【 0 1 3 7 】

本明細書で説明するいくつかの例は、コーディング効率を改善するようにパレット予測ベクトルをコーディングするための方法に関する。たとえば、バイナリ予測ベクトル生成ユニット206によって生成されるバイナリ予測ベクトルが、

$$b=[b_0, b_1, \dots, b_{N-1}], N \geq 0, b_i \in \{0,1\}, 0 \leq i < N$$

によって示されると仮定する。

上の式において、 $b_i \in \{0,1\}, 0 \leq i < N$ は、予測フラグ(バイナリフラグまたはバイナリ予測フラグとも呼ばれる)を示す。 $N=0$ の場合、 $b=$ であり(すなわち、 b は空のベクトルである)、それはシグナリングされる必要がない。したがって、以下の説明では、 $N>0$ であると仮定してよい。

30

【 0 1 3 8 】

図5は、予測子パレットバッファ210および現在パレット220の一例を示す。図5に見られ得るように、現在パレット220は、予測子パレットバッファ210からの、エントリインデックス1、2、5、および9に関連付けられた画素値を再使用する。したがって、図4のバイナリ予測ベクトル生成ユニット206によって生成されるバイナリ予測ベクトルは、 $b=[110010001000]$ であることになる。この例に見られ得るように、バイナリ予測ベクトル b は、予測子パレットバッファ210の中の第1、第2、第5、および第9のインデックスに対応する1としての値を有するフラグを含む。すなわち、予測子パレットバッファ210の中の第1、第2、第5、および第9のエントリだけが、現在パレット220に対して再使用されるエントリである。現在パレット220の中のエントリインデックス5~8に対して、パレットベース符号化ユニット122は、パレットエントリ値を符号化ビデオビットストリームの中でシグナリングするように構成され得る(たとえば、明示的なシグナリングまたは別の予測技法を使用して)。

40

【 0 1 3 9 】

本開示の1つまたは複数の技法によれば、ビデオエンコーダ20は、パレットを符号化ビデオビットストリームの中でシグナリングするのに必要とされるデータの量を低減するために、バイナリ予測ベクトル b を符号化または概して符号化するように構成され得る。図4に示すように、バイナリ予測ベクトル圧縮ユニット209は、符号化バイナリ予測ベクトル

50

ル215を生成およびシグナリングするように構成され得る。しかしながら、本開示のバイナリ予測ベクトル圧縮技法が、エントロピー符号化ユニット118を含むビデオエンコーダ20の他の構造で実施され得ることを理解されたい。

【0140】

本開示の一例では、バイナリ予測ベクトル圧縮ユニット209は、ランレングスペースの符号化技法を使用してバイナリ予測ベクトルを符号化するように構成され得る。たとえば、バイナリ予測ベクトル圧縮ユニット209は、バイナリ予測ベクトルの中の「1」の間の連続した「0」の数を指数ゴロムコードを使用してシグナリングすることによって、バイナリ予測ベクトルを符号化するように構成され得る。一例として、再び $b=[110010001000]$ であると仮定する。この例では、図6に示すように、バイナリ予測ベクトル(すなわち、 b)は、「0個の連続した0」-「1」-「0個の連続した0」-「1」-「2個の連続した0」-「1」-「3個の連続した0」-「1」-および「4個の連続した0」として表現され得る。 $b_i \in \{0,1\}$ であることが知られているので、最後の「連続した0」グループを除いて、各「連続した0」グループに「1」が後続しなければならない。したがって、バイナリ予測ベクトル圧縮ユニット209は、ランレングスシーケンス「0-0-2-3-4」として表現され得る「0個の連続した0」-「0個の連続した0」-「2個の連続した0」-「3個の連続した0」-「4個の連続した0」としてバイナリ予測ベクトル b を表すために、0ベースのランレングスコーディング技法を使用し得る。

【0141】

ランレングスペースのシグナリングに関する本開示の1つまたは複数の例によれば、ランレングスシーケンスをコーディングするために、ゴロムライスコード、任意の次数の指数ゴロムコード、短縮指数ゴロムコード、短縮ライスコード、または短縮された二値化を含む他の二値化が使用され得る。一例では、バイナリ予測ベクトル圧縮ユニット209は、ランレングスコーディング技法として0次の指数ゴロムコードを使用する。

【0142】

短縮された二値化の場合、バイナリベクトルの末尾に移動させると、可能な最大ラン値はベクトル内の位置に応じてベクトルサイズから0に低減されるので、バイナリベクトルの中での「1」の位置およびバイナリベクトルサイズに応じて、最大シンボルはランの可能な最大値であり得る。たとえば、最大シンボルは、バイナリベクトル長、またはランがそこから計数される「1」の位置をバイナリベクトル長から引いたものであり得る。言い換えれば、それはバイナリベクトルの末尾から測定される残りの長さである。特定のサイズとしてのバイナリベクトル b を伴う上述の例の場合、たとえば、13、ランレングスシーケンス「0-0-2-3-4」は、短縮された二値化「0[13]-0[12]-2[11]-3[8]-4[4]」を用いてコーディングされ得、その場合、最大シンボルは大括弧の中に示される。

【0143】

同様に、いくつかの例では、二値化は、要素(0または1)のバイナリベクトルの中での位置またはインデックスに依存し得る。特定の例として、位置が何らかのしきい値よりも小さい場合、あるタイプの二値化が使用され、そうでない場合、別のタイプの二値化が適用される。いくつかの例では、二値化タイプは、異なる二値化コードであってよく、または指数ゴロムコードのような同じコードファミリーであるが異なる次数を有してもよい。

【0144】

一例では、しきい値は、前のブロックまたは前のパレットコード化ブロックからのパレット長であってよい。別の例では、しきい値は、何らかのデフォルト値に固定されてよく、またはブロック、スライス、ピクチャ、または他の場所ごとにシグナリングされてもよい。ラン値をコーディングするためのCABACコンテキストを規定するために、対応する技法が随意に使用され得ることを認識されたい。さらに、パレットベース符号化ユニット122は、シグナリングされる「1」要素の数(すなわち、現在パレット220に対して再使用されるものとして示される予測子パレットバッファ210からのパレットエントリの数)が、可能な最大数に到達したとき、ランレングスシグナリングを停止するように構成され得る。いくつかの例では、可能な最大数は、可能な最大パレットサイズである。

【0145】

本開示のいくつかの例は、バイナリ予測ベクトルbを示すランレングスシーケンスの末尾位置コーディングに関する。本開示の1つまたは複数の例では、バイナリ予測ベクトル圧縮ユニット209は、バイナリ予測ベクトルの末尾位置をコーディングするために、予約済みランレングスLを使用してバイナリ予測ベクトルbを符号化するように構成され得る。一例では、予約済みランレングスとしてL=1が使用される。ビデオエンコーダ20において、ランレングスがL以上である場合、バイナリ予測ベクトル圧縮ユニット209は、ランレングスに1を加算するように構成される。実際のランレングスがLよりも短い場合、バイナリ予測ベクトル圧縮ユニット209は、そのままのランレングスをシグナリングするように構成される。バイナリ予測ベクトル圧縮ユニット209は、予約済みランレングスLを有する末尾位置ランレングスをシグナリングし得る。

10

【0146】

同様に、ビデオデコーダ30において、ランレングスとしての復号された値がLよりも大きい場合、実際のランレングスから1が減算される。復号された値またはランレングスがLよりも小さい場合、復号された値が実際のランレングスとして使用される。復号された値がLに等しい場合、バイナリ予測ベクトルbの中の残りの位置はすべて0である。したがって、復号された値がLに等しい場合、これ以上のランシグナリングは必要でない。

【0147】

上記と同じ例(すなわち、 $b=[110010001000]$)を使用し、 $L=1$ と仮定すると、バイナリ予測ベクトル圧縮ユニット209は、図6のランレングスシーケンス「0-0-2-3-4」を「0-0-3-4-1」としてシグナリングするように構成される。次いで、上の規則を適用すると、ビデオデコーダ30は、ランレングスシーケンスを「0-0-2-3-末尾」として復元するように構成され得る。すなわち、0ランレングスシーケンスの両方が $L=1$ としての予約済みランレングス値よりも小さいので、0としての最初のランレングス値は0として復号され、0としての次のランレングスシーケンスは0として復号される。次のランレングスシーケンスは3であり、したがって、受信された3としての値が $L=1$ としての予約済みランレングス値よりも大きいので、ビデオデコーダ30は、3としての値から1を減算して2を取得するように構成されることになる。同様に、受信された4としての値が $L=1$ としての予約済みランレングス値よりも大きいので、ビデオデコーダ30は、次のランレングスシーケンスに関して、受信された4としての値から1を減算して3を取得するように構成されることになる。最終的に、最後に受信されたランレングス値1は、 $L=1$ としての予約済みランレングス値に等しい。したがって、ビデオデコーダ30はこれ以上「1」としての値がバイナリ予測ベクトルの中に存在しないことを決定し得る。

20

30

【0148】

本開示の別の例では、バイナリ予測ベクトル圧縮ユニット209は、バイナリ予測ベクトルの中の「1」の総数(すなわち、予測子パレットバッファ210から再使用されるパレットエントリの表示の数)が可能な最大パレットサイズよりも小さいときにのみ、末尾位置コーディングを適用するように構成され得る。バイナリ予測ベクトルの中の「1」の総数が可能な最大パレットサイズに等しい場合、バイナリ予測ベクトル圧縮ユニット209は、最後のランレングスのシグナリングをバイパスするように構成され得る。

40

【0149】

上述の例では、可能な最大パレットサイズが4であるとビデオデコーダ30が決定する場合、「0-0-2-3」としてのランレングスシーケンス(または、上述された予約済みランレングス規則によるランレングスシーケンス0-0-3-4)は、バイナリ予測ベクトルbを復元するのに十分であり得る。いくつかの例では、可能な最大パレットサイズは、あらかじめ決定されてよい。したがって、可能な最大パレットサイズは、あらかじめビデオエンコーダ20およびビデオデコーダ30によって決定されてよい。他の例では、可能な最大パレットサイズは、ビデオエンコーダ20によって符号化ビデオビットストリームの中でシグナリングされてよい。可能な最大パレットサイズをシグナリングすることは、任意の適当な従来のデータ通信技法、または本明細書で説明する任意の適当なデータ通信技法を使用して実行さ

50

れ得る。

【0150】

本開示の別の例では、バイナリ予測ベクトル圧縮ユニット209は、ランレングスシーケンスの中の最後のランレングスが0でないときにのみ、上述された末尾位置コーディング技法を適用するように構成され得る。ランレングスシーケンスの中の最後のランレングスが0である場合、末尾位置符号化はバイパスされてよい。この記述は、バイナリ予測ベクトルbの中の最後のフラグが1に等しいときと同等である。この状況では、末尾位置コーディングはバイパスされてよい。たとえば、b=[100000000001]の場合、ランレングスシーケンスは「0-10-0」である。「0-10」としてのランレングスシーケンス(または、上述された予約済みランレングス規則に従ってシグナリングされるシーケンスの値0-11)は、バイナリ予測ベクトルbを復元するのに十分であり得る。さらに、バイナリ予測ベクトルbがゼロベクトルであることを示すためのいかなるランもシグナリングすることなく、末尾位置がシグナリングされてよく、これは予測されているパレットエントリがないことを意味する。

10

【0151】

いくつかの例では、固定値Lを使用して残りのバイナリフラグが0であることをシグナリングする代わりに、Lとしての最小値およびバイナリ予測子ベクトルのサイズが、末尾位置を示すために使用され得る。これは、ランの値がバイナリ予測子ベクトルのサイズよりも常に小さいからである。

【0152】

20

図7は、ビデオデコーダ30のパレットベース復号ユニット165の一例を示すブロック図である。パレットベース復号ユニット165は、図4のパレットベース符号化ユニット122と相反の方式を実行するように構成され得る。パレットベース復号ユニット165は、現在ブロックの中の画素ごとに、パレットに関するエントリが現在ブロックの中の画素のために使用されるか否かを示すマップ312を受信するように構成され得る。加えて、マップ312は、どのパレットエントリが所与の画素に対して使用されるべきであるのかをさらに示し得る。マップユニット302は、マップ312およびパレット生成ユニット304によって生成されたパレットを使用してビデオデータの現在ブロックを復号して、復号ビデオデータ314を生成し得る。

【0153】

30

本開示の技法によれば、パレットベース復号ユニット165はまた、符号化バイナリ予測ベクトル316を受信し得る。上記で説明したように、バイナリ予測ベクトル316は、バイナリ予測ベクトルの中の0値のランを示すランレングスシーケンスを符号化する、ランレングスコーディング技法を使用して符号化され得る。バイナリ予測ベクトル解凍ユニット306は、図4～図6を参照しながら上記で説明したランレングスコーディング技法の任意の組合せを使用して、符号化バイナリ予測ベクトルを復号するように構成され得る。バイナリ予測ベクトルがバイナリ予測ベクトル解凍ユニット306によって復元されると、パレット生成ユニット304は、バイナリ予測ベクトルおよび予測子パレットバッファ310に記憶されている、前に使用されたパレットエントリに基づいて、ビデオデータの現在ブロック用のパレットを生成し得る。パレットベース復号ユニット165は、パレットベース符号化ユニット122が、前に使用されたパレットエントリを予測子パレットバッファ210に記憶したのと同じ方式で、前に使用されたパレットエントリを予測子パレットバッファ310に記憶するように構成され得る。

40

【0154】

図8は、本開示の例示的な符号化方法を示すフローチャートである。図8の技法は、パレットベース符号化ユニット122を含む、ビデオエンコーダ20の1つまたは複数の構造構成要素によって実施され得る。

本開示の一例では、パレットベース符号化ユニット122は、ビデオデータの現在ブロック用のパレットを生成し(800)、ビデオデータの現在ブロック用のパレットに関するバイナリ予測ベクトルを生成する(802)ように構成され得る。バイナリ予測ベクトルは、前に使

50

用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されるか否かを示すエントリを備える。パレットベース符号化ユニット122は、ランレングス符号化技法を使用してバイナリ予測ベクトルを符号化し(804)、パレットを使用してビデオデータの現在ブロックを符号化する(806)ようにさらに構成され得る。一例では、ランレングス符号化技法は、0のランレングスをコーディングすることを備える。

【0155】

本開示の一例では、パレットベース符号化ユニット122は、指数ゴロム符号化技法を使用してバイナリ予測ベクトルを符号化するように構成され得る。一例では、指数ゴロム符号化技法は、0次の指数ゴロム符号化技法である。

【0156】

本開示の別の例では、パレットベース符号化ユニット122は、ランレングス符号化技法および予約済みランレングス値Lを使用してバイナリ予測ベクトルを符号化するように構成され得、予約済みランレングス値Lは、バイナリ予測ベクトルの末尾位置を示す。一例では、予約済みランレングス値Lは1である。

【0157】

本開示の別の例では、パレットベース符号化ユニット122は、ランレングス符号化技法、予約済みランレングス値L、および最大パレットサイズを使用して、バイナリ予測ベクトルを符号化するように構成され得る。予約済みランレングス値Lは、バイナリ予測ベクトルの末尾位置を示す。この例では、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されることを示す、バイナリ予測ベクトルの中のエントリの総数が最大パレットサイズに等しい場合、予約済みランレングス値Lは使用されない。

【0158】

本開示の別の例では、パレットベース符号化ユニット122は、ランレングス符号化技法および予約済みランレングス値Lを使用してバイナリ予測ベクトルを符号化するように構成され得、予約済みランレングス値Lは、バイナリ予測ベクトルの末尾位置を示す。この例では、符号化バイナリ予測ベクトルの中の最後のランレングスが0のランレングスを示さない場合、予約済みランレングス値Lは使用されない。

【0159】

本開示の別の例では、パレットベース符号化ユニット122は、バッファの中のビデオデータの、前に符号化された1つまたは複数のブロックに対して前に使用されたパレットエントリを記憶するように構成され得る。パレットベース符号化ユニット122は、バッファに記憶されている、前に使用されたパレットエントリの全く同じエントリを除去するようにさらに構成され得る。本開示の別の例では、ビデオデータの、前に符号化された1つまたは複数のブロックに対して前に使用されたパレットエントリは、ビデオデータの現在ブロックの上の画素のラインに対して前に使用されたパレットエントリ、およびビデオデータの現在ブロックの左の画素のラインに対して前に使用されたパレットエントリを備える。

【0160】

図9は、本開示の例示的な復号方法を示すフローチャートである。図9の技法は、パレットベース復号ユニット165を含む、ビデオデコード30の1つまたは複数の構造構成要素によって実施され得る。

【0161】

本開示の一例では、パレットベース復号ユニット165は、ビデオデータの現在ブロックに関する符号化バイナリ予測ベクトルを受信し(900)、ランレングス復号技法を使用して符号化バイナリ予測ベクトルを復号する(902)ように構成され得る。一例では、ランレングス復号技法は、0のランレングスをコーディングすることを備える。パレットベース復号ユニット165は、バイナリ予測ベクトルに基づいて、ビデオデータの現在ブロック用のパレットを生成する(904)ようにさらに構成され得る。バイナリ予測ベクトルは、前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用さ

10

20

30

40

50

れるか否かを示すエントリを備える。パレットベース復号ユニット165は、パレットを使用して、ビデオデータの現在ブロックを復号する(906)ようにさらに構成され得る。

【0162】

本開示の別の例では、パレットベース復号ユニット165は、指数ゴロム復号技法を使用して、符号化バイナリ予測ベクトルを復号するようにさらに構成され得る。一例では、指数ゴロム復号技法は、0次の指数ゴロム復号技法である。

【0163】

本開示の別の例では、パレットベース復号ユニット165は、ランレンクス復号技法および予約済みランレンクス値Lを使用して符号化バイナリ予測ベクトルを復号するようにさらに構成され得、予約済みランレンクス値Lは、バイナリ予測ベクトルの末尾位置を示す。一例では、予約済みランレンクス値Lは1である。

10

【0164】

本開示の別の例では、パレットベース復号ユニット165は、ランレンクス復号技法、予約済みランレンクス値L、および最大パレットサイズを使用して、符号化バイナリ予測ベクトルを復号するようにさらに構成され得る。予約済みランレンクス値Lは、バイナリ予測ベクトルの末尾位置を示す。前に使用されたパレットエントリがビデオデータの現在ブロック用のパレットに対して再使用されることを示す、バイナリ予測ベクトルの中のエントリの総数が最大パレットサイズに等しい場合、予約済みランレンクス値Lは使用されない。

【0165】

20

本開示の別の例では、パレットベース復号ユニット165は、ランレンクス復号技法および予約済みランレンクス値Lを使用して符号化バイナリ予測ベクトルを復号するようにさらに構成され得、予約済みランレンクス値Lは、バイナリ予測ベクトルの末尾位置を示す。符号化バイナリ予測ベクトルの中の最後のランレンクスが0のランレンクスを示さない場合、予約済みランレンクス値Lは使用されない。

【0166】

本開示の別の例では、パレットベース復号ユニット165は、バイナリ予測ベクトルによってパレットに対して再使用されるものとして示されるパレットへの、前に使用されたパレットエントリをバッファからコピーし、パレットに対してコピーされた前に使用されたパレットエントリの数が最大パレットサイズよりも少ない場合、追加のパレットエントリを受信するようにさらに構成され得る。

30

【0167】

本開示の別の例では、パレットベース復号ユニット165は、バッファの中のビデオデータの、前に復号された1つまたは複数のブロックに対して前に使用されたパレットエントリを記憶するようにさらに構成され得る。パレットベース復号ユニット165は、バッファに記憶されている、前に使用されたパレットエントリの全く同じエントリを除去するようにさらに構成され得る。一例では、ビデオデータの、前に復号された1つまたは複数のブロックに対して前に使用されたパレットエントリは、ビデオデータの現在ブロックの上の画素のラインに対して前に使用されたパレットエントリ、およびビデオデータの現在ブロックの左の画素のラインに対して前に使用されたパレットエントリを備える。

40

【0168】

以下のセクションは、本開示のさらなる例示的な技法を説明する。第1の例では、ビデオエンコーダ20は、最初に、ゴロムライスコード、指数ゴロムコード、もしくは短縮ライスコード、またはそれらの組合せを使用して、1の総数(「num-of-one」)、すなわち、予測子パレットバッファ210から再使用されるパレットエントリの数をバイナリ予測ベクトルbの中でシグナリングするように構成され得る。次いで、復号された非ゼロのバイナリ予測フラグの数がシグナリングされた「num-of-one」に到達すると、ビデオデコーダ30は、バイナリ予測ベクトルbの中の残りのフラグが0であり、これらのフラグのシグナリングがビデオエンコーダ20によってバイパスされたことを決定し得る。言い換えれば、この第1の例では、上記で説明した末尾位置コーディング手順がバイパスされ得る。この場合、

50

実際のランレングスがシグナリングされる。そのような例では、ランレングス値の上向きの調整は必要でない。

【0169】

第2の例では、上述されたように、バイナリ予測ベクトルの中の0値のランレングスだけがコーディングされる。代替技法として、ランレングスコーディングは、バイナリ予測ベクトルの中の0値のランレングスおよび/またはバイナリ予測ベクトルの中の1値のランレングスの両方を含む(たとえば、許容する)ことができる。初期ランは、「1のランレングス」(または、「0のランレングス」)であるように事前定義され得る。次いで、ビデオエンコーダ20は、0のランおよび1のランのコーディングを交互にし得る。

【0170】

たとえば、上記と同じ例(すなわち、 $b=[110010001000]$)を使用し、最初のランが「1のランレングス」であると仮定すると、 $b=[1100100010000]$ は、

「ラン=2」-「ラン=2」-「ラン=1」-「ラン=3」-「ラン=1」-「ラン=4」

として表現され得る。

しかしながら、「0のラン」に「1のラン」が後続しなければならないこと、また逆も同様であることが知られており、したがって、上の表現は、「210203」としてのランレングスシーケンスとして表現され得る、

「ラン=2」-「ラン=1」-「ラン=0」-「ラン=2」-「ラン=0」-「ラン=3」

として簡略化され得る。

ゴロムライスコーディング、指数ゴロムコーディング、または短縮ライスコーディングが、このシーケンスをコーディングするために使用され得る。「0のラン」および「1のラン」は、ゴロムライスコード、指数ゴロムコード、または短縮ライスコードの異なるパラメータを有し得る。通常のピンがそれらを符号化するために使用される場合、「0のラン」および「1のラン」に対して異なるコンテキストが使用されることがある。この例では、本開示の他の場所で説明される末尾位置コーディング技法も適用され得る。

【0171】

さらに、ビデオエンコーダ20は、シグナリングされる「1」要素の数が、可能な最大数に到達したとき、ランレングスシグナリングを停止するように構成され得、可能な最大数は、可能な最大パレットサイズであってよい。

【0172】

第3の例では、ビデオデコーダ30が、最初のランが「0のラン」であるのか、それとも「1のラン」であるのかを、復号された b_0 値に基づいて決定し得るように、 b_0 、すなわち、バイナリ予測ベクトル b の中の最初の要素が直接シグナリングされ得る。

【0173】

第4の例では、ビデオエンコーダ20は、第1のコーディング技法を使用して、最初の M 個のフラグ、すなわち、バイナリ予測ベクトルの中の $[b_0, b_1, \dots, b_{M-1}]$ を符号化するように構成され得、残りのもの、すなわち、 $[b_M, b_{M+1}, \dots, b_{N-1}]$ は第2のコーディング技法を使用してコーディングされる。第2のコーディング技法は、上で規定されたランレングスベースの方法である。一例では、 $M=4$ である。第1のコーディング技法は、任意の適当な従来のコーディング技法であってよい。いくつかの例では、第1のコーディング技法は、非ランレングスコーディングベースの方法であってよい。他の例では、第1および第2のコーディング技法はともにランレングスベースのコーディング技法であってよいが、第1のコーディング技法および第2のコーディング技法が互いに異なってもよい。

【0174】

上述のような両方のバイナリ要素に対するランレングスベースのシグナリングの一例では、ビデオエンコーダ20は、バイナリ予測ベクトル b に関する「0」または「1」についてのランレングス m をシグナリングするように構成され得る。ただし、両方のバイナリ要素に対してランレングスがシグナリングされる場合、それら2つが組み合わせられてよい。このことは、以下のステップによって達成され得る。すなわち、(1)先頭において、ピン値が選択され、たとえば、ピン=0とし、(2)ピン要素に関するランレングスをシグナリング

10

20

30

40

50

し、(3)ピンが1に等しい場合、1に等しいピンがバイナリベクトルの中の最後の「1」要素であるかどうかを示すラストフラグをシグナリングする。たとえば、1に等しいラストはそれが最後の「1」であることを示し、0に等しければそれ以外である。ラストが1に等しい場合、バイナリベクトル要素をシグナリングすることを停止するとともに終了し、(4)ピン値がスワップされ、すなわち、ピン=1-ピンとし、上のステップ(2)へ進む。

【0175】

最初のピン値が0であるものとして選択される場合、バイナリベクトルが0である場合を示す必要があり得る。本開示は、バイナリベクトルが0であるかどうかを示すフラグを、zero_flagと呼ぶことがある。たとえば、ランレングスシグナリングの先頭における別個のフラグが使用され得る。別個のフラグは1に等しくてよく、これはバイナリベクトルが0であることを示す。この場合、zero_flagの後にランレングスシグナリングはない。zero_flagが0の場合、ランレングス値がシグナリングされる。最初のピンが1に等しいものとして選ばれる場合、1に等しいピンがシグナリングの末尾を示すlast_flagを有するので、このzero_flagをシグナリングする必要がない場合がある。

【0176】

一例では、上述の基本アルゴリズムは、以下のリストの中の技法のうちの1つまたは複数を使用してさらに改良され得る。リストの中の各アイテムは、このリストに続く以下の説明の中で、「改良型アイテム」と呼ばれることがある。

1. ピン選択は0もしくは1に固定されてよく、または適応的であってもよい(すなわち、様々なブロックに対して異なってよい)。たとえば、ピンは、前にコーディングされたブロックの中でどの要素が最多であるかまたはどの最初の要素が最多であるかを識別することによって、前のブロックに基づいて選ばれてよい(すなわち、選択されてよい)。あるいは、ピンは、すべてのブロックに対してシグナリングされてよく、またはブロックのグループに対して(たとえば、タイルごとに、またはスライスヘッダの中で)シグナリングされてもよい。あるいは、ピンは、ビデオシーケンスの中のピクチャごとに、たとえば、ピクチャパラメータセット(PPS)、シーケンスパラメータセット(SPS)、ビデオパラメータセット(VPS)、または別の場所のうちの少なくとも1つの中でシグナリングされてよい。

2. ラストフラグがシグナリングされるべきバイナリベクトル要素「1」が、バイナリベクトルbの中の最後の要素である場合、このラストフラグはシグナリングされなくてよい。

3. バイナリベクトルの中の最初の要素を処理した後、要素が最後の要素でない場合、少なくとも1つの別のバイナリ要素が後続することが知られているので、シグナリングされるランレングス値は実際のランレングス値-1であってよい。

4. バイナリベクトルの中の最後の要素の前に「0」ランレングス値が停止される場合、この要素が「1」であるとともにそれがベクトルの中の最後の「1」要素であることが知られているので、バイナリベクトルの中の最後の「1」要素に対するランレングス値に「0」要素が先行する場合、バイナリベクトルの中の最後の「1」要素に対するランレングス値はシグナリングされなくてよい。

5. あるいは、zero_flagシグナリングまで、バイナリベクトルの中の最後の「1」要素に対するランレングス値が、最初のランレングス値の中に含まれてよく、その場合、これに対して特定の値が割り当てられる。たとえば、最初のランレングス値が0である場合、バイナリベクトルは0であり、その後、他のシグナリングは行われない。非ゼロのバイナリベクトルの場合、あいまいさを回避するために、最初にシグナリングされるランレングス値は、その特定の値、この例では0の後、1だけ増大される必要がある。

6. zero_flagおよびlast_flagは、コンテキスト適応型バイナリ算術コーディング(CABAC)コンテキストコーディングされてよく、その場合、たとえば、別個のコンテキストがフラグごとに割り当てられてよい。コンテキストは、ブロックサイズ、スライスタイプなどに依存し得る。

7. ランレングス値は、単項コード、ゴロムライスコードもしくは指数ゴロム、またはランレングス値がバイナリベクトルサイズよりも大きくなり得ないことを考慮に入れるコー

10

20

30

40

50

ドの短縮バージョンなどの、様々な二値化方法を使用してコーディングされてよい。

8. シグナリングされる「1」要素の数が、可能な最大数に到達したとき、ランレングスシグナリングは停止されてよく、可能な最大数は、可能な最大パレットサイズを表し得る。

【0177】

バイナリ予測ベクトル $b=\{11001110100001\}$ を用いた例の場合、上述された基本アルゴリズムは、 $[0]0-2(0)-2-3(0)-1-1(0)-4-1(1)$ を生成する。先頭において、ピンが0に設定され、最初のランレングスがそのピンに対してシグナリングされ、この例ではそれは0であり、次いで、すべての「-」においてピンが1-ピンにスワップされる。シグナリングされるラストフラグは、中括弧の中に示される。大括弧の中はzero_flagである。一例として、以下のもの、すなわち $1-1(0)-1-2(0)-0-0(0)-3-0(1)$ は、改良型アルゴリズムの結果である。

【0178】

上のアイテム#5に起因してランレングス値1を反映する最初の要素処理の後、他のランレングス値は上の改良型アイテム#3において述べたように1だけデクリメントされる。取り消し線は、上の#2において述べたベクトルの中の最後の「1」に対するラストフラグのシグナリングが行われないこと、および上の#4において提案された最後の「1」要素に対するランレングス値のシグナリングが行われないことを示す。

【0179】

別の例として、 $b=\{11001110100011\}$ および例示的な改良型アルゴリズムは以下のもの、すなわち、 $1-1(0)-1-2(0)-0-0(0)-3-1(1)$ をもたらす。

【0180】

いくつかの例示的な方法は、パレットモードインデックスブロックコーディングにおける上方コピーラン、左側コピーラン、またはその両方の長さをコーディングするために、指数ゴロムコードを使用し得る。たとえば、ランレングスをコーディングするために指数ゴロムコードを使用する方法はまた、パレットモードインデックスブロックコーディングにおける上方コピーラン、左側コピーラン、またはその両方の長さをコーディングするために、指数ゴロムコードを適用し得る。別の例では、「0よりも大きい」フラグ、「1よりも大きい」フラグ、および「2よりも大きい」フラグをシグナリングした後、残りの値(たとえば、 $(\text{ランレングス}-3) \geq 0$ の場合、たとえば、 $(\text{ランレングス}-3)$)をコーディングするために2次のゴロムライスコードが使用され得る。

【0181】

いくつかの例では、「0よりも大きい」フラグ、「1よりも大きい」フラグ、および「2よりも大きい」フラグは、それぞれ、ランレングスが「0よりも大きいこと」、「1よりも大きいこと」、または「2よりも大きいこと」を示すために使用され得る。ランレングスは、ランレングスコード化バイナリ予測ベクトルを使用してコーディングされ得る。

【0182】

本開示のいくつかの追加の例によれば、ビデオコーダ(たとえば、ビデオエンコーダ20またはビデオデコーダ30)は、1つまたは複数の新しいパレット予測子を決定し得る。本開示のこれらの追加の例によれば、ビデオコーダは、パレット予測子をシグナリングするための新しいブルベクトルを決定し得る。これらおよび他の技法が、たとえば、以下の図10に関して、より詳細に説明される。

【0183】

図10は、本開示の技法に一致する、ビデオデータをコーディングするためのパレットを決定することの一例を示す概念図である。図10の例は、パレットモード(PAL)を使用してコーディングされた、第1のパレット184に関連付けられた第1のコーディングユニット(CU)180と、第2のパレット192に関連付けられた第2のCU188とを有する、ピクチャ178を含む。以下でより詳細に説明されるように、本開示の技法によれば、第2のパレット192は第1のパレット184に基づく。ピクチャ178はまた、イントラ予測コーディングモードを用いてコーディングされたブロック196、およびインター予測コーディングモードを用いてコー

10

20

30

40

50

ディングされたブロック200を含む。

【0184】

図10の技法は、説明のために、ビデオエンコーダ20(図1および図2)およびビデオデコーダ30(図1および図3)による実行のコンテキストにおいて、HEVCビデオコーディング規格に関して説明される。しかしながら、本開示の技法はこのようにして限定されず、他のビデオコーディングプロセッサおよび/またはデバイスによって、他のビデオコーディングプロセスおよび/または規格において適用され得ることを理解されたい。

【0185】

概して、パレットは、現在コーディングされているCU、図10の例ではCU188にとって支配的および/または代表的ないくつかの画素値を指す。第1のパレット184および第2のパレット192は、複数のパレットを含むように示される。いくつかの例では、本開示の態様によれば、ビデオコーダ(ビデオエンコーダ20またはビデオデコーダ30などの)は、パレットをCUの色成分ごとに別個にコーディングし得る。たとえば、ビデオエンコーダ20は、CUのルーマ(Y)成分用のパレット、CUのクロマ(U)成分用の別のパレット、およびCUのクロマ(V)成分用のさらに別のパレットを符号化し得る。この例では、YパレットのエントリがCUの画素のY値を表し得、UパレットのエントリがCUの画素のU値を表し得、VパレットのエントリがCUの画素のV値を表し得る。

10

【0186】

他の例では、ビデオエンコーダ20は、CUのすべての色成分用の単一のパレットを符号化し得る。この例では、ビデオエンコーダ20は、 Y_i 、 U_i 、および V_i を含むトリプル値であるi番目のエントリを有するパレットを符号化し得る。この場合、パレットは、画素成分の各々に関する値を含む。したがって、複数の個々のパレットを有するパレットのセットとしてのパレット184および192の表現は、一例にすぎず限定を意図しない。

20

【0187】

図10の例では、第1のパレット184は、それぞれ、エントリインデックス値1、エントリインデックス値2、およびエントリインデックス値3を有する、3つのエントリ202~206を含む。エントリ202~206は、インデックス値を、それぞれ、画素値A、画素値B、および画素値Cを含む画素値に関連付ける。本明細書で説明するように、第1のCU180の実際の画素値をコーディングするのではなく、ビデオコーダ(ビデオエンコーダ20またはビデオデコーダ30などの)は、インデックス1~3を使用してブロックの画素をコーディングするために、パレットベースコーディングを使用し得る。すなわち、第1のCU180の画素位置ごとに、ビデオエンコーダ20は、画素に関するインデックス値を符号化し得、その場合、インデックス値は、第1のパレット184のうちの1つまたは複数の中の画素値に関連付けられる。ビデオデコーダ30は、インデックス値をビットストリームから取得し得、インデックス値および第1のパレット184のうちの1つまたは複数を使用して、画素値を再構成し得る。したがって、パレットベースの復号の際にビデオデコーダ30によって使用するために、第1のパレット184がビデオエンコーダ20によって符号化ビデオデータビットストリームの中で送信される。

30

【0188】

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、第1のパレット184に基づいて第2のパレット192を決定し得る。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、予測パレット、この例では、第1のパレット184がそこから決定される1つまたは複数のブロックの位置を特定し得る。図10に示す例などのいくつかの例では、ビデオエンコーダ20および/またはビデオデコーダ30は、第2のCU188に関する予測パレットを決定するとき、左の隣接CU(第1のCU180)などの、前にコーディングされたCUの位置を特定し得る。

40

【0189】

図10の例では、第2のパレット192は、それぞれ、エントリインデックス値1、エントリインデックス値2、およびエントリインデックス値3を有する、3つのエントリ208~212を含む。エントリ208~212は、インデックス値を、それぞれ、画素値A、画素値B、および画

50

素値Dを含む画素値に関連付ける。この例では、ビデオエンコーダ20は、第1のパレット184のどのエントリが第2のパレット192の中に含まれるのかを示す1つまたは複数のシンタックス要素をコーディングし得る。図10の例では、1つまたは複数のシンタックス要素が、ベクトル216(たとえば、バイナリ予測ベクトル)として示される。ベクトル216は、いくつかの関連付けられたピン(または、ビット)を有し、各ピンは、そのピンに関連付けられたパレット予測子が現在パレットのエントリを予測するために使用されるかどうかを示す。たとえば、ベクトル216は、第1のパレット184の最初の2つのエントリ(202および204)が第2のパレット192の中に含まれることを示し(ベクトル216の中の「1」としての値)、第1のパレット184の第3のエントリは、第2のパレット192の中に含まれない(ベクトル216の中の「0」としての値)。図10の例では、ベクトルはブールベクトルである。

10

【0190】

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、パレット予測を実行するとき、パレット予測子リスト(パレット予測子テーブルと呼ばれることもある)を決定し得る。パレット予測子リストは、現在ブロックをコーディングするためのパレットの1つまたは複数のエントリを予測するために使用される、1つまたは複数の隣接ブロックのパレットからのエントリを含み得る。ビデオエンコーダ20およびビデオデコーダ30は、同じ方式でリストを構成し得る。ビデオエンコーダ20およびビデオデコーダ30は、パレット予測子リストのどのエントリが現在ブロックをコーディングするためのパレットの中に含まれるべきであるのかを示すためのデータ(ベクトル216などの)をコーディングし得る。

【0191】

20

C. Gisquet、G. Laroche、およびP. Onno、「AhG10: Palette predictor stuffing」、JCTVC-Q0063という文書が、パレット予測子リストを決定するための1つの例示的なプロセスを開示する。いくつかの例では、上述のように、現在コーディングされているブロックに対してパレットの中の1つまたは複数のエントリを予測するために、パレット予測子リストの中の各アイテムが使用されるのか(それとも、使用されないのか)を示すために、ビデオエンコーダ20またはビデオデコーダ30はブールベクトル(ベクトル216などの)を使用し得る。

【0192】

いくつかの例では、パレット予測子リストの中のアイテムのすべては、前にコーディングされたパレット(たとえば、前にコーディングされたブロックを用いてコーディングされたパレット)から導出される。しかしながら、そのようなパレットは、現在CUから空間的に遠く離れている場合があり、そのことがパレット相関を比較的弱くさせることがある。概して、パレット予測子テーブル(または、リスト)を拡張することが役に立つ場合がある(たとえば、より正確な予測子を提供し得、そのことが効率の向上をもたらし得る)。しかしながら、比較的大きいパレット予測子テーブル(または、リスト)を決定および使用することは、比較的長いブールベクトルをもたらす。

30

【0193】

様々な例では、本開示で説明する技法は、パレットベースコーディングにおいて、パレットを決定すること、予測すること、および/またはシグナリングすることの様々な組合せのための技法を含み得る。技法は、パレット予測子を決定すること、予測子を予測子の候補リストに追加すること、予測子を予測子の候補リストから取り除くこと、候補予測子の使用の表示をコーディングすること、または本明細書で説明する任意の他の技法の、任意の組合せを含み得る。例示および明快のためにいくつかの例が個々に説明されることがあるが、本開示は、本明細書で説明するパレットベースコーディングのための技法の任意の組合せを意図する。

40

【0194】

本開示のいくつかの態様によれば、ビデオコーダ(ビデオエンコーダ20またはビデオデコーダ30などの)は、1つまたは複数の新しいパレット予測子を決定し得る。たとえば、本開示のいくつかの技法は、1つまたは複数の空間パレット予測子を決定することを含む。一例では、現在CUの上のCU(「上方CU」または「上位CU」と呼ばれ、図10の例ではイント

50

ラコード化CU196として示される)は、パレットベースコーディングを使用し、上位CUからのパレットが、現在CU(第2のCU188)に関するパレット予測子として有効である。すなわち、上位CUのエントリの各々は、現在パレットを予測するためのパレット予測子リスト(たとえば、どのエントリが予測子として使用されるのかを示すベクトルに関連付けられ得る)の中に含まれ得る。

【0195】

しかしながら、上方CUが異なるCTUの一部である場合、予測のために上方CUからパレットを使用することは、追加の記憶領域を必要とし得る。たとえば、インター予測されるブロックの場合、ビデオコードは、画素値を再構成するためにメモリ(たとえば、ビデオコードの外部のメモリ)にアクセスしなければならない場合があり、そのことがレイテンシをもたらし得る。しかしながら、ビデオコードは、いくつかの例では、本明細書によって参照サンプルとして示される、イントラ予測されるブロックに関連付けられた再構成サンプルを考慮することがある。そのような再構成サンプルは、局所的に記憶されてよく、ビデオエンコード20とビデオデコード30の両方にとって利用可能であり得る。

【0196】

一例では、本開示の態様によれば、参照サンプルはパレット予測子リストに追加される。たとえば、空間的に隣接するブロックの画素値(パレットコーディングまたはイントラコーディングされ得る)は、候補パレット予測子として識別され得、ベクトルに関連付けられ得る(上述のように)。いくつかの例では、参照サンプルのサブセットだけが使用され得る。たとえば、パレット予測子リストが、左に隣接するCU(第1のCU180)などの特定のCUからのパレットをすでに含む場合、そのCUからのサンプルがパレット予測子リストの中に含まれないことがある。さらに、いくつかの例では、ビデオコードは、ビデオエンコード20および特にビデオデコード30へのさらなる複雑さを犠牲にして、リストの中の全く同じパレット予測子を除去するための取除き処理を適用し得る。これらの新しいパレット予測子は、予測子リストの先頭またはリストの末尾のいずれかに挿入され得る。他の例では、新しいパレット予測子(たとえば、空間パレット予測子)は、予測子リストサイズ、CUサイズ、または同様のものなどのいくつかの規則に従って、リストの中に適応的に挿入され得る。

【0197】

他の例では、現在CUの左側の列の中の任意のCUの中の画素などの、他の任意の再構成画素も、パレット予測子リストに追加され得る。

【0198】

このようにして、ビデオエンコード20および/またはビデオデコード30は、ビデオデータの現在ブロック用のパレットの1つまたは複数のエントリを予測するための、1つまたは複数の空間パレット予測子を決定することであって、1つまたは複数のエントリの各エントリが画素値を示すことと、1つまたは複数のパレット予測子の各それぞれのパレット予測子がビデオデータの現在ブロック用のパレットのエントリを予測するために使用されるかどうかを示す1つまたは複数のシンタックス要素をコーディングすることとを行い得る。

【0199】

本開示の他の態様によれば、ビデオコード(ビデオエンコード20またはビデオデコード30などの)は、パレット予測子をシグナリングするための新しいブールベクトルを決定し得る。いくつかの例では、本開示の技法は、パレット予測のためのブールベクトルを符号化するために使用され得る。

【0200】

たとえば、本開示の態様によれば、ビデオエンコード20またはビデオデコード30は、パレット予測のための2分木を決定し得る。例示のための一例では、パレット予測子の使用を表す0~1ブールベクトルを、Nによって示す。このベクトルを効率的に符号化するために、2分木が使用される。詳細には、ベクトルNはいくつかの連続する領域に区分され、各領域は1つまたは1つを越えるベクトル要素を含む。その領域内の要素のすべてが0である

10

20

30

40

50

か否かを示すために、1ビットがシグナリングされる。領域がすべて0でない場合、領域は同じ方法で下部領域へさらに分割される。

【0201】

区分は、所与の規則に従って停止され得る。たとえば、ある規則は「領域サイズがしきい値よりも小さい場合、区分プロセスは停止する」であってよい。区分が停止すると、領域サイズが1よりも大きく、領域が非ゼロの要素を含む場合、領域の中の各要素がビットストリームの中でシグナリングされる。この場合、領域サイズがXであると仮定すると、最初のX-1個の要素がすべて0である場合、最後の要素は1でなければならない、その結果、それはビットストリームの中に含められ得ない。

【0202】

別の例では、区分プロセスが停止し、領域サイズが1よりも大きい第2のしきい値よりも小さいとき、ビデオコーデは、その領域の中の要素のすべてが0であるかどうかを示すための1ビットのシグナリングをスキップし得る。代わりに、領域の中の要素ごとに、ビデオコーデは、それぞれの要素が0であるのかそれとも1であるのかを示すための1ビットをシグナリングし得る。

【0203】

別の例では、2分木の代わりに、各ノードにおいて2つよりも多い分岐を有するツリーが使用され得る。同様に、いくつかの例では、ツリーのルートから始めて最初のいくつかのレベルの符号化はバイパスされ得る。そのような例では、値は1であるものと暗黙のうちに想定され得る。

【0204】

本開示の態様によれば、パレット予測子リストからの最後の非ゼロのエントリ(すなわち、ブルベクトルの中の最後の1)の位置が、個々のエントリが0であるのかそれとも1であるのかをシグナリングする前に、ビットストリームの中で明示的にコーディングされ得る。この場合、最後の位置(それを含む)からブルベクトルの末尾までの要素は、ビットストリームの中でシグナリングされない。別の例では、予測のために使用される最初のパレット予測子の位置、すなわち、ブルベクトルの中の最初の1も、ビットストリームの中で明示的にシグナリングされ得る。

【0205】

本開示の態様によれば、パレット予測子リストの中から使用されるエントリ数が、ビットストリームの中で明示的にコーディングされ得る。たとえば、本開示の技法は、上述のブルベクトルの1(「1」)としてのその数をシグナリングすることを含む。いくつかの例では、ビデオエンコーダ20は、最初に、ブルベクトルの中の「1」の数(TotalOnes)をシグナリングし得る。ビデオエンコーダ20は、次いで、シグナリングされた1の数がTotalOnesに等しくなるまで、ブルベクトルの中の各アイテム(0または1)を連続的にシグナリング(または、いくつかの他の方式でシグナリング)し得る。

【0206】

同様に、ビデオデコーダ30は、最初に、受信されるブルベクトルの中に含まれている1の数の表示を復号し得る。ビデオデコーダ30は、次いで、示された「1」の数と一致する数の、ベクトルの中の「1」をビデオデコーダ30が復号するまで、受信されるベクトルの各要素を復号し得る。いくつかの例では、アイテム単位のシグナリング(たとえば、ベクトルの要素に対する)は、ベクトルの先頭から連続的であってよい。他の例では、ベクトルの要素は、いくつかの他の走査順序に従ってシグナリングされてよい。さらに他の例では、アイテム単位のシグナリングは、ランレングスコーディング技術を使用してよい。

【0207】

いくつかの例では、ブルベクトルの中の1の数が最初にシグナリングされるとき(たとえば、ブルベクトルの前にTotalOnesがビットストリームの中でシグナリングされるとき)、ブルベクトルの中の各シンタックス要素(たとえば、フラグ)は、CABACを使用してコンテキストコーディングされ得る。いくつかの事例では、複数のコンテキストが使用され得る。いくつかの例では、ブルベクトルのシンタックス要素のすべてがコンテキスト

10

20

30

40

50

コーディングされてよい。他の例では、ブールベクトルのシンタックス要素のサブセットだけがコンテキストコーディングされてよい。

【0208】

本開示の態様によれば、ブールベクトルのシンタックス要素のためのコンテキストは、シンタックス要素(たとえば、バイナリフラグ)のブールベクトル内での位置、および最初にシグナリングされる1のシグナリングされる数($TotalOnes$)に依存し得る。例示のための一例では、 $((位置>3)\&\&((位置/2+1)>TotalOnes))$ の場合、ビデオコーデ(ビデオエンコーダ20またはビデオデコーダ30などの)は第1のコンテキストを使用し得、そうでない場合(条件が真でないとき)、ビデオコーデは別のコンテキストを使用し得、その場合、位置は、コーディングされているシンタックス要素のブールベクトルの中での相対位置を示す。この場合、ビデオコーデは、ブールベクトルの中のシンタックス要素(たとえば、バイナリフラグ)をコーディングするために2つのコンテキストを使用し、特定のシンタックス要素のブールベクトルの中での相対位置に基づいて、ブールベクトルの特定のシンタックス要素のためのコンテキストを選択する。別の例では、ビデオコーデは、条件 $((位置>3)\&\&((位置/4+1)>=TotalOnes))$ が真である場合、あるコンテキストを使用し得、条件が真でない場合、別のコンテキストを使用し得る。コンテキストを決定するための他の条件も使用され得る。

【0209】

上記の例では、ブールベクトルの特定のシンタックス要素のためのCABACコンテキストは、シンタックス要素の相対位置、およびブールベクトルの中の1の総数($TotalOnes$)に依存する。他の例では、位置および $TotalOnes$ に依存する代わりに、ビデオコーデは、コーディングされているシンタックス要素の相対位置、およびこれまでにシグナリングされた1の総数(たとえば、シンタックス要素がブールベクトルの中でコーディングされる前に発生した1の総数)に基づいて、シンタックス要素のためのコンテキストを決定し得る。さらに他の例では、ビデオコーデは、ブールベクトルのシンタックス要素をCABACコーディングするためのコンテキストを決定するために、上記で説明した条件の組合せを使用し得る。

【0210】

本開示の態様によれば、パレットサイズが、エンコーダによってシグナリングされ、デコーダによって受信される。たとえば、パレットの中のいくつかのエントリが、パレットのサイズを示すためにシグナリングされ得る。いくつかの例では、ビデオエンコーダ20は、最初にパレットサイズ($PLTSize$)をシグナリングし得る。ビデオエンコーダ20、次いで、シグナリングされた1の数が $PLTSize$ に等しくなるまで、ブールベクトルの中の各アイテムを(0または1)をシグナリングし得る。ビデオデコーダ30は、パレットサイズの表示を受信し得、パレットの中のエントリの数がシグナリングされたパレットサイズに到達するまで、(たとえば、ベクトルによって識別されたパレット予測子を使用して)パレットに追加し続け得る。

【0211】

いくつかの例では、ビデオエンコーダ20は、最初にパレットサイズ($PLTSize$)をシグナリングし得る。ビデオエンコーダ20は、次いで、予測されていないパレットアイテムの数($nonPredPLTSize$)をシグナリングし得る。ビデオエンコーダ20は、次いで、シグナリングされた「1」の数が、パレットサイズと予測されていないパレットアイテムの数との間の差分($PLTSize-nonPredPLTSize$)に等しくなるまで、ブールベクトルの中の各アイテム(0または1)をシグナリングし得る。いくつかの例では、ビデオデコーダ30は、パレットサイズの表示、およびパレット予測子(たとえば、ベクトルの「0」の数)を使用する予測されていないパレットの中のエントリの数を受信し得る。ビデオデコーダ30はまた、どのパレット予測子がパレットの中に含まれるべきであることを示すベクトル(たとえば、上述されたブールベクトルなどの)を受信し得る。ビデオデコーダ30は、次いで、「1」の数がパレットサイズと予測されていないパレットアイテムの数との間の差分($PLTSize-nonPredPLTSize$)に等しくなるまで、ベクトルの中の各アイテムを復号し得る。

【0212】

このようにして、ビデオエンコーダ20および/またはビデオデコーダ30は、一例では、ビデオデータの現在ブロック用のパレットの1つまたは複数のエントリを予測するための1つまたは複数のパレット予測子を決定することであって、1つまたは複数のエントリの各エントリが画素値を示すことと、複数のピンを有するベクトルを決定することであって、ベクトルの各ピンが、1つまたは複数のパレット予測子のそれぞれのパレット予測子がビデオデータの現在ブロック用のパレットのエントリを予測するために使用されるかどうかを示すことと、ベクトルを1つまたは複数の領域に区分することと、ベクトルのピンの1つまたは複数の値を区分に基づいて決定することを行い得る。

【0213】

上述された図10の技法はCU(HEVC)のコンテキストにおいて説明されているが、技法が予測ユニット(PU)または他のビデオコーディングプロセスおよび/もしくは規格にも適用され得ることを理解されたい。

【0214】

例に応じて、本明細書で説明する技法のいずれかのいくつかの行為またはイベントは異なる順序で実行されてもよく、一緒に追加され、統合され、または省略されてもよい(たとえば、説明される行為またはイベントのすべてが技法の実施のために必要とは限らない)ことを認識されたい。その上、いくつかの例では、行為またはイベントは、順次的にではなく、たとえばマルチスレッド処理、割り込み処理、または複数のプロセッサを通じて同時に実行され得る。加えて、明快のために本開示のいくつかの態様は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るが、本開示の技法がビデオコーダと関連した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。

【0215】

例示のために、本開示のいくつかの態様は開発中のHEVC規格に関して説明された。しかしながら、本開示で説明する技法は、まだ開発されていない他の標準的なまたは独自のビデオコーディングプロセスを含む、他のビデオコーディングプロセスにとって有用であり得る。

【0216】

上述された技法は、ビデオエンコーダ20(図1および図2)および/またはビデオデコーダ30(図1および図3)によって実行されてよく、それらの両方が概してビデオコーダと呼ばれることがある。同様に、ビデオコーディングは、適用可能なとき、ビデオ符号化またはビデオ復号を指すことがある。

【0217】

本開示で説明する技法は、パレットコーディングの1つまたは複数の異なる態様の様々な組合せのための技法を含んでよい。

【0218】

1つまたは複数の例では、説明される機能は、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た場合、機能は、1つまたは複数の命令またはコードとして、コンピュータ可読媒体に記憶またはコンピュータ可読媒体を介して送信されてもよく、ハードウェアベースの処理ユニットによって実行されてもよい。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って、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含んでよい。このようにして、コンピュータ可読媒体は、概して、(1)非一時的な有形コンピュータ可読記憶媒体、または(2)信号もしくは搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明した技法を実施するための命令、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がコンピュータ可読媒体を含んでもよい。

【 0 2 1 9 】

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、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、または命令もしくはデータ構造の形式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るとともに、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含んでもよい。また、任意の接続が、適切にコンピュータ可読媒体と呼ばれる。たとえば、同軸ケーブル、光ファイバケーブル、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、ウェブサイト、サーバ、または他のリモートソースから命令が送信される場合、同軸ケーブル、光ファイバケーブル、ツイストペア、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は、媒体の定義に含まれる。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、または他の一時媒体を含まず、代わりに非一時的有形記憶媒体を対象とすることを理解されたい。ディスク(disk)およびディスク(disc)は、本明細書で使用される場合、コンパクトディスク(CD)、レーザディスク、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびブルーレイディスクを含み、ディスク(disk)は通常、磁気的にデータを再生し、ディスク(disc)は、レーザを用いて光学的にデータを再生する。上記の組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。

10

【 0 2 2 0 】

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(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、または他の等価の集積論理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てよい。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造、または本明細書で説明した技法の実装に適した任意の他の構造のいずれかを指すことがある。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た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てよく、あるいは組み合わせられたコーデックに組み込まれてもよい。また、技法は、1つまたは複数の回路または論理要素において完全に実装されてもよい。

20

【 0 2 2 1 】

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において実装され得る。様々なコンポーネント、モジュール、またはユニットが、開示される技法を実行するように構成されたデバイスの機能的態様を強調するために本開示で説明されるが、必ずしも異なるハードウェアユニットによる実現を必要としない。むしろ、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において組み合わせられてよく、または適切なソフトウェアおよび/もしくはファームウェアとともに、上記で説明したような1つもしくは複数のプロセッサを含む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てよい。

30

【 0 2 2 2 】

様々な例が説明された。これらおよび他の例は、以下の特許請求の範囲内に入る。上記の方法は、個々に、または組み合わせて使用され得る。

40

【 符号の説明 】

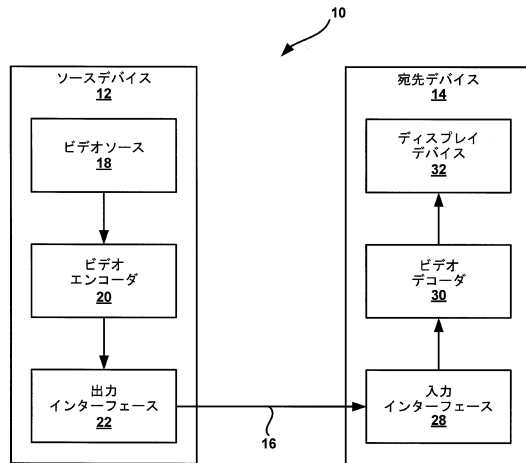
【 0 2 2 3 】

- 10 ビデオコーディングシステム
- 12 ソースデバイス
- 14 宛先デバイス
- 16 チャンネル
- 18 ビデオソース
- 20 ビデオエンコーダ

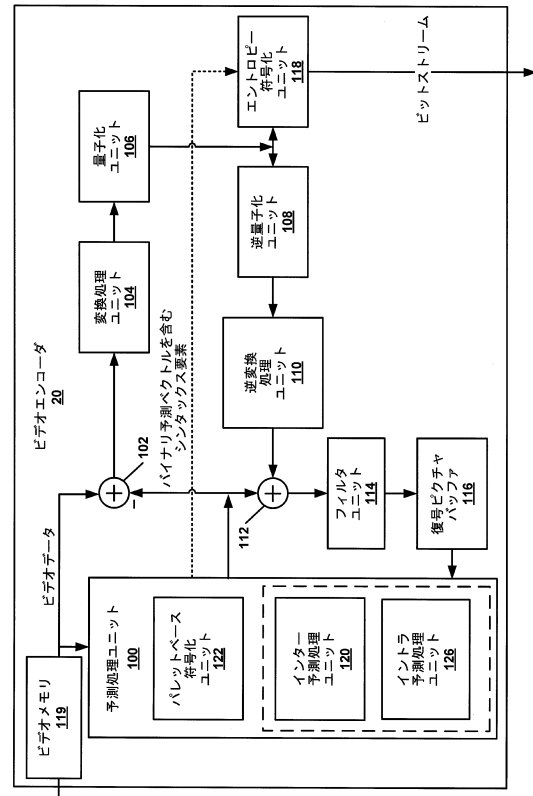
50

22	出力インターフェース	
28	入力インターフェース	
30	ビデオデコーダ	
32	ディスプレイデバイス	
100	予測処理ユニット	
102	残差生成ユニット	
104	変換処理ユニット	
106	量子化ユニット	
108	逆量子化ユニット	
110	逆変換処理ユニット	10
112	再構成ユニット	
114	フィルタユニット	
116	復号ピクチャバッファ	
118	エントロピー符号化ユニット	
119	ビデオメモリ	
120	インター予測処理ユニット	
122	パレットベース符号化ユニット	
126	イントラ予測処理ユニット	
150	エントロピー復号ユニット	
152	予測処理ユニット	20
154	逆量子化ユニット	
156	逆変換処理ユニット	
158	再構成ユニット	
160	フィルタユニット	
162	復号ピクチャバッファ	
163	ビデオメモリ	
164	動き補償ユニット	
165	パレットベース復号ユニット	
166	イントラ予測処理ユニット	
203	パレット生成ユニット	30
204	マップユニット	
206	バイナリ予測ベクトル生成ユニット	
209	バイナリ予測ベクトル圧縮ユニット	
210	予測子パレットバッファ	
212	画素値	
214	マップ	
215	符号化バイナリ予測ベクトル	
302	マップユニット	
304	パレット生成ユニット	
306	バイナリ予測ベクトル解凍ユニット	40
310	予測子パレットバッファ	
312	マップ	
314	復号ビデオデータ	
316	符号化バイナリ予測ベクトル	

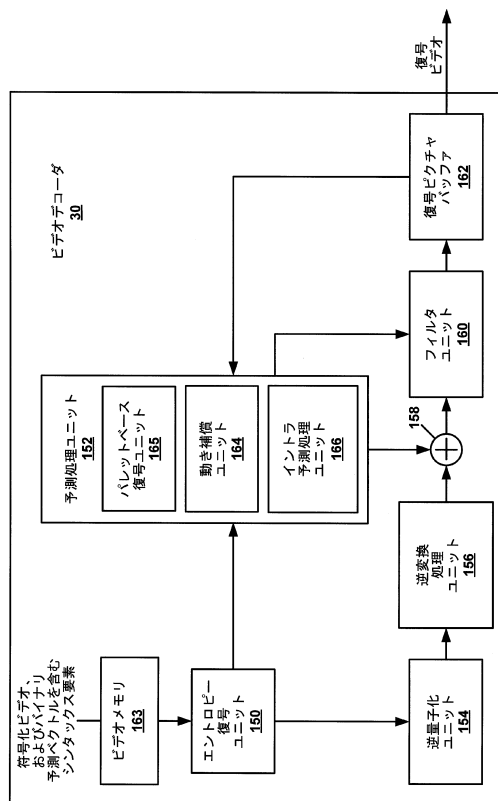
【図 1】



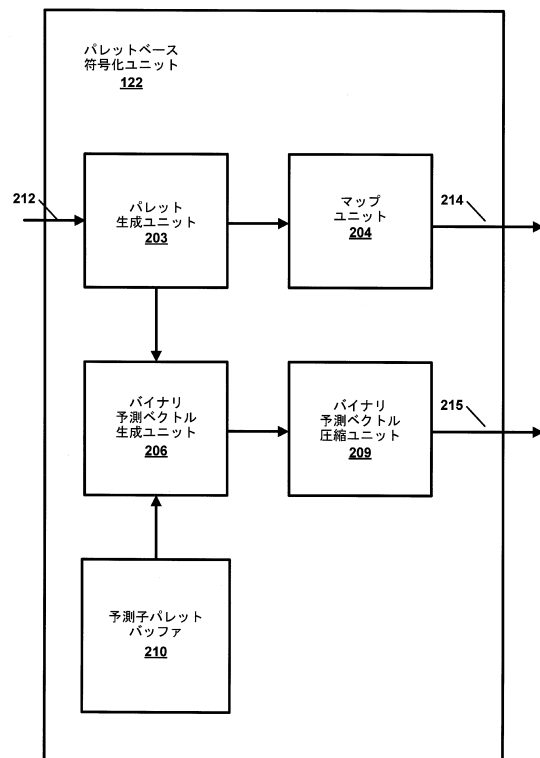
【図 2】



【図 3】



【図 4】



【図 5】

予測子パレットバッファ
210

エントリ インデックス	画素値
1	値A
2	値B
3	値C
4	値D
5	値E
6	値F
7	値G
8	値H
9	値I
10	値J
11	値K
12	値L
13	値M
...	...

現在パレット
220

エントリ インデックス	画素値
1	値A
2	値B
3	値E
4	値I
5	新しい値
6	新しい値
7	新しい値
8	新しい値

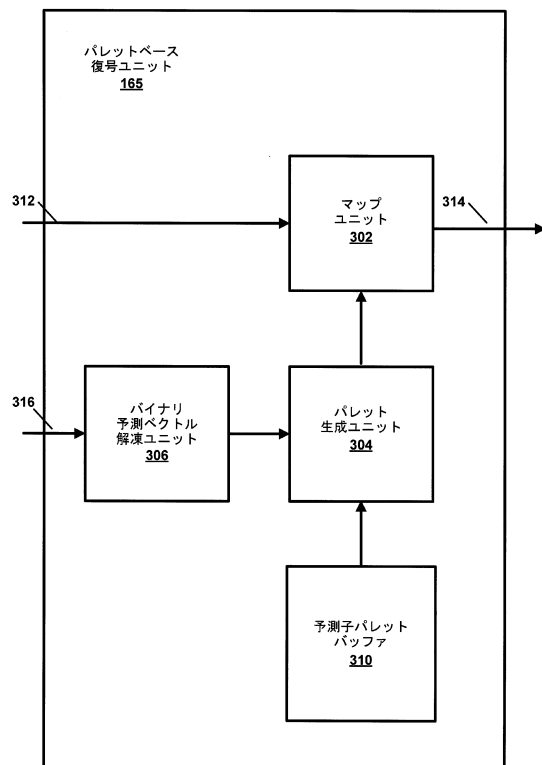
$b = [1100100010000]$

【図 6】

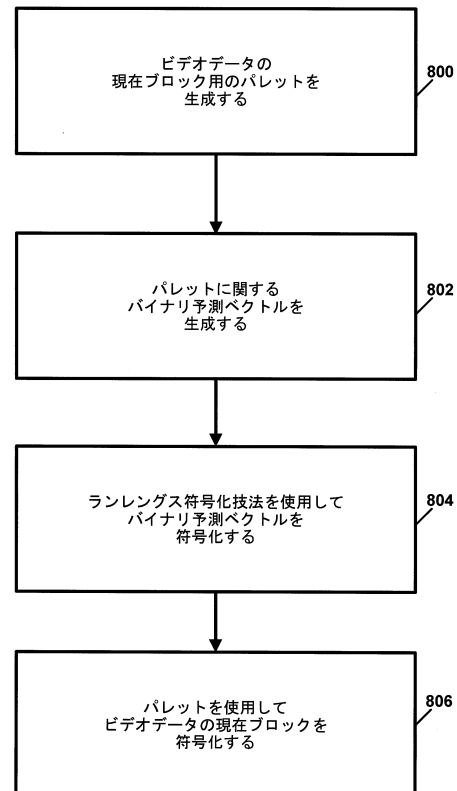
$b = [1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$

ランレングスシーケンス = 00234

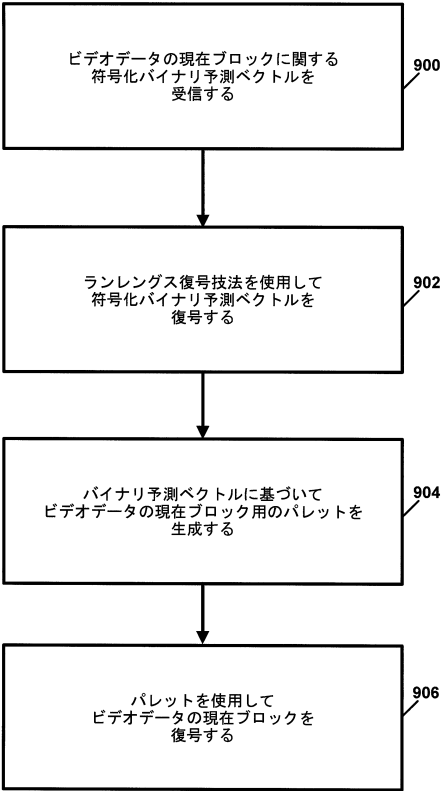
【図 7】



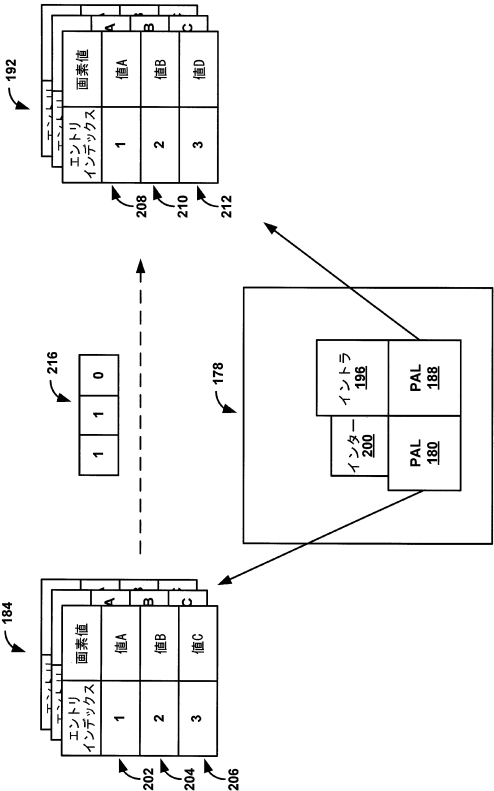
【図 8】



【図 9】



【図 10】



フロントページの続き

- (31)優先権主張番号 62/002,668
 (32)優先日 平成26年5月23日(2014.5.23)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 62/015,327
 (32)優先日 平成26年6月20日(2014.6.20)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 62/018,461
 (32)優先日 平成26年6月27日(2014.6.27)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 62/041,119
 (32)優先日 平成26年8月24日(2014.8.24)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 14/667,411
 (32)優先日 平成27年3月24日(2015.3.24)
 (33)優先権主張国 米国(US)

- (72)発明者 ウェイ・ブ
 アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
 (72)発明者 ヴァディム・セレジン
 アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
 (72)発明者 ラジャン・ラックスマン・ジョシ
 アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
 (72)発明者 ジョエル・ソール・ロジャルス
 アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775
 (72)発明者 フェン・ゾウ
 アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライヴ・5775

審査官 久保 光宏

- (56)参考文献 特表2009-531997(JP,A)
 特開2011-130413(JP,A)
 Chen, J., et al., "Description of screen content coding technology proposal by Qualcomm", [online], Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 2014年3月18日, Document: JCTVC-Q0031 (version 1), Pages 1-18, [retrieved on 2018.12.18], Retrieved from the Internet: , URL, http://phenix.it-sudparis.eu/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0031-v1.zip
 大久保榮監修, 「改訂三版H.264/AVC教科書」, 日本, 株式会社インプレスR&D, 2009年1月1日, 第1版, 第148~162頁, ISBN:978-4-8443-2664-9

- (58)調査した分野(Int.Cl., DB名)
 H04N19/00-19/98
 CSDB(日本国特許庁)
 IEEE Explore(IEEE)