

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7509784号
(P7509784)

(45)発行日 令和6年7月2日(2024.7.2)

(24)登録日 令和6年6月24日(2024.6.24)

(51)国際特許分類		F I	
H O 4 N	19/13 (2014.01)	H O 4 N	19/13
H O 4 N	19/18 (2014.01)	H O 4 N	19/18
H O 4 N	19/136 (2014.01)	H O 4 N	19/136
H O 4 N	19/91 (2014.01)	H O 4 N	19/91
請求項の数 14 (全55頁)			
(21)出願番号	特願2021-538268(P2021-538268)	(73)特許権者	507364838
(86)(22)出願日	令和2年1月2日(2020.1.2)		クアルコム, インコーポレイテッド
(65)公表番号	特表2022-516132(P2022-516132 A)		アメリカ合衆国 カリフォルニア 9 2 1
(43)公表日	令和4年2月24日(2022.2.24)		2 1 サン ディエゴ モアハウス ドライ
(86)国際出願番号	PCT/US2020/012054	(74)代理人	ブ 5 7 7 5
(87)国際公開番号	WO2020/142608		100108453
(87)国際公開日	令和2年7月9日(2020.7.9)	(74)代理人	弁理士 村山 靖彦
審査請求日	令和4年12月16日(2022.12.16)		100163522
(31)優先権主張番号	62/787,707	(72)発明者	弁理士 黒田 晋平
(32)優先日	平成31年1月2日(2019.1.2)		ムハンマド・ゼイド・コバン
(33)優先権主張国・地域又は機関	米国(US)		アメリカ合衆国・カリフォルニア・9 2
(31)優先権主張番号	16/732,008	(72)発明者	1 2 1 - 1 7 1 4 ・サン・ディエゴ・モ
(32)優先日	令和1年12月31日(2019.12.31)		アハウス・ドライヴ・5 7 7 5
最終頁に続く			マルタ・カルチェヴィッチ
			アメリカ合衆国・カリフォルニア・9 2
		最終頁に続く	

(54)【発明の名称】 係数レベルのためのエスケープコーディング

(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、

最大の数の普通モードでコーディングされたビンに達するまで、変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト復号するステップであって、前記コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、前記1つまたは複数の有意性フラグの各々が、変換係数に対する絶対レベルが0に等しいかどうかを示し、前記1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか、または奇数の絶対レベルを有するかを示し、前記1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、コンテキスト復号するステップと、

前記最大の数の普通モードでコーディングされたビンに達したことに応答して、前記変換係数グループの係数の第2のセットに対する追加のシンタックス要素をバイパス復号するステップであって、

変換係数のレベル値に対するプレフィックス値を受信するステップと、

ゴロムライスコーディングを使用して前記プレフィックス値を復号するステップと、

前記プレフィックス値の長さが閾値と等しいことに応答して、前記変換係数に対するサフィックス値を受信するステップであって、前記閾値が全ての係数に対して固定された値である、ステップと、

指数ゴロムコーディングを使用して前記サフィックス値を復号するステップと
を備える、バイパス復号するステップと、

前記コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の前記第1のセットに対する値を決定するステップと、

前記追加のシンタックス要素に基づいて前記変換ユニットの係数の前記第2のセットに対する値を決定するステップであって、

前記復号されたプレフィックス値および前記復号されたサフィックス値に基づいて前記変換係数に対するレベル値を決定するステップを備えるステップと
を備え、

前記追加のシンタックス要素をバイパス復号するステップがさらに、

前記変換係数に対するライスパラメータの値を決定するステップと、

前記ライスパラメータの前記値に基づいて、前記閾値を決定するステップと
を備える、方法。

【請求項2】

前記変換係数に対する前記レベル値を決定するステップが、

前記復号されたプレフィックス値を前記復号されたサフィックス値に加算するステップ
を備える、請求項1に記載の方法。

【請求項3】

前記追加のシンタックス要素をバイパス復号するステップがさらに、

第2の係数に対する第2のプレフィックス値を受信するステップと、

ゴロムライスコーディングを使用して前記第2のプレフィックス値を復号するステップ
と、

前記第2のプレフィックス値の長さが前記閾値未満であることに応答して、前記第2の係数に対するサフィックス値を受信することなく、前記第2の係数に対するレベル値を決定するステップとを備える、請求項1に記載の方法。

【請求項4】

前記変換係数グループの前記シンタックス要素をコンテキスト復号するステップが、前記変換係数グループの前記シンタックス要素を復号するためにコンテキスト適応バイナリ算術復号を実行するステップを備える、請求項1に記載の方法。

【請求項5】

前記変換係数に対する前記レベル値が絶対レベル値を備える、請求項1に記載の方法。

【請求項6】

前記変換係数に対する前記レベル値が残りのレベル値を備える、請求項1に記載の方法。

【請求項7】

前記閾値が5に等しい、請求項1に記載の方法。

【請求項8】

前記閾値が6に等しい、請求項1に記載の方法。

【請求項9】

係数の前記第1のセットに対する前記値および係数の前記第2のセットに対する前記値に基づいて、復号された変換ブロックを決定するステップと、

前記復号された変換ブロックを予測ブロックに加算して再構築されたブロックを決定するステップと、

前記再構築されたブロックに対して1回または複数回のフィルタリング動作を実行して、ビデオデータの復号されたブロックを決定するステップと、

ビデオデータの前記復号されたブロックを含むビデオデータの復号されたピクチャを出力するステップとをさらに備える、請求項1に記載の方法。

【請求項10】

ビデオデータを復号するためのデバイスであって、

ビデオデータを記憶するように構成されるメモリと、

回路において実装された1つまたは複数のプロセッサとを備え、前記1つまたは複数のプ

10

20

30

40

50

ロセッサが、

最大の数の普通モードでコーディングされたビンに達するまで、変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト復号することであって、前記コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、前記1つまたは複数の有意性フラグの各々が、変換係数に対する絶対レベルが0に等しいかどうかを示し、前記1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか、または奇数の絶対レベルを有するかを示し、前記1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、コンテキスト復号することと、

10

前記最大の数の普通モードでコーディングされたビンに達したことに応答して、前記変換係数グループの係数の第2のセットに対する追加のシンタックス要素をバイパス復号することであって、前記追加のシンタックス要素をバイパス復号するために、前記1つまたは複数のプロセッサが、

変換係数のレベル値に対するプレフィックス値を受信することと、

ゴロムライスコーディングを使用して前記プレフィックス値を復号することと、

前記プレフィックス値の長さが閾値と等しいことに応答して、前記変換係数に対するサフィックス値を受信することであって、前記閾値が全ての係数に対して固定された値である、受信することと、

指数ゴロムコーディングを使用して前記サフィックス値を復号することと

20

を行うように構成される、バイパス復号することと、

前記コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の前記第1のセットに対する値を決定することと、

前記追加のシンタックス要素に基づいて前記変換ユニットの係数の前記第2のセットに対する値を決定することであって、前記追加のシンタックス要素に基づいて前記変換ユニットの係数の前記第2のセットに対する前記値を決定するために、前記1つまたは複数のプロセッサが、前記復号されたプレフィックス値および前記復号されたサフィックス値に基づいて前記変換係数に対するレベル値を決定するように構成される、決定することと

を行うように構成され、

前記追加のシンタックス要素をバイパス復号するために、前記1つまたは複数のプロセッサがさらに、

30

前記変換係数に対するライスパラメータの値を決定し、

前記ライスパラメータの前記値に基づいて、前記閾値を決定する

ように構成されるデバイス。

【請求項11】

ビデオデータを符号化する方法であって、

変換係数グループの係数に対する値を決定するステップと、

最大の数の普通モードでコーディングされたビンに達するまで、前記変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化するステップであって、前記コンテキスト符号化されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、前記1つまたは複数の有意性フラグの各々が、変換係数に対する絶対レベルが0に等しいかどうかを示し、前記1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか、または奇数の絶対レベルを有するかを示し、前記1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、コンテキスト符号化するステップと、

40

前記最大の数の普通モードでコーディングされたビンに達したことに応答して、前記変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化するステップであって、

係数の前記第2のセットの変換係数に対するレベル値を決定するステップと、

50

前記レベル値が閾値より大きいことに応答して、ゴロムライスコーディングを使用して前記変換係数の前記レベル値に対するプレフィックス値を符号化するステップと、

前記レベル値が前記閾値より大きいことに応答して、指数ゴロムコーディングを使用してサフィックス値を符号化するステップであって、前記閾値が全ての係数に対して固定された値である、ステップと

を備える、バイパス符号化するステップと
を備え、

前記追加のシンタックス要素をバイパス符号化するステップがさらに、

前記変換係数に対するライスパラメータの値を決定するステップと、

前記ライスパラメータの前記値に基づいて、前記閾値を決定するステップと

を備える、方法。

【請求項 1 2】

前記変換係数に対する前記レベル値から前記プレフィックス値を差し引くことによって、前記サフィックス値を決定するステップをさらに備える、請求項 11 に記載の方法。

【請求項 1 3】

ビデオデータを符号化するためのデバイスであって、

ビデオデータを記憶するように構成されるメモリと、

回路において実装された 1 つまたは複数のプロセッサとを備え、前記 1 つまたは複数のプロセッサが、

変換係数グループの係数に対する値を決定することと、

最大の数の普通モードでコーディングされたビンに達するまで、前記変換係数グループのうちの係数の第 1 のセットに対するシンタックス要素をコンテキスト符号化することであって、前記コンテキスト符号化されたシンタックス要素が、1 つまたは複数の有意性フラグ、1 つまたは複数のパリティレベルフラグ、および 1 つまたは複数の第 1 のフラグを備え、前記 1 つまたは複数の有意性フラグの各々が、変換係数に対する絶対レベルが 0 に等しいかどうかを示し、前記 1 つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか、または奇数の絶対レベルを有するかを示し、前記 1 つまたは複数の第 1 のフラグの各々が、変換係数が 2 より大きい絶対レベルを有するかどうかを示す、コンテキスト符号化することと、

前記最大の数の普通モードでコーディングされたビンに達したことに応答して、前記変換係数グループのうちの係数の第 2 のセットに対する追加のシンタックス要素をバイパス符号化することであって、前記追加のシンタックス要素をバイパス符号化することが、

係数の前記第 2 のセットの変換係数に対するレベル値を決定することと、

前記レベル値が閾値より大きいことに応答して、ゴロムライスコーディングを使用して前記変換係数の前記レベル値に対するプレフィックス値を符号化することと、

前記レベル値が前記閾値より大きいことに応答して、指数ゴロムコーディングを使用してサフィックス値を符号化することであって、前記閾値が全ての係数に対して固定された値である、符号化することと

を含む、バイパス符号化することと

を行うように構成され、

前記追加のシンタックス要素をバイパス符号化するために、前記 1 つまたは複数のプロセッサがさらに、

前記変換係数に対するライスパラメータの値を決定し、

前記ライスパラメータの前記値に基づいて、前記閾値を決定するように構成される、デバイス。

【請求項 1 4】

命令を記憶するコンピュータ可読記憶媒体であって、前記命令が、1 つまたは複数のプロセッサによって実行されると、前記 1 つまたは複数のプロセッサに、請求項 1 から 9、11、および 12 のうちのいずれか一項に記載の方法を実行させる、コンピュータ可読記憶媒体。

10

20

30

40

50

【発明の詳細な説明】

【技術分野】

【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る、2019年1月2日出願された米国仮出願第62/787,707号の利益を主張する、2019年12月31日出願された米国特許出願第16/732,008号の優先権を主張する。

【0002】

本開示は、ビデオ符号化およびビデオ復号に関する。

【背景技術】

【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、ビデオゲーミングデバイス、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, Part 10、Advanced Video Coding (AVC)、High Efficiency Video Coding (HEVC)規格、ITU-T H.265/High Efficiency Video Coding (HEVC)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ビデオコーディング技法などのビデオコーディング技法を実施する。ビデオデバイスは、そのようなビデオコーディング技法を実施することによって、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。

【0004】

ビデオコーディング技法は、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)予測を含む。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(たとえば、ビデオピクチャ、またはビデオピクチャの一部分)は、ビデオブロックに区分されてもよく、ビデオブロックは、コーディングツリーユニット(CTU)、コーディングユニット(CU)、および/またはコーディングノードと呼ばれることもある。ピクチャのイントラコーディングされた(I)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の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コーディングされた(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の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の中の参照サンプルに対する時間的予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

【先行技術文献】

【非特許文献】

【0005】

【文献】Bross他、「Versatile Video Coding (Draft 6)」、ITU-T SG 16 WP 3およびISO/IEC JTC 1/SC 29/WG 11のJoint Video Experts Team (JVET)、第15回会合、ヨーテボリ、スウェーデン、2019年7月3日~12日、JVET-O2001-vE

【文献】H.Schwarz、T.Nguyen、D.Marpe、T.Wiegand、M.Karczewicz、M.Coban、J.Dong、「CE7: Transform coefficient coding with reduced number of regular-coded bins (tests 7.1.3a, 7.1.3b)」、JVET document JVET-L0274、マカオ、中国、2018年10月

【発明の概要】

【課題を解決するための手段】

【0006】

ビデオコーディング(たとえば、ビデオ符号化および/またはビデオ復号)は、通常、同じ

10

20

30

40

50

ピクチャの中のビデオデータのすでにコーディングされたブロック(たとえば、イントラ予測)または異なるピクチャの中のビデオデータのすでにコーディングされたブロック(たとえば、インター予測)のいずれかから、ビデオデータのブロックを予測することを伴う。いくつかの事例では、ビデオエンコーダはまた、予測ブロックを元のブロックと比較することによって残差データを計算する。したがって、残差データは、ビデオデータの予測ブロックと元のブロックとの差を表す。残差データをシグナリングするために必要とされるビットの数を減らすために、ビデオエンコーダは、残差データを変換係数へと変換し、変換係数を量子化し、変換され量子化された係数を符号化されたビットストリームにおいてシグナリングする。変換および量子化プロセスによって達成される圧縮は有損失であることがあり、変換および量子化プロセスが復号されたビデオデータに歪みをもたらし得ることを意味する。本開示は、変換係数のコーディングに関する、

10

より具体的には、ゴロムライス符号と指数ゴロム符号の組合せを使用することに関する技法について説明する。本開示の技法は、変換係数をシグナリングするために必要なビットの数を平均的に減らすことによって、変換係数のコーディング効率を高め得る。加えて、本開示の技法は、変換係数をシグナリングすることと関連付けられる最悪の場合のコーディングシナリオを限定し得る。

【0007】

本開示の例によれば、ビデオデータを復号するための方法は、最大の数の普通のコーディングされるビンに達するまで変換係数グループのうちの変換係数の第1のセットに対するシンタックス要素をコンテキスト復号するステップであって、コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、ステップと、最大の数の普通のコーディングされるビンに達したことに応答して、係数の第2のセットに対する追加のシンタックス要素をバイパス復号するステップであって、追加のシンタックス要素をバイパス復号する、ステップが、変換係数に対するプレフィックス値を受信するステップを含む、ステップと、ゴロムライス符号を使用してプレフィックス値を復号するステップと、プレフィックス値の長さが閾値に等しいことに応答して、変換係数に対するサフィックス値を受信するステップと、指数ゴロム符号を使用してサフィックス値を復号するステップと、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定するステップと、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定するステップであって、追加のシンタックス要素に基づいて変換ユニットの係数の第2のセットに対する値を決定するステップが、復号されたプレフィックス値および復号されたサフィックス値に基づいて、変換係数に対するレベル値を決定するステップを備える、ステップとを含む。

20

30

【0008】

本開示の別の例によれば、ビデオデータを復号するためのデバイスは、ビデオデータを記憶するように構成されるメモリと、1つまたは複数のプロセッサとを含み、プロセッサは、回路において実装され、最大の数の普通のコーディングされたビンに達するまで変換係数グループのうちの変換係数の係数の第1のセットに対するシンタックス要素をコンテキスト復号し、コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示し、最大の数の普通のコーディングされたビンに達したことに応答して、係数の第2のセットに対する追加のシンタック

40

50

ス要素をバイパス復号し、追加のシンタックス要素をバイパス復号するために、1つまたは複数のプロセッサが変換係数に対するプレフィックス値を受信するように構成され、ゴロムライス符号を使用してプレフィックス値を復号し、プレフィックス値の長さが閾値に等しいことに応答して、変換係数に対するサフィックス値を受信し、指数ゴロム符号を使用してサフィックス値を復号し、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定し、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定し、追加のシンタックス要素に基づいて変換ユニットの係数の第2のセットに対する値を決定するために、1つまたは複数のプロセッサが、復号されたプレフィックス値および復号されたサフィックス値に基づいて、変換係数に対するレベル値を決定するように構成される。

10

【0009】

本開示の別の例によれば、コンピュータ可読記憶媒体は命令を記憶し、この命令は、1つまたは複数のプロセッサによって実行されると、1つまたは複数のプロセッサに、最大の数の普通のコーディングされたピンに達するまで変換係数グループのうちの変換係数の第1のセットに対するシンタックス要素をコンテキスト復号させ、コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示し、最大の数の普通のコーディングされたピンに達したことに応答して、係数の第2のセットに対する追加のシンタックス要素をバイパス復号させ、追加のシンタックス要素をバイパス復号するために、命令が、1つまたは複数のプロセッサに、変換係数に対するプレフィックス値を受信させ、ゴロムライス符号を使用してプレフィックス値を復号させ、プレフィックス値の長さが閾値に等しいことに応答して、変換係数に対するサフィックス値を受信させ、指数ゴロム符号を使用してサフィックス値を復号させ、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定させ、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定させ、追加のシンタックス要素に基づいて変換ユニットの係数の第2のセットに対する値を決定することが、復号されたプレフィックス値および復号されたサフィックス値に基づいて、変換係数に対するレベル値を決定することを備える。

20

30

【0010】

本開示の別の例によれば、ビデオデータを復号するための装置は、最大の数の普通のコーディングされたピンに達するまで変換係数グループのうちの変換係数の第1のセットに対するシンタックス要素をコンテキスト復号するための手段であって、コンテキスト復号されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、手段と、最大の数の普通のコーディングされたピンに達したことに応答して、係数の第2のセットに対する追加のシンタックス要素をバイパス復号するための手段であって、追加のシンタックス要素をバイパス復号するための手段が、変換係数に対するプレフィックス値を受信するための手段を備える、手段と、ゴロムライス符号を使用してプレフィックス値を復号するための手段と、プレフィックス値の長さが閾値に等しいことに応答して、変換係数に対するサフィックス値を受信するための手段と、指数ゴロム符号を使用してサフィックス値を復号するための手段と、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定するための手段と、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定するための手段であって、追加のシンタックス要素に基づいて変換ユニ

40

50

ットの係数の第2のセットに対する値を決定することが、復号されたプレフィックス値および復号されたサフィックス値に基づいて、変換係数に対するレベル値を決定することを備える、手段とを含む。

【0011】

本開示の別の例によれば、ビデオデータを符号化する方法は、変換係数グループの係数に対する値を決定するステップと、最大の数の普通のコーディングされたピンに達するまで変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化するステップであって、コンテキスト符号化されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、ステップと、最大の数の普通のコーディングされたピンに達したことに応答して、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化するステップであって、追加のシンタックス要素をバイパス符号化するステップが、係数の第2のセットの変換係数に対するレベル値を決定するステップを含む、ステップと、レベル値が閾値より大きいことに応答して、ゴロムライス符号を使用して変換係数に対するプレフィックス値を符号化するステップと、レベル値が閾値より大きいことに応答して、指数ゴロム符号を使用してサフィックス値を符号化するステップとを含む。

【0012】

本開示の別の例によれば、ビデオデータを符号化するためのデバイスは、ビデオデータを記憶するように構成されるメモリと、1つまたは複数のプロセッサとを含み、プロセッサは、回路において実装され、変換係数グループの係数に対する値を決定し、最大の数の普通のコーディングされたピンに達するまで変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化し、コンテキスト符号化されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示し、最大の数の普通のコーディングされたピンに達したことに応答して、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化し、追加のシンタックス要素をバイパス符号化することが、係数の第2のセットの変換係数に対するレベル値を決定することを含み、レベル値が閾値より大きいことに応答して、ゴロムライス符号を使用して変換係数に対するプレフィックス値を符号化し、レベル値が閾値より大きいことに応答して、指数ゴロム符号を使用してサフィックス値を符号化するように構成される。

【0013】

本開示の別の例によれば、コンピュータ可読記憶媒体は命令を記憶し、この命令は、1つまたは複数のプロセッサによって実行されると、1つまたは複数のプロセッサに、変換係数グループの係数に対する値を決定させ、最大の数の普通のコーディングされたピンに達するまで変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化させ、コンテキスト符号化されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示し、最大の数の普通のコーディングされたピンに達したことに応答して、変換係数グループのうちの係数の第

2のセットに対する追加のシンタックス要素をバイパス符号化させ、追加のシンタックス要素をバイパス符号化するために、命令が、1つまたは複数のプロセッサに、係数の第2のセットの変換係数に対するレベル値を決定させ、レベル値が閾値より大きいことに応答して、ゴロムライス符号を使用して変換係数に対するプレフィックス値を符号化させ、レベル値が閾値より大きいことに応答して、指数ゴロム符号を使用してサフィックス値を符号化させる。

【0014】

本開示の別の例によれば、ビデオデータを符号化するための装置は、変換係数グループの係数に対する値を決定するための手段と、最大の数の普通のコーディングされたビンに達するまで変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化するための手段であって、コンテキスト符号化されたシンタックス要素が、1つまたは複数の有意性フラグ、1つまたは複数のパリティレベルフラグ、および1つまたは複数の第1のフラグを備え、1つまたは複数の有意性フラグの各々が、変換係数の絶対レベルが0に等しいかどうかを示し、1つまたは複数のパリティレベルフラグの各々が、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示し、1つまたは複数の第1のフラグの各々が、変換係数が2より大きい絶対レベルを有するかどうかを示す、手段と、最大の数の普通のコーディングされたビンに達したことに応答して、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化するための手段であって、追加のシンタックス要素をバイパス符号化するための手段が、係数の第2のセットの変換係数に対するレベル値を決定するための手段を含む、手段と、レベル値が閾値より大きいことに応答して、ゴロムライス符号を使用して変換係数に対するプレフィックス値を符号化するための手段と、レベル値が閾値より大きいことに応答して、指数ゴロム符号を使用してサフィックス値を符号化するための手段とを含む。

【0015】

1つまたは複数の例の詳細が、添付の図面および以下の説明に記載される。他の特徴、目的、および利点は、説明、図面、および特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】

【0016】

【図1】本開示の技法を実行することができる例示的なビデオ符号化および復号システムを示すブロック図である。

【図2A】例示的な四分木二分木(QTBT)構造を示す概念図である。

【図2B】QTBT構造に対応するコーディングツリーユニット(CTU)を示す概念図である。

【図3】コーディンググループ(CG)における絶対レベル(absLevel)を表すシンタックス要素の例示的な順序を示す図である。

【図4】確率モデルを選択するために使用されるテンプレートの図である。

【図5】本開示の技法を実行することができ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6】本開示の技法を実行することができる例示的なビデオデコーダを示すブロック図である。

【図7A】バイナリ算術コーディングにおける範囲更新プロセスを示す概念図である。

【図7B】バイナリ算術コーディングにおける範囲更新プロセスを示す概念図である。

【図8】バイナリ算術コーディングにおける出力プロセスを示す概念図である。

【図9】ビデオエンコーダの中のコンテキスト適応型バイナリ算術コーディング(CABAC)コーダを示すブロック図である。

【図10】ビデオデコーダの中のCABACコーダを示すブロック図である。

【図11】ビデオエンコーダの例示的な動作を示すフローチャートである。

【図12】ビデオデコーダの例示的な動作を示すフローチャートである。

【図13】ビデオデータの現在のブロックを符号化するためのビデオエンコーダの例示的な動作を示すフローチャートである。

【図14】図13の変換係数グループのうちの係数の第2のセットに対する追加のシンタッ

クス要素をバイパス符号化するためのビデオエンコーダの例示的な動作を示すフローチャートである。

【図15】ビデオデータの現在のブロックを復号するためのビデオデコーダの例示的な動作を示すフローチャートである。

【図16】図15の係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス復号するためのビデオデコーダの例示的な動作を示すフローチャートである。

【発明を実施するための形態】

【0017】

ビデオコーディング(たとえば、ビデオ符号化および/またはビデオ復号)は、通常、同じピクチャの中のビデオデータのすでにコーディングされたブロック(たとえば、イントラ予測)または異なるピクチャの中のビデオデータのすでにコーディングされたブロック(たとえば、インター予測)のいずれかから、ビデオデータのブロックを予測することを伴う。いくつかの事例では、ビデオエンコーダはまた、予測ブロックを元のブロックと比較することによって残差データを計算する。したがって、残差データは、ビデオデータの予測ブロックと元のブロックとの差を表す。残差データをシグナリングするために必要とされるビットの数を減らすために、ビデオエンコーダは、残差データを変換して量子化し、変換され量子化された残差データを符号化されたビットストリームにおいてシグナリングする。変換および量子化プロセスによって達成される圧縮は有損失であることがあり、変換および量子化プロセスが復号されたビデオデータに歪みをもたらし得ることを意味する。

【0018】

ビデオデコーダは、残差データを復号し予測ブロックに加算して、予測ブロック単独よりも元のビデオブロックとよく一致する再構築されたビデオブロックを生成する。残差データの変換および量子化によりもたらされる損失により、再構築されたブロックには歪みまたはアーティファクトがあることがある。1つのよくあるタイプのアーティファクトまたは歪みはブロックネスと呼ばれ、これは、ビデオデータをコーディングするために使用されるブロックの境界が見えるというものである。

【0019】

復号されたビデオの品質をさらに改善するために、ビデオデコーダは、再構築されたビデオブロックに対して1回または複数回のフィルタリング動作を実行することができる。これらのフィルタリング動作の例は、デブロックフィルタリング、サンプル適応オフセット(SAO)フィルタリング、および適応ループフィルタリング(ALF)を含む。これらのフィルタリング動作のためのパラメータは、ビデオエンコーダによって決定され符号化されたビデオビットストリームにおいて明示的にシグナリングされ得るか、または符号化されたビデオビットストリームにおいてパラメータが明示的にシグナリングされる必要なくビデオデコーダによって暗黙的に決定され得るかのいずれかである。

【0020】

上で紹介されたように、ビデオエンコーダは、残差データを変換して変換係数を生成する。加えて、それらの変換係数は量子化され得る。本開示では、変換係数、または係数という用語は、量子化された変換係数または逆量子化された変換係数のいずれかを指し得る。本開示は、ビデオエンコーダからビデオデコーダに、変換係数、たとえば量子化された変換係数の値をシグナリングするための技法について説明する。そのような技法は、ビットのバイナリ表現を一連の非バイナリ値の量子化された変換係数に変換する、エントロピー復号プロセスに関する技法を含む。より具体的には、本開示は、ゴロムライス/指数ゴロムの組合せとして代わりにコーディングされる、係数レベルの非コンテキストコーディングされた部分を表すエスケープ符号のコーディングのための技法について説明する。全般にエントロピー復号の逆の処理である、対応するエントロピー符号化プロセスも、本開示において説明される。

【0021】

本開示の技法は、High Efficiency Video Coding (HEVC)などの既存のビデオコーデ

10

20

30

40

50

ックの技法とともに実装されてもよく、または、現在開発中のVersatile Video Coding (VVC)規格などの未来のビデオコーデックの一部として含まれてもよい。

【0022】

図1は、本開示の技法を実行することができる例示的なビデオ符号化および復号システム100を示すブロック図である。本開示の技法は全般に、ビデオデータをコーディングする(符号化および/または復号)することを対象とする。一般に、ビデオデータはビデオを処理するためのあらゆるデータを含む。したがって、ビデオデータは、生のコーディングされていないビデオ、符号化されたビデオ、復号された(たとえば、再構築された)ビデオ、およびシグナリングデータなどのビデオメタデータを含み得る。

【0023】

図1に示されるように、この例では、システム100は、宛先デバイス116によって復号され表示されるべき、符号化されたビデオデータを提供するソースデバイス102を含む。具体的には、ソースデバイス102は、コンピュータ可読媒体110を介して宛先デバイス116にビデオデータを提供する。ソースデバイス102および宛先デバイス116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、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、ビデオゲーミングコンソール、ビデオストリーミングデバイスなどを含む、広範囲のデバイスのうちのいずれかであり得る。場合によっては、ソースデバイス102および宛先デバイス116は、ワイヤレス通信に対応し得るので、ワイヤレス通信デバイスと呼ばれ得る。

【0024】

図1の例では、ソースデバイス102は、ビデオソース104、メモリ106、ビデオエンコーダ200、および出力インターフェース108を含む。宛先デバイス116は、入力インターフェース122、ビデオデコーダ300、メモリ120、およびディスプレイデバイス118を含む。本開示によれば、ソースデバイス102のビデオエンコーダ200および宛先デバイス116のビデオデコーダ300は、本開示において説明される係数コーディングのための技法を適用するように構成され得る。したがって、ソースデバイス102はビデオ符号化デバイスの例を表し、一方で宛先デバイス116はビデオ復号デバイスの例を表す。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102は、外部カメラなどの外部ビデオソースからビデオデータを受信し得る。同様に、宛先デバイス116は、統合されたディスプレイデバイスを含むのではなく、外部ディスプレイデバイスとインターフェースしてもよい。

【0025】

図1に示されるようなシステム100は一例にすぎない。一般に、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が、本明細書において説明される係数コーディングのための技法を実行し得る。ソースデバイス102および宛先デバイス116は、ソースデバイス102が宛先デバイス116への送信のためにコーディングされたビデオデータを生成するような、コーディングデバイスの例にすぎない。本開示は、データのコーディング(符号化および/または復号)を実行するデバイスとして、「コーディング」デバイスに言及する。したがって、ビデオエンコーダ200およびビデオデコーダ300は、コーディングデバイス、具体的には、それぞれビデオエンコーダおよびビデオデコーダの例を表す。いくつかの例では、ソースデバイス102および宛先デバイス116は、ソースデバイス102および宛先デバイス116の各々がビデオ符号化および復号構成要素を含むように、実質的に対称的に動作し得る。したがって、システム100は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、またはビデオ電話のための、ソースデバイス102と宛先デバイス116との間での一方向または双方向のビデオ送信をサポートし得る。

【0026】

一般に、ビデオソース104は、ビデオデータのソース(すなわち、生の符号化されていないビデオデータ)を表し、ビデオデータの逐次的な一連のピクチャ(「フレーム」とも呼ばれる)をビデオエンコーダ200に提供し、ビデオエンコーダ200はピクチャのためのデータ

10

20

30

40

50

を符号化する。ソースデバイス102のビデオソース104は、ビデオカメラ、以前にキャプチャされた生のビデオを含むビデオアーカイブ、および/また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などの、ビデオキャプチャデバイスを含み得る。さらなる代替として、ビデオソース104は、ソースビデオとしてのコンピュータグラフィックススペースのデータ、またはライブビデオとアーカイブされたビデオとコンピュータで生成されたビデオとの組合せを生成し得る。各々の場合において、ビデオエンコーダ200は、キャプチャされた、事前にキャプチャされた、またはコンピュータで生成されたビデオデータを符号化する。ビデオエンコーダ200は、受信された順序(「表示順序」と呼ばれることがある)からコーディングのためのコーディング順序へと、ピクチャを並べ替えることができる。ビデオエンコーダ200は、符号化されたビデオデータを含むビットストリームを生成し得る。ソースデバイス102は次いで、たとえば宛先デバイス116の入力インターフェース122によって、受信および/または取り出しのために、出力インターフェース108を介してコンピュータ可読媒体110へと、符号化されたビデオデータを出力し得る。

【0027】

ソースデバイス102のメモリ106および宛先デバイス116のメモリ120は、汎用メモリを表す。いくつかの例では、メモリ106、120は、生のビデオデータ、たとえば、ビデオソース104からの生のビデオと、ビデオデコーダ300からの生の復号されたビデオデータとを記憶し得る。加えて、または代わりに、メモリ106、120は、たとえばビデオエンコーダ200およびビデオデコーダ300によってそれぞれ実行可能な、ソフトウェア命令を記憶し得る。メモリ106およびメモリ120は、この例ではビデオエンコーダ200およびビデオデコーダ300とは別々に示されているが、ビデオエンコーダ200およびビデオデコーダ300は、機能的に同様のまたは等価な目的で内部メモリも含み得ることを理解されたい。さらに、メモリ106、120は、たとえばビデオエンコーダ200から出力されビデオデコーダ300へと入力される、符号化されたビデオデータを記憶し得る。いくつかの例では、メモリ106、120の一部は、たとえば生の、復号された、および/または符号化されたビデオデータを記憶するために、1つまたは複数のビデオバッファとして割り振られ得る。

【0028】

コンピュータ可読媒体110は、ソースデバイス102から宛先デバイス116に符号化されたビデオデータを移すことが可能な任意のタイプの媒体またはデバイスを表し得る。一例では、コンピュータ可読媒体110は、たとえば、無線周波数ネットワークまたはコンピュータベースのネットワークを介して、ソースデバイス102がリアルタイムで宛先デバイス116に符号化されたビデオデータを直接送信することを可能にするための通信媒体を表す。出力インターフェース108は、符号化されたビデオデータを含む送信信号を変調することができ、入力インターフェース122は、ワイヤレス通信プロトコルなどの通信規格に従って、受信された送信信号を復調することができる。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、任意のワイヤレスまたは有線通信媒体を含み得る。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102から宛先デバイス116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。

【0029】

いくつかの例では、ソースデバイス102は、出力インターフェース108からストレージデバイス112に符号化されたデータを出力し得る。同様に、宛先デバイス116は、入力インターフェース122を介してストレージデバイス112からの符号化されたデータにアクセスし得る。ストレージデバイス112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メモリ、または符号化された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の、様々な分散されたまたはローカルでアクセスされるデータ記憶媒体のうちのいずれかを含み得る。

【0030】

いくつかの例では、ソースデバイス102は、ソースデバイス102によって生成された符号化されたビデオを記憶し得るファイルサーバ114または別の中間ストレージデバイスに、符号化されたビデオデータを出力し得る。宛先デバイス116は、ストリーミングまたはダウンロードを介して、ファイルサーバ114からの記憶されたビデオデータにアクセスし得る。ファイルサーバ114は、符号化されたビデオデータを記憶し、その符号化されたビデオデータを宛先デバイス116に送信することが可能な任意のタイプのサーバデバイスであり得る。ファイルサーバ114は、ウェブサーバ(たとえば、ウェブサイトのための)、ファイル転送プロトコル(FTP)サーバ、コンテンツ配信ネットワークデバイス、またはネットワーク接続ストレージ(NAS)デバイスを表し得る。宛先デバイス116は、インターネット接続を含む任意の標準的なデータ接続を通じて、ファイルサーバ114からの符号化されたビデオデータにアクセスし得る。これは、ワイヤレスチャネル(たとえば、Wi-Fi接続)、有線接続(たとえば、(デジタル加入者線(DSL)、ケーブルモデムなど)、またはファイルサーバ114上に記憶されている符号化されたビデオデータにアクセスするのに適した、両方の組合せを含み得る。ファイルサーバ114および入力インターフェース122は、ストリーミング送信プロトコル、ダウンロード送信プロトコル、またはこれらの組合せに従って動作するように構成され得る。

【0031】

出力インターフェース108および入力インターフェース122は、ワイヤレス送信機/受信機、モデム、有線ネットワークコンポーネント(たとえば、イーサネットカード)、種々のIEEE 802.11規格のいずれかに従って動作するワイヤレス通信コンポーネント、または他の物理コンポーネントを表し得る。出力インターフェース108および入力インターフェース122がワイヤレスコンポーネントを備える例では、出力インターフェース108および入力インターフェース122は、4G、4G-LTE(Long-Term Evolution)、LTE Advanced、5Gなどのセルラー通信規格に従って、符号化されたビデオデータなどのデータを転送するように構成され得る。出力インターフェース108がワイヤレス送信機を備えるいくつかの例では、出力インターフェース108および入力インターフェース122は、IEEE 802.11仕様、IEEE 802.15仕様(たとえば、ZigBee(商標))、Bluetooth(商標)規格などの他のワイヤレス規格に従った、符号化されたビデオデータなどのデータを転送するように構成され得る。いくつかの例では、ソースデバイス102および/または宛先デバイス116は、それぞれのシステムオンチップ(SoC)デバイスを含み得る。たとえば、ソースデバイス102は、ビデオエンコーダ200および/または出力インターフェース108に起因する機能を実行するためのSoCデバイスを含んでもよく、宛先デバイス116は、ビデオデコーダ300および/または入力インターフェース122に起因する機能を実行するためのSoCデバイスを含んでもよい。

【0032】

本開示の技法は、電波によるテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、dynamic adaptive streaming over HTTP (DASH)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ているデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の適用例などの、様々なマルチメディア適用例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。

【0033】

宛先デバイス116の入力インターフェース122は、コンピュータ可読媒体110(たとえば、通信媒体、記憶デバイス112、ファイルサーバ114など)から、符号化されたビデオビットストリームを受信する。符号化されたビデオビットストリームは、ビデオブロックまたは他のコーディングされたユニット(たとえば、スライス、ピクチャ、ピクチャグループ、シーケンスなど)の特性および/または処理を記述する値を有するシンタックス要素などの、ビデオデコーダ300によっても使用されるビデオエンコーダ200によって定義されるシグナリング情報を含み得る。ディスプレイデバイス118は、復号されたビデオデータの復号されたピクチャをユーザに表示する。ディスプレイデバイス118は、陰極線管(CRT)、

10

20

30

40

50

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のいずれかを表し得る。

【0034】

図1には示されないが、いくつかの例では、ビデオエンコーダ200およびビデオデコーダ300は各々、オーディオエンコーダおよび/またはオーディオデコーダと統合されることがあり、共通のデータストリームの中のオーディオとビデオの両方を含む多重化されたストリームを扱うために、適切なMUX-DEMUXユニット、または他のハードウェアおよび/もしくはソフトウェアを含むことがある。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。

10

【0035】

ビデオエンコーダ200およびビデオデコーダ300は各々、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、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などの、様々な適切なエンコーダおよび/またはデコーダ回路のいずれかとして実装され得る。本技法が部分的にソフトウェアで実装されるとき、デバイスは、適切な非一時的コンピュータ可読媒体にソフトウェアのための命令を記憶し、本開示の技法を実行するために1つまたは複数のプロセッサを使用してハードウェアでその命令を実行し得る。ビデオエンコーダ200およびビデオデコーダ300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、それらのいずれも、それぞれのデバイスの中で複合エンコーダ/デコーダ(コーデック)の一部として統合されてもよい。ビデオエンコーダ200および/またはビデオデコーダ300を含むデバイスは、集積回路、マイクロプロセッサ、および/または携帯電話などのワイヤレス通信デバイスを備え得る。

20

【0036】

ビデオエンコーダ200およびビデオデコーダ300は、High Efficiency Video Coding (HEVC)としても言及されるITU-T H.265などのビデオコーディング規格、または、マルチビューおよび/もしくはスケーラブルビデオコーディング拡張などのそれらの拡張に従って動作し得る。代わりに、ビデオエンコーダ200およびビデオデコーダ300は、Versatile Video Coding (VVC)とも呼ばれるJoint Exploration Test Model (JEM)またはITU-T H.266などの、他のプロプライエタリ規格または業界規格に従って動作し得る。VVC規格の最新のドラフトは、Bross他、「Versatile Video Coding (Draft 6)」、ITU-T SG 16 WP 3およびISO/IEC JTC 1/SC 29/WG 11のJoint Video Experts Team (JVET)、第15回会合、ヨーテボリ、スウェーデン、2019年7月3~12日、JVET-O2001-vE(以後「VVC Draft 6」)において説明される。しかしながら、本開示の技法は、いかなる特定のコーディング規格にも限定されない。

30

【0037】

一般に、ビデオエンコーダ200およびビデオデコーダ300は、ピクチャのブロックベースのコーディングを実行し得る。「ブロック」という用語は全般に、処理されるべきデータ(たとえば、符号化および/または復号プロセスにおいて符号化される、復号される、または別様に使用される)を含む構造を指す。たとえば、ブロックは、輝度および/または色度データのサンプルの2次元行列を含み得る。一般に、ビデオエンコーダ200およびビデオデコーダ300は、YUV(たとえば、Y、Cb、Cr)フォーマットで表されるビデオデータをコーディングし得る。すなわち、ピクチャのサンプルに対する赤、緑、および青(RGB)のデータをコーディングするのではなく、ビデオエンコーダ200およびビデオデコーダ300は、輝度成分および色度成分をコーディングしてもよく、色度成分は、赤の色調と青の色調の両方の色度成分を含んでもよい。いくつかの例では、ビデオエンコーダ200は、受信されたRGBフォーマットされたデータを符号化の前にYUV表現へと変換し、ビデオデコーダ300は、YUV表現をRGBフォーマットに変換する。代わりに、前処理ユニットおよび後

40

50

処理ユニット(図示せず)は、これらの変換を実行し得る。

【0038】

本開示は一般に、ピクチャのデータを符号化または復号するプロセスを含むものとして、ピクチャのコーディング(たとえば、符号化および復号)に言及し得る。同様に、本開示は、ブロックのためのデータを符号化または復号するプロセス、たとえば予測および/または残差コーディングを含むものとして、ピクチャのブロックのコーディングに言及し得る。符号化されたビデオビットストリームは一般に、コーディングの決定(たとえば、コーディングモード)およびブロックへのピクチャの区分を表す、シンタックス要素に対する一連の値を含む。したがって、ピクチャまたはブロックをコーディングすることへの言及は一般に、ピクチャまたはブロックを形成するシンタックス要素に対する値をコーディングすることとして理解されるべきである。

10

【0039】

HEVCは、コーディングユニット(CU)、予測ユニット(PU)、および変換ユニット(TU)を含む、様々なブロックを定義する。HEVCによれば、ビデオコーデ(ビデオエンコーダ200など)は、四分木構造に従ってコーディングツリーユニット(CTU)をCUへと区分する。すなわち、ビデオコーデは、CTUおよびCUを4つの等しい重複しない正方形へと区分し、四分木の各ノードは、0個または4個のいずれかの子ノードを有する。子ノードのないノードは「リーフノード」と呼ばれることがあり、そのようなリーフノードのCUは、1つまたは複数のPUおよび/または1つまたは複数のTUを含むことがある。ビデオコーデはさらにPUおよびTUを区分し得る。たとえば、HEVCでは、残差四分木(RQT)はTUの区分を表す。HEVCでは、PUはインター予測データを表すが、TUは残差データを表す。イントラ予測されるCUは、イントラモード指示などのイントラ予測情報を含む。

20

【0040】

別の例として、ビデオエンコーダ200およびビデオデコーダ300は、JEMまたはVVCに従って動作するように構成され得る。JEMまたはVVCによれば、ビデオコーデ(ビデオエンコーダ200など)は、複数のコーディングツリーユニット(CTU)へとピクチャを区分する。ビデオエンコーダ200は、四分木二分木(QTBT)構造またはマルチタイプ木(MTT)構造などの木構造に従って、CTUを区分し得る。QTBT構造は、HEVCのCUと、PUと、TUとの分離などの、複数の区分タイプという概念を取り扱う。QTBT構造は、四分木区分に従って区分される第1のレベル、および二分木区分に従って区分される第2のレベルという、2つのレベルを含む。QTBT構造のルートノードはCTUに対応する。二分木のリーフノードはコーディングユニット(CU)に対応する。

30

【0041】

MTT区分構造では、ブロックは四分木(QT)区分、二分木(BT)区分、および1つまたは複数のタイプの三分木(triple tree)(TT)(三分木(ternary tree)(TT)とも呼ばれる)区分を使用して区分され得る。三分木(triple tree)または三分木(ternary tree)区分は、ブロックが3つのサブブロックに分割される区分である。いくつかの例では、三分木(triple tree)または三分木(ternary tree)区分は、中心を通して元のブロックを分割することなく、ブロックを3つのサブブロックへと分割する。MTTにおける区分タイプ(たとえば、QT、BT、およびTT)は、対称的であっても、または非対称であってもよい。

40

【0042】

いくつかの例では、ビデオエンコーダ200およびビデオデコーダ300は、輝度成分および色度成分の各々を表すために単一のQTBT構造またはMTT構造を使用してもよく、他の例では、ビデオエンコーダ200およびビデオデコーダ300は、輝度成分のための1つのQTBT/MTT構造および両方の色度成分のための別のQTBT/MTT構造(またはそれぞれの色度成分のための2つのQTBT/MTT構造)などの、2つ以上のQTBTまたはMTT構造を使用し得る。

【0043】

ビデオエンコーダ200およびビデオデコーダ300は、HEVCに従った四分木区分、QTBT区分、MTT区分、または他の区分構造を使用するように構成され得る。説明を目的に、本

50

開示の技法の説明は、QTBT区分に関連して提示される。しかしながら、本開示の技法は、四分木区分、または他のタイプの区分も使用するように構成される、ビデオコードにも適用され得ることを理解されたい。

【0044】

ブロック(たとえば、CTUまたはCU)は、ピクチャにおいて様々な方法でグループ化され得る。一例として、ブリックは、ピクチャの中の特定のタイル内のCTU行の長方形領域を指し得る。タイルは、ピクチャにおける特定のタイル列および特定のタイル行の中のCTUの長方形領域であり得る。タイル列は、ピクチャの高さに等しい高さおよび(たとえば、ピクチャパラメータセットなどにおいて)シンタックス要素によって指定される幅を有する、CTUの長方形領域を指す。タイル行は、(たとえば、ピクチャパラメータセットなどにおいて)シンタックス要素によって指定される高さおよびピクチャの幅に等しい幅を有する、CTUの長方形領域を指す。

10

【0045】

いくつかの例では、タイルは複数のブリックへと区分されてもよく、ブリックの各々はタイル内の1つまたは複数のCTU行を含んでもよい。複数のブリックへと区分されないタイルは、ブリックとも呼ばれ得る。しかしながら、タイルの真のサブセットであるブリックは、タイルと呼ばれないことがある。

【0046】

ピクチャの中のブリックは、スライスにおいても並べられ得る。スライスは、単一のネットワーク抽象化レイヤ(NAL)ユニットに独占的に含まれ得る、整数個のピクチャのブリックであり得る。いくつかの例では、スライスは、ある数の完全なタイル、または、1つのタイルの完全なブリックの連続的なシーケンスのみの、いずれかを含む。

20

【0047】

本開示は、垂直方向および水平方向の寸法に関して、ブロック(CUまたは他のビデオブロックなど)のサンプル寸法を指すために、「 $N \times N$ 」および「 N 対 N 」を、たとえば 16×16 サンプルまたは16対16サンプルを交換可能に使用することがある。一般に、 16×16 CUは、垂直方向に16個のサンプル($y=16$)と水平方向に16個のサンプル($x=16$)とを有する。同様に、 $N \times N$ CUは、一般に、垂直方向に N 個のサンプルと水平方向に N 個のサンプルとを有し、ここで N は、非負の整数値を表す。CUの中のサンプルは、行および列として並べられ得る。その上、CUは、必ずしも水平方向に垂直方向と同じ数のサンプルを有する必要があるとは限らない。たとえば、CUは $N \times M$ サンプルを備えることがあり、ここで M は必ずしも N と等しいとは限らない。

30

【0048】

ビデオエンコーダ200は、予測情報および/または残差情報、ならびに他の情報を表す、CUのためのビデオデータを符号化する。予測情報は、CUのための予測ブロックを形成するために、CUがどのように予測されるべきかを示す。残差情報は一般に、符号化の前のCUのサンプルと予測ブロックとの間のサンプルごとの差を表す。

【0049】

CUを予測するために、ビデオエンコーダ200は一般に、インター予測またはイントラ予測を通じて、CUのための予測ブロックを形成し得る。インター予測は一般に、以前にコーディングされたピクチャのデータからCUを予測することを指し、一方、イントラ予測は一般に、同じピクチャの以前にコーディングされたデータからCUを予測することを指す。インター予測を実行するために、ビデオエンコーダ200は、1つまたは複数の動きベクトルを使用して予測ブロックを生成し得る。ビデオエンコーダ200は一般に、たとえば、CUと参照ブロックとの差に関して、CUとよく一致する参照ブロックを特定するために、動き探索を実行し得る。ビデオエンコーダ200は、sum of absolute difference (SAD)、sum of squared differences (SSD)、mean absolute difference (MAD)、mean squared differences (MSD)、または他のそのような差分計算を使用して差分メトリックを計算し、参照ブロックが現在のCUとよく一致するかどうかを決定し得る。いくつかの例では、ビデオエンコーダ200は、単方向予測または双方向予測を使用して現在のCUを予測し得る。

40

50

【0050】

JEMおよびVVCのいくつかの例は、アフィン動き補償モードも提供することができ、これはインター予測モードと見なされ得る。アフィン動き補償モードでは、ビデオエンコーダ200は、ズームインもしくはズームアウト、回転、射影運動、または他の不規則な運動タイプなどの、非並進運動を表す2つ以上の動きベクトルを決定し得る。

【0051】

イントラ予測を実行するために、ビデオエンコーダ200は、予測ブロックを生成するためにイントラ予測モードを選択し得る。JEMおよびVVCのいくつかの例は、様々な指向性モード、ならびに平面モードおよびDCモードを含む、67個のイントラ予測モードを提供する。一般に、ビデオエンコーダ200は、現在のブロック(たとえば、CUのブロック)のサンプルをそれから予測すべき、その現在のブロックに対する近隣のサンプルを記述する、イントラ予測モードを選択する。そのようなサンプルは一般に、ビデオエンコーダ200がラスタ走査順序で(左から右、上から下)CTUおよびCUをコーディングすると仮定して、現在のブロックと同じピクチャにおいて、現在のブロックの上、上および左、または左にあり得る。

【0052】

ビデオエンコーダ200は、現在のブロックのための予測モードを表すデータを符号化する。たとえば、インター予測モードでは、ビデオエンコーダ200は、様々な利用可能なインター予測モードのいずれが使用されるかを表すデータ、ならびに、対応するモードのための動き情報を符号化し得る。単方向または双方向のインター予測のために、たとえば、ビデオエンコーダ200は、高度動きベクトル予測(AMVP)モードまたはマージモードを使用して動きベクトルを符号化し得る。ビデオエンコーダ200は、同様のモードを使用して、アフィン動き補償モードのために動きベクトルを符号化し得る。

【0053】

ブロックのイントラ予測またはインター予測などの予測に従って、ビデオエンコーダ200は、ブロックのための残差データを計算し得る。残差ブロックなどの残差データは、対応する予測モードを使用して形成された、ブロックとブロックのための予測ブロックとの間のサンプルごとの差を表す。ビデオエンコーダ200は、サンプル領域の代わりに変換領域において変換されたデータを生成するために、1つまたは複数の変換を残差ブロックに適用し得る。たとえば、ビデオエンコーダ200は、離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または概念的に類似の変換を、残差ビデオデータに適用し得る。加えて、ビデオエンコーダ200は、モード依存非分離可能二次変換(MDNSST: mode-dependent non-separable secondary transform)、信号依存変換、カルーネンレーベ変換(KLT)などの二次的な変換を、最初の変換に続いて適用し得る。ビデオエンコーダ200は、1つまたは複数の変換の適用に続いて変換係数を生成する。

【0054】

上述のように、変換係数を生成するための任意の変換に続いて、ビデオエンコーダ200は、変換係数の量子化を実行し得る。量子化は一般に、変換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。量子化プロセスを実行することによって、ビデオエンコーダ200は、変換係数の一部またはすべてと関連付けられるビット深度を低減し得る。たとえば、ビデオエンコーダ200は、量子化の間に n ビット値を m ビット値に丸めることができ、 n は m より大きい。いくつかの例では、量子化を実行するために、ビデオエンコーダ200は、量子化されるべき値のビットごとの右シフトを実行し得る。

【0055】

量子化に続いて、ビデオエンコーダ200は、変換係数を走査し、量子化された変換係数を含む2次元行列から1次元ベクトルを生成し得る。走査は、より高いエネルギー(それゆえより低い周波数)の変換係数をベクトルの前方に置き、より低いエネルギー(それゆえより高い周波数)の変換係数をベクトルの後方に置くように設計され得る。いくつかの例では、ビデオエンコーダ200は、量子化された変換係数を走査するためにあらかじめ定められ

た走査順序を利用して直列化されたベクトルを生成し、次いで、ベクトルの量子化された変換係数をエントロピー符号化し得る。他の例では、ビデオエンコーダ200は、適応走査を実行し得る。1次元ベクトルを形成するために量子化された変換係数を走査した後、ビデオエンコーダ200は、たとえばコンテキスト適応バイナリ算術コーディング(CABAC)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200はまた、ビデオデータを復号する際にビデオデコーダ300によって使用するための符号化されたビデオデータと関連付けられるメタデータを記述するシンタックス要素に対する値をエントロピー符号化し得る。

【0056】

CABACを実行するために、ビデオエンコーダ200は、送信されるべきシンボルにコンテキストモデル内のコンテキストを割り当て得る。コンテキストは、たとえば、シンボルの隣接する値が0であるかどうかに関連し得る。確率決定は、シンボルに割り当てられたコンテキストに基づき得る。

【0057】

ビデオエンコーダ200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、およびシーケンス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、シーケンスパラメータセット(SPS)、ピクチャパラメータセット(PPS)、もしくはビデオパラメータセット(VPS)などの他のシンタックスデータにおいて、ビデオデコーダ300に対して生成し得る。ビデオデコーダ300は、対応するビデオデータをどのように復号するかを決定するために、そのようなシンタックスデータを同様に復号し得る。

【0058】

このようにして、ビデオエンコーダ200は、符号化されたビデオデータ、たとえば、ブロック(たとえば、CU)へのピクチャの区分ならびにブロックに対する予測および/または残差情報を記述するシンタックス要素を含む、ビットストリームを生成し得る。最終的に、ビデオデコーダ300は、ビットストリームを受信し、符号化されたビデオデータを復号し得る。

【0059】

一般に、ビデオデコーダ300は、ビットストリームの符号化されたビデオデータを復号するために、ビデオエンコーダ200によって実行されたプロセスと逆のプロセスを実行する。たとえば、ビデオデコーダ300は、ビデオエンコーダ200のCABAC符号化プロセスと逆ではあるが実質的に同様の方式で、CABACを使用してビットストリームのシンタックス要素に対する値を復号し得る。シンタックス要素は、CTUのCUを定義するために、CTUへのピクチャの区分情報と、QTBT構造などの対応する区分構造に従った各CTUの区分とを定義し得る。シンタックス要素はさらに、ビデオデータのブロック(たとえば、CU)に対する予測および残差情報を定義し得る。

【0060】

残差情報は、たとえば量子化された変換係数によって表され得る。ビデオデコーダ300は、ブロックの量子化された変換係数を逆量子化し逆変換して、ブロックに対する残差ブロックを再現し得る。ビデオデコーダ300は、シグナリングされた予測モード(イントラ予測またはインター予測)および関連する予測情報(たとえば、インター予測のための動き情報)を使用して、ブロックに対する予測ブロックを形成する。ビデオデコーダ300は次いで、予測ブロックと残差ブロックとを(サンプルごとに)組み合わせて、元のブロックを再現し得る。ビデオデコーダ300は、ブロックの境界に沿った視覚的なアーティファクトを減らすために、デブロック処理を実行することなどの、追加の処理を実行し得る。

【0061】

本開示は一般に、シンタックス要素などのある情報を「シグナリングすること」に言及することがある。「シグナリングすること」という用語は、全般に、シンタックス要素に対する値、および/または符号化された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。すなわち、ビデオエンコーダ200は、ビットストリームにお

10

20

30

40

50

いてシンタックス要素に対する値をシグナリングし得る。一般に、シグナリングは、ビットストリームにおいて値を生成することを指す。上で述べられたように、ソースデバイス102は、実質的にリアルタイムで、または非リアルタイムでビットストリームを宛先デバイス116に転送してもよく、これは、宛先デバイス116により後で取り出すためにシンタックス要素をストレージデバイス112に記憶するときなどに起こることがある。

【0062】

図2Aおよび図2Bは、例示的な四分木二分木(QTBT)構造130および対応するコーディングツリーユニット(CTU)132を示す概念図である。実線は四分木分割を表し、点線は二分木分割を示す。二分木の各分割(すなわち、非リーフ)ノードでは、どちらの分割タイプ(すなわち、水平または垂直)が使用されるかを示すために1つのフラグがシグナリングされ、この例では0が水平の分割を示し、1が垂直の分割を示す。四分木分割では、四分木ノードはブロックをサイズの等しい4つのサブブロックへと水平および垂直に分割するので、分割タイプを示す必要はない。したがって、QTBT構造130の領域木レベル(すなわち、実線)のためのシンタックス要素(分割情報など)およびQTBT構造130の予測木レベル(すなわち、破線)のためのシンタックス要素(分割情報など)を、ビデオエンコーダ200は符号化することができ、ビデオデコーダ300は復号することができる。QTBT構造130の末端リーフノードによって表されるCUのための、予測データおよび変換データなどのビデオデータを、ビデオエンコーダ200は符号化することができ、ビデオデコーダ300は復号することができる。

【0063】

一般に、図2BのCTU132は、第1および第2のレベルにおけるQTBT構造130のノードに対応するブロックのサイズを定義するパラメータと関連付けられ得る。これらのパラメータは、CTUサイズ(例ではCTU132のサイズを表す)、最小四分木サイズ(MinQTSize、最小の許容される四分木リーフノードサイズを表す)、最大二分木サイズ(MaxBTSize、最大の許容される二分木ルートノードサイズを表す)、最大二分木深度(MaxBTDepth、最大の許容される二分木深度を表す)、および最小二分木サイズ(MinBTSize、最小の許容される二分木リーフノードサイズを表す)を含み得る。

【0064】

CTUに対応するQTBT構造のルートノードは、QTBT構造の第1のレベルにおいて4つの子ノードを有してもよく、それらの各々が、四分木区分に従って区分されてもよい。すなわち、第1のレベルのノードは、リーフノード(子ノードを有しない)を有するか、または4つの子ノードを有するかのいずれかである。QTBT構造130の例は、親ノードと、分岐のための実線を有する子ノードとを含むものとして、そのようなノードを表す。第1のレベルのノードが最大の許容される二分木ルートノードサイズ(MaxBTSize)より大きくない場合、ノードはそれぞれの二分木によってさらに区分され得る。分割に起因するノードが最小の許容される二分木リーフノードサイズ(MinBTSize)または最大の許容される二分木深度(MaxBTDepth)に達するまで、1つのノードの二分木分割が繰り返され得る。QTBT構造130の例は、分岐のための破線を有するものとしてそのようなノードを表す。二分木リーフノードはコーディングユニット(CU)と呼ばれ、これは、さらなる区分なしで予測(たとえば、ピクチャ内またはピクチャ間予測)および変換のために使用される。上で論じられたように、CUは「ビデオブロック」または「ブロック」とも呼ばれ得る。

【0065】

QTBT区分構造の一例では、CTUサイズは 128×128 (ルーマサンプルおよび2つの対応する 64×64 クロマサンプル)として設定され、MinQTSizeは 16×16 として設定され、MaxBTSizeは 64×64 として設定され、MinBTSize(幅と高さの両方に対して)は4として設定され、MaxBTDepthは4として設定される。四分木リーフノードを生成するために、四分木区分がまずCTUに適用される。四分木リーフノードは、 16×16 (すなわち、MinQTSize)から 128×128 (すなわち、CTUサイズ)までのサイズを有し得る。リーフ四分木ノードが 128×128 である場合、それは二分木によってさらに分割されず、それはサイズがMaxBTSize(すなわち、この例では 64×64)を超えるからである。それ以外の場合、リーフ四分木ノ

ードはさらに二分木によって区分される。したがって、四分木リーフノードは二分木のルートノードでもあり、0という二分木深度を有する。二分木深度がMaxBTDepth(この例では4)に達するとき、さらなる分割は許可されない。二分木ノードの幅がMinBTSize(この例では4)に等しいとき、それはさらなる水平の分割が許可されないことを示唆する。同様に、MinBTSizeに等しい高さを有する二分木ノードは、さらなる垂直の分割がその二分木ノードに対して許可されないことを示唆する。上で述べられたように、二分木のリーフノードはCUと呼ばれ、さらなる区分なしで予測および変換に従ってさらに処理される。

【0066】

Trellis coded quantization (TCQ)が、H.Schwarz、T.Nguyen、D.Marpe、T.Wiegand、M.Karczewicz、M.Coban、J.Dong、「CE7: Transform coefficient coding with reduced number of regular-coded bins (tests 7.1.3a, 7.1.3b)」、JVET document JVET-L0274、マカオ、中国、2018年10月(以後JVET-L0274)において提案された。JVET-L0274の技法では、2つのスカラー量子化器が、量子化/逆量子化のために切り替え可能に使用される。現在の変換/量子化された係数に対して使用されるスカラー量子化器は、走査順序において現在の変換/量子化された係数に先行する量子化された係数のパリティ(最下位ビット)によって決定される。

【0067】

TCQと組み合わせられる変換係数コーディング方式もJVET-L0274において提案され、その方式では、量子化された係数を復号するためのコンテキスト選択は、使用される量子化器に依存する。具体的には、変換係数が0であるか0ではないかを示す変換係数の有意性フラグ(SIG)は、コンテキストモデルの3つのセットのうちの1つを用いてコーディングされてもよく、特定のSIGに対して選択されるセットは、関連する係数のために使用される量子化器に依存する。したがって、現在の係数のSIGの復号を開始するとき、エントロピーデコーダ(たとえば、ビデオデコーダ300のエントロピーデコーダ)は、以前の走査位置において変換係数のパリティを決定し、これは、現在の係数に対する量子化器を決定し、したがって、その係数のSIGに対するコンテキストセットを決定する。

【0068】

いくつかの例では、TUは、コーディンググループ(CG)と呼ばれる重複しないサブブロックへと分割され、そのサイズは通常4×4サンプルである。本明細書において説明される復号プロセスは、時々4×4 CGに関して説明されることがあるが、任意の他のCGサイズに拡張され得る。本開示の技法、およびしたがって、本明細書に含まれる説明は主に、CGにおける変換係数の絶対レベルのための符号化プロセスおよび復号プロセスに関する。符号などのCGと関連付けられる他の情報が、JVET-L0274において説明される方式で符号化または復号されることがあるが、代替の技法を使用して符号化され復号されることもある。

【0069】

ビデオエンコーダ200およびビデオデコーダ300は、ビットストリームの中のシンタックス要素を処理するように構成され得る。たとえば、変換係数に対する絶対レベル値(absLevel)を表すために、次のシンタックス要素が使用され得る。量子化された変換係数は通常、整数値を有することに留意されたい。絶対レベルは、変換係数の大きさである。

● sig_coeff_flag(SIG): absLevelが0である場合、このフラグは0に等しく、それ以外の場合、このフラグは1に等しい。

● abs_level_gt1_flag: sig_coeff_flagが1に等しい場合、このフラグはビットストリームに存在する。absLevelが1より大きい場合、abs_level_gt1_flagの値は1に等しく、それ以外の場合、abs_level_gt1_flagは0に等しい。

● par_level_flag: abs_level_gt1_flagが1に等しい場合、このフラグはビットストリームに存在する。absLevelが奇数である場合、par_level_flagの値は0に等しく、absLevelが偶数である場合、1に等しい。

● abs_level_gt3_flag: abs_level_gt1_flagが1に等しい場合、このフラグはビットストリームに存在する。absLevelが3より大きい場合、abs_level_gt3_flagの値は1に等しく、それ以外の場合、abs_level_gt3_flagは0に等しい。

10

20

30

40

50

- **abs_remainder**: **abs_level_gt3_flag**が1に等しい場合、このシンタックス要素はビットストリームにおいて存在する。**abs_remainder**シンタックス要素は、ゴロムライス符号を用いてコーディングされる変換係数レベルの残りの絶対値である。
- **abs_level**: これは、ゴロムライス符号を用いてコーディングされる変換係数レベルの残りの絶対値である。

【0070】

以下で、説明を簡単にするために、シンタックス要素**sig_coeff_flag**、**par_level_flag**、**abs_level_gt1_flag**、**abs_level_gt3_flag**、**abs_remainder**、および**abs_level**は、それぞれ、SIG、Par、Gt1、Gt2、remLevel、absLevelと表記される。

【0071】

ビデオエンコーダ200およびビデオデコーダ300は、ビットストリームから構文解析されない上記のシンタックス要素のいずれをも、0などのデフォルト値に設定するように構成され得る。5つのシンタックス要素のうちの最初の値が与えられると、変換係数の絶対レベルに対する値は、次のように計算され得る。

$$\text{absoluteLevel} = \text{SIG} + \text{Gt1} + \text{Par} + (\text{Gt2} \ll 1) + (\text{remLevel} \ll 1) \quad (1)$$

【0072】

代替的に、変換係数がバイパスコーディングモードにおいて完全にコーディングされる場合、**absoluteLevel**は**abs_level**として直接コーディングされ得る。

【0073】

図3は、JVET-L0274におけるような、CGにおける**absoluteLevels**を表すシンタックス要素の例示的な順序を示す。他の順序も使用され得る。図3に見られるように、上で説明されたすべての5つのシンタックス要素は、**absLevel**が4より大きいとき、ビットストリームから構文解析される。

【0074】

図3の例では、ビデオデコーダ300は、最高で4つのパスで、CGの中の位置を走査する。最初のパス136において、ビデオデコーダ300は、SIG、Par、およびGt1シンタックス要素に対する値を構文解析する。対応するGt1およびParシンタックス要素の前には、0ではないSIGのみがある。すなわち、SIGが0という値を有することをビデオデコーダ300が決定する場合、これは、変換係数レベルが0に等しいことを意味し、ビデオデコーダ300は、その係数に対するGt1およびParシンタックス要素のインスタンスを受信しない。第1のパス136の後で、各位置に対する、**absLevel1**と表記される部分的な**absoluteLevel**の値は、式(2)に示されるように再構築され得る。

$$\text{absLevel1} = \text{SIG} + \text{Par} + \text{Gt1} \quad (2)$$

【0075】

いくつかの実装形態では、ビデオデコーダ300は、4×4サブブロックに対して、第1のパス136において最大で28個の普通のコーディングされたピンを構文解析し、2×2サブブロックに対して、最大で6個の普通のコーディングされたピンを構文解析するように構成され得る。普通のコーディングされたピンの数に対する制約は、SIG、Gt1、およびParピンのグループにおいて実施されてもよく、これは、SIG、Gt1、およびParピンの各グループがセットとしてコーディングされ、セットの途中でのバイパスコーディングへの切り替えが許容されることを意味する。すなわち、いくつかの実装形態では、SIG、Gt1、およびParセットの中の1つ1つのシンタックス要素が、普通のコーディングされたピンとしてコーディングされ(たとえば、コンテキストコーディングされ)、またはバイパスモードでコーディングされる。

【0076】

第1のパス136において少なくとも1つの0ではないGt1がある場合、ビデオデコーダ300は、第2のパス138を走査するように構成され得る。第2のパス138において、ビデオデコーダ300は、0ではないGt1シンタックス要素を伴うCGの中の位置に対するGt2シンタックス要素を構文解析する。CGの中の位置は、変換係数と関連付けられる。第1のパス136および第2のパス138におけるピンは、すべて普通にコーディングされてもよく、これは

10

20

30

40

50

、ビンの確率分布が選択されたコンテキストモデルによってモデル化されることを意味する。第2のパス138において少なくとも1つの0ではないGt2シンタックス要素がある場合、ビデオデコーダ300は第3のパス140を走査する。第3のパス140の間、ビデオデコーダ300は、0ではないGt2を伴う位置のremLevelシンタックス要素を構文解析する。remLevelシンタックス要素は二値ではなく、ビデオデコーダ300はremLevelの二値化されたバージョンのピンをバイパスコーディングしてもよく、これは、ピンが均一に分布していると想定され、コンテキスト選択が不要であることを意味する。

【0077】

第4のパス142において、ビデオデコーダ300は、以前の3つのパスにおいて普通のコーディングされたピンを用いて部分的に表されていない、すべての残りの係数を走査する。第4のパス142の変換係数レベルは、バイパスコーディングされたピンを使用して絶対値としてコーディングされる。

【0078】

ビデオエンコーダ200およびビデオデコーダ300は、上で説明されたシンタックス要素の一部をエントロピーコーディングするために、コンテキストモデリングを実行し得る。JVET-L0274において使用されるコンテキストモデリングが、本開示によって提案される修正とともに、ここでも簡単に紹介される。以下でより詳細に論じられるコンテキストモデリングは一般に、符号化または復号されているピンのための、コンテキストとも呼ばれる確率モデルの選択を指す。JVET-L0274において、シンタックス要素SIG、Par、Gt1、およびGt2は、コンテキストモデリングを使用してコーディングされる。コンテキストの選択は、Nと表記される、近傍におけるabsLevel1シンタックス要素の値に依存する。テンプレートの内部にあるが現在のTUの外部にある位置は、Nから除外され得る。

【0079】

図4は、確率モデルを選択するために使用されるテンプレートの図を示す。「X」でマークされた正方形は現在の走査位置を指定し、「Y」でマークされた正方形は使用される近傍を表す。

【0080】

現在の位置(図4のXを伴う正方形を参照)に対して、ビデオデコーダ300は、ctxIdxSIG、ctxIdxPar、ctxIdxGt1、およびctxIdxGt2と表記される、SIG、Par、Gt1、およびGt2シンタックス要素のコンテキストインデックスを決定する。コンテキストインデックスを決定するために、ビデオデコーダ300は、3つの変数、すなわちnumSIG、sumAbs1、およびdをまず決定し得る。変数numSIGは、Nの中の0ではないSIGの数を表し、これは以下の式(3)によって表される。

$$\text{sumSIG} = \sum_{i \in N} \text{SIG}(i) \quad (3)$$

【0081】

変数sumAbs1はNの中のabsLevel1の合計を表し、これは以下の式(4)によって表される。

$$\text{sumAbs1} = \sum_{i \in N} \text{absLevel1}(i) \quad (4)$$

【0082】

変数dは、以下の式(5)によって表されるような、TUの内部の現在の位置の対角方向の尺度を表す。

$$d = x + y \quad (5)$$

ここで、xおよびyは、TUの内部の現在の位置の座標を表す。

【0083】

sumAbs1およびdが与えられると、ビデオデコーダ300は、次のようにSIGを復号するためのコンテキストインデックスを決定する。

● ルマに対して、ctxIdxSIGは式(6)によって決定される。

$$\text{ctxIdxSIG} = 18 * \max(0, \text{state} - 1) + \min(\text{sumAbs1}, 5) + (d < 2 ? 12 : (d < 5 ? 6 : 0)) \quad (6)$$

● クロマに対して、ctxIdxSIGは式(7)によって決定される。

10

20

30

40

50

$ctxIdxSIG = 12 * \max(0, state - 1) + \min(\text{sumAbs1}, 5) + (d < 2 ? 6 : 0) \quad (7)$
 【0084】

式(6)および(7)において、変数「state」は、JVET-L0274において定義されるような、状態機械の現在の状態を表す。状態機械は以下でより詳しく論じられる。

【0085】

sumSIG、sumAbs1、およびdが与えられると、ビデオデコーダ300は、次のようにParを復号するためのコンテキストインデックスを決定する。

● 現在の走査位置が最後の0ではない係数の位置に等しい場合、ctxIdxParは0である。

● それ以外の場合、

○ ルマに対して、ctxIdxParは式(8)によって決定される。

$ctxIdxPar = 1 + \min(\text{sumAbs1} - \text{numSIG}, 4) + (d == 0 ? 15 : (d < 3 ? 10 : (d < 10 ? 5 : 0))) \quad (8)$

○ クロマに対して、ctxIdxParは(9)によって決定される。

$ctxIdxPar = 1 + \min(\text{sumAbs1} - \text{numSIG}, 4) + (d == 0 ? 5 : 0) \quad (9)$

ctxIdxGt1およびctxIdxGt2は、ctxIdxParの値に設定される。

【0086】

ビデオエンコーダ200およびビデオデコーダ300は、RemLevelコーディングを実行するように構成され得る。ビデオデコーダ300は、非バイナリシンタックス要素remRemainder(remLevel)およびabsLevelを次のようにコーディングするためのライスパラメータ(ricePar)を導出する。

● 各サブブロックの最初において、riceParは0に等しく設定される。

● 残りに対するシンタックス要素をコーディングした後で、ライスパラメータ(ricePar)は次のように修正される。

riceParが3より小さく、残りの最後のコーディングされた値が $((3 < \text{ricePar}) - 1)$ より大きい場合、riceParは1だけインクリメントされる。

【0087】

完全にバイパスコーディングされる絶対量子化インデックスを表す、非バイナリシンタックス要素absLevelをコーディングするために、次のことが当てはまる。

● ローカルテンプレートの中の絶対値の合計subAbsが決定される。

● 変数riceParおよびposZeroが、

$\text{ricePar} = \text{riceParTable}[\min(31, \text{sumAbs})]$

$\text{posZero} = \text{posZeroTable}[\max(0, state - 1)][\min(31, \text{sumAbs})]$

に従ってテーブルルックアップによって決定され、変数状態は、従属量子化のための状態(従属量子化が無効であるとき、0に等しい)を表し、テーブルriceParTable[]およびposZeroTable[][]は、以下により与えられる。

$\text{riceParTable}[32] = \{$

0,0,0,0,0,0,0,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3

$\};$

$\text{posZeroTable}[3][32] = \{$

{0,0,0,0,0,1,2,2,2,2,2,2,4,4,4,4,4,4,4,4,4,4,8,8,8,8,8,8,8,8,8,8},

{1,1,1,1,2,3,4,4,4,6,6,6,8,8,8,8,8,8,12,12,12,12,12,12,12,12,12,16,16,16,16,16,16},

{1,1,2,2,2,3,4,4,4,6,6,6,8,8,8,8,8,8,12,12,12,12,12,12,12,12,12,16,16,16,16,16,16},

$\};$

● 中間変数codeValueは、次のように導出される。

○ absLevelが0に等しい場合、codeValueはposZeroに等しく設定される。

○ それ以外の場合、absLevelがposZero以下である場合、codeValueはabsLevel-1に等しく設定される。

○ それ以外の場合(absLevelがposZeroより大きい)、codeValueはabsLevelに等しく

10

20

30

40

50

設定される。

● codeValueの値は、ライスパラメータriceParを用いてゴロムライス符号を使用してコーディングされる。

【0088】

ビデオエンコーダ200およびビデオデコーダ300は、残りのレベルコーディングのためにゴロムライスコーディングと指数ゴロムコーディングの組合せを実行するように構成され得る。すなわち、コンテキストコーディングされない変換係数レベルの部分は、ゴロムライスおよび指数ゴロム符号の組合せによって表され得る。変換係数の非コンテキストコーディングされた部分は、上で論じられたremLevelまたはabsLevelに対する値に対応し得る。

【0089】

係数に対するレベル値をコーディングするとき、非コンテキストコーディングされるレベル値が比較的小さいことが一般的に予想される。したがって、本開示はゴロムライスコーディングを使用することを提案し、これは、小さい値をコーディングするための効率的なコーディング技法であり得る。変換係数に対するレベル値が小さくない事例では、指数ゴロムコーディングが使用されてもよく、これは、より大きい値をコーディングするための効率的なコーディング技法であり得る。

【0090】

残りの係数値に対して、ゴロムライスコーディングから指数ゴロムコーディングに移行するための切替点は、以下の閾値に基づいて決定され得る。

threshold = g_auiGoRiceRange[m_goRicePar] << m_goRicePar

ここで、m_goRiceParはライスパラメータであり、

g_auiGoRiceRange[MAX_GR_ORDER_RESIDUAL] = {6, 5, 6, 3}

は、各ライスパラメータに対するプレフィックス長(すなわち、長さ-1)を決定するルックアップテーブルである。閾値以下の係数値は、ゴロムライス符号としてコーディングされ得るが、閾値より大きい係数値は、ゴロムライス符号と指数ゴロム符号の両方を使用してコーディングされ得る。

【0091】

ビデオデコーダ300は、absoluteLevelの再構築を実行するように構成されてもよく、これは、様々なシンタックス要素に基づいてレベル値を決定するプロセスを全般に指す。absoluteLevelの再構築はJVET-L0274におけるものと同じであり、これは、ビットストリームの中のシンタックス要素に関して上で論じられた。

【0092】

本開示は、残りのレベルコーディングのための技法について説明する。本開示の技法によれば、ビデオエンコーダ200およびビデオデコーダ300は、ゴロムライス符号の使用と指数ゴロム符号の使用との間の遷移点を決定するための、すべてのライスパラメータに対するRiceRangeにより定義される固定されたプレフィックス長を使用するように構成され得る。一例では、閾値は次のようである。

threshold = RiceRange << m_goRicePar

【0093】

RiceRangeに対する5または6という値は、現在のVVC設計におけるコーディング性能に関して良好な結果を与え得る。本開示の技法は、たとえば、コーディング効率を不当に損なうことなく、ルックアップテーブルを使用する必要をなくし得る。

【0094】

いくつかの例では、異なるRiceRangeが、変換係数のremLevel部分およびabsLevel部分のコーディングのために使用されてもよく、それは、remLevelが残りのレベルの半分を表す一方で、absLevelが完全なレベル(より大きい値)を表すからである。

【0095】

他の例では、ビデオエンコーダ200およびビデオデコーダ300は、最大のライスパラメータを除くすべてのライスパラメータに対して、5または6などの一定のRiceRangeパラメ

10

20

30

40

50

ータを使用するように構成されてもよく、最大のライスパラメータは、3などのより小さいRiceRangeパラメータを有し得る。

【0096】

本開示の技法は、提案される最悪の場合のエスケープ符号長の制限を含む。係数の残りのレベルに対する現在のゴロムライス/指数ゴロム表現は、長さが32ビットを超え得る。たとえば、riceParam=0という最悪の場合に対して、remLevelコードは、16381という最大値を有することがあり、これは33ビットで表される。absLevelに対して、 2^{15} という最大値は35ビットで表される。本開示は、最悪の場合の符号長を32に制限するための技法を提案する。

【0097】

最大のプレフィックス符号長は、以下により与えられる。

$\text{maximumPrefixLength} = 32 - (\text{RiceRange} + \text{MAX_TR_DYNAMIC_RANGE})$

このプレフィックス長に達すると、対応するサフィックス長が以下により与えられる。

$\text{suffixLength} = \text{MAX_TR_DYNAMIC_RANGE} - \text{rParam}$

この方式は、abs_remainder(remLevel)およびdec_abs_level(absLevel)シンタックス要素のコーディングのために、最悪の場合のエスケープ符号長を32ビットに制限する。

【0098】

以下のテーブルは、提案される二値化cRiceParam = 0を示す。

10

20

30

40

50

【表 1】

入力値	符号語プレフィックス	符号語サフィックス	プレフィックス符号長	サフィックス符号長	全体の符号長
0	0		1	0	1
1	10		2	0	2
2	110		3	0	3
3	1110		4	0	4
4	11110		5	0	5
5	111110		6	0	6
[6, 7]	1111110	x	7	1	8
[8, 11]	11111110	xx	8	2	10
[12, 19]	111111110	xxx	9	3	12
[20, 35]	1111111110	xxxx	10	4	14
[36, 67]	11111111110	xxxxx	11	5	16
[68, 131]	111111111110	xxxxxx	12	6	18
[132, 259]	1111111111110	xxxxxxx	13	7	20
[260, 515]	11111111111110	xxxxxxxx	14	8	22
[516, 1027]	111111111111110	xxxxxxxxx	15	9	24
[1028, 2051]	1111111111111110	xxxxxxxxxx	16	10	26
[2052, 4099]	11111111111111110	xxxxxxxxxxx	17	11	28
[4100, 32768]	1111111111111111	xxx… 15 times	17	15	32

【0099】

上の表において、入力値5は、ゴロムライスコーディングと指数ゴロムコーディングとの切替点を表す。入力値に対する符号語は、11111に0が付加されたゴロムライス符号として見られてもよく、これは5+0に相当する。次の入力6は、11111に10xが付加されたゴロムライス符号として見られてもよく、これは5+1=6に相当する。残りの入力値は、11111と指数ゴロム値を足したゴロムライス符号として同様に見られてもよい。

【0100】

以下の表は全般に、上と同じであるが提案される二値化がcRiceParam = 1である、コーディング方式を示す。

10

20

30

40

50

【表 2】

入力値	符号語プレフィックス	符号語サフィックス	プレフィックス符号長	サフィックス符号長	全体の符号長
[0, 1]	0	x	1	1	2
[2, 3]	10	x	2	1	3
[4, 5]	110	x	3	1	4
[6, 7]	1110	x	4	1	5
[8, 9]	11110	x	5	1	6
[10, 11]	111110	x	6	1	7
[12, 15]	1111110	xx	7	2	9
[16, 23]	11111110	xxx	8	3	11
[24, 39]	111111110	xxxx	9	4	13
[40, 71]	1111111110	xxxxx	10	5	15
[72, 135]	11111111110	xxxxxx	11	6	17
[136, 263]	111111111110	xxxxxxx	12	7	19
[264, 519]	1111111111110	xxxxxxxx	13	8	21
[520, 1031]	11111111111110	xxxxxxxxx	14	9	23
[1032, 2055]	111111111111110	xxxxxxxxxx	15	10	25
[2056, 4103]	1111111111111110	xxxxxxxxxxx	16	11	27
[4014, 8199]	11111111111111110	xxxxxxxxxxxx	17	12	29
[8200, 32768]	1111111111111111	xxx... 15 times	17	15	32

【0 1 0 1】

以下の表は全般に、上と同じであるが提案される二値化がcRiceParam = 2である、コーディング方式を示す。

10

20

30

40

50

【表 3】

入力値	符号語プレフィックス	符号語サフィックス	プレフィックス符号長	サフィックス符号長	全体の符号長
[0, 3]	0	xx	1	2	3
[4, 7]	10	xx	2	2	4
[8, 11]	110	xx	3	2	5
[12, 15]	1110	xx	4	2	6
[16, 19]	11110	xx	5	2	7
[20, 23]	111110	xx	6	2	8
[24, 31]	1111110	xxx	7	3	10
[32, 47]	11111110	xxxx	8	4	12
[48, 79]	111111110	xxxxx	9	5	14
[80, 143]	1111111110	xxxxxx	10	6	16
[144, 271]	11111111110	xxxxxxx	11	7	18
[272, 527]	111111111110	xxxxxxxx	12	8	20
[528, 1039]	1111111111110	xxxxxxxxx	13	9	22
[1040, 2063]	11111111111110	xxxxxxxxxx	14	10	24
[2064, 4111]	111111111111110	xxxxxxxxxxx	15	11	26
[4112, 8207]	1111111111111110	xxxxxxxxxxxx	16	12	28
[8028, 16399]	11111111111111110	xxxxxxxxxxxxx	17	13	30
[16400, 32768]	1111111111111111	xxx … 15 times	17	15	32

【0102】

以下の表は全般に、上と同じであるが提案される二値化がcRiceParam = 3である、コーディング方式を示す。

10

20

30

40

50

【表 4】

入力値	符号語プレフィックス	符号語サフィックス	プレフィックス符号長	サフィックス符号長	全体の符号長
[0, 7]	0	xxx	1	3	4
[8, 15]	10	xxx	2	3	5
[16, 23]	110	xxx	3	3	6
[24, 31]	1110	xxx	4	3	7
[32, 39]	11110	xxx	5	3	8
[40, 47]	111110	xxx	6	3	9
[48, 63]	1111110	xxxx	7	4	11
[64, 95]	11111110	xxxxx	8	5	13
[96, 159]	111111110	xxxxxx	9	6	15
[160, 287]	1111111110	xxxxxxx	10	7	17
[288, 543]	11111111110	xxxxxxxx	11	8	19
[544, 1055]	111111111110	xxxxxxxxx	12	9	21
[1056, 2079]	1111111111110	xxxxxxxxxx	13	10	23
[2080, 4127]	11111111111110	xxxxxxxxxxx	14	11	25
[4128, 8223]	111111111111110	xxxxxxxxxxxx	15	12	27
[8224, 16415]	1111111111111110	xxxxxxxxxxxxx	16	13	29
[16416, 32768]	11111111111111110	xxxxxxxxxxxxxx	17	14	31

【0103】

図5は、本開示の技法を実行することができる例示的なビデオエンコーダ200を示すブロック図である。図5は、説明のために提供され、本開示において広く例示および説明されるような技法の限定と見なされるべきでない。説明を目的に、本開示は、HEVCビデオコーディング規格および開発中のH.266ビデオコーディング規格などの、ビデオコーディング規格の文脈においてビデオエンコーダ200について説明する。しかしながら、本開示の技法は、これらのビデオコーディング規格に限定されず、ビデオ符号化および復号に全般に適用可能である。

【0104】

図5の例では、ビデオエンコーダ200は、ビデオデータメモリ230、モード選択ユニット202、残差生成ユニット204、変換処理ユニット206、量子化ユニット208、逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、フィルタユニット216、復号ピクチャバッファ(DPB)218、およびエントロピー符号化ユニット220を含む。

【0105】

ビデオデータメモリ230は、ビデオエンコーダ200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。ビデオエンコーダ200は、ビデオデータメモリ230に記憶されるビデオデータを、たとえば、ビデオソース104(図1)から受信し得る。DPB218は、ビデオエンコーダ200による後続のビデオデータの予測において使用するための参照ビデオデータを記憶する、参照ピクチャメモリとして作動し得る。ビデオデータメモリ230およびDPB218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)、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。ビデオデータメモリ230およびDPB218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提供され得る。様々な例では、ビデオデータメモリ230は、示されるように、ビデオエンコーダ200の他の構成要素とともにオンチップであることがあり、またはそれらの構成要素に対してオフチップであることがある。

10

【0106】

本開示では、ビデオデータメモリ230への言及は、そのように別段記述されない限り、ビデオエンコーダ200の内部のメモリに限定されるものとして解釈されるべきではなく、または、そのように別段記述されない限り、ビデオエンコーダ200の外部のメモリに限定されるものとして解釈されるべきではない。むしろ、ビデオデータメモリ230への言及は、ビデオエンコーダ200が符号化のために受信するビデオデータ(たとえば、符号化されるべき現在のブロックに対するビデオデータ)を記憶する参照メモリとして理解されるべきである。図1のメモリ106はまた、ビデオエンコーダ200の様々なユニットからの出力の一時的な記憶を行い得る。

20

【0107】

図5の様々なユニットは、ビデオエンコーダ200によって実行される動作を理解するのを助けるために示されている。ユニットは、固定機能の回路、プログラム可能回路、またはこれらの組合せとして実装され得る。固定機能の回路は、特定の機能を提供する回路を指し、実行され得る動作があらかじめ設定される。プログラム可能回路は、様々なタスクを実行して、実行され得る動作においてフレキシブルな機能を提供するようにプログラムされ得る、回路を指す。たとえば、プログラム可能回路は、ソフトウェアまたはファームウェアの命令によって定義される方式で、プログラム可能回路を動作させるソフトウェアまたはファームウェアを実行し得る。固定機能の回路は、(たとえば、パラメータを受信するために、またはパラメータを出力するために)ソフトウェア命令を実行し得るが、固定機能の回路が実行する動作のタイプは、一般に不変である。いくつかの例では、ユニットの1つまたは複数は、別個の回路ブロック(固定機能またはプログラム可能)であってもよく、いくつかの例では、1つまたは複数のユニットは集積回路であってもよい。

30

【0108】

ビデオエンコーダ200は、算術論理ユニット(ALU)、基本機能ユニット(EFU)、デジタル回路、アナログ回路、および/または、プログラム可能回路から形成されるプログラム可能コアを含み得る。プログラム可能回路によって実行されるソフトウェアを使用してビデオエンコーダ200の動作が実行される例では、メモリ106(図1)が、ビデオエンコーダ200が受信して実行するソフトウェアのオブジェクトコードを記憶してもよく、または、ビデオエンコーダ200内の別のメモリ(図示せず)が、そのような命令を記憶してもよい。

40

【0109】

ビデオデータメモリ230は、受信されたビデオデータを記憶するように構成される。ビデオエンコーダ200は、ビデオデータメモリ230からビデオデータのピクチャを取り出し、ビデオデータを残差生成ユニット204およびモード選択ユニット202に提供し得る。ビデオデータメモリ230の中のビデオデータは、符号化されるべき生のビデオデータであり得る。

【0110】

モード選択ユニット202は、動き推定ユニット222と、動き補償ユニット224と、イントラ予測ユニット226とを含む。モード選択ユニット202は、他の予測モードに従ってビデオ予測を実行するための追加の機能ユニットを含み得る。例として、モード選択ユニッ

50

ト202は、パレットユニット、イントラブロックコピーユニット(これは、動き推定ユニット222および/または動き補償ユニット224の一部であり得る)、アフィンユニット、線形モデル(LM)ユニットなどを含み得る。

【0111】

モード選択ユニット202は全般に、複数の符号化パスを協調させて、符号化パラメータの組合せと、そのような組合せに対して得られるレート歪み値を試験する。符号化パラメータは、CUへのCTUの区分、CUに対する予測モード、CUの残差データに対する変換タイプ、CUの残差データに対する量子化パラメータなどを含み得る。モード選択ユニット202は最終的に、他の試験された組合せより良いレート歪み値を有する符号化パラメータの組合せを選択し得る。

10

【0112】

ビデオエンコーダ200は、ビデオデータメモリ230から取り出されたピクチャを一連のCTUへと区分し、スライス内に1つまたは複数のCTUをカプセル化し得る。モード選択ユニット202は、QTBT構造または上で説明されたHEVCの四分木構造などの、木構造に従ってピクチャのCTUを区分し得る。上で説明されたように、ビデオエンコーダ200は、木構造に従ってCTUを区分することから1つまたは複数のCUを形成し得る。そのようなCUは、一般に「ビデオブロック」または「ブロック」とも呼ばれ得る。

【0113】

一般に、モード選択ユニット202はまた、現在のブロック(たとえば、現在のCU、またはHEVCでは、PUとTUの重複部分)に対する予測ブロックを生成するように、その構成要素(たとえば、動き推定ユニット222、動き補償ユニット224、およびイントラ予測ユニット226)を制御する。現在のブロックのインター予測のために、動き推定ユニット222は、1つまたは複数の参照ピクチャ(たとえば、DPB218に記憶されている1つまたは複数の以前にコーディングされたピクチャ)の中の1つまたは複数のよく一致する参照ブロックを特定するために、動き探索を実行し得る。具体的には、動き推定ユニット222は、たとえば、sum of absolute difference (SAD)、sum of squared differences (SSD)、mean absolute difference (MAD)、mean squared differences (MSD)などに従って、潜在的な参照ブロックが現在のブロックとどれだけ類似しているかを表す値を計算し得る。動き推定ユニット222は一般に、現在のブロックと検討されている参照ブロックとの間のサンプルごとの差を使用して、これらの計算を実行し得る。動き推定ユニット222は、現在のブロックと最もよく一致する参照ブロックを示す、これらの計算に起因する最低の値を有する参照ブロックを特定し得る。

20

30

【0114】

動き推定ユニット222は、現在のピクチャの中の現在のブロックの位置に対する相対的な参照ピクチャの中の参照ブロックの位置を定義する、1つまたは複数の動きベクトル(MV)を形成し得る。動き推定ユニット222は次いで、動きベクトルを動き補償ユニット224に提供し得る。たとえば、単方向のインター予測のために、動き推定ユニット222は単一の動きベクトルを提供してもよく、一方、双方向インター予測のために、動き推定ユニット222は2つの動きベクトルを提供してもよい。動き補償ユニット224は次いで、動きベクトルを使用して予測ブロックを生成し得る。たとえば、動き補償ユニット224は、動きベクトルを使用して参照ブロックのデータを取り出し得る。別の例として、動きベクトルが分数サンプル精度を有する場合、動き補償ユニット224は、1つまたは複数の補間フィルタに従って、予測ブロックに対する値を補間し得る。その上、双方向インター予測のために、動き補償ユニット224は、それぞれの動きベクトルによって特定される2つの参照ブロックに対するデータを取り出し、たとえば、サンプルごとの平均または荷重平均を通じて、取り出されたデータを組み合わせ得る。

40

【0115】

別の例として、イントラ予測またはイントラ予測コーディングのために、イントラ予測ユニット226は、現在のブロックに隣接するサンプルから予測ブロックを生成し得る。たとえば、指向性モードでは、イントラ予測ユニット226は一般に、隣接するサンプルの値

50

を数学的に組み合わせて、現在のブロックにわたって定められた方向においてこれらの計算された値を埋めて、予測ブロックを生み出し得る。別の例として、DCモードでは、イントラ予測ユニット226は、現在のブロックに隣接するサンプルの平均を計算し、予測ブロックの各サンプルに対するこの得られた平均を含むように予測ブロックを生成し得る。

【0116】

モード選択ユニット202は、予測ブロックを残差生成ユニット204に提供する。残差生成ユニット204は、ビデオデータメモリ230からの現在のブロックおよびモード選択ユニット202からの予測ブロックの生のコーディングされていないバージョンを受信する。残差生成ユニット204は、現在のブロックと予測ブロックとのサンプルごとの差を計算する。得られるサンプルごとの差は、現在のブロックに対する残差ブロックを定義する。いくつかの例では、残差生成ユニット204はまた、残差差分パルス符号変調(RDPCM: residual differential pulse code modulation)を使用して残差ブロックを生成するために、残差ブロックの中のサンプル値間の差を決定し得る。いくつかの例では、残差生成ユニット204は、二進減算を実行する1つまたは複数の減算器回路を使用して形成され得る。

【0117】

モード選択ユニット202がCUをPUへと区分する例では、各PUは、ルマ予測ユニットおよび対応するクロマ予測ユニットと関連付けられ得る。ビデオエンコーダ200およびビデオデコーダ300は、様々なサイズを有するPUをサポートし得る。上で示されたように、CUのサイズは、CUのル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のルマ予測ユニット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200は、イントラ予測に対して $2N \times 2N$ または $N \times N$ というPUサイズ、およびインター予測に対して $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 、または類似の対称的なPUサイズをサポートし得る。ビデオエンコーダ200およびビデオデコーダ300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 $nR \times 2N$ というPUサイズのための非対称区分をサポートし得る。

【0118】

モード選択ユニットがさらにCUをPUへと区分する例では、各CUは、ルマコーディングブロックおよび対応するクロマコーディングブロックと関連付けられ得る。上のように、CUのサイズは、CUのルマコーディングブロックのサイズを指し得る。ビデオエンコーダ200およびビデオデコーダ120は、 $2N \times 2N$ 、 $2N \times N$ 、または $N \times 2N$ というCUサイズをサポートし得る。

【0119】

少数の例として、イントラブロックコピーモードコーディング、アフィンモードコーディング、および線形モデル(LM)モードコーディングなどの他のビデオコーディング技法では、モード選択ユニット202は、コーディング技法と関連付けられるそれぞれのユニットを介して、符号化されている現在のブロックに対する予測ブロックを生成する。パレットモードコーディングなどのいくつかの例では、モード選択ユニット202は、予測ブロックを生成しなくてもよく、代わりに、選択されたパレットに基づいてブロックを再構築する方式を示すシンタックス要素を生成してもよい。そのようなモードでは、モード選択ユニット202は、符号化されるべきエントロピー符号化ユニット220にこれらのシンタックス要素を提供し得る。

【0120】

上で説明されたように、残差生成ユニット204は、現在のブロックおよび対応する予測ブロックに対するビデオデータを受信する。残差生成ユニット204は次いで、現在のブロックに対する残差ブロックを生成する。残差ブロックを生成するために、残差生成ユニット204は、予測ブロックと現在のブロックとのサンプルごとの差を計算する。

【0121】

変換処理ユニット206は、1つまたは複数の変換を残差ブロックに適用して、変換係数のブロック(「変換係数ブロック」と本明細書で呼ばれる)を生成する。変換処理ユニット206は、残差ブロックに様々な変換を適用して、変換係数ブロックを形成し得る。たとえば

10

20

30

40

50

、変換処理ユニット206は、離散コサイン変換(DCT)、方向変換、カルーネンレーベ変換(KLT)、または概念的に類似の変換を、残差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206は、複数の変換、たとえば、回転変換などの一次変換および二次変換を残差ブロックに対して実行し得る。いくつかの例では、変換処理ユニット206は、残差ブロックに変換を適用しない。

【0122】

量子化ユニット208は、変換係数ブロックの中の変換係数を量子化して、量子化された変換係数ブロックを生成し得る。量子化ユニット208は、現在のブロックと関連付けられる量子化パラメータ(QP)値に従って、変換係数ブロックの変換係数を量子化し得る。ビデオエンコーダ200は(たとえば、モード選択ユニット202を介して)、CUと関連付けられるQP値を調整することによって、現在のブロックと関連付けられる変換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、量子化された変換係数は、変換処理ユニット206によって生成される元の変換係数より精度が低いことがある。

【0123】

逆量子化ユニット210および逆変換処理ユニット212は、それぞれ、量子化された変換係数ブロックに逆量子化および逆変換を適用して、変換係数ブロックから残差ブロックを再構築し得る。再構築ユニット214は、モード選択ユニット202によって生成される再構築された残差ブロックおよび予測ブロックに基づいて、(ある程度の歪みを伴う可能性があるが)現在のブロックに対応する再構築されたブロックを生成し得る。たとえば、再構築ユニット214は、モード選択ユニット202によって生成される予測ブロックからの対応するサンプルに再構築された残差ブロックのサンプルを加算して、再構築されたブロックを生成し得る。

【0124】

フィルタユニット216は、再構築されたブロックに対して1回または複数回のフィルタ動作を実行し得る。たとえば、フィルタユニット216は、CUの端部に沿ったブロック内スライスアर्टィファクトを減らすために、デブロッキング動作を実行し得る。いくつかの例では、フィルタユニット216の動作はスキップされ得る。

【0125】

ビデオエンコーダ200は、DPB218に再構築されたブロックを記憶する。たとえば、フィルタユニット216の動作がスキップされる例では、すなわち、フィルタリングが実行されない例では、再構築ユニット214は、再構築されたブロックをDPB218に記憶し得る。フィルタユニット216の動作が実行される例では、フィルタユニット216は、フィルタリングされた再構築されたブロックをDPB218に記憶し得る。動き推定ユニット222および動き補償ユニット224は、後で符号化されるピクチャのブロックをインター予測するために、再構築された(および場合によってはフィルタリングされた)ブロックから形成される、DPB218から参照ピクチャを取り出し得る。加えて、イントラ予測ユニット226は、現在のピクチャの中の他のブロックをイントラ予測するために、現在のピクチャのDPB218の中の再構築されたブロックを使用し得る。

【0126】

一般に、エントロピー符号化ユニット220は、係数コーディングのための上で説明されたシンタックス要素を含む、ビデオエンコーダ200の他の機能的なコンポーネントから受信されるシンタックス要素をエントロピー符号化し得る。エントロピー符号化ユニット220は、たとえば、変換係数をシグナリングするための上で論じられた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、エントロピー符号化ユニット220は、量子化ユニット208から量子化された変換係数ブロックをエントロピー符号化し得る。別の例として、エントロピー符号化ユニット220は、モード選択ユニット202からの予測シンタックス要素(たとえば、インター予測のための動き情報またはイントラ予測のためのイントラモード情報)をエントロピー符号化し得る。エントロピー符号化ユニット220は、1つまたは複数のエントロピー符号化動作を、ビデオデータの別の例であるシンタックス要素に対して

実行して、エントロピー符号化されたデータを生成し得る。たとえば、エントロピー符号化ユニット220は、コンテキスト適応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、シンタックスベースコンテキスト適応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することができる。いくつかの例では、エントロピー符号化ユニット220は、シンタックス要素がエントロピー符号化されないバイパスモードで動作し得る。

【0127】

ビデオエンコーダ200は、スライスまたはピクチャのブロックを再構築するために必要とされる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。具体的には、エントロピー符号化ユニット220はビットストリームを出力し得る。

10

【0128】

上で説明された動作は、ブロックに関して説明される。そのような説明は、ルマコーディングブロックおよび/またはクロマコーディングブロックのための動作であるものとして理解されるべきである。上で説明されたように、いくつかの例では、ルマコーディングブロックおよびクロマコーディングブロックは、CUのルマ成分およびクロマ成分である。いくつかの例では、ルマコーディングブロックおよびクロマコーディングブロックは、PUのルマ成分およびクロマ成分である。

【0129】

いくつかの例では、ルマコーディングブロックに関して実行される動作は、クロマコーディングブロックに対して繰り返される必要はない。一例として、ルマコーディングブロックのための動きベクトル(MV)および参照ピクチャを特定するための動作が、クロマブロックのためのMVおよび参照ピクチャを特定するために繰り返される必要はない。むしろ、ルマコーディングブロックのためのMVは、クロマブロックのためのMVを決定するためにスケーリングされてもよく、参照ピクチャは同じであってもよい。別の例として、イントラ予測処理は、ルマコーディングブロックおよびクロマコーディングブロックに対して同じであってもよい。

20

【0130】

ビデオエンコーダ200は、ビデオデータを記憶するように構成されるメモリと、回路において実装され本開示において説明されるように係数を符号化するように構成される1つまたは複数の処理ユニットとを含む、ビデオデータを符号化するように構成されるデバイスの例を表す。

30

【0131】

ビデオエンコーダ200は、たとえば、変換係数グループの第1のシンタックス要素に対する第1の符号化パスを実行し、第1のシンタックス要素が変換係数グループの中の係数の第1のセットに対応し、変換係数グループの追加のシンタックス要素に対する追加の符号化パスを実行し、追加のシンタックス要素が係数の第1のセットの変換係数の残りのレベルを示す少なくとも1つのシンタックス要素を備え、本開示の任意の技法に従って変換係数の残りのレベルを示す少なくとも1つのシンタックス要素を処理し得る。第1のシンタックス要素は、たとえば、1つまたは複数の有意性フラグを含んでもよく、1つまたは複数の有意性フラグの各々は、変換係数に対する絶対レベルが0に等しいかどうかを示す。第1のシンタックス要素は、1つまたは複数のパリティレベルフラグを含んでもよく、1つまたは複数のパリティレベルフラグの各々は、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す。ビデオエンコーダ200はまた、追加の符号化パスを実行する前に、変換係数グループの第2のシンタックス要素に対して第2の符号化パスを実行し得る。

40

【0132】

ビデオエンコーダ200はまた、ビデオデータを記憶するように構成されるメモリと、回路において実装され、最大の数の普通のコーディングされたピンに達するまで、変換係数グループの係数に対する値を決定し、変換係数グループのうちの係数の第1のセットに対

50

するシンタックス要素をコンテキスト符号化するように構成される、1つまたは複数の処理ユニットとを含む、ビデオデータを符号化するように構成されるデバイスの例を表す。コンテキスト符号化されたシンタックス要素は、変換係数に対する絶対レベルが0に等しいかどうかを示す1つまたは複数の有意性フラグと、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す1つまたは複数のパリティレベルフラグと、変換係数が2より大きい絶対レベルを有するかどうかを示す1つまたは複数の第1のフラグとを含む。変換係数グループのシンタックス要素をコンテキスト符号化するために、ビデオエンコーダ200は、コンテキスト適応バイナリ算術符号化を実行して、変換係数グループのシンタックス要素を符号化するように構成され得る。

【0133】

ビデオエンコーダ200は、最大の数の普通のコーディングされたビンに達したことに応答して、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化するように構成され得る。追加のシンタックス要素をバイパス符号化するために、ビデオエンコーダ200は、係数の第2のセットの変換係数に対するレベル値を決定し、レベル値が閾値より大きいことに応答して、ゴロムライスコーディングを使用して変換係数に対するプレフィックス値を符号化し、レベル値が閾値より大きいことに応答して、指数ゴロムコーディングを使用してサフィックス値を符号化する。ビデオエンコーダ200は、たとえば、変換係数に対するレベル値からプレフィックス値を差し引くことによって、サフィックス値を決定し得る。変換係数に対するレベル値は、絶対レベル値または残りのレベル値のいずれかであり得る。

【0134】

追加のシンタックス要素をバイパス符号化するために、ビデオエンコーダ200はまた、係数の第2のセットの第2の係数に対する第2のレベル値を決定し、第2のレベル値が閾値より小さいことに応答して、第2の係数に対するサフィックス値を符号化することなく、ゴロムライスコーディングを使用して第2の係数に対するプレフィックス値を符号化するように構成され得る。閾値は、すべての係数に対する固定された値、または変化する値であり得る。たとえば、ビデオエンコーダ200は、たとえば、変換係数に対するライスパラメータの値を決定し、ライスパラメータの値に基づいて閾値を決定するように構成され得る。閾値は、たとえば5または6であってもよいが、他の値も使用されてもよい。

【0135】

図6は、本開示の技法を実行することができる例示的なビデオデコーダ300を示すブロック図である。図6は説明のために提供され、本開示において広く例示および説明されるような技法を限定するものではない。説明のために、本開示は、JEMおよびHEVCの技法に従ったビデオデコーダ300を説明する。しかしながら、本開示の技法は、他のビデオコーディング規格に従ったビデオコーディングデバイスによって実行され得る。

【0136】

図6の例では、ビデオデコーダ300は、コーディングピクチャバッファ(CPB)メモリ320、エン트로ピー復号ユニット302、予測処理ユニット304、逆量子化ユニット306、逆変換処理ユニット308、再構築ユニット310、フィルタユニット312、および復号ピクチャバッファ(DPB)314を含む。予測処理ユニット304は、動き補償ユニット316およびイントラ予測ユニット318を含む。予測処理ユニット304は、他の予測モードに従って予測を実行するための追加のユニットを含み得る。例として、予測処理ユニット304は、パレットユニット、イントラブロックコピーユニット(これは、動き補償ユニット316の一部を形成し得る)、アフィンユニット、線形モデル(LM)ユニットなどを含み得る。他の例では、ビデオデコーダ300は、より多数の、より少数の、または異なる機能コンポーネントを含み得る。

【0137】

CPBメモリ320は、ビデオデコーダ300のコンポーネントによって復号されるべき、符号化されたビデオビットストリームなどのビデオデータを記憶し得る。CPUメモリ320に記憶されるビデオデータは、たとえば、コンピュータ可読媒体110(図1)から取得され得る

10

20

30

40

50

。CPUメモリ320は、符号化されたビデオビットストリームからの符号化されたビデオデータ(たとえば、シンタックス要素)を記憶するCPBを含み得る。また、CPBメモリ320は、ビデオデコーダ300の様々なユニットからの出力を表す一時データなどの、コーディングされたピクチャのシンタックス要素以外のビデオデータを記憶し得る。DPB314は一般に復号されたピクチャを記憶し、ビデオデコーダ300は、符号化されたビデオビットストリームの後続のデータまたはピクチャを復号するとき、参照ビデオデータとしてこの復号されたピクチャを出力および/または使用し得る。CPBメモリ320およびDPB314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)、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。CPUメモリ320およびDPB314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提供され得る。様々な例では、CPBメモリ320は、ビデオデコーダ300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。

【0138】

加えて、または代わりに、いくつかの例では、ビデオデコーダ300は、メモリ120(図1)からコーディングされたビデオデータを取り出し得る。すなわち、メモリ120は、CPBメモリ320とともに上で論じられたようなデータを記憶し得る。同様に、ビデオデコーダ300の機能の一部またはすべてが、ビデオデコーダ300の処理回路によって実行されるようにソフトウェアで実装されるとき、メモリ120は、ビデオデコーダ300によって実行されるべき命令を記憶し得る。

【0139】

図6に示される様々なユニットは、ビデオデコーダ300によって実行される動作を理解するのを助けるために示されている。ユニットは、固定機能の回路、プログラム可能回路、またはこれらの組合せとして実装され得る。図5と同様に、固定機能の回路は、特定の機能を提供する回路を指し、実行され得る動作があらかじめ設定される。プログラム可能回路は、様々なタスクを実行して、実行され得る動作においてフレキシブルな機能を提供するようにプログラムされ得る、回路を指す。たとえば、プログラム可能回路は、ソフトウェアまたはファームウェアの命令によって定義される方式で、プログラム可能回路を動作させるソフトウェアまたはファームウェアを実行し得る。固定機能の回路は、(たとえば、パラメータを受信するために、またはパラメータを出力するために)ソフトウェア命令を実行し得るが、固定機能の回路が実行する動作のタイプは、一般に不変である。いくつかの例では、ユニットの1つまたは複数は、別個の回路ブロック(固定機能またはプログラム可能)であってもよく、いくつかの例では、1つまたは複数のユニットは集積回路であってもよい。

【0140】

ビデオデコーダ300は、ALU、EFU、デジタル回路、アナログ回路、および/または、プログラム可能回路から形成されるプログラム可能コアを含み得る。ビデオデコーダ300の動作がプログラム可能回路上で実行されるソフトウェアによって実行される例では、オンチップメモリまたはオフチップメモリは、ビデオデコーダ300が受信して実行するソフトウェアの命令(たとえば、オブジェクトコード)を記憶し得る。

【0141】

エントロピー復号ユニット302は、CPBから符号化されたビデオデータを受信し、係数コーディングのために上で説明されたシンタックス要素を含むシンタックス要素を再現するためにビデオデータをエントロピー復号し得る。予測処理ユニット304、逆量子化ユニット306、逆変換処理ユニット308、再構成ユニット310、およびフィルタユニット312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、復号されたビデオデータを生成し得る。

【0142】

一般に、ビデオデコーダ300は、ブロックごとにピクチャを再構築する。ビデオデコーダ300は、各ブロックに対する再構築動作を個別に実行し得る(ここで、現在再構築されて

10

20

30

40

50

いる、すなわち復号されているブロックは、「現在のブロック」と呼ばれ得る)。

【0143】

エントロピー復号ユニット302は、量子化された変換係数ブロックの量子化された変換係数を定義するシンタックス要素、ならびに、量子化パラメータ(QP)および/または変換モード指示などの変換情報をエントロピー復号し得る。エントロピー復号ユニット302は、たとえば、変換係数をシグナリングするための上で論じられたシンタックス要素をエントロピー復号し得る。逆量子化ユニット306は、量子化の程度と、同様に、逆量子化ユニット306が適用すべき逆量子化の程度とを決定するために、量子化された変換係数ブロックと関連付けられるQPを使用し得る。逆量子化ユニット306は、たとえば、ビットごとの左シフト演算を実行して、量子化された変換係数を逆量子化し得る。こうして、逆量子化ユニット306は、変換係数を含む変換係数ブロックを形成し得る。

10

【0144】

逆量子化ユニット306が変換係数ブロックを形成した後、逆変換処理ユニット308は、現在のブロックと関連付けられる残差ブロックを生成するために、変換係数ブロックに1つまたは複数の逆変換を適用し得る。たとえば、逆変換処理ユニット308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、変換係数ブロックに適用し得る。

【0145】

さらに、予測処理ユニット304は、エントロピー復号ユニット302によってエントロピー復号された予測情報シンタックス要素に従って、予測ブロックを生成する。たとえば、現在のブロックがインター予測されることを、予測情報シンタックス要素が示す場合、動き補償ユニット316は、予測ブロックを生成し得る。この場合、予測情報シンタックス要素は、参照ブロックをそこから取り出すべきDPB314の中の参照ピクチャ、ならびに、現在のピクチャの中の現在のブロックの位置に対する相対的な参照ピクチャの中の参照ブロックの位置を特定する動きベクトルを示し得る。動き補償ユニット316は一般に、動き補償ユニット224(図5)に関して説明されるものと実質的に同様の方式で、インター予測処理を実行し得る。

20

【0146】

別の例として、現在のブロックがイントラ予測されることを、予測情報シンタックス要素が示す場合、イントラ予測ユニット318は、予測情報シンタックス要素によって示されるイントラ予測モードに従って、予測ブロックを生成し得る。再び、イントラ予測ユニット318は一般に、イントラ予測ユニット226(図5)に関して説明されるものと実質的に同様の方式で、イントラ予測処理を実行し得る。イントラ予測ユニット318は、DPB314から、現在のブロックに隣接するサンプルのデータを取り出し得る。

30

【0147】

再構築ユニット310は、予測ブロックと残差ブロックとを使用して現在のブロックを再構築し得る。たとえば、再構築ユニット310は、予測ブロックの対応するサンプルに残差ブロックのサンプルを追加して、現在のブロックを再構築し得る。

【0148】

フィルタユニット312は、再構築されたブロックに対して1回または複数回のフィルタ動作を実行し得る。たとえば、フィルタユニット312は、再構築されたブロックの端部に沿ったブロックネスアーティファクトを減らすために、デブロック動作を実行し得る。すべての例において、フィルタユニット312の動作が実行されるとは限らない。

40

【0149】

ビデオデコーダ300は、DPB314に再構築されたブロックを記憶し得る。上で論じられたように、DPB314は、イントラ予測のための現在のピクチャおよび以後の動き補償のための以前に復号されたピクチャのサンプルなどの参照情報を、予測処理ユニット304に提供し得る。その上、ビデオデコーダ300は、図1のディスプレイデバイス118などのディスプレイデバイス上で後で提示するために、復号されたピクチャをDPBから出力し得る。

【0150】

50

このようにして、ビデオデコーダ300は、ビデオデータを記憶するように構成されるメモリと、回路において実装され本開示において説明されるように係数を復号するように構成される1つまたは複数の処理ユニットとを含む、ビデオ復号デバイスの例を表す。

【0151】

ビデオデコーダ300は、たとえば、変換係数グループの第1のシンタックス要素に対する第1の復号パスを実行し、第1のシンタックス要素が変換係数グループの中の係数の第1のセットに対応し、変換係数グループの追加のシンタックス要素に対する追加の復号パスを実行し、追加のシンタックス要素が係数の第1のセットの変換係数の残りのレベルを示す少なくとも1つのシンタックス要素を備え、本開示の任意の技法に従って変換係数の残りのレベルを示す少なくとも1つのシンタックス要素を処理し得る。ビデオデコーダ300は、第1のシンタックス要素および追加のシンタックス要素に基づいて、変換係数の値を決定し得る。

【0152】

第1のシンタックス要素は、たとえば、1つまたは複数の有意性フラグを含んでもよく、1つまたは複数の有意性フラグの各々は、変換係数に対する絶対レベルが0に等しいかどうかを示す。第1のシンタックス要素は、たとえば、1つまたは複数のパリティレベルフラグを含んでもよく、1つまたは複数のパリティレベルフラグの各々は、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す。ビデオデコーダ300はまた、追加の復号パスを実行する前に、変換係数グループの第2のシンタックス要素に対して第2の復号パスを実行し得る。

【0153】

このようにして、ビデオデコーダ300はまた、ビデオデータを記憶するように構成されるメモリと、回路において実装され、最大の数の普通のコーディングされたビンに達するまで、変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト復号するように構成される、1つまたは複数の処理ユニットとを含む、ビデオ復号デバイスの例を表す。コンテキスト符号化されたシンタックス要素は、変換係数に対する絶対レベルが0に等しいかどうかを示す1つまたは複数の有意性フラグと、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す1つまたは複数のパリティレベルフラグと、変換係数が2より大きい絶対レベルを有するかどうかを示す1つまたは複数の第1のフラグとを含む。変換係数グループのシンタックス要素をコンテキスト復号するために、ビデオデコーダは、コンテキスト適応バイナリ算術復号を実行して、変換係数グループのシンタックス要素を復号するように構成され得る。

【0154】

ビデオデコーダ300は、最大の数の普通のコーディングされたビンに達したことに応答して、係数の第2のセットに対する追加のシンタックス要素をバイパス復号するように構成され得る。追加のシンタックス要素をバイパス復号するために、ビデオデコーダ300は、変換係数に対するプレフィックス値を受信し、ゴロムライスコーディングを使用してプレフィックス値を復号し、プレフィックス値の長さが閾値に等しいことに応答して、変換係数に対するサフィックス値を受信し、指数ゴロムコーディングを使用してサフィックス値を復号するように構成され得る。

【0155】

ビデオデコーダ300は、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定し、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定するように構成され得る。追加のシンタックス要素に基づいて変換ユニットの係数の第2のセットに対する値を決定するために、ビデオデコーダ300は、復号されたプレフィックス値および復号されたサフィックス値に基づいて、変換係数に対するレベル値を決定するように構成され得る。変換係数に対するレベル値を決定するために、ビデオデコーダ300は、復号されたプレフィックス値を復号されたサフィックス値に加算するように構成され得る。変換係数に対するレベル値は、絶対レベル値または残りのレベル値のいずれかであり得る。

【0156】

第2の係数に対して、ビデオデコーダ300は、第2の係数に対する第2のプレフィックス値を受信し、ゴロムライスコーディングを使用して第2のプレフィックス値を復号し、第2のプレフィックス値の長さが閾値より小さいことに応答して、第2の係数に対するサフィックス値を受信することなく第2の係数に対するレベル値を決定するように構成され得る。

【0157】

閾値は、すべての係数に対する固定された値、または変化する値であり得る。たとえば、ビデオデコーダ300は、たとえば、変換係数に対するライスパラメータの値を決定し、ライスパラメータの値に基づいて閾値を決定するように構成され得る。閾値は、たとえば5または6であってもよいが、他の値も使用されてもよい。

【0158】

ビデオデコーダは、係数の第1のセットに対する値および係数の第2のセットに対する値に基づいて、復号された変換ブロックを決定し、復号された変換ブロックを予測ブロックに追加して再構築されたブロックを決定し、再構築されたブロックに対する1回または複数回のフィルタリング動作を実行してビデオデータの復号されたブロックを決定し、ビデオデータの復号されたブロックを含むビデオデータの復号されたピクチャを出力するように構成され得る。

【0159】

図7Aおよび図7Bは、ピン n におけるCABACプロセスの例を示す。図7Aの例400では、ピン n において、ピン2における範囲は、あるコンテキスト状態(σ)が与えられると、最低確率シンボル(LPS)(p_σ)の確率によって与えられるRangeMPSおよびRangeLPSを含む。例400は、ピン n の値が最高確率シンボル(MPS)に等しいときの、ピン $n+1$ における範囲の更新を示す。この例では、ローは同じままあるが、ピン $n+1$ における範囲の値は、ピン n においてRangeMPSの値まで下げられる。図7Bの例402は、ピン n の値がMPSに等しくない(すなわち、LPSに等しい)とき、ピン $n+1$ における範囲の更新を示す。この例では、ローが、ピン n においてRangeLPSのより低い範囲値に移される。加えて、ピン $n+1$ における範囲の値は、ピン n におけるRangeLPSの値まで下げられる。

【0160】

HEVCビデオコーディングプロセスの一例では、範囲は9ビットで表され、ローは10ビットで表される。範囲およびローの値を十分な精度で維持するための再正規化プロセスがある。再正規化は、範囲が256未満であるときはいつでも起こる。したがって、範囲は、再正規化の後には常に256以上である。範囲およびローの値に応じて、バイナリ算術コーダ(BAC)は、ビットストリームに、「0」、もしくは「1」を出力し、または今後の出力に備えるために、内部変数(BO:未解決ビットと呼ばれる)を更新する。図8は、範囲に応じたBAC出力の例を示す。たとえば、範囲およびローがある閾値(たとえば、512)を上回るとき、「1」がビットストリームに出力される。範囲およびローがある閾値(たとえば、512)を下回るとき、「0」がビットストリームに出力される。範囲およびローがある閾値と閾値の間にあるとき、ビットストリームには何も出力されない。そうではなく、BO値がインクリメントされ、次のピンが符号化される。

【0161】

H.264/AVCのCABACコンテキストモデルでは、およびHEVCのいくつかの例では、128個の状態がある。0~63であり得る64個の可能なLPS確率(状態 σ によって表記される)がある。各MPSは、0または1であり得る。したがって、128個の状態は、64個の状態確率 $\times$ MPSの2つの可能な値(0または1)である。したがって、状態は、7ビットでインデクシングされ得る。

【0162】

LPS範囲(rangeLPS_σ)を導出する計算を減らすために、すべての事例についての結果が、ルックアップテーブルにおいて近似としてあらかじめ計算され記憶され得る。したがって、LPS範囲は、単純なテーブルルックアップを使用することによって、いかなる乗算も用いずに取得され得る。乗算を避けることは、いくつかのデバイスまたは用途にとって重

10

20

30

40

50

要であることがあり、それは、この演算が、多くのハードウェアアーキテクチャにおいて多大な待ち時間を引き起こし得るからである。

【0163】

4列のあらかじめ計算されたLPS範囲テーブルが、乗算の代わりに使用され得る。範囲は、4つのセグメントに分割される。セグメントインデックスが、質問(range>6)&3によって導出され得る。実際には、セグメントインデックスは、実際の範囲からビットをシフトし、除去することによって導出される。以下のTable 1(表5)は、可能な範囲と、それらの対応するインデックスとを示す。

【表5】

TABLE 1 - 範囲インデックス

範囲	256-319	320-383	384-447	448-511
(range>>6) & 3	0	1	2	3

【0164】

そして、LPS範囲テーブルは、64個のエントリ(各確率状態に対して1つ)×4(各範囲インデックスに対して1つ)を有する。各エントリは、RangeLPS、すなわち、範囲とLPS確率を乗算した値である。このテーブルの一部の例が、以下のTable 2(表6)に示される。Table 2(表6)は、確率状態9～12を示す。HEVCに対する1つの提案では、確率状態は0～63にわたり得る。

【表6】

TABLE 2 - RangeLPS

確率状態(σ)	RangeLPS			
	インデックス0	インデックス1	インデックス2	インデックス3
...	...	...	...	...
9	90	110	130	150
10	85	104	123	142
11	81	99	117	135
12	77	94	111	128
...	...	...	...	...

【0165】

各セグメント(すなわち、範囲値)の中で、各確率状態 σ のLPS範囲があらかじめ定義される。言い換えると、確率状態 σ のLPS範囲は、4つの値(すなわち、各範囲インデックスに対して1つの値)へと量子化される。所与の点において使用される具体的なLPS範囲は、どのセグメントに範囲が属すかに依存する。テーブルにおいて使用される可能なLPS範囲の数は、テーブル列の数(すなわち、可能なLPS範囲値の数)とLPS範囲精度との間のトレードオフである。一般に、列が多いほど、LPS範囲値の量子化誤差は小さくなるが、テーブルを記憶するための、より多くのメモリの必要性も増す。列が少ないと量子化誤差が増大するが、テーブルを記憶するのに必要とされるメモリも減る。

【0166】

上で説明されたように、各LPS確率状態は、対応する確率を有する。各状態の確率 p は、次のように導出される。

$$p_{\sigma} = \alpha p_{\sigma-1}$$

ここで、状態 σ は0～63である。定数 α は、各コンテキスト状態間の確率変化の量を表す。一例では、 $\alpha=0.9493$ であり、または、より厳密には、 $\alpha=(0.01875/0.5)^{1/63}$ である。状態 $\sigma=0$ における確率は、0.5(すなわち、 $p_0=1/2$)に等しい。つまり、コンテキスト状態0において、LPSおよびMPSは等しく起こり得る。各々の連続する状態における確率は、前の状態を α と乗じることによって導出される。したがって、LPSがコンテキスト状態 $\alpha=1$ において起こる確率は、 $p_0 \cdot 0.9493$ ($0.5 \cdot 0.9493=.47465$)である。したがって、状態 α のインデックスが増大すると、LPSが起こる確率が下がる。

10

【0167】

CABACは、確率状態が信号統計(すなわち、前にコーディングされたビンの値)に従うために更新されるので、適応的である。更新プロセスは、次の通りである。所与の確率状態について、更新は、状態インデックスと、LPSまたはMPSのいずれかとして識別された符号化されたシンボルの値とに依存する。更新プロセスの結果として、新しい確率状態が導出され、これは、修正された可能性のあるLPS確率推定値と、必要であれば、修正されたMPS値とからなる。

【0168】

ビン値がMPSに等しい場合、所与の状態インデックスが1だけインクリメントされ得る。これは、MPSが、LPS確率がすでにその最低値である(または等価的に、最高のMPS確率に達している)、状態インデックス62において起こるときを除いて、すべての状態に当てはまる。この場合、状態インデックス62は、LPSが見つかるか、または最後のビン値が符号化される(最後のビン値の特殊な場合に対しては、状態63が使用される)まで、固定されたままである。LPSが起こると、状態インデックスは、以下の式に示されるように、状態インデックスをある量だけデクリメントすることによって変更される。この規則は全般に、以下を例外として、LPSの各発生に対して適用される。確率が等しい事例に対応するインデックス $\sigma=0$ の状態において、LPSが符号化されていると仮定すると、状態インデックスは固定されたままであるが、MPS値は、LPSおよびMPSの値が交換されるように切り替えられる。すべての他の事例では、どのシンボルが符号化されていても、MPS値は変えられない。LPS確率に対する遷移規則の導出は、所与のLPS確率 p_{old} と、その更新された確率 p_{new} との間の以下の関係に基づく。

20

$$p_{new} = \max(\alpha p_{old}, p_{62}) \quad \text{MPSが起こる場合}$$

$$p_{new} = (1 - \alpha) + \alpha p_{old} \quad \text{LPSが起こる場合}$$

30

【0169】

CABACにおける確率推定プロセスの現実的な実装形態に関して、すべての遷移規則が、6ビットの符号なし整数値の63個のエントリを各々有する、最大で2つのテーブルによって実現され得ることに留意することが重要である。いくつかの例では、単一のテーブルTransIdxLPSを用いて状態遷移が決定されてもよく、このテーブルは、所与の状態インデックス σ に対して、LPSが確認されている事例における新しい更新された状態インデックスTransIdxLPS[σ]を決定する。1という固定値による、状態インデックスの単純な(飽和した)インクリメントによって、MPSにより引き起こされる遷移を得ることができ、更新された状態インデックス $\min(\sigma+1, 62)$ をもたらす。以下のTable 3(表7)は、部分的TransIdxLPSテーブルの例である。

40

【表 7】

TABLE 3 - TransIdxLPS

確率状態(σ)	新しい状態 TransIdxLPS [σ]
...	...
9	6
10	8
11	8
12	8
...	...

【0170】

図7A、図7B、および図8に関して上で説明された技法は、CABACの1つの例示的な実装形態を表すだけである。本開示の技法は、CABACのこの説明された実装形態だけに限定されないことを理解されたい。たとえば、より古いBAC手法(たとえば、H.264/AVCにおいて使用されるBAC手法)では、テーブルRangeLPSおよびTransIdxLPSが、低解像度ビデオ(すなわち、共通中間フォーマット(CIF)およびクォーターCIF(QCIF)ビデオ)のために調整された。HEVCおよびVVCなどの未来のコーデックでは、大量のビデオコンテンツが高精細(HD)であり、場合によっては、HDよりも大きい。HDであるか、またはHD解像度よりも大きいビデオコンテンツは、H.264/AVCを開発するのに使用された、10年もののQCIFシーケンスとは異なる統計値を有する傾向がある。したがって、H.264/AVCからのテーブルRangeLPSおよびTransIdxLPSは、状態の間で、速すぎる適応を引き起こすことがある。つまり、特にLPSが起るとき、確率状態の間の遷移は、HDビデオのより滑らかな、より高解像度のコンテンツにとっては大きすぎることがある。したがって、従来技術に従って使用される確率モデルは、HDおよびエクストラHDコンテンツに対して正確ではないことがある。加えて、HDビデオコンテンツは、より大きい範囲のピクセル値を含むので、H.264/AVCテーブルは、HDコンテンツに存在し得る、より極端な値を考慮するための十分なエントリを含まない。

【0171】

したがって、HEVCおよびVVCなどの未来のコーディング規格に対して、RangeLPSおよびTransIdxLPSテーブルは、この新規コンテンツの特性を考慮するように修正され得る。具体的には、HEVCおよび未来のコーディング規格に対するBACプロセスは、より遅い適応プロセスを可能にするテーブルを使用することができ、より極端な場合(すなわち、歪んでいる確率)を考慮することができる。したがって、一例として、RangeLPSおよびTransIdxLPSテーブルは、H.264/AVCまたはHEVCについてBACにおいて使用されるものより多くの確率状態および範囲を含むことにより、これらの目標を達成するように修正され得る。

【0172】

図9は、本開示の技法による、CABACを実行するように構成され得る例示的なエントロピー符号化ユニット220のブロック図である。シンタックス要素418が、エントロピー符号化ユニット220へと入力される。シンタックス要素が、すでにバイナリ値シンタックス要素(すなわち、0および1の値のみを有するシンタックス要素)である場合、バイナリ化のステップはスキップされてもよい。シンタックス要素が非バイナリ値のシンタックス要素(たとえば、変換係数レベルなどの、複数のビットによって表されるシンタックス要素)であ

る場合、非バイナリ値のシンタックス要素はバイナライザ420によってバイナリ化される。バイナライザ420は、非バイナリ値のシンタックス要素の、バイナリ決定のシーケンスへのマッピングを実行する。これらのバイナリ決定はしばしば、「ビン」と呼ばれる。たとえば、変換係数レベルに対して、レベルの値を連続するビンに分解することができ、各ビンは、係数レベルの絶対値が何らかの値よりも大きいかどうかを示す。たとえば、ビン0(有意性フラグと呼ばれることがある)は、変換係数レベルの絶対値が0より大きいかどうかを示す。ビン1は、変換係数レベルの絶対値が1より大きいかどうかを示し、以下同様である。各非バイナリ値のシンタックス要素のために、固有のマッピングが開発されてもよい。

【0173】

バイナライザ420によって生成された各ビンは、エントロピー符号化ユニット220のバイナリ算術コーディング側に供給される。つまり、非バイナリ値のシンタックス要素の所定のセットに対して、各ビンタイプ(たとえば、ビン0)は、次のビンタイプ(たとえば、ビン1)の前にコーディングされる。コーディングは、普通モードまたはバイパスモードのいずれかで実行され得る。バイパスモードでは、バイパスコーディングエンジン426は、固定確率モデルを使用して、たとえば、ゴロムライスコーディングまたは指数ゴロムコーディングを使用して、算術コーディングを実行する。バイパスモードは一般に、より予測可能なシンタックス要素のために使用される。

【0174】

普通モードでのコーディングは、CABACを実行することを伴う。普通モードCABACは、前にコーディングされたビンの値を与えられるとビンの値の確率が予測可能である場合のビン値をコーディングするためである。ビンがLPSである確率は、コンテキストモデラ422によって決定される。コンテキストモデラ422は、ビン値およびコンテキストモデル(たとえば、確率状態 σ)を出力する。コンテキストモデルは、一連のビンのための初期コンテキストモデルであってもよく、前にコーディングされたビンのコーディングされた値に基づいて決定されてもよい。上で説明されたように、コンテキストモデラは、前にコーディングされたビンがMPSであったか、またはLPSであったかに基づいて状態を更新することができる。

【0175】

コンテキストモデルおよび確率状態 σ がコンテキストモデラ422によって決定された後、普通コーディングエンジン424は、ビン値に対してBACを実行する。本開示の技法によれば、普通コーディングエンジン424は、64よりも多い確率状態 σ を含むTransIdxLPSテーブル430を使用してBACを実行する。一例では、確率状態の数は128である。TransIdxLPSは、前のビン(ビン n)がLPSであるとき、どの確率状態が次のビン(ビン $n+1$)に使用されるかを決定するために使用される。普通コーディングエンジン424はまた、特定の確率状態 σ を与えられると、LPSに対する範囲値を決定するために、RangeLPSテーブル128を使用し得る。しかしながら、本開示の技法によれば、TransIdxLPSテーブル430のすべての可能な確率状態 σ を使用するのではなく、確率状態インデックス σ は、RangeLPSテーブルにおける使用のために、グループ化されたインデックスにマッピングされる。つまり、RangeLPSテーブル428への各インデックスは、確率状態の総数のうちの2つ以上を表し得る。確率状態インデックス σ の、グループ化されたインデックスへのマッピングは、線形(たとえば、2で除算することによる)であってもよく、または非線形(たとえば、対数関数もしくはマッピングテーブル)であってもよい。

【0176】

本開示の他の例では、連続する確率状態の間の差は、パラメータ α を、0.9493よりも大きくなるように設定することによって、より小さくすることができる。一例では、 $\alpha=0.9689$ である。本開示の別の例では、LPSが起こる最も高い確率(p_0)は、0.5よりも低くなるように設定され得る。一例では、 p_0 は0.493に等しくてもよい。

【0177】

本開示の1つまたは複数の技法によれば、バイナリ算術コーディングプロセスにおいて

10

20

30

40

50

確率状態を更新するのに使用される変数(たとえば、ウィンドウサイズ、スケーリング因子(α)、および確率更新速度のうちの1つまたは複数)の同じ値を使用するのは反対に、エントロピー符号化ユニット220は、異なるコンテキストモデルおよび/または異なるシンタックス要素に対して、変数の異なる値を使用してもよい。たとえば、エントロピー符号化ユニット220は、複数のコンテキストモデルのうちのあるコンテキストモデルに対して、バイナリ算術コーディングプロセスにおいて確率状態を更新するのに使用される変数の値を決定し、決定された値に基づいて確率状態を更新することができる。

【0178】

図10は、本開示の技法による、CABACを実行するように構成され得る例示的なエントロピー復号ユニット302のブロック図である。図10のエントロピー復号ユニット302は、図9において説明されるエントロピー符号化ユニット220のものとは逆方式で、CABACを実行する。ビットストリーム448からのコーディングされたビットが、エントロピー復号ユニット302へと入力される。コーディングされたビットは、コーディングされたビットが、バイパスモードを使用してエントロピーコーディングされたか、または普通モードを使用してエントロピーコーディングされたかに基づいて、コンテキストモデラ450またはバイパス復号エンジン452のいずれかに供給される。コーディングされたビットがバイパスモードでコーディングされた場合、バイパス復号エンジン452は、たとえば、バイナリ値のシンタックス要素または非バイナリシンタックス要素のビンを取り出すために、ゴロムライス復号または指数ゴロム復号を使用し得る。

【0179】

コーディングされたビットが普通モードでコーディングされた場合、コンテキストモデラ450は、コーディングされたビットに対する確率モデルを決定してもよく、普通復号エンジン454は、非バイナリ値のシンタックス要素のビン(または、バイナリ値である場合はシンタックス要素自体)を生成するように、コーディングされたビットを復号してもよい。コンテキストモデルおよび確率状態 σ がコンテキストモデラ450によって決定された後、普通復号エンジン454は、ビン値に対してBACを実行する。本開示の技法によれば、普通復号エンジン454は、64個よりも多い確率状態 σ を含むTransIdxLPSテーブル458を使用してBACを実行する。一例では、確率状態の数は128であるが、本開示の技法と整合する、他の数の確率状態が定義されてもよい。TransIdxLPSテーブル458は、前のビン(ビン n)がLPSであるとき、どの確率状態が次のビン(ビン $n+1$)に使用されるかを決定するために使用される。普通復号エンジン454はまた、特定の確率状態 σ を与えられると、LPSに対する範囲値を決定するために、RangeLPSテーブル456を使用し得る。しかしながら、本開示の技法によれば、TransIdxLPSテーブル458のすべての可能な確率状態 σ を使用するのではなく、確率状態インデックス σ は、RangeLPSテーブル456における使用のために、グループ化されたインデックスにマッピングされる。つまり、RangeLPSテーブル456への各インデックスは、確率状態の総数のうちの2つ以上を表し得る。確率状態インデックス σ の、グループ化されたインデックスへのマッピングは、線形(たとえば、2で除算することによる)であってもよく、または非線形(たとえば、対数関数もしくはマッピングテーブル)であってもよい。

【0180】

本開示の他の例では、連続する確率状態の間の差は、パラメータ α を、0.9493よりも大きくなるように設定することによって、より小さくすることができる。一例では、 $\alpha=0.9689$ である。本開示の別の例では、LPSが起こる最も高い確率(p_0)は、0.5よりも低くなるように設定され得る。一例では、 p_0 は0.493に等しくてもよい。

【0181】

ビンが普通復号エンジン454によって復号された後、逆バイナライザ460が、ビンを非バイナリ値のシンタックス要素の値へ戻すように変換するために、逆方向マッピングを実行し得る。

【0182】

図11は、ビデオデータの現在のブロックを符号化するためのビデオエンコーダの例示的

な動作を示すフローチャートである。現在のブロックは現在のCUを含み得る。ビデオエンコーダ200(図1および図5)に関して説明されるが、他のデバイスが、図11の動作に類似する動作を実行するように構成されてもよいことを理解されたい。

【0183】

この例では、ビデオエンコーダ200は、最初に、現在のブロックを予測する(550)。たとえば、ビデオエンコーダ200は、現在のブロックに対する予測ブロックを形成し得る。次いで、ビデオエンコーダ200は、現在のブロックに対する残差ブロックを計算し得る(552)。残差ブロックを計算するために、ビデオエンコーダ200は、元のコーディングされていないブロックと現在のブロックに対する予測ブロックとの差を計算し得る。次いで、ビデオエンコーダ200は、残差ブロックの係数を変換して量子化し得る(554)。次に、ビデオエンコーダ200は、残差ブロックの量子化された変換係数を走査し得る(556)。走査の間、または走査に続いて、ビデオエンコーダ200は、変換係数をエントロピー符号化し得る(558)。たとえば、ビデオエンコーダ200は、CAVLCまたはCABACを使用して変換係数を符号化し得る。次いで、ビデオエンコーダ200は、ブロックのエントロピーコーディングされたデータを出力し得る(560)。

【0184】

図12は、ビデオデータの現在のブロックを復号するためのビデオデコーダの例示的な動作を示すフローチャートである。現在のブロックは現在のCUを含み得る。ビデオデコーダ300(図1および図6)に関して説明されるが、他のデバイスが、図12の動作に類似する動作を実行するように構成されてもよいことを理解されたい。

【0185】

ビデオデコーダ300は、現在のブロックに対応する残差ブロックの係数に対するエントロピーコーディングされた予測情報およびエントロピーコーディングされたデータなどの、現在のブロックに対するエントロピーコーディングされたデータを受信し得る(570)。ビデオデコーダ300は、エントロピーコーディングされたデータをエントロピー復号して現在のブロックに対する予測情報を決定し、残差ブロックの係数を再現し得る(572)。ビデオデコーダ300は、現在のブロックに対する予測ブロックを計算するために、たとえば、現在のブロックに対する予測情報によって示されるようなイントラ予測モードまたはインター予測モードを使用して、現在のブロックを予測し得る(574)。次いで、ビデオデコーダ300は、量子化された変換係数のブロックを作り出すために、再現された係数を逆走査し得る(576)。次いで、ビデオデコーダ300は、変換係数を逆量子化し逆変換して、残差ブロックを生成し得る(578)。ビデオデコーダ300は、最終的に、予測ブロックと残差ブロックとを組み合わせることによって現在のブロックを復号し得る(580)。

【0186】

図13は、ビデオデータの現在のブロックを符号化するためのビデオエンコーダの例示的な動作を示すフローチャートである。現在のブロックは、たとえば、変換ユニットまたは変換ブロックであり得る。ビデオエンコーダ200(図1および図5)に関して説明されるが、他のデバイスが、図13の動作に類似する動作を実行するように構成されてもよいことを理解されたい。

【0187】

図13の例では、ビデオエンコーダ200は、変換係数グループの係数に対する値を決定し(602)、最大の数の普通のコーディングされたビンに達するまで、変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト符号化する(604)。コンテキスト符号化されたシンタックス要素は、変換係数に対する絶対レベルが0に等しいかどうかを示す1つまたは複数の有意性フラグと、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す1つまたは複数のパリティレベルフラグと、変換係数が2より大きい絶対レベルを有するかどうかを示す1つまたは複数の第1のフラグとを含む。変換係数グループのシンタックス要素をコンテキスト符号化するために、ビデオエンコーダ200は、コンテキスト適応バイナリ算術符号化を実行して、変換係数グループのシンタックス要素を符号化するように構成され得る。ビデオエンコーダ200は、最大の数の普

通のコーディングされたピンに達したことに応答して、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化する(606)。

【0188】

図14は、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス符号化するためのビデオエンコーダの例示的な動作を示すフローチャートである。この点について、図14は、図13のブロック606に対応し得る例示的なプロセスを表す。図14の例では、ビデオエンコーダ200は、係数の第2のセットの変換係数に対するレベル値を決定する(608)。ビデオエンコーダ200は、変換係数に対するレベル値が閾値より低いかどうかを決定する(610)。変換係数が閾値より小さい場合(610)、ビデオエンコーダ200は、ゴロムライスコーディングを使用して、変換係数レベルに等しいプレフィックス値を符号化する(612)。したがって、ビデオエンコーダ200は、変換係数に対するサフィックス値を符号化することなく、閾値未満の変換係数に対するレベルを符号化する。

【0189】

変換係数が閾値より大きい場合(610、No)、ビデオエンコーダ200は、ゴロムライスコーディングを使用して、閾値に等しいプレフィックス値を符号化し(614)、指数ゴロムコーディングを使用してサフィックス値を符号化する(616)。ビデオエンコーダ200は、たとえば、変換係数に対するレベル値からプレフィックス値を差し引くことによって、サフィックス値を決定し得る。変換係数に対するレベル値は、絶対レベル値または残りのレベル値のいずれかであり得る。

【0190】

図15は、ビデオデータの現在のブロックを復号するためのビデオデコーダの例示的な動作を示すフローチャートである。現在のブロックは、たとえば、変換ユニットまたは変換ブロックであり得る。ビデオデコーダ300(図1および図6)に関して説明されるが、他のデバイスが、図14の動作に類似する動作を実行するように構成されてもよいことを理解されたい。

【0191】

図15の例では、ビデオデコーダ300は、最大の数の普通のコーディングされたピンに達するまで、変換係数グループのうちの係数の第1のセットに対するシンタックス要素をコンテキスト復号する(620)。コンテキスト符号化されたシンタックス要素は、変換係数に対する絶対レベルが0に等しいかどうかを示す1つまたは複数の有意性フラグと、変換係数が偶数の絶対レベルを有するか奇数の絶対レベルを有するかを示す1つまたは複数のパリティレベルフラグと、変換係数が2より大きい絶対レベルを有するかどうかを示す1つまたは複数の第1のフラグとを含む。変換係数グループのシンタックス要素をコンテキスト復号するために、ビデオデコーダは、コンテキスト適応バイナリ算術復号を実行して、変換係数グループのシンタックス要素を復号するように構成され得る。

【0192】

ビデオデコーダ300は、最大の数の普通のコーディングされたピンに達したことに応答して、係数の第2のセットに対する追加のシンタックス要素をバイパス復号する(622)。ビデオデコーダ300は、コンテキスト復号されたシンタックス要素に基づいて、変換ユニットの係数の第1のセットに対する値を決定し(624)、追加のシンタックス要素に基づいて、変換ユニットの係数の第2のセットに対する値を決定する(626)。ビデオデコーダ300は、係数の第1のセットに対する値および係数の第2のセットに対する値に基づいて、復号された変換ブロックを決定する(628)。

【0193】

図16は、変換係数グループのうちの係数の第2のセットに対する追加のシンタックス要素をバイパス復号するためのビデオデコーダの例示的な動作を示すフローチャートである。この点について、図16は、図15のブロック622に対応し得る例示的なプロセスを表す。図14の例では、ビデオデコーダ300は、変換係数に対するプレフィックス値を受信し(630)、ゴロムライスコーディングを使用してプレフィックス値を復号する(632)。ビデオデコーダ300は、復号されたプレフィックス値が閾値未満であるかどうかを決定する(634)

。復号されたプレフィックス値が閾値未満である場合(634、Yes)、ビデオデコーダ300は、変換係数レベルが復号されたプレフィックス値に等しいと決定する。この場合、ビデオデコーダ300は、サフィックス値を受信することなく、変換係数に対するレベル値を決定する。

【0194】

復号されたプレフィックス値が閾値より大きい場合(634、No)、ビデオデコーダ300は、変換係数に対するサフィックス値を受信し(638)、指数ゴロムコーディングを使用してサフィックス値を復号する(640)。ビデオデコーダ300は、復号されたプレフィックス値と復号されたサフィックス値を足したものに等しいものとして、変換係数に対するレベルを決定する(642)。

【0195】

例に応じて、本明細書において説明された技法のいずれかのいくつかの行為またはイベントが、異なる順序で実行されてもよく、追加され、併合され、または完全に除外されてもよい(たとえば、説明されたすべての行為またはイベントが技法の実践に必要であるとは限らない)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、逐次的にではなく、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同時に実行されてもよい。

【0196】

1つまたは複数の例では、説明された機能は、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として実装され得る。ソフトウェアで実装される場合、機能は、1つ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上に記憶されるか、またはコンピュータ可読媒体を通じて送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体、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み得る。このように、コンピュータ可読媒体は一般に、(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、本開示において説明された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つまたは複数のコンピュータまたは1つまたは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であり得る。コンピュータプログラム製品はコンピュータ可読媒体を含み得る。

【0197】

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、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のうちの1つまたは複数を含み得る。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、命令が、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。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体が、接続、搬送波、信号、または他の一時的媒体を含まず、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。本明細書で使用されるディスク(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)、フロッピーディスク(disk)およびBlu-ray(登録商標)ディスク(disc)を含み、ディスク(disk)は通常、データを磁氣的に再生し、ディスク(disc)は、レーザーを用いてデータを光学的に再生する。上記の組合せも、コンピュータ可読媒体の範囲に含まれるべきである。

10

20

30

40

50

【0198】

命令は、1つまたは複数のDSP、汎用マイクロプロセッサ、ASIC、FPGA、または他の等価な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得る。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、または本明細書において説明された技法の実装に適した任意の他の構造のいずれかを指し得る。加えて、いくつかの態様では、本明細書において説明された機能は、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/もしくはソフトウェアモジュール内に提供されるか、または複合コーデックに組み込まれ得る。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得る。

【0199】

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。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、様々な構成要素、モジュール、またはユニットが本開示において説明されたが、それらは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するとは限らない。むしろ、上で説明されたように、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もよく、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、上で説明されたような1つまた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もよい。

【0200】

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。

【符号の説明】

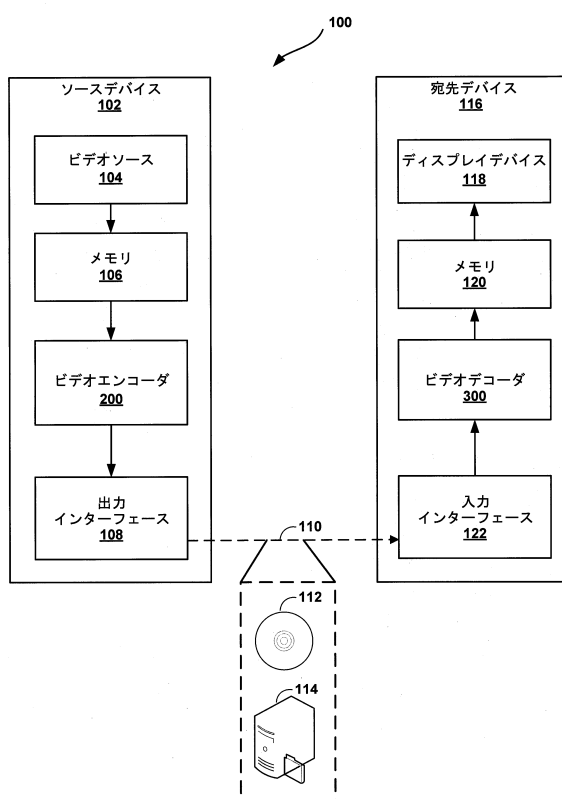
【0201】

- 102 ソースデバイス
- 104 ビデオソース
- 106 メモリ
- 108 出力インターフェース
- 110 コンピュータ可読媒体
- 112 ストレージデバイス
- 114 ファイルサーバ
- 116 宛先デバイス
- 118 ディスプレイデバイス
- 120 メモリ
- 122 入力インターフェース
- 130 QTBT構造
- 132 コーディングツリーユニット
- 136 第1のパス
- 138 第2のパス
- 140 第3のパス
- 142 第4のパス
- 200 ビデオエンコーダ
- 202 モード選択ユニット
- 204 残差生成ユニット
- 206 変換処理ユニット
- 208 量子化ユニット
- 210 逆量子化ユニット
- 212 逆変換処理ユニット
- 214 再構築ユニット
- 216 フィルタユニット
- 218 復号ピクチャバッファ

220	エントロピー符号化ユニット	
222	動き推定ユニット	
224	動き補償ユニット	
226	イントラ予測ユニット	
230	ビデオデータメモリ	
300	ビデオデコーダ	
302	エントロピー復号ユニット	
304	予測処理ユニット	
306	逆量子化ユニット	
308	逆変換処理ユニット	10
310	再構築ユニット	
312	フィルタユニット	
316	動き補償ユニット	
318	イントラ予測ユニット	
320	CPBメモリ	
418	シンタックス要素	
420	バイナライザ	
422	コンテキストモデラ	
424	普通コーディングエンジン	
426	バイパスコーディングエンジン	20
428	RangeLPSテーブル	
430	TransIdxLPSテーブル	
448	ビットストリーム	
450	コンテキストモデラ	
452	バイパス復号エンジン	
454	普通復号エンジン	
456	RangeLPSテーブル	
458	TransIdxLPSテーブル	
460	逆バイナライザ	30

【図面】

【図 1】



【図 2 A】

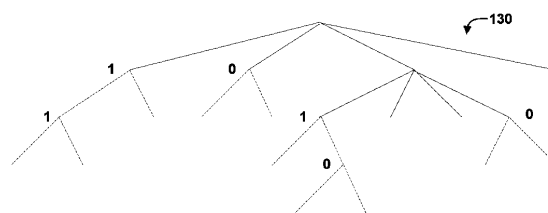


FIG. 2A

10

20

【図 2 B】

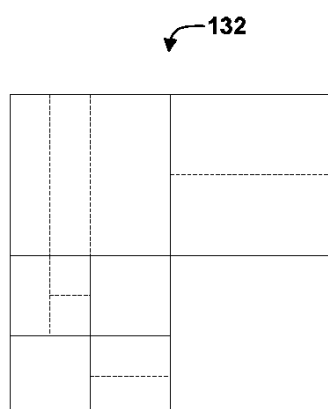
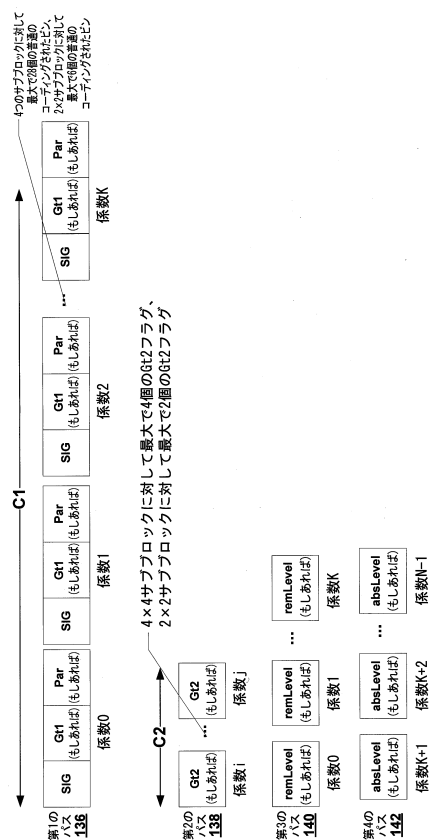


FIG. 2B

【図 3】



30

40

50

【図 4】

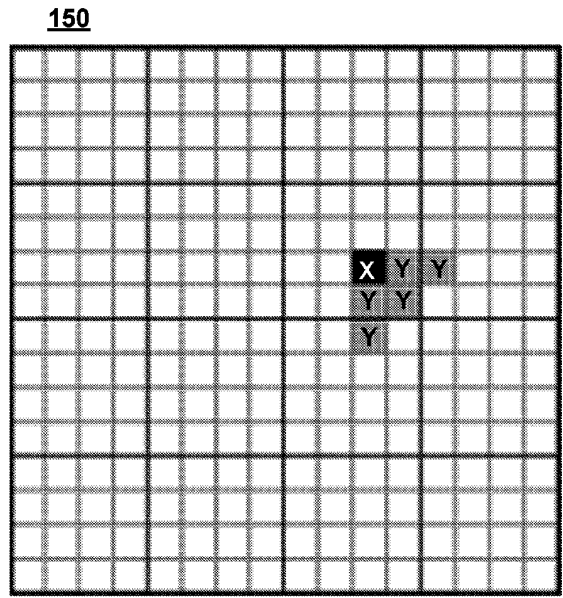
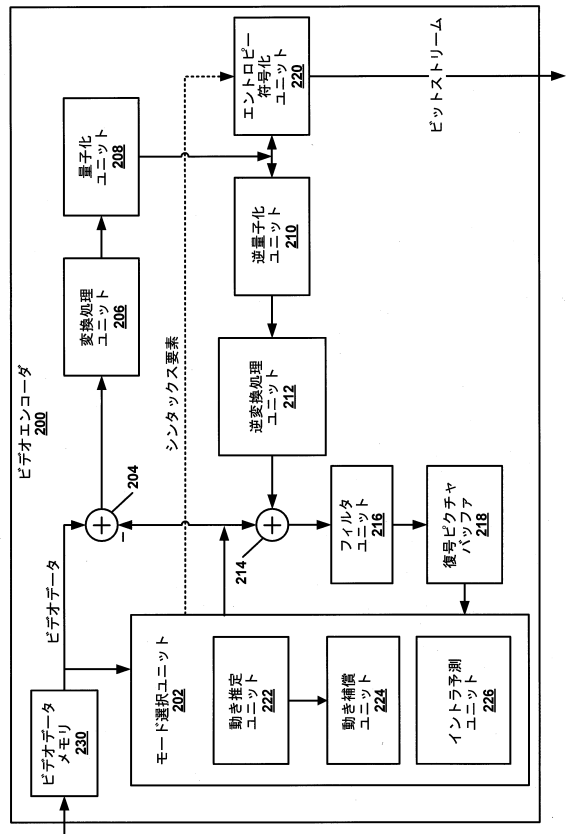


FIG. 4

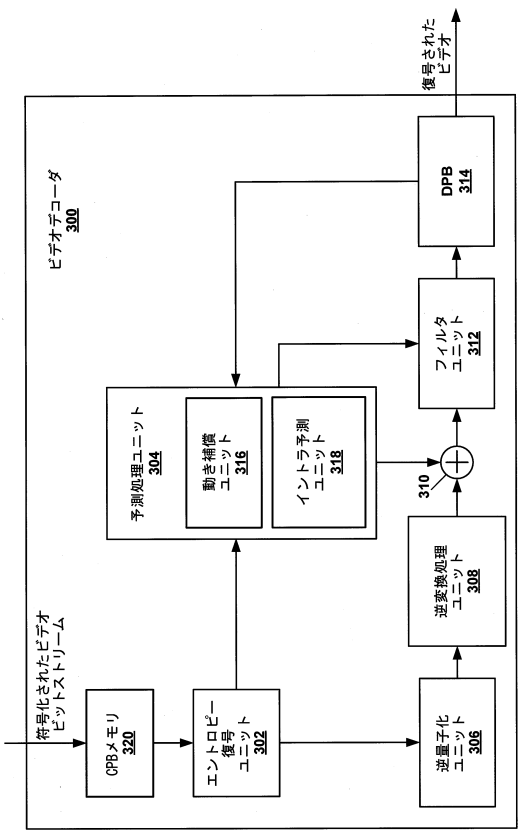
【図 5】



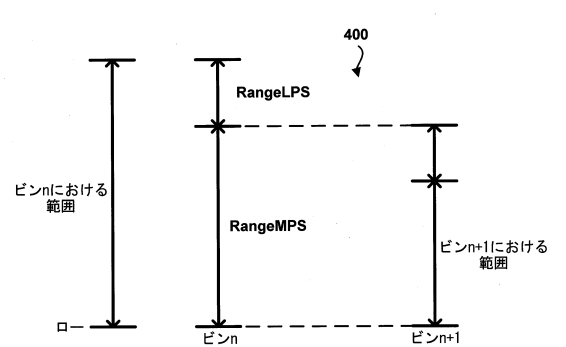
10

20

【図 6】



【図 7 A】

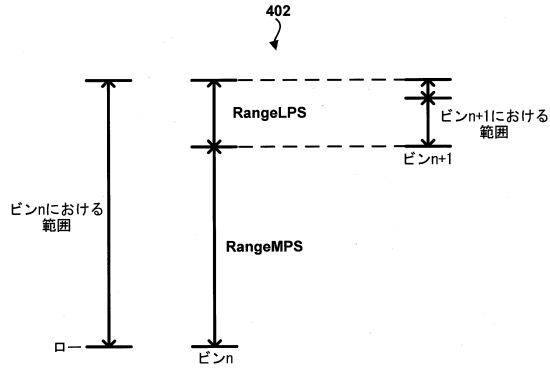


30

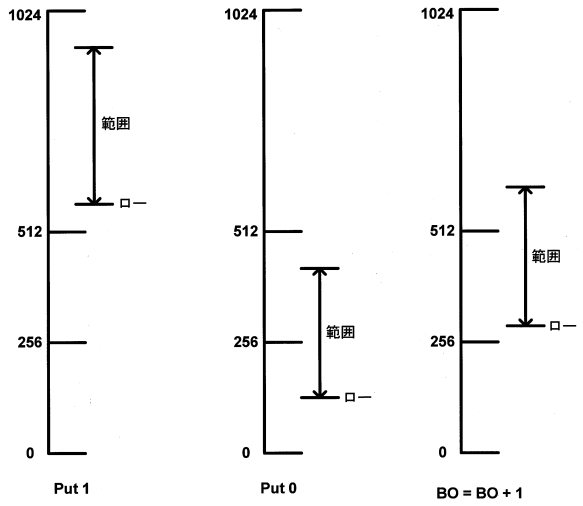
40

50

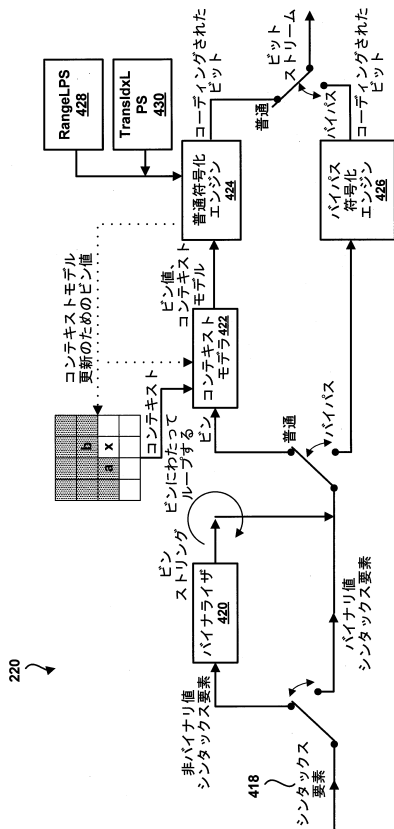
【図 7 B】



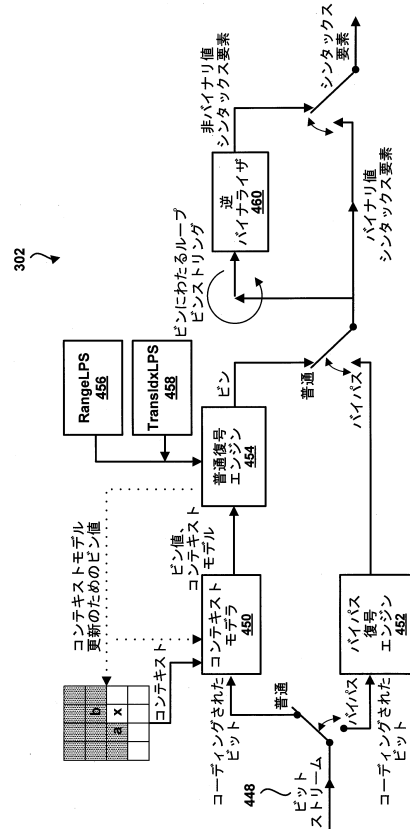
【図 8】



【図 9】



【図 10】



10

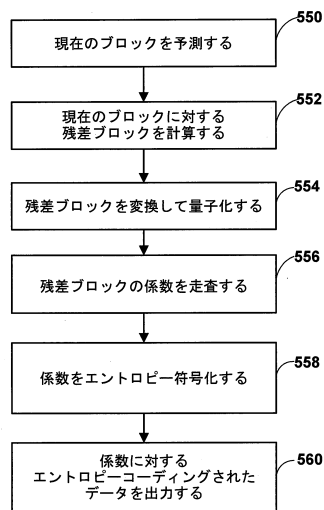
20

30

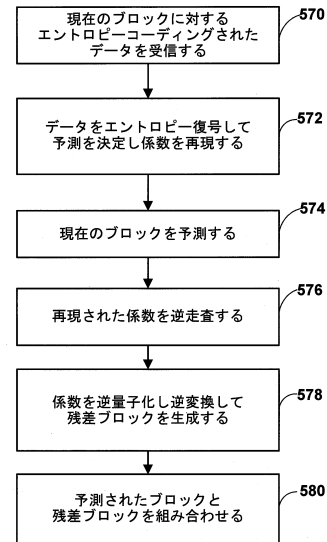
40

50

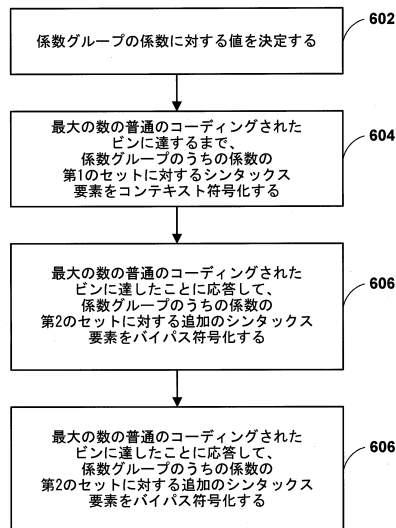
【図 1 1】



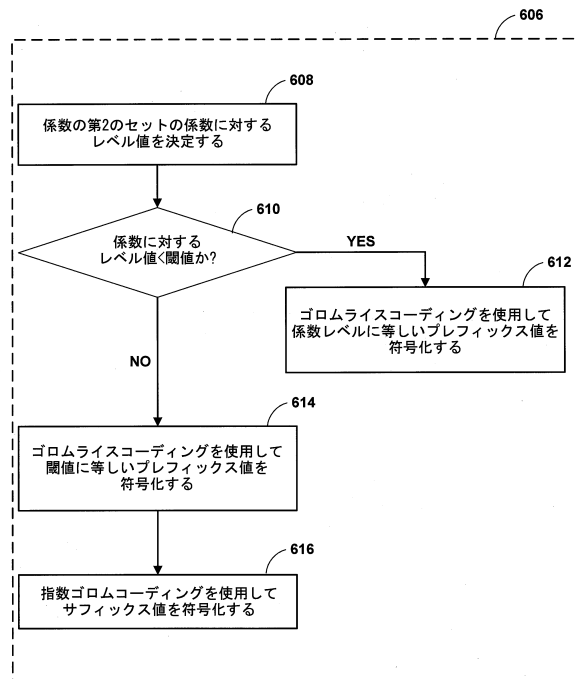
【図 1 2】



【図 1 3】



【図 1 4】



10

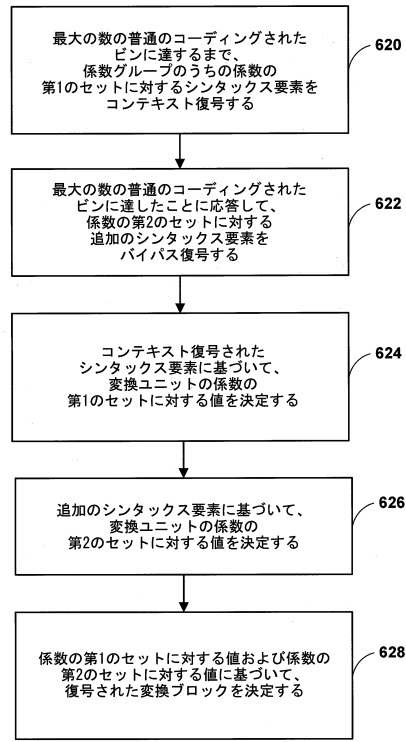
20

30

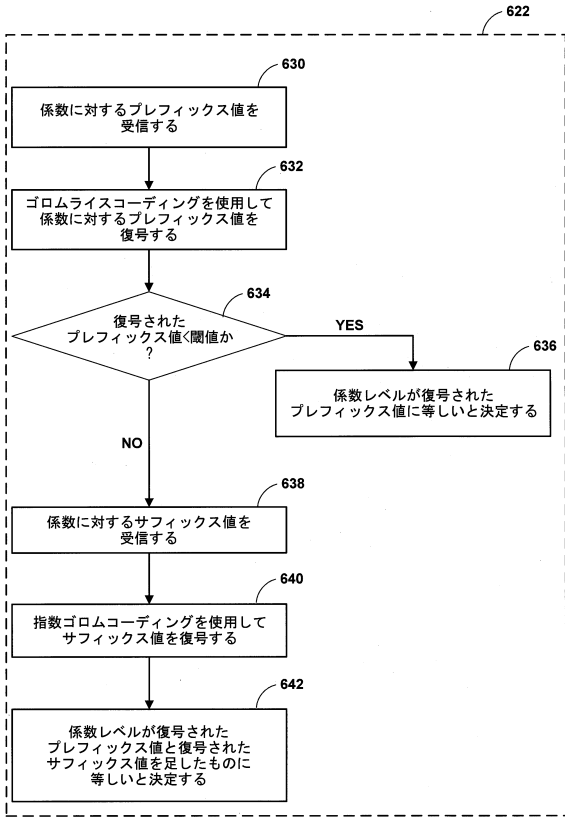
40

50

【図 15】



【図 16】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

1 2 1 - 1 7 1 4 ・ サン ・ ディエゴ ・ モアハウス ・ ドライヴ ・ 5 7 7 5

審査官 田部井 和彦

(56)参考文献

特表 2 0 1 8 - 5 2 9 2 7 3 (J P , A)

特表 2 0 1 5 - 5 1 6 7 4 5 (J P , A)

Draft text for JVET-L0274 (version B) [online], JVET-L0274 (JVET-L0274_DraftTextForVVC-versionB.docx) , [2024年1月25日検索], インターネット <URL: https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0274-v3.zip> , 2018年10月03日
Heiko Schwarz et al., CE7: Transform coefficient coding with reduced number of regular-coded bins (tests 7.1.3a, 7.1.3b) [online], JVET-L0274-v3 (JVET-L0274-v3 .docx) , [2024年1月25日検索], インターネット <URL: [URL: https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0274-v3.zip](https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0274-v3.zip)> , 2018年10月03日

Muhammed Coban et al., CE7-related: Golomb-Rice/exponential Golomb coding for abs_remainder and dec_abs_level syntax elements [online], JVET-M0470 (JVET-M0470.docx) , [2024年1月25日検索], インターネット <URL: https://jvet-experts.org/doc_end_user/documents/13_Marrakech/wg11/JVET-M0470-v1.zip> , 2019年01月03日

(58)調査した分野 (Int.Cl., D B 名)

H 0 4 N 1 9 / 1 3

H 0 4 N 1 9 / 1 8

H 0 4 N 1 9 / 1 3 6

H 0 4 N 1 9 / 9 1