

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2020-504506

(P2020-504506A)

(43) 公表日 令和2年2月6日(2020.2.6)

(51) Int.Cl.	F I	テーマコード (参考)
HO 4 N 19/122 (2014.01)	HO 4 N 19/122	5 C 1 5 9
HO 4 N 19/136 (2014.01)	HO 4 N 19/136	
HO 4 N 19/176 (2014.01)	HO 4 N 19/176	
HO 4 N 19/70 (2014.01)	HO 4 N 19/70	

審査請求 未請求 予備審査請求 有 (全 56 頁)

(21) 出願番号	特願2019-533621 (P2019-533621)	(71) 出願人	507364838 クアルコム、インコーポレイテッド アメリカ合衆国 カリフォルニア 921 21 サン ディエゴ モアハウス ドラ イブ 5775
(86) (22) 出願日	平成30年1月5日 (2018.1.5)	(74) 代理人	100108453 弁理士 村山 靖彦
(85) 翻訳文提出日	令和1年6月20日 (2019.6.20)	(74) 代理人	100163522 弁理士 黒田 晋平
(86) 国際出願番号	PCT/US2018/012589	(72) 発明者	シアン・リ アメリカ合衆国・カリフォルニア・921 30・サン・ディエゴ・ゲイルモント・レ ーン・10574
(87) 国際公開番号	W02018/129322		
(87) 国際公開日	平成30年7月12日 (2018.7.12)		
(31) 優先権主張番号	62/443,569		
(32) 優先日	平成29年1月6日 (2017.1.6)		
(33) 優先権主張国・地域又は機関	米国 (US)		
(31) 優先権主張番号	15/862,203		
(32) 優先日	平成30年1月4日 (2018.1.4)		
(33) 優先権主張国・地域又は機関	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マルチタイプツリーフレームワーク

(57) 【要約】

ビデオを復号する方法は、ビデオデータの符号化ブロックを受信するステップと、ビデオデータの符号化ブロックに対する変換を判定するステップであって、変換が、2の冪ではないサイズSを有する、判定するステップと、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するステップと、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用するステップと、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号するステップとを含む。

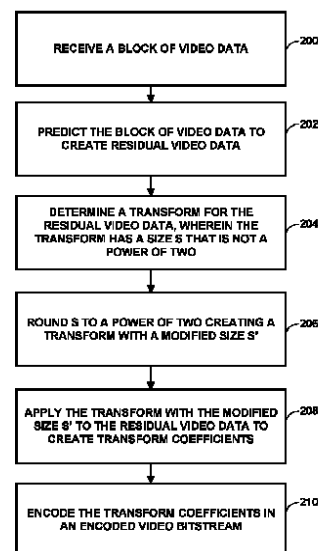


FIG. 12

【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、
前記ビデオデータの符号化ブロックを受信するステップと、
前記ビデオデータの前記符号化ブロックに対する変換を判定するステップであって、前記変換が、2の冪ではないサイズSを有する、判定するステップと、
Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成するステップと、
残差ビデオデータを生成するために、前記修正されたサイズS'を有する前記逆変換を前記ビデオデータの前記符号化ブロックに適用するステップと、
前記ビデオデータの復号ブロックを生成するために、前記残差ビデオデータを復号するステップと
を含む、方法。 10

【請求項 2】

Sを2の冪に前記丸めるステップが、Sを2の最も近い冪に丸めるステップを含む、請求項1に記載の方法。

【請求項 3】

前記ビデオデータの前記符号化ブロックが、逆量子化変換係数を含む、請求項1に記載の方法。

【請求項 4】

前記ビデオデータの前記符号化ブロックが、非正方形形状を有する、請求項1に記載の方法。 20

【請求項 5】

Sが12であり、Sを2の冪に前記丸めるステップが、12を16に丸めるステップを含み、前記修正されたサイズS'が16である、請求項1に記載の方法。

【請求項 6】

Sが24であり、Sを2の冪に前記丸めるステップが、24を32に丸めるステップを含み、前記修正されたサイズS'が32である、請求項1に記載の方法。

【請求項 7】

Sが前記変換の幅である、請求項1に記載の方法。

【請求項 8】

Sが前記変換の高さである、請求項1に記載の方法。 30

【請求項 9】

S'に基づいて、前記逆変換に対するシフト値を判定するステップ
をさらに含む、請求項1に記載の方法。

【請求項 10】

ビデオデータを符号化する方法であって、
前記ビデオデータのブロックを受信するステップと、
残差ビデオデータを生成するために、前記ビデオデータの前記ブロックを予測するステップと、
前記残差ビデオデータに対する変換を判定するステップであって、前記変換が、2の冪ではないサイズSを有する、判定するステップと、
Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するステップと、
変換係数を生成するために、前記修正されたサイズS'を有する前記変換を前記残差ビデオデータに適用するステップと、
符号化ビデオビットストリーム内の前記変換係数を符号化するステップと
を含む、方法。 40

【請求項 11】

Sを2の冪に前記丸めるステップが、Sを2の最も近い冪に丸めるステップを含む、請求項10に記載の方法。

【請求項 12】

前記変換係数を量子化するステップ
をさらに含む、請求項10に記載の方法。

【請求項13】

前記ビデオデータの前記ブロックが、非正方形形状を有する、請求項10に記載の方法。

【請求項14】

Sが12であり、Sを2の冪に前記丸めるステップが、12を16に丸めるステップを含み、前記修正されたサイズS'が16である、請求項10に記載の方法。

【請求項15】

Sが24であり、Sを2の冪に前記丸めるステップが、24を32に丸めるステップを含み、前記修正されたサイズS'が32である、請求項10に記載の方法。

10

【請求項16】

Sが前記変換の幅である、請求項10に記載の方法。

【請求項17】

Sが前記変換の高さである、請求項10に記載の方法。

【請求項18】

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、
前記ビデオデータを記憶するように構成されたメモリと、
前記メモリと通信している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、

20

前記ビデオデータの符号化ブロックを受信することと、

前記ビデオデータの前記符号化ブロックに対する変換を判定することであって、前記変換が、2の冪ではないサイズSを有する、判定することと、

Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成することと、

残差ビデオデータを生成するために、前記修正されたサイズS'を有する前記逆変換を前記ビデオデータの前記符号化ブロックに適用することと、

前記ビデオデータの復号ブロックを生成するために、前記残差ビデオデータを復号することと

を行うように構成される、装置。

【請求項19】

Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、Sを2の最も近い冪に丸めるように構成される、請求項18に記載の装置。

30

【請求項20】

前記ビデオデータの前記符号化ブロックが、逆量子化変換係数を含む、請求項18に記載の装置。

【請求項21】

前記ビデオデータの前記符号化ブロックが、非正方形形状を有する、請求項18に記載の装置。

【請求項22】

Sが12であり、Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、12を16に丸めるように構成され、前記修正されたサイズS'が16である、請求項18に記載の装置。

40

【請求項23】

Sが24であり、Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、24を32に丸めるように構成され、前記修正されたサイズS'が32である、請求項18に記載の装置。

【請求項24】

Sが前記変換の幅である、請求項18に記載の装置。

【請求項25】

Sが前記変換の高さである、請求項18に記載の装置。

【請求項26】

前記1つまたは複数のプロセッサが、

S'に基づいて、前記逆変換に対するシフト値を判定する

50

ようにさらに構成される、請求項18に記載の装置。

【請求項 27】

前記ビデオデータの前記復号ブロックを表示するように構成されたディスプレイをさらに備える、請求項18に記載の装置。

【請求項 28】

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、
前記ビデオデータを記憶するように構成されたメモリと、
前記メモリと通信している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、

前記ビデオデータのブロックを受信することと、

10

残差ビデオデータを生成するために、前記ビデオデータの前記ブロックを予測することと、

前記残差ビデオデータに対する変換を判定することであって、前記変換が、2の冪ではないサイズSを有する、判定することと、

Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成することと、

変換係数を生成するために、前記修正されたサイズS'を有する前記変換を前記残差ビデオデータに適用することと、

符号化ビデオビットストリーム内の前記変換係数を符号化することと
を行うように構成される、装置。

【請求項 29】

20

Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、Sを2の最も近い冪に丸めるように構成される、請求項28に記載の装置。

【請求項 30】

前記1つまたは複数のプロセッサが、
前記変換係数を量子化する
ようにさらに構成される、請求項28に記載の装置。

【請求項 31】

前記ビデオデータの前記ブロックが、非正方形形状を有する、請求項28に記載の装置。

【請求項 32】

Sが12であり、Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、12を16に丸めるように構成され、前記修正されたサイズS'が16である、請求項28に記載の装置。

30

【請求項 33】

Sが24であり、Sを2の冪に丸めるために、前記1つまたは複数のプロセッサが、24を32に丸めるように構成され、前記修正されたサイズS'が32である、請求項28に記載の装置。

【請求項 34】

Sが前記変換の幅である、請求項28に記載の装置。

【請求項 35】

Sが前記変換の高さである、請求項28に記載の装置。

【請求項 36】

前記ビデオデータをキャプチャするように構成されるカメラ
をさらに備える、請求項28に記載の装置。

40

【請求項 37】

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、
前記ビデオデータの符号化ブロックを受信するための手段と、
前記ビデオデータの前記符号化ブロックに対する変換を判定するための手段であって、
前記変換が、2の冪ではないサイズSを有する、判定するための手段と、

Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成するための手段と、

残差ビデオデータを生成するために、前記修正されたサイズS'を有する前記逆変換を前記ビデオデータの前記符号化ブロックに適用するための手段と、

前記ビデオデータの復号ブロックを生成するために、前記残差ビデオデータを復号する

50

ための手段と
を備える、装置。

【請求項 38】

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、
前記ビデオデータのブロックを受信するための手段と、
残差ビデオデータを生成するために、前記ビデオデータの前記ブロックを予測するための手段と、

前記残差ビデオデータに対する変換を判定するための手段であって、前記変換が、2の
冪ではないサイズSを有する、判定するための手段と、

Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するための手段と、
変換係数を生成するために、前記修正されたサイズS'を有する前記変換を前記残差ビデオ
データに適用するための手段と、

符号化ビデオビットストリーム内の前記変換係数を符号化するための手段と
を備える、装置。

【請求項 39】

実行されると、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、

前記ビデオデータの符号化ブロックを受信することと、
前記ビデオデータの前記符号化ブロックに対する変換を判定することであって、前記変
換が、2の冪ではないサイズSを有する、判定することと、

Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成することと、
残差ビデオデータを生成するために、前記修正されたサイズS'を有する前記逆変換を前
記ビデオデータの前記符号化ブロックに適用することと、

前記ビデオデータの復号ブロックを生成するために、前記残差ビデオデータを復号する
ことと
を行わせる命令を記憶した、コンピュータ可読記憶媒体。

【請求項 40】

実行されると、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、

前記ビデオデータのブロックを受信することと、
残差ビデオデータを生成するために、前記ビデオデータの前記ブロックを予測すること
と、

前記残差ビデオデータに対する変換を判定することであって、前記変換が、2の冪では
ないサイズSを有する、判定することと、

Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成することと、
変換係数を生成するために、前記修正されたサイズS'を有する前記変換を前記残差ビデオ
データに適用することと、

符号化ビデオビットストリーム内の前記変換係数を符号化することと
を行わせる命令を記憶した、コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、その内容全体が参照によって本明細書に組み込まれている、2017年1月6日に
出願された米国仮出願第62/443,569号の利益を主張する。

【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

10

20

30

40

50

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、アドバンスドビデオコーディング(AVC:Advanced Video Coding)、高効率ビデオコーディング(HEVC:High Efficiency Video Coding)規格によって定義された規格、およびそのような規格の拡張に記載されている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。

10

【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(イントラピクチャ)予測および/または時間(インターピクチャ)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、ビデオフレーム、またはビデオフレームの一部)は、ビデオブロックに区分されてよく、ビデオブロックは、ツリーブロック、コーディングユニット(CU:coding unit)、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。ピクチャは、フレームと呼ばれることがある。参照ピクチャは、参照フレームと呼ばれることがある。

20

【0005】

空間予測または時間予測は、コーディングされるべきブロックの予測ブロックをもたらす。残差データは、コーディングされるべき元のブロックと予測ブロックとの間のピクセル差分を表す。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残差変換係数をもたらし得、残差変換係数は、次いで、量子化され得る。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てよい。

【発明の概要】

【課題を解決するための手段】

【0006】

本開示は、マルチタイプツリー(MTT:multi-type tree)フレームワークを使用してビデオデータのブロックを区分するための技法について説明する。本開示の技法は、ツリー構造の様々なノードにおける複数の区分技法のうちの1つを判定することを含む。複数の区分技法の例は、ブロックの中心を通してブロックを対称的に分割する区分技法、ならびにブロックの中心が分割されないように、対称的または非対称的のいずれかで、ブロックを分割する区分技法を含み得る。このようにして、ビデオブロックの区分は、ブロックの中心内にある、ビデオデータ内のオブジェクトをより良くキャプチャする区分を含めて、より効率的なコーディングをもたらす様式で実行され得る。

30

【0007】

本開示は、MTTフレームワークに従って区分されたブロックに変換を適用するための技法、MTTフレームワークに従ってブロックがどのように分割されるかを示すシンタックス要素を生成してパースするための技法、MTTフレームワークに従ってルーマブロックおよびクロマブロックを区分するための技法、およびMTTフレームワークに従って区分されたブロックをコーディング(すなわち、符号化および/または復号)するための技法についてさらに説明する。本開示で説明する技法は、個々に、または任意の組合せで一緒に使用され得る。

40

【0008】

本開示の一例では、ビデオデータを復号する方法は、ビデオデータの符号化ブロックを受信するステップと、ビデオデータの符号化ブロックに対する変換を判定するステップであって、変換が、2の冪ではないサイズSを有する、判定するステップと、Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成するステップと、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用

50

するステップと、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号するステップとを含む。

【0009】

本開示の別の例では、ビデオデータを符号化する方法は、ビデオデータのブロックを受信するステップと、残差ビデオデータを生成するために、ビデオデータのブロックを予測するステップと、残差ビデオデータに対する変換を判定するステップであって、変換が、2の冪ではないサイズSを有する、判定するステップと、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するステップと、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用するステップと、符号化ビデオビットストリーム内の変換係数を符号化するステップとを含む。

10

【0010】

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と、メモリと通信している1つまたは複数のプロセッサとを備え、1つまたは複数のプロセッサは、ビデオデータの符号化ブロックを受信することと、ビデオデータの符号化ブロックに対する変換を判定することであって、変換が、2の冪ではないサイズSを有する、判定することと、Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成することと、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用することと、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号することとを行うように構成される。

20

【0011】

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と、メモリと通信している1つまたは複数のプロセッサとを備え、1つまたは複数のプロセッサが、ビデオデータのブロックを受信することと、残差ビデオデータを生成するために、ビデオデータのブロックを予測することと、残差ビデオデータに対する変換を判定することであって、変換が、2の冪ではないサイズSを有する、判定することと、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成することと、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用することと、符号化ビデオビットストリーム内の変換係数を符号化することとを行うように構成される。

30

【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、ビデオデータの符号化ブロックを受信するための手段と、ビデオデータの符号化ブロックに対する変換を判定するための手段であって、変換が、2の冪ではないサイズSを有する、判定するための手段と、Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成するための手段と、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用するための手段と、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号するための手段とを含む。

【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、ビデオデータのブロックを受信するための手段と、残差ビデオデータを生成するために、ビデオデータのブロックを予測するための手段と、残差ビデオデータに対する変換を判定するための手段であって、変換が、2の冪ではないサイズSを有する、判定するための手段と、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するための手段と、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用するための手段と、符号化ビデオビットストリーム内の変換係数を符号化するための手段とを備える。

40

【0014】

別の例では、本開示は、実行されると、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、ビデオデータの符号化ブロックを受信することと、ビデオデータの符号化ブロックに対する変換を判定することであって、変換が、2の冪

50

ではないサイズSを有する、判定することと、Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成することと、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用することと、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号することとを行わせる命令を記憶した、コンピュータ可読記憶媒体について説明する。

【0015】

別の例では、本開示は、実行されると、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、ビデオデータのブロックを受信することと、残差ビデオデータを生成するために、ビデオデータのブロックを予測することと、残差ビデオデータに対する変換を判定することとであって、変換が、2の冪ではないサイズSを有する、判定することと、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成することと、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用することと、符号化ビデオビットストリーム内の変換係数を符号化することとを行わせる命令を記憶した、コンピュータ可読記憶媒体について説明する。

10

【0016】

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】

【0017】

【図1】本開示の技法を実施するように構成された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。

20

【図2】高効率ビデオコーディング(HEVC)内のコーディングユニット(CU)構造を示す概念図である。

【図3】インター予測モードの場合の例示的な区分タイプを示す概念図である。

【図4A】4分木2分木(QTBT:quad-tree-binary-tree)構造を使用したブロック区分の例を示す概念図である。

【図4B】図4AのQTBT構造を使用したブロック区分に対応する例示的なツリー構造を示す概念図である。

【図5A】4分木区分を示す概念図である。

【図5B】垂直2分木区分を示す概念図である。

30

【図5C】水平2分木区分を示す概念図である。

【図5D】垂直中心-側部(center-side)3分木区分を示す概念図である。

【図5E】水平中心-側部木区分を示す概念図である。

【図6】本開示の技法による、コーディングツリーユニット(CTU)区分の例を示す概念図である。

【図7】QTBT区分の一例による、例示的な非対称区分を示す概念図である。

【図8】デッドゾーンプラス一様量子化(deadzone plus uniform quantization)方式を示す概念図である。

【図9】例示的な非対称的区分タイプを示す図である。

【図10】ビデオエンコーダの一例を示すブロック図である。

40

【図11】ビデオデコーダの一例を示すブロック図である。

【図12】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13】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。

【発明を実施するための形態】

【0018】

本開示は、ブロックベースのビデオコーディングにおけるビデオデータのブロック(たとえば、コーディングユニット)の区分および/または編成に関する。本開示の技法は、ビデオコーディング規格に適用され得る。下記で説明する様々な例では、本開示の技法は、3つ以上の異なる区分構造を使用して、ビデオデータのブロックを区分するステップを含む。いくつかの例では、3つ以上の異なる区分構造は、コーディングツリー構造の各深度

50

において使用され得る。そのような区分技法は、マルチタイプツリー (MTT) 区分と呼ばれることがある。MTT区分を使用することによって、ビデオデータをより柔軟に区分することが可能であり、これにより、より高いコーディング効率を可能にする。

【 0 0 1 9 】

本開示は、MTTフレームワークに従って区分されたブロックに変換を適用するための技法、MTTフレームワークに従ってブロックがどのように分割されるかを示すシンタックス要素を生成してパースするための技法、MTTフレームワークに従ってルーマブロックおよびクロマブロックを区分するための技法、およびMTTフレームワークに従って区分されたブロックをコーディング(すなわち、符号化および/または復号)するための技法についてさらに説明する。本開示で説明する技法は、個々に、または任意の組合せで一緒に使用され得る。

10

【 0 0 2 0 】

図1は、ビデオデータのブロックを区分し、区分タイプをシグナリングしてパースし、変換およびさらなる変換区分を適用するために本開示の技法を利用し得る、例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示すように、システム10は、宛先デバイス14によって後で復号されるべき符号化ビデオデータを提供するソースデバイス12を含む。詳細には、ソースデバイス12は、コンピュータ可読媒体16を介して宛先デバイス14にビデオデータを提供する。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、タブレットコンピュータ、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を備え得る。場合によっては、ソースデバイス12および宛先デバイス14は、ワイヤレス通信のために装備され得る。したがって、ソースデバイス12および宛先デバイス14は、ワイヤレス通信デバイスであり得る。ソースデバイス12は、例示的なビデオ符号化デバイス(すなわち、ビデオデータを符号化するためのデバイス)である。宛先デバイス14は例示的なビデオ復号デバイス(たとえば、ビデオデータを復号するためのデバイスまたは装置)である。

20

【 0 0 2 1 】

図1の例では、ソースデバイス12は、ビデオソース18、ビデオデータを記憶するように構成された記憶媒体20、ビデオエンコーダ22、および出力インターフェース24を含む。宛先デバイス14は、入力インターフェース26、符号化ビデオデータを記憶するように構成された記憶媒体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。他の例では、ソースデバイス12および宛先デバイス14は、他の構成要素または構成を含む。たとえば、ソースデバイス12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14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。

30

【 0 0 2 2 】

図1の図示されるシステム10は一例にすぎない。ビデオデータを処理するための技法は、任意のデジタルビデオ符号化および/または復号デバイスまたは装置によって実行され得る。概して、本開示の技法はビデオ符号化デバイスおよびビデオ復号デバイスによって実行されるが、技法はまた、典型的には「コーデック」と呼ばれる複合ビデオエンコーダ/デコーダによって実行され得る。ソースデバイス12および宛先デバイス14は、ソースデバイス12が宛先デバイス14への送信のために符号化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。いくつかの例では、ソースデバイス12および宛先デバイス14は、ソースデバイス12および宛先デバイス14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、実質的に対称的様式で動作する。したがって、システム10は、たとえば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスティング、またはビデオ電話のための、ソースデバイス12と宛先デバイス14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。

40

50

【0023】

ソースデバイス12のビデオソース18は、ビデオカメラ、以前にキャプチャされた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データを受信するためのビデオフィードインターフェースなどの、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18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。ソースデバイス12は、ビデオデータを記憶するように構成された1つまたは複数のデータ記憶媒体(たとえば、記憶媒体20)を備え得る。本開示で説明する技法は、ビデオコーディングに概して適用可能であり得、ワイヤレスおよび/または有線の適用例に適用され得る。各々の場合において、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が、ビデオエンコーダ22によって符号化され得る。出力インターフェース24は、符号化ビデオ情報をコンピュータ可読媒体16に出力することができる。

10

【0024】

宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化ビデオデータを受信し得る。コンピュータ可読媒体16は、ソースデバイス12から宛先デバイス14に符号化ビデオデータを移動することが可能な任意のタイプの媒体またはデバイスを備え得る。いくつかの例では、コンピュータ可読媒体16は、ソースデバイス12がリアルタイムで宛先デバイス14に符号化ビデオデータを直接送信することを可能にする通信媒体を含む。符号化ビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス14に送信され得る。通信媒体は、無線周波数(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12から宛先デバイス14への通信を容易にするために有用であり得る任意の他の機器を含む場合がある。宛先デバイス14は、符号化ビデオデータおよび復号ビデオデータを記憶するように構成された1つまたは複数のデータ記憶媒体を備え得る。

20

【0025】

いくつかの例では、符号化データ(たとえば、符号化ビデオデータ)は、出力インターフェース24から記憶デバイスに出力され得る。同様に、符号化されたデータは、入力インターフェース26によって記憶デバイスからアクセスされ得る。記憶デバイスは、ハードドライブ、Blu-ray(登録商標)ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性メモリもしくは不揮発性メモリ、サーバ、または符号化ビデオデータを記憶するための任意の他の好適なデジタル記憶媒体などの、分散されるかまたはローカルにアクセスされる様々なデータ記憶媒体のいずれかを含み得る。さらなる例では、記憶デバイスは、ソースデバイス12によって生成された符号化されたビデオを記憶し得るファイルサーバまたは別の中間記憶デバイスに対応し得る。宛先デバイス14は、ストリーミングまたはダウンロードを介して記憶デバイスからの記憶されたビデオデータにアクセスし得る。ファイルサーバは、符号化ビデオデータを記憶するとともにその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、任意のタイプのサーバであってよい。例示的なファイルサーバは、(たとえば、ウェブサイトのための)ウェブサーバ、FTPサーバ、ネットワークアタッチストレージ(NAS)デバイス、またはローカルディスクドライブを含む。宛先デバイス14は、インターネット接続を含む任意の標準的なデータ接続を通して符号化ビデオデータにアクセスし得る。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、DSL、ケーブルモデムなど)、またはファイルサーバ上に記憶された符号化ビデオデータにアクセスするのに適した両方の組合せを含み得る。記憶デバイスからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組合せであり得る。

30

40

【0026】

本開示の技法は、オーバージエアテレビジョンブロードキャスト、ケーブルテレビジ

50

ン送信、衛星テレビジョン送信、動的適応ストリーミングオーバーHTTP(DASH:dynamic adaptive streaming over HTTP)などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化され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。いくつかの例では、システム10は、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト、および/またはビデオ電話などの適用例をサポートするために、一方向または双方向のビデオ送信をサポートするように構成され得る。

【0027】

コンピュータ可読媒体16は、ワイヤレスブロードキャストもしくは有線ネットワーク送信などの一時的媒体、またはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray(登録商標)ディスク、サーバ、もしくは他のコンピュータ可読媒体などの記憶媒体(すなわち、非一時的記憶媒体)を含み得る。いくつかの例では、ネットワークサーバ(図示せず)が、たとえば、ネットワーク送信を介して、ソースデバイス12から符号化ビデオデータを受信してよく、符号化ビデオデータを宛先デバイス14に提供してよい。同様に、ディスクスタンピング設備などの媒体製造設備のコンピューティングデバイスが、ソースデバイス12から符号化ビデオデータを受信してよく、符号化ビデオデータを含むディスクを製造してよい。したがって、コンピュータ可読媒体16は、様々な例において、様々な形態の1つまたは複数のコンピュータ可読媒体を含むものと理解されてよい。

【0028】

宛先デバイス14の入力インターフェース26は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ブロックおよび他のコーディングされたユニット、たとえば、ピクチャグループ(GOP:groups of pictures)の特性および/または処理を記述するシンタックス要素を含む、ビデオエンコーダ22によって定義されるとともに、ビデオデコーダ30によって同様に使用されるシンタックス情報を含み得る。記憶媒体28は、入力インターフェース26によって受信された符号化ビデオデータを記憶し得る。ディスプレイデバイス32は、復号ビデオデータをユーザに表示する。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のいずれかを備え得る。

【0029】

ビデオエンコーダ22およびビデオデコーダユニット30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回路またはデコーダ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内にソフトウェア用の命令を記憶することができ、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において命令を実行することができる。ビデオエンコーダ22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内に含まれてよく、そのいずれもが、それぞれのデバイスにおいて複合エンコーダ/デコーダ(コーデック)の一部として統合されてよい。

【0030】

いくつかの例では、ビデオエンコーダ22およびビデオデコーダ30は、ビデオコーディング規格に従って動作し得る。例示的なビデオコーディング規格は、限定はしないが、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 Visual、ITU-T H.262またはISO/IEC MPEG-2 Visual、ITU-T H.263、ISO/IEC MPEG-4 Visual、およびITU-T H.264(ISO/IEC MPEG-4 AVCとしても知られている)を含む。ビデオコーディング規格、高効率ビデオコーディング(HE

10

20

30

40

50

VC)またはITU-T H.265が、その範囲拡張、スクリーンコンテンツコーディング拡張、3Dビデオコーディング(3D-HEVC)拡張、およびマルチビュー拡張(MV-HEVC)ならびにスケーラブル(SHVC)拡張を含めて、ITU-Tビデオコーディングエキスパートグループ(VCEG:Video Coding Experts Group)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:Motion Picture Experts Group)のビデオコーディング共同研究部会(JCT-VC:Joint Collaboration Team on Video Coding)によって最近開発された。ビデオエンコーダ22およびビデオデコーダ30は、ジョイントビデオエクスプロレーションチーム(JVET:Joint Video Exploration Team)グループによって開発されているビデオコーディング規格など、将来のビデオコーディング規格に従って動作するように構成されてもよい。JEMソフトウェアは、HEVCモデル(HM)ソフトウェアに基づき、JVETに対する参照ソフトウェアである。

10

【0031】

HEVCおよび他のビデオコーディング仕様では、ビデオシーケンスは、典型的には、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、本明細書では「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ームであることがあり、ルーマサンプルのアレイしか含まないことがある。

【0032】

さらに、HEVCおよび他のビデオコーディング仕様では、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22は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を備え得る。モノクロームピクチャ、または3つの別個の色平面を有するピクチャでは、CTUは、単一の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を備え得る。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)と呼ばれる場合もある。HEVCのCTUは、H.264/AVCなどの他の規格のマクロブロックに概して類似し得る。しかしながら、CTUは、必ずしも特定のサイズに限定されとは限らず、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、ラスタ走査順序で連続的に順序付けられた整数個のCTUを含み得る。

20

30

【0033】

HEVCに従って動作する場合、コーディングされたCTUを生成するために、ビデオエンコーダ22は、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することができ、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を備え得る。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、CUは、単一のコーディングブロック、および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を備え得る。

40

【0034】

ビットストリーム内のシンタックスデータも、CTUに関するサイズを定義し得る。スライスは、コーディング順序において連続するいくつかのCTUを含む。ビデオフレームまたはピクチャは、1つまたは複数のスライスに区分され得る。上述のように、各ツリーブロックは、4分木に従ってコーディングユニット(CU)に分割され得る。概して、4分木データ構造はCU当たり1つのノードを含み、ルートノードはツリーブロックに対応する。CUが4つ

50

のサブCUに分割された場合、CUに対応するノードは4つのリーフノードを含み、リーフノードの各々はサブCUのうちの1つに対応する。

【0035】

4分木データ構造の各ノードは、対応するCUにシンタックスデータを提供し得る。たとえば、4分木内のノードは、ノードに対応するCUがサブCUに分割されるかどうかを示す分割フラグを含み得る。CUのためのシンタックス要素は再帰的に定義されることがあり、CUがサブCUに分割されるかどうかに依存することがある。CUがさらに分割されない場合、それはリーフCUと呼ばれる。CUのブロックがさらに分割される場合、それは、概して、非リーフCUと呼ばれ得る。本開示のいくつかの例では、リーフCUの4つのサブCUはまた、元のリーフCUの明示的な分割がなくてもリーフCUと呼ばれる。たとえば、 16×16 サイズにおけるCUがさらに分割されない場合、 16×16 CUはまったく分割されなかったが、4つの 8×8 サブCUもリーフCUと呼ばれ得る。

【0036】

CUは、CUがサイズの区別をもたないことを除いて、H.264規格のマクロブロックと同様の目的を有する。たとえば、ツリーブロックは4つの子ノード(サブCUとも呼ばれる)に分割されてよく、各子ノードは、次には親ノードであってよく別の4つの子ノードに分割され得る。最後の、4分木のリーフノードと呼ばれる分割されていない子ノードは、リーフCUとも呼ばれるコーディングノードを備える。コーディングされたビットストリームに関連するシンタックスデータは、最大CU深度と呼ばれる、ツリーブロックが分割され得る最大回数を定義し得、コーディングノードの最小サイズも定義し得る。ツリーブロック構造の深度は、ブロックが分割されている回数を示し得る。たとえば、深度0は、いずれかの分割の前のブロックに関係し得、深度1は、親ブロックの1つの分割から生成されるブロックに関係し得、深度2は、深度1におけるブロックの1つの分割から生成されるブロックに関係し得る、などである。ビットストリームはまた、最小コーディングユニット(SCU)を定義し得る。本開示は、HEVCのコンテキストにおけるCU、PU、もしくはTUのいずれか、または、他の規格のコンテキストにおける同様のデータ構造(たとえば、JEMにおけるコーディングユニット、H.264/AVCにおけるマクロブロックおよびそのサブブロックなど)を指すために、「ブロック」という用語を使用する。

【0037】

CUは、コーディングノード、ならびにコーディングノードに関連する予測ユニット(PU)および変換ユニット(TU)を含む。CUのサイズは、コーディングノードのサイズに対応し、いくつかの例では、形状が正方形であり得る。HEVCの例では、CUのサイズは、 8×8 ピクセルから、最大 64×64 ピクセル以上のツリーブロックのサイズまでの範囲であってもよい。各CUは、1つまたは複数のPUおよび1つまたは複数のTUを含み得る。CUに関連するシンタックスデータは、たとえば、1つまたは複数のPUへのCUの区分を記述し得る。区分モードは、CUがスキップモード符号化または直接モード符号化されているのか、イントラ予測モード符号化されているのか、またはインター予測モード符号化されているのかの間で異なり得る。PUは、形状が非正方形であるように区分されてよい。CUに関連するシンタックスデータはまた、たとえば、4分木に従った1つまたは複数のTUへのCUの区分を記述してもよい。TUは、形状が正方形または非正方形(たとえば、矩形)であることが可能である。

【0038】

HEVC規格は、TUに従った変換を可能にする。TUは、異なるCUに対して異なり得る。TUは、典型的には、区分されたLCUについて定義された所与のCU内のPUのサイズに基づいてサイズが決められるが、これは必ずしもそうではないことがある。TUは、典型的には、PUと同じサイズであるか、またはPUよりも小さい。いくつかの例では、CUに対応する残差サンプルは、「残差4分木」(RQT)と呼ばれることがある4分木構造を使用して、より小さい単位に再度分割されてよい。RQTのリーフノードは、TUと呼ばれることがある。TUに関連するピクセル差分値は、量子化されてもよい変換係数を生成するために変換されてもよい。

【0039】

リーフCUは、1つまたは複数のPUを含み得る。概して、PUは、対応するCUのすべてまた

10

20

30

40

50

は一部分に対応する空間領域を表し、PUのための参照サンプルを取り出すためのデータを含む場合がある。その上、PUは予測に関連するデータを含む。たとえば、PUがイントラモード符号化されるとき、PUに対するデータは、PUに対応するTU用のイントラ予測モードを記述するデータを含み得るRQT内に含まれてよい。別の例として、PUがインターモード符号化されるとき、PUは、PUに関する1つまたは複数の動きベクトルを定義するデータを含む場合がある。PUに関する動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度(たとえば、4分の1ピクセル精度もしくは8分の1ピクセル精度)、動きベクトルが指す参照ピクチャ、および/または動きベクトルのための参照ピクチャリスト(たとえば、リスト0、リスト1、もしくはリストC)を記述してもよい。

10

【0040】

1つまたは複数のPUを有するリーフCUはまた、1つまたは複数のTUを含み得る。TUは、上で論じたように、RQT(TU4分木構造とも呼ばれる)を使用して指定され得る。たとえば、分割フラグは、リーフCUが4つの変換ユニットに分割されるかどうかを示し得る。いくつかの例では、各変換ユニットは、さらなるサブTUにさらに分割され得る。TUは、それ以上分割されないとき、リーフTUと呼ばれ得る。概して、イントラコーディングの場合、リーフCUに属するすべてのリーフTUは、同じイントラ予測モードから作り出された残差データを含む。すなわち、同じイントラ予測モードは、概して、リーフCUのすべてのTU内で変換されることになる予測される値を計算するために適用される。イントラコーディングの場合、ビデオエンコーダ22は、各リーフTUに対する残差値を、TUに対応するCUの部分と元のブロックとの間の差としてイントラ予測モードを使用して計算してもよい。TUは、必ずしもPUのサイズに限定されるとは限らない。したがって、TUは、PUよりも大きくてもまたは小さくてもよい。イントラコーディングの場合、PUは、同じCUに対して対応するリーフTUとコロケートされてよい。いくつかの例では、リーフTUの最大サイズは、対応するリーフCUのサイズに対応し得る。

20

【0041】

その上、リーフCUのTUはまた、それぞれのRQT構造に関連付けられ得る。すなわち、リーフCUは、リーフCUがTUにどのように区分されるかを示す4分木を含み得る。TU4分木のルートノードは、概して、リーフCUに対応し、一方、CU4分木のルートノードは、概して、ツリーブロック(またはLCU)に対応する。

30

【0042】

上で論じたように、ビデオエンコーダ22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矩形(すなわち、正方形または非正方形)ブロックである。CUのPUは、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を予測するために使用されるシンタックス構造を備え得る。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、PUは、単一の予測ブロック、および予測ブロックを予測するために使用されるシンタックス構造を備え得る。ビデオエンコーダ22は、CUの各PUの予測ブロック(たとえば、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、予測ブロック(たとえば、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)を生成し得る。

40

【0043】

ビデオエンコーダ22は、イントラ予測またはインター予測を使用して、PU用の予測ブロックを生成し得る。ビデオエンコーダ22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22は、PUを含む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。

【0044】

ビデオエンコーダ22がCUの1つまたは複数のPUに対する予測ブロック(たとえば、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)を生成した後、ビデオエンコーダ22は、CUに対して1つまたは複数の残差ブロックを生成し得る。たとえば、ビデオエン

50

コード22は、CUの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内の各サンプルは、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内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22は、CUに対してCb残差ブロックを生成し得る。CUのCb残差ブロック内の各サンプルは、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内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22はまた、CUに対してCr残差ブロックを生成し得る。CUのCr残差ブロック内の各サンプルは、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内の対応するサンプルとの間の差分を示し得る。

【0045】

さらに、上記で論じたように、ビデオエンコーダ22は、4分木区分を使用して、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。変換ブロックとは、同じ変換が適用されるサンプルの矩形(たとえば、正方形または非正方形)ブロックである。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、およびそれらの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を備え得る。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変換ブロックを有し得る。TUの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。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、TUは、単一の変換ブロック、および変換ブロックの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を含み得る。

【0046】

ビデオエンコーダ22は、TUに対して係数ブロックを生成するために、1つまたは複数の変換をTUの変換ブロックに適用し得る。たとえば、ビデオエンコーダ22は、TUに対してルーマ係数ブロックを生成するために、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。ビデオエンコーダ22は、TUに対するCb係数ブロックを生成するために、TUのCb変換ブロックに1つまたは複数の変換を適用し得る。ビデオエンコーダ22は、TUに対するCr係数ブロックを生成するために、TUのCr変換ブロックに1つまたは複数の変換を適用し得る。

【0047】

いくつかの例では、ビデオエンコーダ22は、変換ブロックへの変換の適用をスキップする。そのような例では、ビデオエンコーダ22は、変換係数と同じ方法で残差サンプル値を扱い得る。したがって、ビデオエンコーダ22が変換の適用をスキップする例では、変換係数および係数ブロックの以下の議論は、残差サンプルの変換ブロックに適用可能であり得る。

【0048】

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22は、係数ブロックを量子化して、場合によっては、係数ブロックを表すために使用されるデータの量を低減し、潜在的に、さらなる圧縮を実現し得る。量子化は、概して、値の範囲が単一の値に圧縮されるプロセスを指す。たとえば、量子化は、定数によって値を除算し、次いで、最も近い整数に丸めることによって行われ得る。係数ブロックを量子化するために、ビデオエンコーダ22は、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22が係数ブロックを量子化した後、ビデオエンコーダ22は、量子化された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することができる。たとえば、ビデオエンコーダ22は、量子化された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: Context-Adaptive Binary Arithmetic Coding)または他のエントロピーコーディング技法を実行すること

10

20

30

40

50

ができる。

【 0 0 4 9 】

ビデオエンコーダ22は、コーディングされたピクチャの表現および関連するデータを形成するビットのシーケンスを含むビットストリームを出力することができる。したがって、ビットストリームは、ビデオデータの符号化表現を含む。ビットストリームは、ネットワーク抽象化レイヤ(NAL)ユニットのシーケンスを含み得る。NALユニットは、NALユニット内のデータのタイプの指示、および必要に応じてエミュレーション防止ビットが散在させられているローバイトシーケンスペイロード(RBSP:raw byte sequence payload)の形態でそのデータを含むバイトを含む、シンタックス構造である。NALユニットの各々は、NALユニットヘッダを含むことができ、RBSPをカプセル化し得る。NALユニットヘッダは、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指定される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ている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ビットを含む。

10

【 0 0 5 0 】

ビデオデコーダ30は、ビデオエンコーダ22によって生成されたビットストリームを受信し得る。ビデオデコーダ30は、ビットストリームを復号してビデオデータのピクチャを再構築し得る。ビットストリームを復号することの一部として、ビデオデコーダ30は、ビットストリームをパースしてビットストリームからシンタックス要素を取得し得る。ビデオデコーダ30は、ビットストリームから取得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築することができる。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22によって実行されるプロセスの逆であってよい。たとえば、ビデオデコーダ30は、現在CUのPUの予測ブロックを判定するために、PUの動きベクトルを使用することができる。加えて、ビデオデコーダ30は、現在CUのTUの係数ブロックを逆量子化し得る。ビデオデコーダ30は、係数ブロックに対して逆変換を実行して、現在CUのTUの変換ブロックを再構築し得る。ビデオデコーダ30は、現在CUのPUの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築し得る。ピクチャのCUごとにコーディングブロックを再構築することによって、ビデオデコーダ30はピクチャを再構築し得る。

20

30

【 0 0 5 1 】

HEVCの一般的な概念および一定の設計態様について、ブロック区分に関する技法に焦点を当てて、以下で説明する。HEVCでは、スライス内の最大のコーディングユニットはCTBと呼ばれる。CTBは、そのノードがコーディングユニットである4分木構造に従って分割される。4分木構造内の複数のノードは、リーフノードおよび非リーフノードを含む。リーフノードは、ツリー構造内に子ノードを有さない(すなわち、リーフノードはさらに分割されない)。非リーフノードは、ツリー構造のルートノードを含む。ルートノードは、ビデオデータの最初のビデオブロック(CTB)に対応する。複数のノードの各それぞれの非ルートノードごとに、それぞれの非ルートノードは、それぞれの非ルートノードのツリー構造内の親ノードに対応するビデオブロックのサブブロックであるビデオブロックに対応する。複数の非リーフノードの各それぞれの非リーフノードは、ツリー構造内に1つまたは複数の子ノードを有する。

40

【 0 0 5 2 】

CTBのサイズは、HEVCメインプロファイルにおいて16×16から64×64の範囲である(ただし、技術的には8×8CTBサイズをサポートすることができる)。CTBは、W. J. Han他、「Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools」、IEEE Transaction on Circuits and Systems for Video Technology、vol. 20、no.12、1709～1720頁、2010年12月で記述され、図2に示すように、4分木様式で再帰的にCUに分割され得る。図2に示すように、区分の各レベルは、4つのサブブロックへの4分木分割である。黒いブロックは、リーフノード(すなわち

50

、さらに分割されないブロック)の一例である。

【 0 0 5 3 】

いくつかの例では、CUは、CTBと同じサイズであり得るが、CUは 8×8 程度の小ささであってよい。各CUは、たとえば、イントラコーディングモードまたはインターコーディングモードであり得る、1つのコーディングモードでコーディングされる。スクリーンコンテンツに対するコーディングモード(たとえば、イントラブロックコピーモード、パレットベースのコーディングモードなど)を含めて、他のコーディングモードも可能である。CUがインターコーディングされる(すなわち、インターモードが適用される)とき、CUは予測ユニット(PU)にさらに区分され得る。たとえば、CUは2つまたは4つのPUに区分され得る。別の例では、さらなる区分が適用されないとき、CU全体が単一のPUとして扱われる。HEVCの例では、1つのCU内に2つのPUが存在するとき、それらのPUは、CUの半分のサイズの矩形であってもよく、またはCUの $1/4$ もしくは $3/4$ のサイズを有する2つの矩形サイズであってよい。

10

【 0 0 5 4 】

図3に示すように、HEVCにおいて、インター予測モードでコーディングされるCUに対して8つの予測モード、すなわち、 $PART_2N \times 2N$ 、 $PART_2N \times N$ 、 $PART_N \times 2N$ 、 $PART_N \times N$ 、 $PART_2N \times nU$ 、 $PART_2N \times nD$ 、 $PART_nL \times 2N$ 、および $PART_nR \times 2N$ がある。図3に示すように、区分モード $PART_2N \times 2N$ でコーディングされたCUはさらに分割されない。すなわち、CU全体が単一のPU(PU0)として扱われる。区分モード $PART_2N \times N$ でコーディングされたCUは、2つのPU(PU0およびPU1)に対称的に水平に分割される。区分モード $PART_N \times 2N$ でコーディングされたCUは、2つのPUに対称的に垂直に分割される。区分モード $PART_N \times N$ でコーディングされたCUは、4つの等しいサイズのPU(PU0、PU1、PU2、PU3)に対称的に分割される。

20

【 0 0 5 5 】

区分モード $PART_2N \times nU$ でコーディングされたCUは、CUのサイズの $1/4$ を有する1つのPU0(上側PU)およびCUのサイズの $3/4$ を有する1つのPU1(下側PU)に非対称的に水平に分割される。区分モード $PART_2N \times nD$ でコーディングされたCUは、CUのサイズの $3/4$ を有する1つのPU0(上側PU)およびCUのサイズの $1/4$ を有する1つのPU1(下側PU)に非対称的に水平に分割される。区分モード $PART_nL \times 2N$ でコーディングされたCUは、CUのサイズの $1/4$ を有する1つのPU0(左PU)およびCUのサイズの $3/4$ を有する1つのPU1(右PU)に非対称的に垂直に分割される。区分モード $PART_nR \times 2N$ でコーディングされたCUは、CUのサイズの $3/4$ を有する1つのPU0(左PU)およびCUのサイズの $1/4$ を有する1つのPU1(右PU)に非対称的に垂直に分割される。

30

【 0 0 5 6 】

CUがインターコーディングされるとき、PUごとに動き情報の1つのセット(たとえば、動きベクトル、予測方向、および参照ピクチャ)が存在する。加えて、各PUは、動き情報のセットを導出するために一意のインター予測モードでコーディングされる。しかしながら、2つのPUが一意にコーディングされる場合ですら、2つのPUは、いくつかの状況において、依然として同じ動き情報を有し得ることを理解されたい。

【 0 0 5 7 】

J. An他、「Block partitioning structure for next generation video coding」、国際電気通信連合、COM16-C966、2015年9月(以後、「VCEG提案COM16-C966」)において、HEVCを超えた将来のビデオコーディング規格のための4分木2分木(QTBT)区分技法が提案された。提案されたQTBT構造は、使用されるHEVCにおいて4分木構造より効率的であることを、シミュレーションが示している。

40

【 0 0 5 8 】

VCEG提案COM16-C966の提案されたQTBT構造では、CTBはまず4分木区分技法を使用して区分され、ノードが最小の許容される4分木リーフノードサイズに達するまで1つのノードの4分木分割が繰り返され得る。最小の許容される4分木リーフノードサイズは、シンタックス要素MinQTSIZEの値によってビデオデコードに示され得る。4分木リーフノードサイズが最大の許容される2分木ルートノードサイズ(たとえば、シンタックス要素MaxBTSIZEによ

50

り示されるような)より大きくない場合、4分木リーフノードはさらに、2分木区分を使用して区分され得る。ノードが最小の許容される2分木リーフノードサイズ(たとえば、シンタックス要素MinBTSIZEにより示されるような)または最大の許容される2分木深度(たとえば、シンタックス要素MaxBTDDepthにより示されるような)に達するまで、1つのノードの2分木区分が繰り返され得る。VCEG提案COM16-C966は、2分木リーフノードを指すために「CU」という用語を使用する。VCEG提案COM16-C966では、CUは、さらなる区分を伴わない予測(たとえば、イントラ予測、インター予測など)および変換のために使用される。概して、QTBt技法によれば、対称的な水平の分割および対称的な垂直の分割という、2分木分割のための2つの分割タイプがある。各々の場合において、ブロックは、水平または垂直のいずれかに、ブロックを中心で分割することによって分割される。

10

【0059】

QTBt区分構造の一例では、CTUサイズは 128×128 として設定され(たとえば、 128×128 のルーマブロックおよび2つの対応する 64×64 のクロマブロック)、MinQTSIZEは 16×16 として設定され、MaxBTSIZEは 64×64 として設定され、MinBTSIZE(幅と高さの両方のための)は4として設定され、MaxBTDDepthは4として設定される。4分木リーフノードを生成するために、4分木区分がまずCTUに適用される。4分木リーフノードは、 16×16 (すなわち、MinQTSIZEは 16×16)から 128×128 (すなわち、CTUサイズ)のサイズを有し得る。QTBt区分の一例によれば、リーフ4分木ノードが 128×128 である場合、リーフ4分木ノードをさらに2分木によって分割することはできず、それは、リーフ4分木ノードのサイズがMaxBTSIZE(すなわち、 64×64)を超えるからである。それ以外の場合、リーフ4分木ノードはさらに2分木によって区分される。したがって、4分木リーフノードは2分木のルートノードでもあり、2分木深度を0として有する。MaxBTDDepth(たとえば、4)に達する2分木深度は、さらなる分割がないことを示唆する。MinBTSIZE(たとえば、4)に等しい幅を有する2分木ノードは、さらなる水平の分割がないことを示唆する。同様に、MinBTSIZEに等しい高さを有する2分木ノードは、さらなる垂直の分割がないことを示唆する。2分木のリーフノード(CU)はさらに、さらなる区分なしで(たとえば、予測プロセスおよび変換プロセスを実行することによって)処理される。

20

【0060】

図4Aは、QTBt区分技法を使用して区分されたブロック50(たとえば、CTB)の一例を示す。図4Aに示すように、QTBt区分技法を使用して、得られるブロックの各々は、各ブロックの中心を通过对称的に分割される。図4Bは、図4Aのブロック区分に対応するツリー構造を示す。図4Bの実線は4分木分割を示し、点線は2分木分割を示す。一例では、2分木の各分割(すなわち、非リーフ)ノードにおいて、実行される分割のタイプ(たとえば、水平または垂直)を示すために、シンタックス要素(たとえば、フラグ)がシグナリングされ、ここで0は水平の分割を示し、1は垂直の分割を示す。4分木分割では、4分木分割は常に等しいサイズの4つのサブブロックにブロックを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。

30

【0061】

図4Bに示すように、ノード70において、ブロック50は、QT区分を使用して、図4Aに示す4つのブロック51、52、53、および54に分割される。ブロック54はさらに分割されず、したがって、リーフノードである。ノード72において、ブロック51は、BT区分を使用して2つのブロックにさらに分割される。図4Bに示すように、ノード72には、垂直分割を示す1がつけられている。したがって、ノード72での分割は、ブロック57と、ブロック55と56の両方を含むブロックとをもたらす。ブロック55および56は、ノード74におけるさらなる垂直分割によって生成される。ノード76において、ブロック52は、BT区分を使用して2つのブロック58および59にさらに分割される。図4Bに示すように、ノード76には、水平分割を示す1がつけられている。

40

【0062】

ノード78において、ブロック53は、QT区分を使用して4つの等しいサイズのブロックに分割される。ブロック63および66は、このQT区分から生成され、さらに分割されない。ノ

50

ード80において、左上のブロックがまず垂直の2分木分割を使用して分割されて、ブロック60および右側の垂直ブロックをもたらす。次いで、右側の垂直ブロックが水平の2分木分割を使用してブロック61および62に分割される。ノード78における4分木分割から生成される右下のブロックはノード84において、水平の2分木分割を使用してブロック64および65に分割される。

【0063】

上記で説明したQTBT構造はHEVCで使用される4分木構造よりもより良いコーディング性能を示すが、QTBT構造は柔軟性に欠ける。たとえば、上記で説明したQTBT構造では、4分木ノードは2分木でさらに分割され得るが、2分木ノードを4分木を用いてさらに分割することはできない。別の例では、4分木および2分木は均等分割(すなわち、ブロックの中心を分割すること)のみを達成することができ、オブジェクトが分割されるべきブロックの中心にあるとき、これは効率的ではない。したがって、将来のビデオコーディング規格に対してQTBTのコーディング性能は十分でない場合がある。

【0064】

上述の問題に対処するために、その両方が参照により本明細書に組み込まれている、2017年1月12日に出版し米国特許公開第20170208336号、および2017年3月20日に出版した米国特許公開第20170272782号は、マルチタイプツリー(MTT)区分構造の様々な例について説明している。MTT区分構造に従って、2分木、対称的な中心-側部3分木(triple-tree)、および4分木などの複数木のタイプを用いて、木ノードをさらに分割することができる。マルチタイプツリー構造が4分木2分木構造よりはるかに効率的であることをシミュレーションが示した。

【0065】

CTUに対してより柔軟な区分をより良く達成するために、QT、BT、および/またはQTBTベースのCU構造を置換するためのMTTベースのCU構造が提案される。本開示のMTT区分構造は、依然として再帰的ツリー構造である。しかしながら、複数の異なる区分構造(たとえば、3つ以上)が使用される。たとえば、本開示のMTT技法によれば、ツリー構造の各深度において、3つ以上の異なる区分構造が使用され得る。このコンテキストで、ツリー構造内のノードの深度は、ノードからツリー構造のルートまでの経路の長さ(たとえば、分割の数)を指すことがある。

【0066】

本開示の技法による一例では、ビデオエンコーダ22および/またはビデオデコーダ30は、ビデオデータのピクチャを受信し、3つ以上の異なる区分構造を使用して、ビデオデータのピクチャを複数のブロックに区分し、ビデオデータのピクチャの複数のブロックを再構成/符号化するように構成され得る。一例では、ビデオデータのピクチャを区分することは、3つ以上の異なる区分構造を使用して、ビデオデータのピクチャを複数のブロックに区分することを含み、3つ以上の異なる区分構造のうちの少なくとも3つが、ビデオデータのピクチャがどのように区分されるかを表すツリー構造の各深度において使用され得る。一例では、3つ以上の異なる区分構造は3分木区分構造を含み、ビデオエンコーダ22および/またはビデオデコーダ30は、3分木区分タイプの3分木区部構造を使用して、ビデオデータの複数のブロックのうちの1つを区分するように構成され得、3分木区分構造は、複数のブロックのうちの1つの中心を通して分割せずに、複数のブロックのうちの1つを3つのサブブロックに分割する。本開示のさらなる例では、3つ以上の異なる区分構造は、4分木区分構造および2分木区分構造をさらに含む。

【0067】

したがって、一例では、ビデオエンコーダ22は、ビデオデータの最初のビデオブロック(たとえば、コーディングツリーブロックすなわちCTU)の符号化表現を生成し得る。最初のビデオブロックの符号化表現を生成する一部として、ビデオエンコーダ22は、複数のノードを含むツリー構造を判定する。たとえば、ビデオエンコーダ22は、本開示のMTT区分構造を使用して、ツリーブロックを区分することができる。

【0068】

10

20

30

40

50

MTT区分構造内の複数のノードは、複数のリーフノードおよび複数の非リーフノードを含む。リーフノードは、ツリー構造内に子ノードを有さない。非リーフノードはツリー構造のルートノードを含む。ルートノードは、最初のビデオブロックに対応する。複数のノードのそれぞれの非ルートノードごとに、それぞれの非ルートノードは、それぞれの非ルートノードのツリー構造内の親ノードに対応するビデオブロックのサブブロックであるビデオブロック(たとえば、コーディングブロック)に対応する。複数の非リーフノードの各それぞれの非リーフノードは、ツリー構造内に1つまたは複数の子ノードを有する。いくつかの例では、ピクチャ境界における非リーフノードは、強制分割により1つの子ノードのみを含む場合があり、子ノードのうちの1つは、ピクチャ境界外部のブロックに対応する。

10

【0069】

本開示の技法によれば、ツリー構造の各深度レベルにおけるツリー構造のそれぞれの非リーフノードごとに、それぞれの非リーフノードに対して複数の許容された分割パターン(たとえば、区分構造)が存在する。たとえば、ツリー構造の各深度に対して許容される3つ以上の区分構造が存在し得る。ビデオエンコーダ22は、複数の許容可能な区分構造のうちの1つに従って、それぞれの非リーフノードに対応するビデオブロックをそれぞれの非リーフノードの子ノードに対応するビデオブロックに区分するように構成され得る。複数の許容される区分構造の各それぞれの許容される区分構造は、それぞれの非リーフノードに対応するビデオブロックをそれぞれの非リーフノードの子ノードに対応するビデオブロックに区分する様々な方法に対応し得る。さらに、この例では、ビデオエンコーダ22は、ビデオデータの符号化表現を含むビットストリーム内に最初のビデオブロックの符号化表現を含み得る。

20

【0070】

同様の例では、ビデオデコーダ30は、複数のノードを含むツリー構造を判定することができる。前の例にあるように、複数のノードは、複数のリーフノードおよび複数の非リーフノードを含む。リーフノードは、ツリー構造内に子ノードを有さない。非リーフノードはツリー構造のルートノードを含む。ルートノードはビデオデータの最初のビデオブロックに対応する。複数のノードの各それぞれの非ルートノードごとに、それぞれの非ルートノードは、それぞれの非ルートノードのツリー構造内の親ノードに対応するビデオブロックのサブブロックであるビデオブロックに対応する。複数の非リーフノードの各それぞれの非リーフノードは、ツリー構造内に1つまたは複数の子ノードを有する。ツリー構造の各深度レベルにおけるツリー構造の各それぞれの非リーフノードごとに、それぞれの非リーフノードに対して複数の許容可能な分割パターンがあり、それぞれの非リーフノードに対応するビデオブロックは、複数の許容可能な分割パターンのうちの1つに従って、それぞれの非リーフノードの子ノードに対応するビデオブロックに区分される。複数の許容される分割パターンの各それぞれの許容される分割パターンは、それぞれの非リーフノードに対応するビデオブロックをそれぞれの非リーフノードの子ノードに対応するビデオブロックに区分する様々な方法に対応し得る。さらに、この例では、ツリー構造の各(または、少なくとも1つの)それぞれのリーフノードに対して、ビデオデコーダ30は、それぞれのリーフノードに対応するビデオブロックを再構成する。

30

40

【0071】

いくつかのそのような例では、ルートノード以外のツリー構造の各それぞれの非リーフノードごとに、それぞれの非リーフノードに対して複数の許容される分割パターン(たとえば、区分構造)は、それに従ってそれぞれの非リーフノードの親ノードに対応するビデオブロックがそれぞれの非リーフノードの親ノードの子ノードに対応するビデオブロックに区分される区分構造とは無関係である。

【0072】

本開示の他の例では、ツリー構造の各深度において、ビデオエンコーダ22は、3つのさらなる区分構造のうちの1つの中から特定の区分タイプを使用してサブツリーをさらに分割するように構成され得る。たとえば、ビデオエンコーダ22は、QT、BT、3分木(TT)、お

50

よび他の区分構造から特定の区分タイプを判定するように構成され得る。一例では、QT区分構造は、正方形4分木および矩形4分木の区分タイプを含み得る。ビデオエンコーダ22は、正方形4分木区分を使用して、ブロックを分割することによって、正方形ブロックの中心を、水平および垂直の両方に、4つの等しいサイズの正方形ブロックに区分することができる。同様に、ビデオエンコーダ22は、矩形4分木区分を使用して、矩形ブロックを分割することによって、矩形(たとえば、非正方形)ブロックの中心を、水平および垂直の両方に、4つの等しいサイズの矩形ブロックに区分することができる。

【0073】

BT区分構造は、水平対称2分木、垂直対称2分木、水平非対称2分木、および垂直非対称2分木の区分タイプを含み得る。水平対称2分木区分タイプの場合、ビデオエンコーダ22は、ブロックの中心を同じサイズの2つの対称ブロックに水平に分割するように構成され得る。垂直対称2分木区分タイプの場合、ビデオエンコーダ22は、ブロックの中心を同じサイズの2つの対称ブロックに垂直に分割するように構成され得る。水平非対称2分木区分タイプの場合、ビデオエンコーダ22は、異なるサイズの2つのブロックにブロックを水平に分割するように構成され得る。たとえば、図3のPART_2N×nUまたはPART_2N×nD区分タイプにあるように、1つのブロックは、親ブロックの1/4のサイズであってよく、他のブロックは、親ブロックの3/4のサイズであってよい。垂直非対称2分木区分タイプの場合、ビデオエンコーダ22は、異なるサイズの2つのブロックにブロックを垂直に分割するように構成され得る。たとえば、図3のPART_nL×2NまたはPART_nR×2N区分タイプにあるように、1つのブロックは、親ブロックの1/4のサイズであってよく、他のブロックは、親ブロックの3/4のサイズであってよい。

【0074】

他の例では、非対称2分木区分タイプは、親ブロックを異なるサイズの部分に分割することができる。たとえば、1つのサブブロックは、親ブロックの3/8であってよく、他のサブブロックは、親ブロックの5/8であってよい。当然、そのような区分タイプは、垂直または水平のいずれかであってよい。

【0075】

TT区分構造はブロックの中心を分割しないという点で、TT区分構造は、QTまたはBT構造のTT区分構造とは異なる。ブロックの中心領域は同じサブブロック内で一緒に残る。4つのブロックを生成するQT、または2つのブロックを生成する2分木とは異なり、TT区分構造による分割は、3つのブロックを生成する。TT区分構造による例示的な区分タイプは、水平対称3分木、垂直対称3分木、水平非対称3分木、および垂直非対称3分木の区分タイプを含み得る。水平対称3分木区分タイプの場合、ビデオエンコーダ22は、ブロックの中心を分割せずに、ブロックを3つのブロックに水平に分割するように構成され得る。水平対称3分木区分に従って分割されるとき、中心のサブブロックの上下のブロックはミラーであり、すなわち、それらのブロックは同じサイズである。ブロックが3で直接分割可能である(たとえば、12個のサンプルの高さ)場合、中心ブロックは、上側ブロックおよび下側ブロックと同じサイズであり得る。ブロックが3で直接分割可能でない(たとえば、8つのサンプルの高さ)場合、中心ブロックは、上側ブロックおよび下側ブロックと異なるサイズであり得る。たとえば、8つのサンプルの高さのブロックの場合、上側および下側のブロックは、3つのサンプルの高さであり得、中心ブロックは2つのサンプルの高さであり得る。別の例では、8つのサンプルの高さのブロックの場合、上側および下側のブロックは、2つのサンプルの高さであり得、中心ブロックは4つのサンプルの高さであり得る。図5Eは、水平3分木区分の一例を示す。

【0076】

垂直対称3分木区分タイプの場合、ビデオエンコーダ22は、ブロックの中心を分割せずに、ブロックを3つのブロックに垂直に分割するように構成され得る。垂直対称3分木区分に従って分割されるとき、中心のサブブロックの左または右に対するブロックはミラーであり、すなわち、それらのブロックは同じサイズである。ブロックが3で直接分割可能である(たとえば、12個のサンプルの幅)場合、中心ブロックは、左ブロックおよび右ブロッ

クと同じサイズであり得る。ブロックが3で直接分割可能でない(たとえば、8つのサンプルの幅)場合、中心ブロックは、左ブロックおよび右ブロックと異なるサイズであり得る。たとえば、8つのサンプルの幅のブロックの場合、左ブロックおよび右ブロックは、3つのサンプルの幅であり得、中心ブロックは2つのサンプルの幅であり得る。別の例では、8つのサンプルの幅のブロックの場合、左ブロックおよび右ブロックは、2つのサンプルの幅であり得、中心ブロックは4つのサンプルの幅であり得る。図5Dは、垂直3分木区分の一例を示す。

【0077】

水平非対称3分木区分タイプの場合、ビデオエンコーダ22は、対称的なミラーではない3つのブロックにブロックを水平に分割するように構成され得る。いくつかの例では、水平非対称3分木区分タイプは、ブロックの中心を分割することができ、他の例では、水平非対称3分木区分タイプは、ブロックの中心を分割しない。垂直非対称3分木区分タイプの場合、ビデオエンコーダ22は、対称的なミラーではない3つのブロックにブロックを垂直に分割するように構成され得る。いくつかの例では、垂直非対称3分木区分タイプは、ブロックの中心を分割することができ、他の例では、垂直非対称3分木区分タイプは、ブロックの中心を分割しない。

【0078】

ブロックが(たとえば、サブツリーノードにおいて)非対称3分木区分タイプに分割される例では、ビデオエンコーダ22および/またはビデオデコーダ30は、3つの区分のうちの2つが同じサイズを有するように制限を適用し得る。そのような制限は、ビデオエンコーダ22が、ビデオデータを符号化するときに準拠しなければならない制限に対応し得る。さらに、いくつかの例では、ビデオエンコーダ22およびビデオデコーダ30は、それにより非対称3分木区分タイプに従って分割するときに2つの区分のエリアの和が残りの区分のエリアに等しくなる制限を適用し得る。たとえば、ビデオエンコーダ22は、ツリー構造のノードに対応するビデオブロックが非対称3分木パターンに従って区分されるとき、ノードが第1の子ノード、第2の子ノード、および第3の子ノードを有することを指定する規制に準拠する最初のビデオブロックの符号化表現を、ビデオエンコーダ22は生成することができるか、またはビデオデコーダ30は受信することができ、第2の子ノードは、第1および第3の子ノードに対応するビデオブロック同士の間のビデオブロックに対応し、第1および第3の子ノードに対応するビデオブロックは同じサイズを有し、第1および第3の子ノードに対応するビデオブロックのサイズの和は、第2の子ノードに対応するビデオブロックのサイズに等しい。

【0079】

本開示のいくつかの例では、ビデオエンコーダ22は、QT、BT、およびTTの区分構造の各々に対する前述の区分タイプのすべての中から選択するように構成され得る。他の例では、ビデオエンコーダ22は、前述の区分タイプのサブセットの中からのみ区分タイプを判定するように構成され得る。たとえば、4分木構造の一定のブロックサイズに対して、または一定の深度に対して、上で論じた区分タイプ(または、他の区分タイプ)のサブセットが使用され得る。サポートされる区分タイプのサブセットは、ビデオデコーダ30が使用するためにビットストリーム内でシグナリングされ得るか、またはビデオエンコーダ22およびビデオデコーダ30が、何のシグナリングもなしに、サブセットを判定することができるよう事前に定義され得る。

【0080】

他の例では、サポートされる区分タイプの数は、すべてのCTU内のすべての深度に対して固定され得る。すなわち、ビデオエンコーダ22およびビデオデコーダ30は、CTUの任意の深度に対して同じ数の区分タイプを使用するように事前構成され得る。他の例では、サポートされる区分タイプの数は、異なってよく、深度、スライスタイプ、または他の以前にコーディングされた情報に依存し得る。一例では、ツリー構造の深度0または深度1において、QT区分構造のみが使用される。1を超える深度において、QT、BT、およびTT区分構造の各々が使用され得る。

【0081】

いくつかの例では、ビデオエンコーダ22および/またはビデオデコーダ30は、ビデオフレームの一定の領域またはCTUの領域に対して重複区分を回避するために、サポートされる区分タイプに事前構成された制約を適用することができる。一例では、ブロックが非対称区分タイプで分割されるとき、ビデオエンコーダ22および/またはビデオデコーダ30は、現在ブロックから分割された最大サブブロックをさらに分割しないように構成され得る。たとえば、非対称区分タイプ(たとえば、図3のPART_2N×nU区分タイプ)に従って正方形ブロックが分割されるとき、すべてのサブブロック内の最大サブブロック(たとえば、図3のPART_2N×nU区分タイプのPU1)は、指摘されるリーフノードであり、さらに分割され得ない。しかしながら、より小さいサブブロック(たとえば、図3のPART_2N×nU区分タイプのPU0)は、さらに分割され得る。

10

【0082】

一定の領域に対する重複区分を回避するために、サポートされる区分タイプに関する制約が適用され得る別の例として、ブロックが非対称区分タイプで分割されるとき、現在ブロックから分割された最大サブブロックは、同じ方向にさらに分割され得ない。たとえば、正方形ブロックが非対称区分タイプ(たとえば、図3のPART_2N×nU区分タイプ)で分割されるとき、ビデオエンコーダ22および/またはビデオデコーダ30は、すべてのサブブロック内の大きいサブブロック(たとえば、図3のPART_2N×nU区分タイプのPU1)を水平方向に分割しないように構成され得る。しかしながら、この例において、ビデオエンコーダ22および/またはビデオデコーダ30は、PU1を垂直方向に再度分割することができる。

20

【0083】

さらに分割する際の困難を回避するために、サポートされる区分タイプに関する制約が適用され得る別の例として、ビデオエンコーダ22および/またはビデオデコーダ30は、ブロックの幅/高さが2の冪でないとき(たとえば、幅/高さが2、4、8、16などではないとき)ブロックを水平または垂直のいずれにも分割しないように構成され得る。

【0084】

上記の例は、ビデオエンコーダ22が、本開示の技法に従ってMTT区分を実行するようにどのように構成され得るかについて説明している。ビデオデコーダ30は、ビデオエンコーダ22によって実行されたのと同じMMT区分を適用することができる。いくつかの例では、ビデオデコーダ30において事前定義された規則の同じセットを適用することによって、ビデオデータのフレームがビデオエンコーダ22によってどのように区分されたかが判定され得る。しかしながら、多くの状況において、ビデオエンコーダ22は、コーディングされているビデオデータの特定のフレームに対するレートひずみの基準に基づいて、使用する特定の区分構造および区分タイプを判定することができる。したがって、ビデオデコーダ30が特定のフレームに対する区分を判定するために、ビデオエンコーダ22は、フレーム、およびフレームのCTUがどのように区分されることになるかを示すシンタックス要素を符号化ビットストリーム内でシグナリングすることができる。ビデオデコーダ30は、そのようなシンタックス要素をパースし、それに応じて、フレームおよびCTUを区分することができる。

30

【0085】

本開示の一例では、ビデオエンコーダ22は、サポートされる区分タイプの特定のサブセットをハイレベルシンタックス要素として、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)、スライスヘッダ、適応パラメータセット(APS)、または任意の他のハイレベルシンタックスパラメータセット内でシグナリングするように構成され得る。たとえば、最大数の区分タイプ、およびどのタイプがサポートされるかを事前定義すること、またはビットストリーム内でハイレベルシンタックス要素としてシーケンスパラメータセット(PPS)、ピクチャパラメータセット(PPS)、または任意の他のハイレベルシンタックスパラメータセット内でシグナリングすることができる。ビデオデコーダ30は、そのようなシンタックス要素を受信してパースして、使用される区分タイプの特定のサブセットおよび/またはサポートされる区分構造(たとえば、QT、BT、TTなど)およびタイプの最

40

50

大数を判定するように構成され得る。

【0086】

いくつかの例では、各深度において、ビデオエンコーダ22は、ツリー構造のその深度において使用される、選択された区分タイプを示すインデックスをシグナリングするように構成され得る。さらに、いくつかの例では、ビデオエンコーダ22は、各CUにおいてそのような区分タイプインデックスを適応的にシグナリングすることができ、すなわち、インデックスは異なるCUに対して異なってよい。たとえば、ビデオエンコーダ22は、1つまたは複数のレートひずみ計算に基づいて、区分タイプのインデックスを設定することができる。一例では、一定の条件が満たされる場合、区分タイプ(たとえば、区分タイプのインデックス)のシグナリングはスキップしてよい。たとえば、ビデオエンコーダ22は、特定の深度に関連して1つのサポートされる区分タイプのみが存在するとき、区分タイプのシグナリングをスキップされてよい。この例では、ピクチャ境界に近いとき、コーディングされるべき領域はCTUよりも小さくてよい。したがって、この例では、CTUは、ピクチャ境界に適合するように分割されるように強制され得る。一例では、強制分割に対して対称2分木のみが使用され、区分タイプはシグナリングされない。いくつかの例では、一定の深度において、スライスタイプ、CTU深度、CU位置など、以前にコーディングされた情報に基づいて、区分タイプが導出され得る。

10

【0087】

本開示の別の例では、各CU(リーフノード)に対して、ビデオエンコーダ22は、シンタックス要素(たとえば、1ビットのtransform_splitフラグ)をシグナリングし、CUと同じサイズに変換が実行されることになるか否かを示す(すなわち、フラグは、TUがCUと同じサイズであるかどうか、またはさらに分割されるかどうかを示す)ようにさらに構成され得る。transform_splitフラグが真としてシグナリングされる場合、ビデオエンコーダ22は、CUの残差を複数のサブブロックにさらに分割するように構成され得、各サブブロックに関して変換が行われる。ビデオデコーダ30は、逆のプロセスを実行し得る。

20

【0088】

一例では、transform_splitフラグが真としてシグナリングされるとき、以下が実行される。CUが正方形ブロックに対応する(すなわち、CUが正方形である)場合、ビデオエンコーダ22は、4分木分割を使用して、残差を4つの正方形サブブロックに分割し、各正方形サブブロックに変換が実行される。CUが非正方形ブロックに対応する場合、たとえば、 $M \times N$ である場合、ビデオエンコーダ22は、残差を2つのサブブロックに分割し、サブブロックサイズは、 $M > N$ であるとき、 $0.5M \times N$ であり、 $M < N$ であるとき、 $M \times 0.5N$ である。別の例として、transform_splitフラグが真としてシグナリングされ、CUが非正方形ブロックに対応する、たとえば、 $M \times N$ である(すなわち、CUは非正方形である)場合、ビデオエンコーダ22は、残差をサイズ $K \times K$ のサブブロックに分割するように構成され得、各サブブロックに対して $K \times K$ 正方形変換が使用され、 K は、 M および N の最大係数に等しい。別の例として、CUが正方形ブロックであるとき、transform_splitフラグはシグナリングされない。

30

【0089】

いくつかの例では、予測の後にCU内に残差が存在するとき、分割フラグはシグナリングされず、1つの導出サイズを有する変換のみが使用される。たとえば、 $M \times N$ に等しいサイズを有するCUの場合、 $K \times K$ 正方形変換が使用され、 K は、 M および N の最大係数に等しい。したがって、この例では、サイズ 16×8 を有するCUの場合、CUの残差データの2つの 8×8 サブブロックに対して同じ 8×8 変換が適用され得る。「分割フラグ」は、ツリー構造内のノードがツリー構造内に子ノードを有することを示すシンタックス要素である。

40

【0090】

いくつかの例では、各CUに関して、CUが正方形4分木または対称2分木に分割されない場合、ビデオエンコーダ22は、区分のサイズ(たとえば、CUのサイズ)に等しい変換サイズを常に設定するように構成される。

【0091】

ビデオエンコーダ22を参照して説明した上記の例の各々に関して、ビデオデコーダ30は

50

、逆のプロセスを実行するように構成され得ることを理解されたい。シンタックス要素のシグナリングに関して、ビデオデコーダ30は、そのようなシンタックス要素を受信してパースし、それに応じて、関連するビデオデータを区分して復号するように構成され得る。

【0092】

本開示の1つの特定の例では、ビデオデコーダは、各深度において許容される5つの異なる区分タイプを用いて、3つの異なる区分構造(QT、BT、およびTT)に従ってビデオブロックを区分するように構成され得る。区分タイプは、図5A～図5Eに示すように、4分木区分(QT区分構造)、水平2分木区分(BT区分構造)、垂直2分木区分(BT区分構造)、水平中心-側部3分木区分(TT区分構造)、および垂直中心-側部3分木区分(TT区分構造)を含む。

【0093】

5つの例示的な区分タイプの定義は、次の通りである。正方形は矩形の特殊例と見なされることに留意されたい。

- ・4分木区分:ブロックが4つの同じサイズの矩形ブロックにさらに分割される。図5Aは、4分木区分の一例を示す。

- ・垂直2分木区分:ブロックが2つの同じサイズの矩形ブロックに垂直に分割される。図5Bは、垂直2分木区分の一例である。

- ・水平2分木区分:ブロックが2つの同じサイズの矩形ブロックに水平に分割される。図5Cは、水平2分木区分の一例である。

- ・垂直中心-側部3分木区分:2つの側部ブロックが同じサイズを共有し、中心ブロックのサイズが2つの側部ブロックの和であるように、ブロックが3つの矩形ブロックに垂直に分割される。図5Dは、垂直中心-側部3分木区分の一例である。

- ・水平中心-側部3分木区分:2つの側部ブロックが同じサイズを共有し、中心ブロックのサイズが2つの側部ブロックの和であるように、ブロックが3つの矩形ブロックに水平に分割される。図5Eは、水平中心-側部3分木区分の一例である。

【0094】

特定の深度に関連するブロックの場合、ビデオエンコーダ22は、(さらなる分割がないものも含めて)どの区分タイプが使用されるかを判定し、判定された区分タイプを明示的または暗示的にビデオデコーダ30にシグナリングする(たとえば、区分タイプは、事前判定された規則から導出され得る)。ビデオエンコーダ22は、異なる区分タイプを使用するブロックに対するレートひずみコストの検査に基づいて使用する区分タイプを判定することができる。レートひずみコストを得るために、ビデオエンコーダ22は、そのブロックに関して考えられる区分タイプを再帰的に検査することが必要な場合がある。

【0095】

図6は、コーディングツリーユニット(CTU)区分の一例を示す概念図である。言い換えれば、図6は、CTUに対応するCTB80の区分を示す。図6の例では、

- ・深度0において、CTB80(すなわち、CTB全体)が(単一の点によって分離された破線を用いたライン82によって示す)水平2分木区分を用いて2つのブロックに分割される。

- ・深度1において:

- ・上側ブロックが(小さな破線を用いたライン84および86によって示す)垂直中心-側部3分木区分を用いて3つのブロックに分割される。

- ・下側ブロックが(2つの点によって分離された破線を用いたライン88および90によって示す)4分木区分を用いて4つのブロックに分割される。

- ・深度2において:

- ・深度1における上側ブロックの左側ブロックが(短い破線によって分離された長い破線を用いたライン92および94によって示す)水平中心-側部3分木区分を用いて3つのブロックに分割される。

- ・深度1において上側ブロックの中心ブロックおよび右ブロックに対してさらなる分割はない。

- ・深度1において下側ブロックの4つのブロックに対してさらなる分割はない。

【0096】

10

20

30

40

50

図6の例でわかるように、4つの異なる区分タイプ(水平2分木区分、垂直中心-側部3分木区分、4分木区分、および水平中心-側部3分木区分)を用いた3つの異なる区分構造(BT、QT、およびTT)が使用される。

【0097】

別の例では、一定の深度において、または一定のサイズに伴って、ブロックに追加の制約が適用され得る。たとえば、ブロックの高さ/幅が16ピクセル未満である場合、4ピクセル未満の高さ/幅を有するブロックを回避するために、垂直/水平中心-側部木を用いてそのブロックを分割することはできない。

【0098】

F. Le Leannee、T. Poirier、F. Urban、「Asymmetric Coding Units in QTBT」、JVET-D0064、成都、2016年10月(以下で「JVET-D0064」)で、QTBTに関連して非対称コーディングユニットを使用することが提案された。新しい分割構成を可能にするために、4つの新しい2分木分割モード(たとえば、区分タイプ)がQTBTフレームワークに導入された。図7に示すような、QTBTにおいてすでに利用可能な分割モードに加えて、いわゆる非対称分割モードが提案された。図7に示すような、HOR_UP、HOR_DOWN、VER_LEFT、およびVER_RIGHT区分タイプは非対称分割モードの例である。

【0099】

追加された非対称分割モードによれば、サイズSを有するコーディングユニットは、水平(たとえば、HOR_UPまたはHOR_DOWN)方向または垂直(たとえば、VER_LEFTまたはVER_RIGHT)方向のいずれかで、サイズS/4および3.S/4を有する2つのサブCUに分割される。JVET-D0064では、新たに追加されたCU幅または高さは、12または24のみであり得る。

【0100】

順変換および逆変換に関する技法についてここで論じる。画像/ビデオコーディングでは、変換は主に2D入力データソースに適用される。2D入力データに対して変換を適用する例示的な方法は、分離可能な2D変換および分離不能な2D変換を含む。分離可能な2D変換は、非分離可能な2D変換と比較して少ない動作(加算および乗算)カウントを必要とするため、分離可能な2D変換が通常使用される。

【0101】

一例では、変数Xは入力W×Hデータアレイであり、Wはアレイの幅であり、Hはアレイの高さである。例示的な分離可能な2D変換は、下記で定式化されるように、Xの水平ベクトルおよび垂直ベクトルに1D変換を順次適用する。

$$Y=C \cdot X \cdot R^T$$

式中、YはXの変換アレイであり、CおよびRは、整数精度または倍精度のいずれかであり得る、W×W変換行列およびH×H変換行列をそれぞれ示す。(以下の等式(1)におけるような)Tは、整数値の変換行列を表す。この公式から、Cは、Xの列ベクトルに対して1D垂直(列、左)変換を適用し、Rは、Xの行ベクトルに対して1D水平(行、右)変換を適用することがわかる。

【0102】

HEVCでは、WはHに等しい(かつ、Sに等しい)。したがって、W×Hデータアレイは $2^{2 \cdot K}$ によって示すことができ、式中、Kは整数であり、Sは 2^K に等しい。変換行列(CおよびR)は、次のように、Tとして生成される。

【0103】

【数1】

$$T = \text{int}(\sqrt{S} \cdot M \cdot 2^N), \quad (1)$$

【0104】

式中、Tは、変換行列を整数値で表し、int()は、行列の各浮動小数点要素値に対して最

10

20

30

40

50

も近い整数値を得るための関数であり、Sは、(8点変換または32点変換など)変換のサイズを示し、Mは、変換タイプに対するユニタリー行列を浮動点で示し、 2^N は、(HEVC変換においてN=6が使用されるなど)整数変換行列Tの精度を制御するスケーリング係数である。その共役変換 U^* がやはりその逆である場合(すなわち、 $U^*U=U U^*=I$ 、式中、Iは識別行列である)場合、複素正方行列Uはユニタリーである。加えて、HEVCでは、数個の要素は、+1または-1によってわずかに調整されているため、変換行列はTとして導出される。

【 0 1 0 5 】

HEVCでは、水平変換および垂直変換が適用された後、変換の出力は、ユニタリー変換(すなわち、M)と比較して、おおまかに、

【 0 1 0 6 】

10

【 数 2 】

$$(\sqrt{S} \cdot 2^N)^2$$

【 0 1 0 7 】

だけ拡大される。HEVCでは、Sは2の冪であるため、

【 0 1 0 8 】

20

【 数 3 】

$$(\sqrt{S} \cdot 2^N)^2$$

【 0 1 0 9 】

の値も、2の冪になる。したがって、変換行列において導入された

【 0 1 1 0 】

30

【 数 4 】

$$\sqrt{S}$$

【 0 1 1 1 】

は、変換の間にビットシフトで補償され得る。HEVC行列は、正規直交DCT変換と比較して、

【 0 1 1 2 】

40

【 数 5 】

$$(\sqrt{S} \cdot 2^N = 2^{N+\frac{K}{2}} = 2^{6+\frac{K}{2}}, \text{suppose } S = 2^K)$$

【 0 1 1 3 】

だけスケーリングされるため、順方向2次元変換および逆方向2次元変換を通して残差ブロックのノルムを保存するために、追加のスケール係数fwd_shift1およびfwd_shift2が適用される。同様に、逆変換の場合、

【 0 1 1 4 】

【数 6】

$$\sqrt{S} \cdot 2^N$$

【0 1 1 5】

だけスケーリングされた逆変換行列により、スケーリング係数がやはり使用される。したがって、2次元順変換および2次元逆変換を通してノルムを保存するために、スケール係数の積は、

【0 1 1 6】

【数 7】

$$(1/(2^{N+\frac{K}{2}}))^4 = 1/2^{24+2*K}$$

【0 1 1 7】

に等しいものとする。たとえば、HEVC参照ソフトウェアにおける順変換では、ビデオエンコード22は、それぞれ、次のように水平(fwd_shift1)順変換および垂直(fwd_shift2)順変換の後でシフトを適用して、各変換の後に、Dが8に等しい場合、出力が16ビットに適合することを確実にする。

$$\text{fwd_shift1} = \log_2 S + D + 6 - r \quad (2)$$

$$\text{fwd_shift2} = \log_2 S + 6 \quad (3)$$

HEVC参照ソフトウェアにおける逆変換では、ビデオデコード30は、それぞれ、垂直(inv_shift1)逆変換および水平(inv_shift2)逆変換の後で次のように、シフトを適用する。

$$\text{inv_shift1} = 7 \quad (4)$$

$$\text{inv_shift2} = 5 - D + r \quad (5)$$

式中、Dは、ビデオを再構成するために使用されるビット深度を表す。4つのシフトの和は、 $(2 * \log_2 S + 24 = 2 * K + 24)$ である。ビット深度は、SPSにおいて指定され得る。D=8およびD=10のビット深度値は、それぞれ、8ビットおよび10ビットのピクセル再構成をもたらす。パラメータrは、水平順変換の精度を制御する。より大きなr値は、より高い精度を与える。いくつかの例では、rの値は、SPSで指定されるような構成に応じて、固定値15、もしくは最大で15、またはD+6(たとえば、 $\max(15, D+6)$)であり得る。

【0 1 1 8】

上記において、Sの同じ値がshift1およびshift2の計算において使用されるが、これは、HEVCでは、正方形変換(水平変換および垂直変換のサイズが同じである)のみが使用されるためであることに留意されたい。

【0 1 1 9】

QTBT区分を使用するビデオコーディングのいくつかの例では、いくつかの新しい変換、たとえば、非正方形8×4変換がある。この場合、 $\log_2(W) + \log_2(H)$ の出力は偶数ではない。この場合、追加の係数

【0 1 2 0】

【数 8】

$$\sqrt{2}$$

【0 1 2 1】

が導入されるが、変換の間に単純なビットシフトで補償され得ない。したがって、2016年11月22日に出願した、米国特許公開第20170150176号、および2016年11月22日に出願した、米国特許公開第20170150183号に記載されるように、(たとえば、変換行列を変更するのとは対称的に)、値

【 0 1 2 2 】

【 数 9 】

$$\sqrt{2}$$

10

【 0 1 2 3 】

を量子化プロセスに吸収することが提案された。

【 0 1 2 4 】

非対称コーディングユニット(たとえば、図7に示すような非対称コーディングユニット)では、サイズ12および24を有する変換など、2の冪に等しくないサイズを有する変換が使用され得る。したがって、そのような非対称的コーディングユニットは、変換プロセスにおいて容易に補償され得ない、より多くの係数を導入する。そのような非対称コーディングユニットに関して、変換または逆変換を実行するために、追加の処理が必要な場合がある。

【 0 1 2 5 】

20

順および/または逆量子化に関する技法について次に論じる。残差エネルギーを圧縮して周波数係数を低減する変換プロセスの後、ビデオエンコーダ22は量子化を適用して、残差再構成のひずみを制御する。それに応じて、ビデオデコーダ30において、逆変換に先立って、逆量子化(inverse quantize)(逆量子化(de-quantize))プロセスが実行される。

【 0 1 2 6 】

順量子化の場合、HEVC参照ソフトウェアにおいて、ビデオエンコーダ22は、下記で説明するように、デッドゾーンプラス一様量子化方式を適用する。

$$y' = \text{sign}(y) \cdot (|y| \cdot Q + f \cdot 2^{q_{\text{bits}}}) \quad q_{\text{bits}} \quad (6)$$

式中、 y は、入力変換係数であり、 Q は、量子化スケーリング係数であり、 f は、デッドゾーンサイズのサイズを制御する丸めオフセットである(デッドゾーンが範囲 $[-(1-f) \cdot \dots, (1-f) \cdot \dots]$ 内である、図8に示すように)、 $\text{sign}(y) = y > 0 ? 1 : -1$ 、 q_{bits} は、シフティングパラメータであり、 y' は、出力量子化変換係数を与える。デッドゾーン領域に分類されるすべての値は、0になるように量子化されることになる。図8は、デッドゾーンプラス一様量子化方式を示す概念図である。一例では、デルタの値(図8の)は $= 2^{q_{\text{bits}}}$ であり得る。

30

【 0 1 2 7 】

HEVCでは、イントラスライスの場合、 f は171/512である、それ以外の場合、 f は85/512である。

【 0 1 2 8 】

上記の量子化スケーリング係数 Q およびシフティングパラメータ q_{bits} は、下記のように指定される。

40

$$Q = g_{\text{quantScales}}[QP\%6] \quad (7)$$

式中、 QP は、ユーザによって定義された量子化パラメータであり、 $g_{\text{quantScales}}$ は、下記のHEVC参照ソフトウェアにおいて指定されるような定数配列である。

```
const Int g_quantScales[SCALING_LIST_REM_NUM]=
{
    26214,23302,20560,18396,16384,14564
};
```

である。

【 0 1 2 9 】

加えて、 q_{bits} は下記のように導出される。

50

【 0 1 3 0 】

【 数 1 0 】

$$qbits = 14 + \lfloor QP/6 \rfloor + iTransformShift \quad (8)$$

【 0 1 3 1 】

式中、 $iTransformShift = r - D - \log_2(S)$ であり、 S は、ブロックサイズであり、 D および r は、上記の等式(2)および等式(5)で定義されたのと同じである。 10

【 0 1 3 2 】

HEVCにおける逆量子化の場合、逆量子化プロセスは下記のように指定される。

【 0 1 3 3 】

【 数 1 1 】

$$\hat{y} = \text{sign}(y') \cdot (|y'| \cdot DQ + 2^{qbits'-1}) \gg qbits' \quad (9)$$

【 0 1 3 4 】

20

式中、 y' は、入力量子化変換係数であり、

【 0 1 3 5 】

【 数 1 2 】

$$\hat{y}$$

【 0 1 3 6 】

は、逆量子化変換係数であり、 DQ は下記のように指定される。

$$Q = g_invQuantScales[QP\%6] \quad (10)$$

30

$g_invQuantScales$ は、下記のように指定された定数配列である、

```
const Int g_invQuantScales[SCALING_LIST_REM_NUM]=
{
    40,45,51,57,64,72
};
```

【 0 1 3 7 】

加えて、 $qbits'$ は下記のように導出される。

【 0 1 3 8 】

【 数 1 3 】

40

$$qbits' = 6 - \lfloor QP/6 \rfloor - iTransformShift \quad (11)$$

【 0 1 3 9 】

式中、 $iTransformShift = r - D - \log_2(S)$ であり、 S は、ブロックサイズであり、 D および r は、等式(2)および等式(5)で定義されたのと同じである。

【 0 1 4 0 】

等式(8)の $iTransformShift$ の上記の定義から、逆量子化プロセスはブロックサイズに依存することがわかる。 50

【0141】

イントラスライスに対して分離されたツリー構造について次に論じる。VCEG提案COM16-C966において、イントラスライスに対して分離されたツリー構造が提案された。特に、クロマ成分に対して、コーディング性能をさらに改善するために、イントラスライス内のルーマ成分およびクロマ成分に対して異なるツリー構造を有することが提案された。すなわち、クロマブロックは、ルーマブロックとは異なる様式で区分され得る。イントラスライスは、イントラコーディングされたコーディングユニットを含むスライスである。

【0142】

様々なQBTB区分構造およびMTT区分構造に従ってビデオデータをコーディングするための現在の提案に関して、以下の問題が観測されている。そのサイズが2の冪ではない変換をMTTフレームワークに組み込むとき、変換および/または逆変換プロセスの間にシフト動作で補償され得ない係数をより効率的に処理するために特別な処理が必要とされる場合がある。MTTフレームワークの下で分離されたルーマツリー構造およびクロマツリー構造を使用するとき、クロマ成分に対する複雑なツリー構造が常に有利であるとは限らない。分離されたルーマツリーおよびクロマツリーをインターフレームに拡張するとき、ルーマとクロマの両方に対して動き情報がシグナリングされ、これはかなりのシグナリングコストをもたらす。分離されたルーマツリー構造およびクロマツリー構造を使用するとき、ルーマツリーおよびクロマツリーは同じ分割パターンを有することがある。この場合、ルーマ区分およびクロマ区分を別個にシグナリングすることは効率的でない場合がある。

【0143】

上述の問題に対処するために、以下の技法が提案される。ビデオエンコーダ22およびビデオデコーダ30は、以下の技法を逆の様式で実行するように構成され得る。以下の箇条書きにされる技法は、個別に適用され得る。加えて、以下の技法の各々は、任意の組合せで一緒に使用されてもよい。

【0144】

本開示の第1の例では、そのサイズが2の冪ではない変換を組み込むとき、本開示は、ビデオデコーダ30において、かつ/またはビデオエンコーダ22とビデオデコーダ30の両方において、等式(1)の真のサイズSの代わりに、丸められた修正されたS'を使用して変換行列を生成することを提案する。2の冪ではないサイズを有する変換は、ブロックが非正方形形状を有するときに生じ得る。そのような変更に伴って、ビットシフティングによって補償され得ない変換のスケーリングは、変換行列内に吸収される。したがって、(

【0145】

【数14】

$$\sqrt{2}$$

【0146】

問題が上記で論じた様式で処理されると仮定すると)変換および量子化処理の他の処理技法に変更する必要がない可能性がある。

【0147】

第1の例の一態様では、ビデオエンコーダ22およびビデオデコーダ30は、真のサイズSを2の冪である値に丸めることによって、修正された値S'を得るために、真のサイズSを変換に丸めるように構成され得る。たとえば、12のSの値を16に丸め、24のSの値を32に丸める。概して、S'の値は、Sを、2の上もしくは下、または2の最も近い冪に丸めることによって得ることができる。

【0148】

一例では、ビデオデコーダ30は、ビデオデータの符号化ブロックを受信するように構成され得る。一例では、ビデオデータの符号化ブロックは、逆量子化変換係数を含む。いくつかの例では、ビデオデータのブロックは非正方形形状を有し得る。ビデオデコーダ30は

、ビデオデータの符号化ブロックに対する変換を判定するようにさらに構成されてよく、変換は、2の冪ではないサイズSを有する。ビデオデコーダ30は、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成するようにさらに構成され得る。ビデオデコーダ30は、その場合、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用し、残差ビデオデータを復号して、ビデオデータの復号ブロックを生成し得る。

【0149】

同様に、ビデオエンコーダ22は、ビデオデータのブロックを受信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22は、ビデオデータのブロックを非正方形形状に区分した。ビデオエンコーダ22は、残差ビデオデータを生成するために、(たとえば、インター予測および/またはイントラ予測を使用して)ビデオデータのブロックを予測することができる。ビデオエンコーダ22は、残差ビデオデータに対する変換を判定することができ、変換は、2の冪ではないサイズSを有する。ビデオエンコーダ22は、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成し、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用し得る。いくつかの例では、ビデオエンコーダ22は、変換係数を量子化することもできる。ビデオエンコーダ22は、その場合、(たとえば、CABACなどのエントロピーコーディングを使用して)符号化ビットストリーム内の変換係数を符号化することができる。

10

【0150】

第1の例の別の態様では、等式(2)および等式(3)の適応シフトは、Sの代わりに、S'に基づく。たとえば、修正された等式(2)'および(3)'は次のように修正され得る：

20

$$\text{fwd_shift1} = \log_2 S + D + 6 - r$$

(2)'

$$\text{fwd_shift2} = \log_2 S' + 6$$

(3)'

【0151】

第1の例の別の態様では、整数変換行列Tを導出するとき、等式(1)に示したようなユニタリー変換の上部のスケーリング係数

【0152】

【数15】

$$(\sqrt{S} \cdot 2^N)$$

30

【0153】

は、事前定義された固定値、たとえば、256、512、または1024によって置換される。一例では、等式(2)および等式(3)で説明したような右シフト演算は、fwd_shift1および/またはfwd_shift2の値がもはやSに依存しないような方法で修正され、すなわち、 $\log_2 S$ は、等式(2)および/または等式(3)から除去される。この場合、Sは、変換サイズとは無関係になり、すなわち、等式(2)および等式(3)のシフトは、変換サイズにかかわらず、固定値に設定される。

40

【0154】

第1の例の別の態様では、逆変換の場合、元のシフト演算は変更されない状態に維持される。すなわち、ビデオデコーダ30は、等式(4)および等式(5)で説明したように、逆シフトを実行するように構成され得る。

$$\text{inv_shift1} = 7$$

(4)

$$\text{inv_shift2} = 5 - D + r$$

(5)

【0155】

第一の例の別の態様では、等式(8)および/または等式(11)で説明したような右シフト演算は、qbitsおよび/またはqbits'の値がもはやSに依存しないような方法で修正され、すなわち、 $\log_2 S$ は、等式(8)および/または等式(11)から除去される。修正された等式(8)'および等式(11)'を下記に示す。

50

【 0 1 5 6 】

本開示の一例では、変換係数の量子化を実行するとき、ビデオエンコーダ22は、下記に示すようにqbitの値を判定することができる。

【 0 1 5 7 】

【 数 1 6 】

$$qbits = 14 + \lfloor QP/6 \rfloor + iTransformShift \quad (8)'$$

10

【 0 1 5 8 】

式中、 $iTransformShift=r-D$ であり、 D および r は、上記の等式(2)および等式(5)で定義されたのと同じである。

【 0 1 5 9 】

加えて、変換係数の逆量子化を実行するとき、ビデオデコーダ30は、下記に示すようにqbit'の値を判定することができる。

【 0 1 6 0 】

【 数 1 7 】

$$qbits' = 6 - \lfloor QP/6 \rfloor - iTransformShift \quad (11)'$$

20

【 0 1 6 1 】

式中、 $iTransformShift=r-D$ であり、 D および r は、等式(2)および等式(5)で定義されたのと同じである。

【 0 1 6 2 】

第一の例の別の態様では、等式(4)および等式(5)で説明したような右シフト演算は、 inv_shift1 および/または inv_shift2 の値が、以下など、 S に依存するような方法で修正される。

$$inv_shift1 = 7 + \log_2 S \quad (4)''$$

$$inv_shift2 = 5 - D + r + \log_2 S \quad (5)''$$

【 0 1 6 3 】

本開示の第2の例では、2分木タイプ、対称中心-側部3分木タイプ、4分木タイプ、および非対称木ツリータイプを使用する構造など、MTT構造に従ってブロックを区分するとき、ビデオエンコーダ22およびビデオデコーダ30は、2レベルMTTを使用してブロックを区分するように構成され得る。例示的な2レベルMTT構造は、2017年3月20日に出願した、米国特許公開第20170272782号に記載されている。2レベルMTTでは、第1のレベル(「領域木レベル」と呼ばれる)において、ビデオデータのピクチャまたはブロックは領域に分割され、領域の各々が、大きいブロックを小さいブロックに迅速に(たとえば、4分木または16分木を使用して)区分することが可能な、単一木タイプまたは複数木タイプを伴う。第2のレベル(「予測レベル」と呼ばれる)において、領域はMMT技法(さらなる分割を含まない)を用いてさらに分割される。予測ツリーのリーフノードは、本開示ではコーディングユニット(CU)と呼ばれる。加えて、以下の技法が使用され得る。

【 0 1 6 4 】

第2の例の一態様では、さらに分割されるとしてシグナリングされている予測ツリーの場合、ビデオエンコーダ22は、垂直分割または水平分割を示すためのフラグをまずシグナリングすることができる。ビデオエンコーダ22は、その場合、そのような分割が対称分割

30

40

50

(たとえば、2分木または対称中心-側部3分木)であるかどうかを示すためのフラグをシグナリングし得る。分割が対称分割である場合、ビデオエンコーダ22は、2分木または対称中心-側部3分木など、様々な許容される対称区分タイプのタイプインデックスをシグナリングし得る。それ以外の場合(たとえば、分割が非対称分割である)、ビデオエンコーダ22は、非対称分割が上分割または下分割である(たとえば、分割が水平分割であるとき)かどうかを示すための、または分割が左分割または右分割である(たとえば、分割が垂直分割であるとき)ことを示すためのフラグをシグナリングし得る。ビデオデコーダ30は、前述のフラグを受信してパースし、それに応じて、ビデオブロックを区分するように構成され得る。

【0165】

第2の例の別の態様では、さらに分割されるとしてシグナリングされている予測ツリーの場合、ビデオエンコーダ22は、垂直分割または水平分割を示すためのフラグをまずシグナリングすることができる。次に、ビデオエンコーダ22は、分割が2分木区分タイプであるかどうかを示すためのフラグをシグナリングし得る。分割が2分木区分タイプでない場合、ビデオエンコーダ22は、対称中心-側部3分木または非対称木など、他のツリータイプのタイプインデックスをシグナリングし得る。分割が非対称木である場合、ビデオエンコーダ22は、分割が上分割または下分割であること(たとえば、分割が水平分割であるとき)を示すための、または分割が左分割または右分割である(たとえば、分割が垂直分割であるとき)を示すためのフラグをシグナリングし得る。ビデオデコーダ30は、前述のフラグを受信してパースし、それに応じて、ビデオブロックを区分するように構成され得る。

【0166】

第2の例の別の態様では、さらに分割されているとしてシグナリングされている予測ツリーの場合、ビデオエンコーダ22は、垂直分割または水平分割を示すためのフラグをシグナリングすることができる。ビデオエンコーダ22は、その場合、分割が対称中心-側部3分木であるかどうかを示すためのフラグをシグナリングし得る。分割が対称中心-側部3分木区分タイプでない場合、ビデオエンコーダ22は、2分木または非対称木など、他のツリータイプのタイプインデックスをシグナリングし得る。分割が非対称木である場合、ビデオエンコーダ22は、非対称木分割が上分割または下分割である(たとえば、分割が水平分割であるとき)かどうかを示すための、または非対称木分割が左分割または右分割である(たとえば、分割が垂直分割であるとき)ことを示すためのフラグをシグナリングし得る。ビデオデコーダ30は、前述のフラグを受信してパースし、それに応じて、ビデオブロックを区分するように構成され得る。

【0167】

本開示の別の態様では、ビデオエンコーダ22は、近隣ブロックの関連するインジケータ/区分タイプ、またはスライス/ピクチャタイプなど、コーディングされた情報の特性に従って、垂直/水平分割区分タイプ、および/または上/下分割区分タイプ、および/または左/右分割区分タイプ、および/またはツリータイプ区分タイプを示すシンタックス要素をシグナリングする順番を適応的に変更するように構成され得る。一例では、異なるスライス/ピクチャは、シグナリングされたシンタックス要素の異なる順序を使用して、区分(たとえば、ブロックがどのように分割されるか)を示すことができる。別の例では、ビデオエンコーダ22は、ブロック単位でシンタックス要素の順序を変更するように構成され得る。ビデオデコーダ30は、ビデオエンコーダ22によって判定された同じ順序で前述のシンタックス要素を受信するように構成され得る。ビデオデコーダ30は、ビデオエンコーダ22と同じ様式でシンタックス要素の順序を判定することができる。

【0168】

第2の例の別の態様では、非対称木区分タイプ(たとえば、上/下区分または左/右区分)を指示するシンタックス要素をエントリピーコーディングするために使用されるコンテキストは、次のように導出され得る。図9は、本開示のこの例の例示的な非対称区分タイプを示す。A、B、Cを、図9に示した区分の各々の中心位置のすぐ上のロケーション(左上)をカバーするブロックのブロックサイズであるとする。この例では、ビデオエンコーダ22お

よびビデオデコーダ30は、ゼロに初期化されるカウンタを使用してコンテキストモデルのインデックスを判定することができる。カウンタの値は、コンテキストを判定するために使用される。

【0169】

次の条件を検討されたい。

条件1: AがBに等しくなく、BがCに等しい場合。

条件2: 上記のCUが、境界が左半分内にある、垂直に区分された非対称ツリーである場合。

【0170】

一例では、条件1または2のいずれかが満たされる場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。別の例では、条件1と2の両方が満たされる場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。別の例では、ビデオエンコーダ22およびビデオデコーダ30は、条件1が満たされる場合、カウンタを1だけ増分するように構成され得る。同様に、ビデオエンコーダ22およびビデオデコーダ30は、条件2が満たされる場合、カウンタを1だけ増分するように構成され得る。

【0171】

別の例では、A、B、Cは、区分に対して他の位置であってよい。一例では、A、B、Cは、図9に示した区分の各々の左上角のすぐ上のロケーション(上の中央)をカバーするブロックのブロックサイズである。別の例では、A、B、Cを、図9に示した区分の各々の右上角のすぐ上のロケーション(右上)をカバーするブロックのブロックサイズであるとする。

【0172】

別の例では、D、E、Fを、図9に示した区分の各々の中心位置のすぐ左のロケーション(左下)をカバーするブロックのブロックサイズであるとする。

【0173】

この例では、以下の条件を検討する。

条件1: DがEに等しくなく、EがFに等しい場合。

条件2: 左CUが、境界が上半分内にある、水平に区分された非対称ツリーである場合。

【0174】

一例では、条件1または2のいずれかが満たされる場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。別の例では、条件1と2の両方が満たされる場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。別の例では、ビデオエンコーダ22およびビデオデコーダ30は、条件1が満たされる場合、カウンタを1だけ増分するように構成され得る。同様に、ビデオエンコーダ22およびビデオデコーダ30は、条件2が満たされる場合、カウンタを1だけ増分するように構成され得る。

【0175】

別の例では、D、E、Fは、他の位置にあり得る。一例では、D、E、Fは、図9に示した区分の各々の左上角のすぐ左のロケーション(下の中央)をカバーするブロックのブロックサイズである。別の例では、D、E、Fを、図9に示した区分の各々の左下角のすぐ左のロケーション(右下)をカバーするブロックのブロックサイズであるとする。

【0176】

別の例では、ビデオエンコーダ22およびビデオデコーダ30は、事前定義された様式でA~Fのロケーションを判定するように構成され得る。別の例では、ビデオエンコーダ22は、SPS、PPS、またはスライスヘッダ内でブロックA~Fのロケーションをシグナリングするように構成され得る。

【0177】

第2の例の別の態様では、ビデオエンコーダ22およびビデオデコーダ30は、次のように、ツリー対称性をシグナリングするために使用されるフラグのコンテキストを導出するように構成され得る。一例では、単一のコンテキストモデルが使用され得る。別の例では、

ビデオエンコーダ22およびビデオデコーダ30は、カウンタに基づいてマルチレベルコンテキストモデルを使用するように構成され得る。一例では、カウンタの初期値はゼロである。上記のCUが非対称ブロックである場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。左CUが非対称ブロックである場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。左上CUが非対称ブロックである場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。右上CUが非対称ブロックである場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを1だけ増分するように構成され得る。上記の4つのブロックのいずれも非対称ブロックに属さない場合、ビデオエンコーダ22およびビデオデコーダ30は、カウンタを5に設定するように構成され得る。

10

【0178】

第2の例の別の態様では、ビデオエンコーダ22は、予測ツリーがさらに分割されるかどうかを示すためのフラグを条件付きで生成してシグナリングするように構成され得る。予測ツリーのサイズが、さらなる分割なしに、そのサイズがサポートされない変換を使用するとき、ビデオエンコーダ22は、分割フラグをシグナリングしないように構成され得る。むしろ、予測ツリーのサイズが、さらなる分割なしに、そのサイズがサポートされない変換を使用するという条件に基づいて、ビデオエンコーダ22とビデオデコーダ30は両方とも、さらに分割されるとして予測を推論するように構成され得る。

【0179】

第2の例の別の態様では、領域ツリー(RT)に基づく可変予測ツリー(PT)深度を用いたビデオシステムコーディングでは、最小の許容されるブロックサイズはRT依存であり得る。したがって、PTの分割フラグ、分割方向フラグ(水平/垂直)、ツリー対称性フラグ、または他の前述のツリータイプフラグのシグナリングを回避することができる。また、フレーム境界上にあるCTUが一定のまたはすべてのタイプのPTで分割されることが許容される場合、そのCTUに同様の制約を課すことができる。

20

【0180】

本開示の一例では、ビデオエンコーダ22は、2レベルマルチタイプツリー区分構造を使用してビデオデータを区分することと、2レベルマルチタイプツリー区分構造の予測ツリーがどのように構造化されているかを示すシンタックス要素を生成することとによって、シンタックス要素が、垂直分割または水平分割を示すためのフラグ、分割が対称分割であるかどうかを示すためのフラグ、タイプインデックス、または非対称分割が上分割もしくは下分割であるか、または左分割もしくは右分割であるかを示すためのフラグのうちの1つまたは複数を含む、生成することを行うように構成され得る。一例では、ビデオエンコーダ22は、近隣ブロックのブロックサイズに基づいて、シンタックス要素に関するコンテキストを判定するように構成され得る。本開示の別の例では、ビデオエンコーダ22は、近隣ブロックの区分タイプに基づいて、シンタックス要素に関するコンテキストを判定するように構成され得る。

30

【0181】

本開示の第3の例では、ルーマ成分およびクロマ成分に対して分離された(たとえば、ルーマブロックおよびクロマブロックが別個に区分された)ツリー構造を使用するとき、ビデオエンコーダ22は、ルーマ成分およびクロマ成分に対して許容されるツリータイプを別個に示すシンタックス要素を生成してシグナリングするように構成され得る。すなわち、ビデオエンコーダ22は、ルーマブロックとクロマブロックの両方に対して許容されるツリータイプを示す別個のシンタックス要素を生成することができる。シンタックス要素の値は、2つ以上のツリータイプのどれが特定のルーマブロックおよびクロマブロックに対して許容されるかを示し得る。例示的なツリータイプは、対称2分木タイプおよび非対称2分木タイプ、4分木タイプ、ならびに対称3分木タイプおよび非対称3分木タイプを含めて、上記で論じたツリータイプのうちのいずれかであり得る。ビデオデコーダ30は、許容されるツリータイプを示すシンタックス要素をパースするように構成され得る。

40

【0182】

50

ビデオエンコーダ22は、許容されるツリータイプのうちのどれを特定のブロックに使用するかを示す追加のシンタックス要素をシグナリングするようにさらに構成され得る。ビデオデコーダは、追加のシンタックス要素をパースし、許容されるツリータイプを示すシンタックス要素および許容されるツリータイプから使用する特定のツリータイプを示す追加のシンタックス要素から特定のブロックをどのように区分するかを判定するように構成され得る。

【0183】

一例では、ビデオエンコーダ22は、ビデオパラメータセット(VPS)、シーケンスパラメータセット(PPS)、ピクチャパラメータセット(PPS)、適応パラメータセット(APS)、または任意の他のシーケンス/ピクチャ/スライスレベルシンタックス要素本体において、ルーマ成分およびクロマ成分に対して許容されるツリータイプを別個にシグナリングすることができる。一例では、ツリータイプは、2分木、対称中心-側部3分木、4分木、または非対称木タイプのうちの少なくとも2つを含み得る。別の例では、2分木および/または対称中心-側部3分木は、常に、オンであり得(たとえば、常に可能にされ)、対称中心-側部3分木および/または非対称CU/ツリー区分タイプは、任意であり、ビットストリーム内でシグナリングされる。

【0184】

本開示の第4の例では、分離されたルーマ/クロマツリー構造をインタースライスに拡大するとき、ビデオエンコーダ22は、ルーマツリー(プライマリツリーとも呼ばれる)に対してのみなど、動き情報を一度シグナリングするように構成され得る。ビデオデコーダ30は、その場合、ルーマブロックに対してコロケートされた位置においてブロックに対する他のツリー構造に関する(たとえば、クロマブロックに対するセカンダリツリー構造に関する)動き情報を継承する(たとえば、再使用する)ように構成され得る。コロケートされたクロマブロックが単一のコロケートされたルーマブロックよりも大きい場合、ビデオデコーダ30は、すべてのコロケートされたルーマブロックに関する動き情報を再使用するよう構成され得、すなわち、1つのクロマコーディングブロックは、すべてのコロケートされたルーマブロックからの動き情報のいくつかのセットを含み得る。別の例では、ビデオデコーダ30は、他のツリーの動き情報に関する予測子としてプライマリツリー(たとえば、クロマブロックに対するツリー構造)の動き情報を使用するように構成され得る。

【0185】

概して、本開示のこの例によれば、ビデオデコーダ30は、ビデオデータのルーマブロックを区分し、ビデオデータのルーマブロックを区分することとは別個に、ビデオデータの1つまたは複数のクロマブロックを区分するように構成され得る。ビデオデータのルーマブロックに対する区分構造は、マルチタイプツリー区分構造であってよく、ビデオデータの1つまたは複数のクロマブロックに対する区分構造も、マルチタイプツリー区分構造であってよい。ビデオデコーダ30は、インタースライスに対してビデオデータのルーマブロックに関する動き情報を判定し、ビデオデータのルーマブロックに関して判定された動き情報から、ビデオデータの1つまたは複数のクロマブロックに関する動き情報を推論するようにさらに構成され得る。

【0186】

本開示の第5の例では、分離されたツリーコーディング(たとえば、ルーマブロックおよびクロマブロックが、潜在的に異なる区分を用いて、別個に区分される)では、クロマツリー(セカンダリツリーと呼ばれることもある)のツリー分割パターンは、プライマリツリー内にコロケートされたブロックがイントラコーディングされたブロックとインターコーディングされたブロックの両方を含むとき、ルーマツリー(プライマリツリーと呼ばれることもある)から継承される。すなわち、ビデオエンコーダ22およびビデオデコーダ30は、ルーマブロックがイントラコーディングされたブロックとインターコーディングされたブロックの両方を含むとき、関連するルーマブロックと同じ分割パターンを使用してクロマブロックを区分するように構成され得る。

【0187】

この例では、ビデオエンコーダ22は、プライマリツリー内のコロケートされたブロック(たとえば、ルーマブロック)が同じタイプのブロックを含むときのみ、セカンダリツリーに対するツリー分割パターンを単にシグナリングするように構成され得る。すなわち、コロケートされたルーマブロックは、すべてのインターコーディングされたブロックまたはすべてのイントラコーディングされたブロックのいずれかを含む。一例では、ツリー分割パターンは、限定はしないが、ツリータイプ(非分割も特殊ツリータイプと見なされ得る)を含む。ツリータイプは、2分木、3分木、および4分木など、対称ツリータイプと非対称ツリータイプの両方を含み得る。

【0188】

概して、本開示のこの例によれば、ビデオデコーダ30は、ビデオデータのルーマブロックを区分し、ルーマブロックがイントラコーディングされたブロックとインターコーディングされたブロックの両方を含む場合、ビデオデータの1つまたは複数のクロマブロックに対する区分がビデオデータのルーマブロックに対するのと同じになると推論し、ルーマブロックが同じタイプのコーディングされたブロックのすべてを含むとき、シグナリングされたシンタックス要素から、ビデオデータの1つまたは複数のクロマブロックに対する区分を判定するように構成され得る。

【0189】

本開示の第6の例では、PT分割フラグのシグナリングコストを削減するために、ビデオエンコーダ22およびビデオデコーダ30は、現在ブロックのサイズに対する近隣ブロックの相対的サイズを使用して、PT分割フラグのコンテキストを判定するように構成され得る。

【0190】

一例では、ビデオエンコーダ22およびビデオデコーダ30は、その近隣ブロックに対するブロックのサイズを使用して、PTツリータイプコーディングに関するコンテキストを選択するように構成され得る。一例では、現在ブロックの幅がその上の近隣の幅よりも大きいとき、現在ブロックがさらに分割される可能性が高い。同様に、現在ブロックの高さがその左の近隣の高さよりも大きいとき、現在ブロックがさらに分割される可能性が高い。さらに、現在ブロックサイズに対する、左上、右上、および左下の近隣の相対的サイズも、現在ブロックがさらに分割されるべきかどうかを判定するための有用な情報を提供する。現在ブロックサイズがその近隣のブロックサイズよりも大きい場合、現在ブロックがさらに分割される可能性はやはり高い。ビデオエンコーダ22およびビデオデコーダ30は、PT分割フラグに関するコンテキストに対するインデックスとして、前述のイベントの発生集約数を使用するように構成され得る。また、個々のイベントも、PT分割フラグに関するコンテキストのセットを形成し得る。

【0191】

一例では、ビデオエンコーダ22およびビデオデコーダ30は、上の近隣ブロックの幅および左の近隣ブロックの高さを使用して、PT分割方向(たとえば、水平分割または垂直分割)に関するコンテキストを判定するように構成され得る。上の近隣ブロックの幅が現在ブロックの幅よりも小さく、左の近隣ブロックの高さが現在ブロックの高さ以上である場合、現在ブロックが垂直に分割される可能性が高い。同様に、左の近隣ブロックの高さが現在ブロックの高さよりもより小さく、上の近隣ブロックの幅が現在ブロックの幅以上である場合、現在ブロックが水平に分割される可能性が高い。

【0192】

別の例では、ビデオエンコーダ22およびビデオデコーダ30は、上の近隣ブロックの幅および左の近隣ブロックの高さを使用して、PT分割モードに関するコンテキストを判定するように構成され得る(たとえば、2分木と中心-側部3分木との間など、分割モード同士の間の判定)。上の近隣ブロックの幅が現在ブロックの幅よりも小さく、現在ブロックが垂直に分割される場合、または左の近隣ブロックの高さが現在ブロックの高さよりも小さく、現在ブロックが水平に分割される場合、現在ブロックが3分木に分割される可能性が高い。

【0193】

10

20

30

40

50

一例では、近隣ブロックが利用可能でない場合、ビデオエンコーダ22およびビデオデコーダ30は、現在コンテキストの導出の間にデフォルトコンテキスト値を使用するように構成され得る。別の例として、異なる成分(Y、Cb、Cr、または深度成分)に対して異なるRT分割またはPT分割が許容されるとき、他の成分内の関連ブロックを近隣ブロックとして使用することによって、すべての前述の方法が適用され得る。

【0194】

別の例では、ビデオエンコーダ22およびビデオデコーダ30は、現在ブロックの深度に対する近隣ブロックの相対深度を使用して、PT分割シンタックス要素のコンテキストを判定するように構成され得る。

【0195】

別の例では、最小ブロックサイズの3倍に等しい幅/高さを有するブロックの片側に対して、ビデオエンコーダ22およびビデオデコーダ30は、上または左の近隣ブロックの中央ロケーションのブロックサイズまたはPT深度を使用して、コンテキストを導出するように構成され得る。

【0196】

別の例では、ブロックの幅/高さが最小ブロックサイズの3倍に等しいとき、ビデオエンコーダ22およびビデオデコーダ30は、3つの近隣ブロックのブロックサイズまたはPT深度の平均値を使用して、コンテキストを導出するように構成され得る。

【0197】

別の例では、ブロックの幅/高さが最小ブロックサイズの3倍に等しいとき、ビデオエンコーダ22およびビデオデコーダ30は、3つの近隣ブロックのブロックサイズまたはPT深度の最大値を使用して、コンテキストを導出するように構成され得る。

【0198】

別の例では、ブロックの幅/高さが最小ブロックサイズの3倍に等しいとき、ビデオエンコーダ22およびビデオデコーダ30は、3つの近隣ブロックのブロックサイズまたはPT深度の最小値を使用して、コンテキストを導出するように構成され得る。

【0199】

別の例では、ブロックの幅/高さが最小ブロックサイズの3倍に等しいとき、ビデオエンコーダ22およびビデオデコーダ30は、3つの近隣ブロックのブロックサイズまたはPT深度の中間値を使用して、コンテキストを導出するように構成され得る。

【0200】

別の例では、(上記で説明したような)ブロックサイズまたはPT深度の集約値は、近隣ブロックのサイズの表現である。一例では、上の近隣の幅が現在ブロックの幅よりも小さい場合、コンテキスト1が使用される。それ以外の場合、コンテキスト0が使用される。同様に、左の近隣の高さが現在ブロックの高さよりも小さい場合、コンテキスト1が使用される。それ以外の場合、コンテキスト0が使用される。一例では、ブロックサイズまたはPT深度の集約値を使用して、使用するコンテキストモデルのインデックスに等しいかまたはそのインデックスを制御するカウンタを増分することができる。また、集約値は、単独で位置する値の置換として、下記の等式内の設計にリンクされ得る。近隣ブロックの各々に対して、コンテキスト(CTX)の以下の設定プロセスを順序正しく実行することができる。コンテキストインデックスを選択するために、CTXの和を利用することができる。別の例として、下記の最初の2つの等式は、まず、近隣ブロックの各々に適用され、次いで、最後の等式を利用して、コンテキストインデックスを選択し、入力されたCTXは、すべての近隣ブロックからのCTXの和である。

$$CTX = (W > W_{T2}) + (H > H_{L2}) + (W * H > STL) + (W * H > STR)$$

$$CTX = ((W < W_{T2}) \&\& (H < H_{L2}) \&\& (W * H < STL) \&\& (W * H < STR)) ? 0 : CTX$$

$$CTX = (CTX - 3) ? 3 : CTX$$

【0201】

概して、本開示のこの例によれば、ビデオエンコーダ22およびビデオデコーダ30は、現在ブロックに対する近隣ブロックの相対サイズに基づいて、現在ブロックの分割フラグに

10

20

30

40

50

関するコンテキストと、判定されたコンテキストに基づいて分割フラグをコーディングするコンテキストとを判定するように構成され得る。

【0202】

本開示の第7の例では、MTTなど、任意のツリー構造フレームワークにおいて、ビデオエンコーダ22および/またはビデオデコーダ30は、ブロックサイズに基づいて変換を適用するように構成され得る。いくつかの例では、一定のブロックサイズは、関連する変換を有さない場合がある(すなわち、同じサイズの変換はサポートされない)。たとえば、ビデオエンコーダ22および/またはビデオデコーダ30は、 64×48 CUを区分するように構成され得るが、 48×48 変換を使用するように構成されない場合がある。別の例では、ビデオエンコーダ22および/またはビデオデコーダ30は、 256×256 CUを区分するように構成され得るが、最大のサポートされる変換は単に 128×128 である。

10

【0203】

これらの例では、ビデオエンコーダ22および/またはビデオデコーダ30は、そのようなCUに対して一定のコーディングモードのみを可能にするように構成され得る。一例では、ビデオエンコーダ22および/またはビデオデコーダ30は、CUに対して関連変換が存在しない場合、CUに対してスキップモードを単に使用するように構成され得る。この例では、関連変換は、CUの少なくとも1つの寸法に等しいサイズを有する変換である。この例では、ビデオエンコーダ22は、スキップモードフラグをシグナリングしなくてよい。むしろ、ビデオデコーダ30は、CUが何の関連変換も有さないサイズであることに基づいて、スキップフラグの値が真であると推論するように構成され得る。

20

【0204】

別の例では、ビデオエンコーダ22および/またはビデオデコーダ30は、CUのサイズに対してサポートされる変換が存在しない場合、CUが任意の非ゼロ残差(残差値)を有することを許容しないように構成され得る。この例では、ビデオエンコーダは、コーディングされたブロックフラグ(CBF)をシグナリングするように構成され得ない。コーディングされたブロックフラグは、ブロックが任意の非ゼロ変換係数を含むか否かを示すフラグである。この例では、CUがサポートされる変換を有さない場合、CUは任意の非ゼロ残差値を有しないため、ビデオデコーダ30は、CBFフラグがゼロである(すなわち、非ゼロ変換係数が存在しない)ことを示し得る。

【0205】

30

一例では、ビデオエンコーダ22およびビデオデコーダ30は、ビデオデータのブロックのサイズに基づいて、ビデオデータのブロックに対するコーディングモードを判定するように構成され得る。詳細には、ビデオエンコーダ22およびビデオデコーダ30は、ビデオデータのブロックのサイズ、およびビデオエンコーダ22およびビデオデコーダ30によってサポートされる変換に基づいて、ビデオデータのブロックに対するコーディングモードを判定するように構成され得る。ビデオデータのブロックのサイズに関してサポートされる変換が存在しない場合、ビデオエンコーダ22およびビデオデコーダ30は、ビデオデータのブロックに対して一定の事前判定されたコーディングモードを判定するように構成され得る。一例では、コーディングモードはスキップモードであってよい。本開示の別の例では、コーディングモードは別のコーディングモード(たとえば、合併モード、AMVPモード、イントラモード)であってよいが、CBFフラグはゼロになると推論される。

40

【0206】

概して、ビデオエンコーダ22およびビデオデコーダ30は、ブロックのサイズに基づいて、ブロックが関連する変換を有するかどうかを判定し、ブロックが関連する変換を有さない場合、ブロックに対してコーディングモードを再開するように構成され得る。

【0207】

図10は、本開示の技法を実装することができる例示的なビデオエンコーダ22を示すブロック図である。図10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において説明されるような技法の限定と見なされるべきでない。本開示の技法は、様々なコーディング規格または方法に適用可能であり得る。

50

【0208】

図10の例では、ビデオエンコーダ22は、予測処理ユニット100、ビデオデータメモリ101、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116、およびエントロ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)を含み得る。

【0209】

ビデオデータメモリ101は、ビデオエンコーダ22の構成要素によって符号化されるべきビデオデータを記憶するように構成され得る。ビデオデータメモリ101内に記憶されるビデオデータは、たとえば、ビデオソース18から取得され得る。復号ピクチャバッファ116は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオエンコーダ22によってビデオデータを符号化するために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であり得る。ビデオデータメモリ101および復号ピクチャバッファ116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)、抵抗性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。ビデオデータメモリ101および復号ピクチャバッファ116は、同一の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。様々な例において、ビデオデータメモリ101は、ビデオエンコーダ22の他の構成要素とともにオンチップであってもよく、または、これらの構成要素に対してオフチップであってもよい。ビデオデータメモリ101は、図1の記憶媒体20と同じであることがあり、またはその一部であることがある。

【0210】

ビデオエンコーダ22は、ビデオデータを受け取る。ビデオエンコーダ22は、ビデオデータのピクチャのスライス内の各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と関連付けられ得る。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、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することができる。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に関連するCTBをツリー構造に従って区分し得る。本開示の1つまたは複数の技法によれば、ツリー構造の各深度レベルにおけるツリー構造のそれぞれの非リーフノードごとに、それぞれの非リーフノードに対して複数の許容分割パターンがあり、それぞれの非リーフノードに対応するビデオブロックは、複数の許容可能な分割パターンのうちの1つに従って、それぞれの非リーフノードの子ノードに対応するビデオブロックに区分される。一例では、予測処理ユニット100またはビデオエンコーダ22の別の処理ユニットは、上記で説明したMTT区分技法の任意の組合せを実行するように構成され得る。

【0211】

ビデオエンコーダ22は、CTUのCUを符号化して、CUの符号化表現(すなわち、コーディングされた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の間で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。本開示の技法によれば、CUは単一のPUのみを含み得る。すなわち、本開示のいくつかの例では、CUは、別個の予測ブロックに分割されず、むしろ、CU全体に対して予測プロセスが実行される。したがって、各CU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と関連付けられ得る。ビデオエンコーダ22およびビデオデコーダ30は、様々なサイズを有するCUをサポートし得る。上記のように、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してよく、ルーマ予測ブロックのサイズを指してもよい。上記で論じたように、ビデオエンコーダ22およびビデオデコーダ30は、上記で説明した例示的なMTT区分タイプの任意の組合せによって定義されるCUサイズをサポートし得る。

【0212】

インター予測処理ユニット120は、CUの各PUに対してインター予測を実行することによ

10

20

30

40

50

って、PUに関する予測データを生成し得る。上記で説明したように、本開示のいくつかのMTT例では、CUは、単一のPUのみを含んでよく、すなわち、CUおよびPUは同期し得る。PUに対する予測データは、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。インター予測処理ユニット120は、PUがIスライス内にあるのか、Pスライス内にあるのか、それともBスライス内にあるのかに応じて、PUまたはCUに対して異なる動作を実行し得る。Iスライス内では、すべてのPUがイントラ予測される。したがって、PUがIスライス内にある場合、インター予測処理ユニット120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、同じフレーム内の以前に符号化された近隣ブロックから空間予測を使用して形成される。PUがPスライス内にある場合、インター予測処理ユニット120は、単方向インター予測を使用してPUの予測ブロックを生成し得る。PUがBスライス内にある場合、インター予測処理ユニット120は、単方向または双方向インター予測を使用してPUの予測ブロックを生成し得る。

10

【0213】

イントラ予測処理ユニット126は、PUに対してイントラ予測を実行することによって、PUの予測データを生成し得る。PUに関する予測データは、PUの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スライス内のPUに対して、イントラ予測を実行し得る。

【0214】

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の予測データの複数のセットを生成し得る。イントラ予測処理ユニット126は、近隣PUのサンプルブロックからのサンプルを使用して、PUに対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、近隣PUは、PUの上、右上、左上、または左であり得る。イ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個の方向性イントラ予測モードを使用することができる。いくつかの例では、イントラ予測モードの数は、PUに関連する領域のサイズに依存し得る。

20

【0215】

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によって生成された予測データ、またはPUに対してイントラ予測処理ユニット126によって生成された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/ひずみの尺度に基づいて、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、本明細書で選択予測ブロックと呼ばれることがある。

30

【0216】

残差生成ユニット102は、CUに対する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に対する選択予測ブロック(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に対する残差ブロック(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、残差ブロック内の各サンプルがCUのコーディングブロック内のサンプルとCUのPUの対応する選択予測ブロック内の対応するサンプルとの間の差分に等しい値を有するように、CUの残差ブロックを生成し得る。

40

【0217】

変換処理ユニット104は、4分木区分を実行して、CUに関連する残差ブロックをCUのTUに関連する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関連付けられ得る。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基づかなくてもよい。「残差4分木」(RQT)として知られる4分木構造が、領域の各々に関連するノードを含み得る。CUのTUは、RQTのリーフノードに相当し得る。別の例では、変換処理ユニット104は、上記で説明したMTT技法に従って、TUを区分するように構成され

50

得る。たとえば、ビデオエンコーダ22は、RQT構造を使用して、CUをTUにさらに分割することができない。したがって、一例では、CUは単一のTUを含む。

【0218】

変換処理ユニット104は、TUの変換ブロックに1つまたは複数の変換を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に関連する変換ブロック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得る。

【0219】

量子化ユニット106は、係数ブロック内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、量子化中にmビットの変換係数に切り捨てられてよく、nはmよりも大きい。量子化ユニット106は、CUに関連する量子化パラメータ(QP)値に基づいて、CUのTUに関連する係数ブロックを量子化し得る。ビデオエンコーダ22は、CUに関連するQP値を調整することによって、CUに関連する係数ブロックに適用される量子化の程度を調整し得る。量子化は、情報の喪失をもたらし得る。したがって、量子化された変換係数は、元の精度よりも精度が低いことがある。

【0220】

逆量子化ユニット108および逆変換処理ユニット110は、係数ブロックから残差ブロックを再構成するために、係数ブロックにそれぞれ逆量子化および逆変換を適用することができる。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に関連する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築することによって、ビデオエンコーダ22は、CUのコーディングブロックを再構築し得る。

【0221】

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に関連する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、再構成されたコーディングブロックに対して1つまたは複数のデブロッキング動作を実行した後、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、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て、他のピクチャのPUに対してインター予測を実行し得る。加えて、イントラ予測処理ユニット126は、復号ピクチャバッファ116内の再構成されたコーディングブロックを使用して、CUと同じピクチャ内の他のPUに対してイントラ予測を実行し得る。

【0222】

エントロピー符号化ユニット118は、ビデオエンコーダ22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、量子化ユニット106から係数ブロックを受信してよく、予測処理ユニット100からシンタックス要素を受信してよい。エントロピー符号化ユニット118は、データに対して1つまたは複数のエントロピー符号化動作を実行して、エントロピー符号化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118は、CABAC動作、コンテキスト適応型可変長コーディング(CAVLC)動作、可変長対可変(V2V)長コーディング動作、シンタックスベースコンテキスト適応型バイナリ算術コーディング(SBAC)動作、確率区間区分エントロピー(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22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、本開示の技法によるCUに対する区分構造を表すデータを含み得る。

【0223】

10

20

30

40

50

図11は、本開示の技法を実施するように構成された例示的なビデオデコーダ30を示すブロック図である。図11は説明のために提供され、本開示において広く例示され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いてビデオデコーダ30について説明する。しかしながら、本開示の技法は、他のコーディング標準規格または方法に適用可能とすることができる。

【0224】

図11の例において、ビデオデコーダ30は、エントロピー復号ユニット150、ビデオデータメモリ151、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、および復号ピクチャバッファ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。他の例では、ビデオデコーダ30は、より多数の、より少数の、または異なる機能構成要素を含み得る。

【0225】

ビデオデータメモリ151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、符号化ビデオビットストリームなどの符号化ビデオデータを記憶し得る。ビデオデータメモリ151内に記憶されるビデオデータは、たとえば、カメラなどのローカルビデオソースから、ビデオデータの有線ネットワーク通信もしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって、たとえば、コンピュータ可読媒体16から取得され得る。ビデオデータメモリ151は、符号化ビデオビットストリームからの符号化ビデオデータを記憶するコーディングされたピクチャバッファ(CPB:coded picture buffer)を形成し得る。復号ピクチャバッファ162は、たとえば、イントラコーディングモードまたはインターコーディングモードでビデオデコーダ30によってビデオデータを復号する際に使用するための、または出力のための、参照ビデオデータを記憶する参照ピクチャメモリであってよい。ビデオデータメモリ151および復号ピクチャバッファ162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)、抵抗性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。ビデオデータメモリ151および復号ピクチャバッファ162は、同じメモリデバイスまたは別個のメモリデバイスによって設けられてよい。様々な例では、ビデオデータメモリ151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。ビデオデータメモリ151は、図1の記憶媒体28と同じであってよく、またはその一部であってよい。

【0226】

ビデオデータメモリ151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)を受信して記憶する。エントロピー復号ユニット150は、符号化ビデオデータ(たとえば、NALユニット)をビデオデータメモリ151から受信し得、NALユニットをパースしてシンタックス要素を取得し得る。エントロピー復号ユニット150は、NALユニット内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。エントロピー復号ユニット150は、エントロピー符号化ユニット118のプロセスとは全般的に逆のプロセスを実行し得る。

【0227】

本開示のいくつかの例によれば、エントロピー復号ユニット150、またはビデオデコーダ30の別の処理ユニットは、ビットストリームからシンタックス要素を取得することの一部として、ツリー構造を判定することができる。ツリー構造は、CTBなどの最初のビデオブロックが、コーディングユニットなどのより小さいビデオブロックにどのように区分されるのかを指定し得る。本開示の1つまたは複数の技法によれば、ツリー構造の各深度レベルにおけるツリー構造のそれぞれの非リーフノードごとに、それぞれの非リーフノードに対して複数の許容区分タイプがあり、それぞれの非リーフノードに対応するビデオプロ

ックは、複数の許容可能な分割パターンのうちの1つに従って、それぞれの非リーフノードの子ノードに対応するビデオブロックに区分される。

【0228】

ビットストリームからシンタックス要素を取得することに加えて、ビデオデコーダ30は、区分されていないCUに対して再構築動作を実行し得る。CUに対して再構築動作を実行するために、ビデオデコーダ30は、CUの各TUに対して再構築動作を実行し得る。CUのTUごとに再構築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築し得る。上記で論じたように、本開示の一例では、CUは単一のTUを含む。

【0229】

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に関連する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に関連する残差ブロックを生成するために、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。

【0230】

CUまたは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の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、空間的に近隣するブロックのサンプルに基づいて、PUの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、ビットストリームから取得された1つまたは複数のシンタックス要素に基づいて、PU用のイントラ予測モードを判定することができる。

【0231】

PUがインター予測を使用して符号化されている場合、エントロピー復号ユニット150は、PUに関する動き情報を判定することができる。動き補償ユニット164は、PUの動き情報に基づいて、1つまたは複数の参照ブロックを判定することができる。動き補償ユニット164は、1つまたは複数の参照ブロックに基づいて、PUに対する予測ブロック(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。上記で論じたように、MTT区分を使用する、本開示の一例では、CUは単一のPUのみを含む。すなわち、CUを複数のPUに分割することはできない。

【0232】

再構成ユニット158は、CUのTUに対する変換ブロック(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、ルーマブロック、Cbブロック、およびCrブロック)、すなわち、適用可能な場合、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に対する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、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、予測ブロック(たとえば、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)の対応するサンプルに加算して、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。

【0233】

フィルタユニット160は、デブロッキング動作を実行して、CUのコーディングブロックに関連するブロッキ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を復号ピクチャバッファ162内に記憶し得る。復号ピクチャバッファ162は、後続の動き補償、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内のブロックに基づいて、他のCUのPUに対してイン

10

20

30

40

50

トラ予測動作またはインター予測動作を実行し得る。

【0234】

図12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12の技法は、変換処理ユニット104および/または量子化ユニット106を含む、ビデオエンコーダ22によって実行され得る。

【0235】

本開示の一例では、ビデオエンコーダ22は、ビデオデータのブロックを受信し(200)、残差ビデオデータを生成するために、ビデオデータのブロックを予測する(202)ように構成され得る。ビデオエンコーダ22は、残差ビデオデータに対する変換を判定することであって、変換が、2の冪ではないサイズSを有する、判定すること(204)と、Sを2の冪に丸めて、修正されたサイズS'を有する変換を生成することと(206)とを行うようにさらに構成され得る。ビデオエンコーダ22はまた、変換係数を生成するために、修正されたサイズS'を有する変換を残差ビデオデータに適用し(208)、符号化されたビデオストリーム内の変換係数を符号化する(210)ことができる。

【0236】

別の例では、ビデオエンコーダ22は、Sを2の最も近い冪に丸めるように構成され得る。

【0237】

別の例では、ビデオエンコーダ22は、変換係数を量子化するように構成され得る。

【0238】

別の例では、ビデオデータのブロックは、非正方形形状を有する。

【0239】

別の例では、Sは12であり、ビデオエンコーダ22は、12を16に丸めるように構成され得、修正されたサイズS'は16である。別の例では、Sは24であり、ビデオエンコーダ22は、24を32に丸めるように構成され得、修正されたサイズS'は32である。

【0240】

一例では、Sは変換の幅である。別の例では、Sは変換の高さである。

【0241】

図13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13の技法は、逆変換処理ユニット156および/または逆量子化ユニット154を含む、ビデオデコーダ30によって実行され得る。

【0242】

本開示の一例では、ビデオデコーダ30は、ビデオデータの符号化ブロックを受信することと(300)と、ビデオデータの符号化ブロックに対する変換を判定することであって、変換が、2の冪ではないサイズSを有する、判定すること(302)と、Sを2の冪に丸めて、修正されたサイズS'を有する逆変換を生成することと(304)とを行うように構成され得る。ビデオデコーダ30は、残差ビデオデータを生成するために、修正されたサイズS'を有する逆変換をビデオデータの符号化ブロックに適用し(306)、ビデオデータの復号ブロックを生成するために、残差ビデオデータを復号する(308)ようさらに構成され得る。

【0243】

一例では、ビデオデコーダ30は、Sを2の最も近い冪に丸めるように構成され得る。別の例では、Sは12であり、ビデオデコーダ30は、12を16に丸めるように構成され得、修正されたサイズS'は16である。別の例では、Sは24であり、ビデオデコーダ30は、24を32に丸めるように構成され得、修正されたサイズS'は32である。

【0244】

別の例では、ビデオデータの符号化ブロックは、逆量子化変換係数を含む。別の例では、ビデオデータの符号化ブロックは、非正方形形状を有する。一例では、Sは変換の幅である。別の例では、Sは変換の高さである。

【0245】

本開示の別の例では、ビデオデコーダ30は、S'に基づいて逆変換に対するシフト値を判定するように構成され得る。

【0246】

本開示のいくつかの態様は、説明を目的にHEVC規格の拡張に関して説明されている。しかしながら、本開示において説明される技法は、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。

【0247】

本開示で説明したようなビデオコードは、ビデオエンコードまたはビデオデコードを指すことがある。同様に、ビデオコーディングユニットは、ビデオエンコードまたはビデオデコードを指すことがある。同様に、ビデオコーディングは、適用可能な場合、ビデオ符号化またはビデオ復号を指すことがある。本開示では、「に基づいて」という句は、「だけにに基づいて」、「に少なくとも一部基づいて」、または「何らかの形で基づいて」を示し得る。本開示は、サンプルの1つまたは複数の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1つまたは複数のサンプルブロックおよびシンタックス構造を指すために、「ビデオユニット」または「ビデオブロック」または「ブロック」という用語を使用することがある。例示的なタイプのビデオユニットは、CTU、CU、PU、変換ユニット(TU)、マクロブロック、マクロブロック区分などを含み得る。いくつかのコンテキストでは、PUの議論はマクロブロックまたはマクロブロック区分の議論と相互に交換され得る。例示的なタイプのビデオブロックは、コーディングツリーブロック、コーディングブロック、およびビデオデータの他のタイプのブロックを含み得る。

【0248】

例に応じて、本明細書で説明した技法のいずれかのいくつかの行為またはイベントが、異なるシーケンスで実行されてよく、追加、併合、または完全に除外されてよい(たとえば、説明したすべての行為またはイベントが技法の実践にとって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、連続的にではなく、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並行して実行されてよい。

【0249】

1つまたは複数の例では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せ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、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する場合がある。データ記憶媒体は、本開示で説明した技法の実装のための命令、コード、および/またはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスされ得る、任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【0250】

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記憶デバイス、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を備え得る。また、いかなる接続もコンピュータ可読媒体と適切に呼ばれる。たとえば、命令が、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、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、光ファイバーケーブル、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。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)および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。上記の組合せもまた、コンピュータ可読媒体の範囲内に含まれるべきである。

【0251】

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ロジックアレイ(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、本明細書で使用する「プロセッサ」という用語は、上記の構造、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、いくつかの例では、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/もしくはソフトウェアモジュール内で与えられることがあり、または複合コーデックに組み込まれることがある。また、技法は、1つまたは複数の回路または論理要素で全体的に実装され得る。

【0252】

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、またはユニットについて本開示で説明したが、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よい。

【0253】

様々な例が記載されている。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】

【0254】

- 10 ビデオ符号化および復号システム
- 12 ソースデバイス
- 14 宛先デバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 20 記憶媒体
- 22 ビデオエンコーダ
- 24 出力インターフェース
- 26 入力インターフェース
- 28 記憶媒体
- 30 ビデオデコーダ
- 32 ディスプレイデバイス
- 50 ブロック
- 51 ブロック
- 52 ブロック
- 53 ブロック
- 54 ブロック

10

20

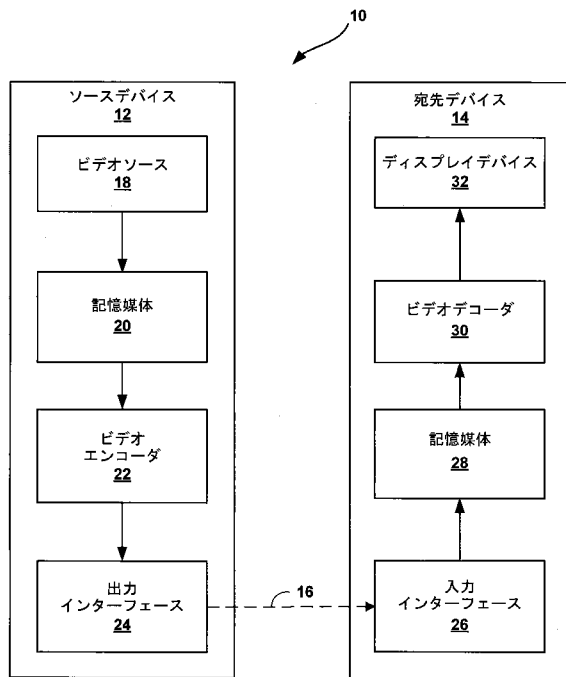
30

40

50

55	ブロック	
56	ブロック	
57	ブロック	
58	ブロック	
59	ブロック	
60	ブロック	
61	ブロック	
62	ブロック	
63	ブロック	
64	ブロック	10
65	ブロック	
66	ブロック	
70	ノード	
72	ノード	
74	ノード	
76	ノード	
78	ノード	
78	ノード	
80	ノード	
84	ノード	20
100	予測処理ユニット	
101	ビデオデータメモリ	
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	
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	30
118	エントロピー符号化ユニット	
120	インター予測処理ユニット	
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	
151	ビデオデータメモリ	
152	予測処理ユニット	
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	40
162	復号ピクチャバッファ	
164	動き補償ユニット	
166	イントラ予測処理ユニット	

【図 1】



【図 2】

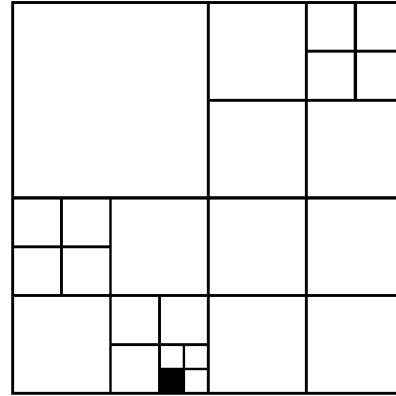


FIG. 2

【図 3】

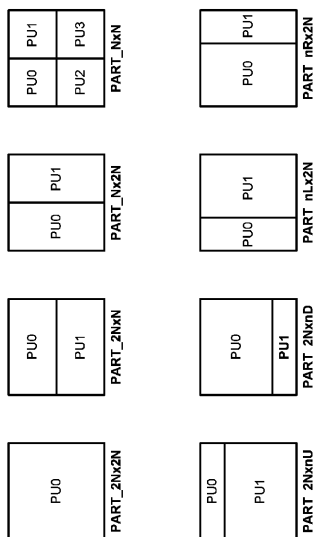


FIG. 3

【図 4 A】

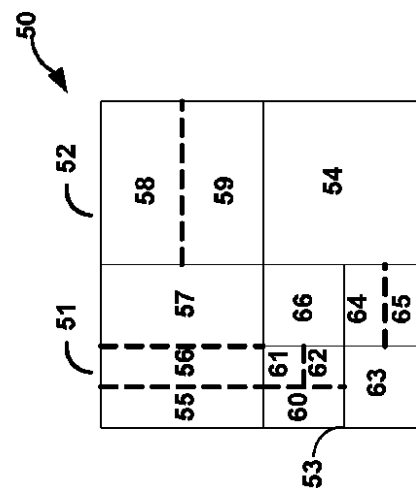


FIG. 4A

【図 7】

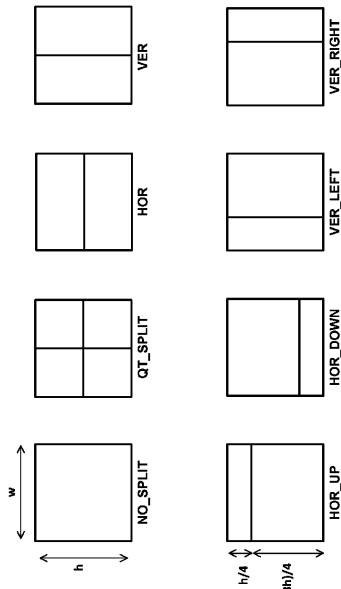


FIG. 7

【図 9】

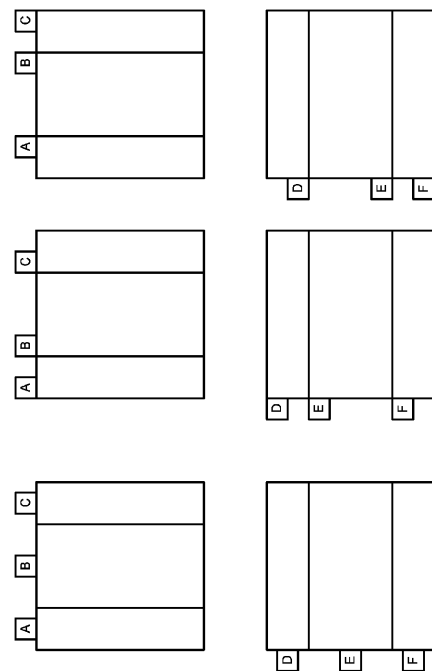
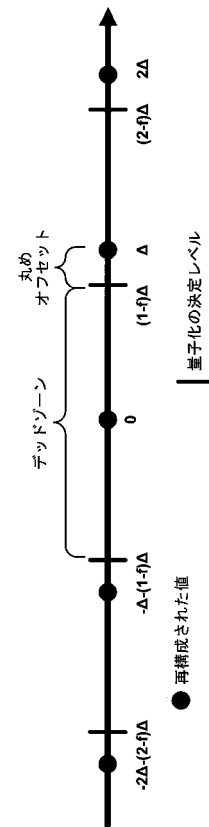
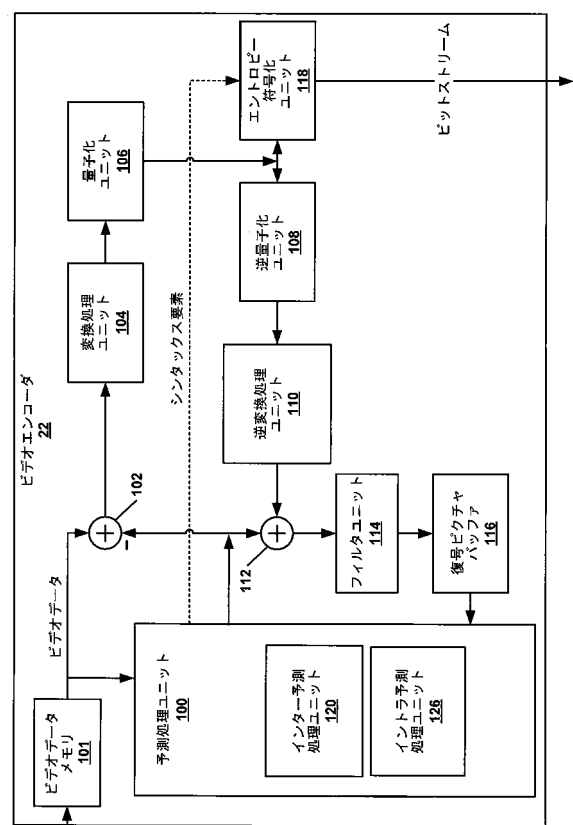


FIG. 9

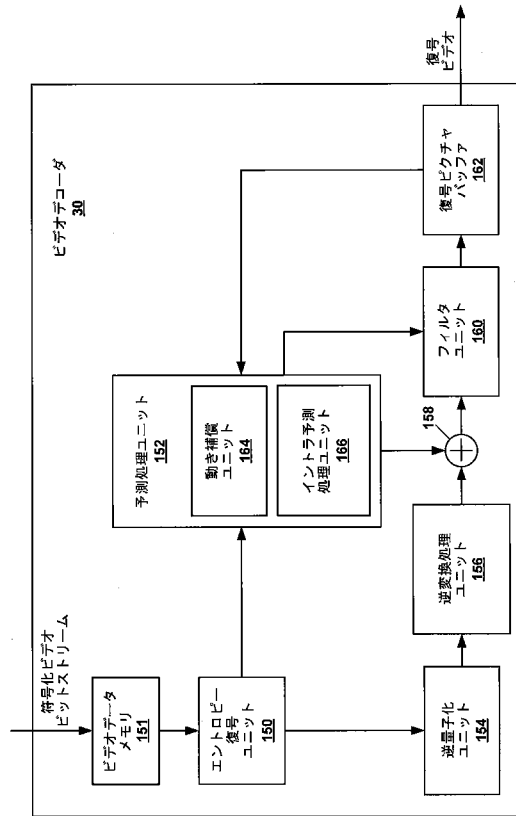
【図 8】



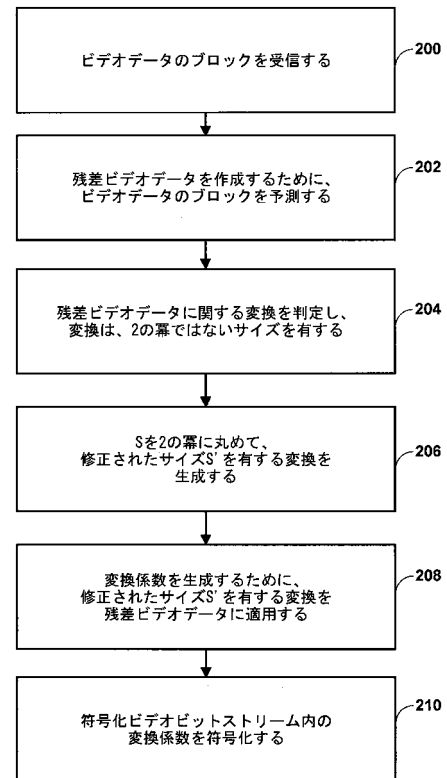
【図 10】



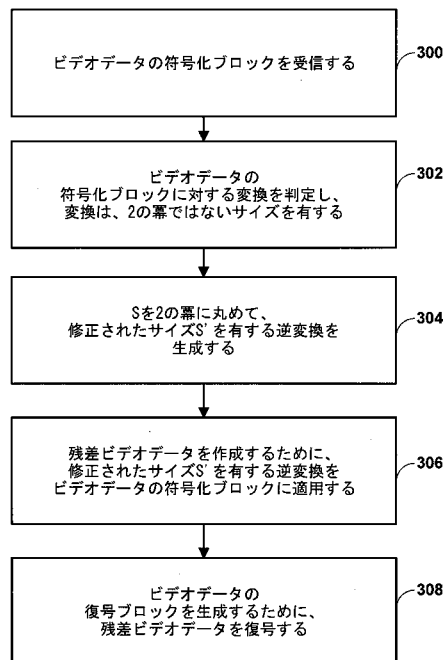
【図 1 1】



【図 1 2】



【図 1 3】



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/012589

A. CLASSIFICATION OF SUBJECT MATTER INV. H04N19/122 H04N19/126 H04N19/176 H04N19/186 H04N19/42 H04N19/70 ADD. According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H04N Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	F. Le L?annec ET AL: "Asymmetric Coding Units in QTBT", 4. JVET MEETING; 15-10-2016 - 21-10-2016; CHENGDU; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/, JEVT-D0064r1, 10 October 2016 (2016-10-10), pages 1-10, XP055417365, Chengdu,CN Retrieved from the Internet: URL:file:///C:/data-heising/MPEG/JEVT_D0064_10Oct17.pdf [retrieved on 2017-10-19] the whole document ----- -/--	1-40
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 28 March 2018		Date of mailing of the international search report 10/04/2018
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Lombardi, Giancarlo

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2018/012589

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>John C. Russ ET AL: "Introduction to Image Processing and Analysis" In: "Introduction to Image Processing and Analysis", 31 October 2007 (2007-10-31), CRC Press, Boca Raton, FL, USA, XP055462417, ISBN: 978-0-8493-7073-1 pages 133-142, the whole document</p>	1-40
A	<p>MADHUKAR BUDAGAVI ET AL: "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard", IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, vol. 7, no. 6, 1 December 2013 (2013-12-01), pages 1029-1041, XP055200337, ISSN: 1932-4553, DOI: 10.1109/JSTSP.2013.2270429 the whole document</p>	1-40
A	<p>FLYNN DAVID ET AL: "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, USA, vol. 26, no. 1, 1 January 2016 (2016-01-01), pages 4-19, XP011592176, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2015.2478707 [retrieved on 2016-01-05] pages 11-13, paragraph V.D</p>	1-40

フロントページの続き

(81)指定国・地域 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT

(72)発明者 シン・ジャオ

アメリカ合衆国・カリフォルニア・9 2 1 2 7・サン・ディエゴ・オールド・ストーンフィールド
・チェイス・8 5 2 6

(72)発明者 リ・ジャン

アメリカ合衆国・カリフォルニア・9 2 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

(72)発明者 ジエンレ・チェン

アメリカ合衆国・カリフォルニア・9 2 1 3 0・サン・ディエゴ・コルテ・デ・タイバーアン・1
0 7 5 6

(72)発明者 シャオ・チアン・チュアン

アメリカ合衆国・カリフォルニア・9 2 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

(72)発明者 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 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

Fターム(参考) 5C159 MA04 MA05 MA21 MC11 ME01 ME11 PP04 SS26 TA12 TB08

TC26 UA02 UA05