

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6612767号  
(P6612767)

(45) 発行日 令和1年11月27日(2019.11.27)

(24) 登録日 令和1年11月8日(2019.11.8)

(51) Int.Cl.	F 1
HO4N 19/126 (2014.01)	HO4N 19/126
HO4N 19/70 (2014.01)	HO4N 19/70
HO4N 19/186 (2014.01)	HO4N 19/186
HO4N 19/176 (2014.01)	HO4N 19/176
HO4N 19/136 (2014.01)	HO4N 19/136

請求項の数 15 (全 50 頁)

(21) 出願番号	特願2016-557043 (P2016-557043)
(86) (22) 出願日	平成27年3月13日 (2015.3.13)
(65) 公表番号	特表2017-513323 (P2017-513323A)
(43) 公表日	平成29年5月25日 (2017.5.25)
(86) 國際出願番号	PCT/US2015/020543
(87) 國際公開番号	W02015/138962
(87) 國際公開日	平成27年9月17日 (2015.9.17)
審査請求日	平成30年2月23日 (2018.2.23)
(31) 優先権主張番号	61/953,573
(32) 優先日	平成26年3月14日 (2014.3.14)
(33) 優先権主張国・地域又は機関	米国(US)
(31) 優先権主張番号	61/981,645
(32) 優先日	平成26年4月18日 (2014.4.18)
(33) 優先権主張国・地域又は機関	米国(US)

(73) 特許権者	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
(74) 代理人	100108453 弁理士 村山 靖彦
(74) 代理人	100163522 弁理士 黒田 晋平
(72) 発明者	リ・ジャン アメリカ合衆国・カリフォルニア・921 21-1714・サン・ディエゴ・モアハ ウス・ドライブ・5775

最終頁に続く

(54) 【発明の名称】色空間変換コーディングのための量子化パラメータ

## (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、

前記ビデオデータが色空間変換を使用して符号化されたかどうかを決定するステップと

、第1の量子化パラメータ(QP)に対する第1の値を決定するステップと、

第2のQPに対する第2の値を決定するステップと、

色空間変換が前記ビデオデータの符号化に使用されなかつたものと決定することに応答して、

前記第1のQPの前記第1の値を使用して第1の色成分のデータを逆量子化するステップ 10  
、および前記第2のQPの前記第2の値を使用して第2の色成分のデータを逆量子化するステップ  
と、

前記ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して

、前記第1のQPの第1のオフセットを使用して前記ビデオデータの前記第1の色成分の前  
記データを逆量子化するステップ、および前記第2のQPの第2のオフセットを使用して前記ビデオデータの第2の色成分の前記デ  
ータを逆量子化するステップであって、前記第2の色成分が前記第1の色成分と異なり、前  
記第2のQPの前記第2のオフセットが前記第1のQPの前記第1のオフセットと異なる、ステッ  
20

と、

を含み、

前記第1のオフセットが前記第1のQPに関連付けられ、前記第2のオフセットが前記第2のQPに関連付けられ、前記第1のオフセットおよび前記第2のオフセットが前記ビデオデータを符号化するために使用される色空間変換プロセスによって決まる、方法。

【請求項2】

前記ビデオデータが色空間変換を使用して符号化されたとき、前記ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加するステップをさらに含み、前記第1のオフセットが8に等しいか、または、

前記ビデオデータが色空間変換を使用して符号化されたとき、前記ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加するステップをさらに含み、前記第1のオフセットが2に等しく、

前記第1の色成分がルーマ成分であり、前記第2の色成分がクロマ成分である、

請求項1に記載の方法。

【請求項3】

前記ビデオデータが色空間変換を使用して符号化されたかどうかを決定するステップが、

前記ビデオデータのコーディングユニットの、前記コーディングユニットが色空間変換を使用して符号化されたかどうかを示す構文要素を復号するステップと、

前記構文要素の値が、前記ビデオデータが色空間変換を使用して符号化されたことを示すかどうかを決定するステップと、

前記構文要素が、前記コーディングユニットが色空間変換を使用してコーディングされたことを示すものと決定することに応答して、色空間逆変換プロセスを適用するステップとを含み、前記構文要素が1ビットフラグを含み、前記構文要素に対する1の値が、前記コーディングユニットが色空間変換を使用して符号化されたことを示す、

請求項1に記載の方法。

【請求項4】

前記構文要素が、前記コーディングユニットがイントラパルス符号変調(IPCM)モード以外のモードでコーディングされるときにのみ、前記コーディングユニットが色空間変換を用いてコーディングされたことを示すようにシグナリングされ、かつ／または、

前記構文要素が、前記コーディングユニットの変換単位内に非ゼロ係数が存在するときに前記コーディングユニットが色空間変換を使用して符号化されたことを示し、かつ／または、

前記構文要素が、前記コーディングユニットがイントラコーディングされるとき、および前記コーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるとき、前記コーディングユニットが色空間変換を使用して符号化されなかったことを示し、前記構文要素が、前記ビデオデータを含む受信されたビットストリーム内に存在せず、前記構文要素を復号するステップが、前記構文要素の前記値を推測するステップを含み、かつ／または、

前記構文要素が、前記コーディングユニットがパレットモードでコーディングされるときに前記コーディングユニットが色空間変換を使用して符号化されなかったことを示し、前記構文要素が、前記ビデオデータを含む受信されたビットストリーム内に存在せず、前記構文要素を復号するステップが、前記構文要素の前記値を推測するステップを含む、

請求項3に記載の方法。

【請求項5】

前記ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、色空間逆変換プロセスをコーディングユニットに適用するステップであって、前記色空間逆変換プロセスが、元のYCoCg-R逆変換行列を含むか、または、修正されたYCoCg逆変換行列を含むか、または、元のYCoCg変換行列を含む、ステップをさらに含み、

前記色空間逆変換プロセスが修正されたYCoCg逆変換行列を含む場合、前記修正されたY

10

20

30

40

50

CoCg逆変換行列の係数が、丸めオフセットによって2ビット右シフトした後、元のYCoCg逆変換行列の係数に等しく設定され、前記修正されたYCoCg行列が、2に等しいオフセットを有し、または、

前記色空間逆変換プロセスが元のYCoCg変換行列を含む場合、前記方法はさらに、正規化プロセスを前記ビデオデータに適用するステップを含む、  
請求項1に記載の方法。

**【請求項 6】**

ビデオデータを記憶するように構成されたメモリと、

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、

前記ビデオデータが色空間変換を使用して符号化されたかどうかを決定することと、

前記ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して

、  
第1の量子化パラメータ(QP)の第1のオフセットを使用して前記ビデオデータの第1の色成分のデータを逆量子化することと、および

第2のQPの第2のオフセットを使用して前記ビデオデータの第2の色成分のデータを逆量子化することとを行うように構成され、前記第2の色成分が前記第1の色成分と異なり、前記第2のQPの前記第2のオフセットが、前記第1のQPの前記第1のオフセットと異なり、前記第1のオフセットおよび前記第2のオフセットが、前記ビデオデータを符号化するために使用される色空間変換プロセスによって決まる、ビデオ復号デバイス。

**【請求項 7】**

前記ビデオデータが色空間変換を使用して符号化されたかどうかを決定することが、前記ビデオデータのコーディングユニットの、前記コーディングユニットが色空間変換を使用して符号化されたかどうかを示す構文要素を復号することを含み、

前記構文要素が、IPCMモード以外のモードを使用するコーディングユニットに対してだけシグナリングされる、請求項6に記載のビデオ復号デバイス。

**【請求項 8】**

ビデオデータを符号化する方法であって、

前記ビデオデータを符号化するために色空間変換を使用するかどうかを決定するステップと、

前記ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、

第1の量子化パラメータ(QP)の第1のオフセットを使用して前記ビデオデータの第1の色成分のデータを量子化するステップ、および

第2のQPの第2のオフセットを使用して前記ビデオデータの第2の色成分のデータを量子化するステップとを含み、前記第2の色成分が前記第1の色成分と異なり、前記第2のQPの前記第2のオフセットが、前記第1のQPの前記第1のオフセットと異なり、前記第1のオフセットおよび前記第2のオフセットが、前記ビデオデータを符号化するために使用される色空間変換プロセスによって決まる、方法。

**【請求項 9】**

前記ビデオデータの符号化に色空間変換を使用することを決定することに応答して、

前記ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加するステップであって、前記第1のオフセットが8に等しい、ステップ、または、

前記ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加するステップであって、前記第1のオフセットが2に等しい、ステップ、

をさらに含み、

前記第1の色成分がルーマ成分であり、前記第2の色成分がクロマ成分である、

請求項8に記載の方法。

**【請求項 10】**

10

20

30

40

50

前記ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、前記ビデオデータが色空間変換を使用して符号化されたことを示す構文要素の値を設定するステップをさらに含み、前記構文要素が1ビットフラグを含み、前記構文要素に対する1の値が、コーディングユニットが色空間変換を用いてコーディングされたことを示す、請求項8に記載の方法。

【請求項11】

前記構文要素が、前記コーディングユニットがイントラパルス符号変調(IPCM)モード以外のモードでコーディングされるときにのみ、前記コーディングユニットが色空間変換を用いてコーディングされたことを示すようにシグナリングされ、かつ／または、

前記構文要素が、前記コーディングユニットの変換単位内に非ゼロ係数が存在するときに前記コーディングユニットが色空間変換を使用して符号化されたことを示し、かつ／または、

前記構文要素が、前記コーディングユニットがイントラコーディングされるとき、および前記コーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるとき、シグナリングされず、かつ／または、

前記構文要素が、前記コーディングユニットがパレットモードでコーディングされるときにシグナリングされない、

請求項10に記載の方法。

【請求項12】

前記ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、前記色空間変換プロセスをコーディングユニットに適用するステップであって、前記色空間変換プロセスが、元のYCoCg-R変換行列を含むか、または、修正されたYCoCg逆変換行列を含むか、または、元のYCoCg変換行列を含む、ステップをさらに含み、

前記色空間変換プロセスが修正されたYCoCg逆変換行列を含む場合、前記修正されたYCoCg変換行列の係数が、丸めオフセットによって2ビット右シフトした後、元のYCoCg逆変換行列の係数に等しく設定され、前記修正されたYCoCg行列が、2に等しいオフセットを有し、または、

前記色空間変換プロセスが元のYCoCg変換行列を含む場合、前記方法はさらに、正規化プロセスを前記ビデオデータに適用するステップを含む、

請求項8に記載の方法。

【請求項13】

ビデオデータを記憶するように構成されたメモリと、

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、

前記ビデオデータを符号化するために色空間変換を使用するかどうかを決定することと、

前記ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、

第1の量子化パラメータ(QP)の第1のオフセットを使用して前記ビデオデータの第1の色成分のデータを量子化すること、および

第2のQPの第2のオフセットを使用して前記ビデオデータの第2の色成分のデータを量子化することとを行うように構成され、前記第2の色成分が前記第1の色成分と異なり、前記第2のQPの前記第2のオフセットが、前記第1のQPの前記第1のオフセットと異なり、前記第1のオフセットおよび前記第2のオフセットが、前記ビデオデータを符号化するために使用される色空間変換プロセスによって決まる、ビデオ符号化デバイス。

【請求項14】

前記第1のQPと前記第2のQPとの間の差が、前記ビデオデータを符号化するために使用される色空間変換プロセスのノルムによって決まる、請求項1または8に記載の方法、または、請求項6に記載のビデオ復号デバイスまたは請求項13に記載のビデオ符号化デバイス。

【請求項15】

命令を含むコンピュータ可読記憶媒体であって、前記命令が、実行されるとき、ビデオ

10

20

30

40

50

復号デバイスの1つまたは複数のプロセッサに、請求項1乃至5および8乃至12の何れか1項に記載の方法を実施させる、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】

本出願は、2014年3月14日に出願された米国仮出願第61/953,573号、2014年4月18日に出願された米国仮出願第61/981,645号、および2014年10月10日に出願された米国仮出願第62/062,637号の利益を主張し、これらの米国仮出願の各々の内容全体が参照によって本明細書に組み込まれる。

【0002】

10

本開示はビデオコーディングに関し、より詳細には色空間変換を使用するビデオコーディングに関する。

【背景技術】

【0003】

デジタルビデオ機能は、デジタルテレビ、デジタル直接放送システム、ワイヤレス放送システム、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、電子書籍リーダ、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲームデバイス、ビデオゲームコンソール、セルラーまたは衛星無線電話、いわゆる「スマートフォン」ビデオ会議デバイス、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ることが可能である。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、パート10、アドバンストビデオコーディング(AVC:Advanced Video Coding)、現在開発中の高効率ビデオコーディング(HEVC:High Efficiency Video Coding)規格、およびそのような規格の拡張によって定義された規格に記載されるものなどのビデオコーディング技法を実装する。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装することによって、デジタルビデオ情報をより効率的に送信、受信、符号化、復号、および/または記憶することがある。

20

【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ために、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もよく、ビデオブロックはまた、ツリーブロック、コーディング単位(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がある。ピクチャのイントラコード化(I)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに対する空間予測を用いて符号化される。ピクチャのインターフォーマット(PまたはB)スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルに対する空間予測、または他の参照ピクチャ内の参照サンプルに対する時間予測を使用してもよい。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。

30

【0005】

空間または時間予測は、コーディングされるべきブロックのため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を表す。インターフォーマットブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルと、コード化ブロックと予測ブロックとの間の差を示す残差データとに従って符号化される。イントラコード化ブロックは、イントラコーディングモードと残差データとに従って符号化される。さらなる圧縮のために、残差データは、ピクセル領域から変換領域に変換されてもよく、次いで量子化されてもよい残差変換係数をもたらす。最初に2次元アレイに配置される量子化変換係数は、変換係数の1次元ベクトルを生成するために、走査されてもよく、エントロピーコーディングが、さらに多くの圧縮を達成するために適用されてもよい。

40

【先行技術文献】

50

## 【特許文献】

## 【0006】

【特許文献1】米国仮出願第61/810,649号

## 【発明の概要】

## 【課題を解決するための手段】

## 【0007】

一般に、本開示は、色空間変換プロセスを使用してビデオブロックをコーディングするための技法を説明する。ビデオエンコーダまたはビデオデコーダなどのビデオコーダは、ビデオデータを符号化するために色空間変換を使用するかどうかを決定し得る。ビデオコーダは、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化し得、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。ビデオコーダは、さらに、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る。ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、ビデオコーダは、第1のQPの第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化し得る。ビデオデータが色空間変換を使用して符号化されなかったものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに第1のQPおよび第2のQPを使用してビデオデータを逆量子化し得る。

## 【0008】

一例では、ビデオデータを復号する方法について説明する。方法は、ビデオデータが色空間変換を使用して符号化されたかどうかを決定するステップを含む。方法は、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化するステップと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化するステップとをさらに含み、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

## 【0009】

別の例では、デバイスは、ビデオデータを記憶するように構成されたメモリと、ビデオデータが色空間変換を使用して符号化されたかどうかを決定するように構成された1つまたは複数のプロセッサとを含む。加えて、1つまたは複数のプロセッサはまた、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化することとを行うように構成され、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

## 【0010】

別の例では、デバイスは、ビデオデータが色空間変換を使用して符号化されたかどうかを決定するための手段を含む。デバイスは、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化するための手段と、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化するための手段とをさらに含み、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

## 【0011】

別の例では、コンピュータ可読記憶媒体は命令を含み、その命令は、実行されるとき、1つまたは複数のプロセッサに、ビデオデータが色空間変換を使用して符号化されたかどうかを決定することを行わせる。命令はまた、実行されるとき、プロセッサに、ビデオデ

10

20

30

40

50

ータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化することを行わせ、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

【0012】

別の例では、ビデオデータを符号化するための方法について説明する。方法は、ビデオデータを符号化するために色空間変換を使用するかどうかを決定するステップを含む。方法は、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化するステップと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化するステップとをさらに含み、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

10

【0013】

別の例では、デバイスは、ビデオデータを記憶するように構成されたメモリと、ビデオデータを符号化するために色空間変換を使用するかどうかを決定するように構成された1つまたは複数のプロセッサとを含む。1つまたは複数のプロセッサは、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化することを行なうようにさらに構成され、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

20

【0014】

別の例では、デバイスは、ビデオデータを符号化するために色空間変換を使用するかどうかを決定するための手段を含む。デバイスは、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化するための手段と、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化するための手段とをさらに含み、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

30

【0015】

別の例では、コンピュータ可読記憶媒体は命令を含み、その命令は、実行されるとき、1つまたは複数のプロセッサに、ビデオデータを符号化するために色空間変換を使用するかどうかを決定することを行わせる。命令は、さらに、1つまたは複数のプロセッサに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化することを行わせ、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

40

【0016】

本開示の1つまたは複数の例の詳細を添付の図面および以下の説明に記載する。本開示の他の特徴、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】本開示に記載の技法を利用してよい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2】本開示に記載の技法を実装してもよい例示的なビデオエンコーダを示すブロック図である。

【図3】本開示に記載の技法を実装してもよい例示的なビデオデコーダを示すブロック図

50

である。

【図4】本開示の1つまたは複数の技法による35のHEVC予測モードを示す概念図である。

【図5】本開示の1つまたは複数の技法による、マージのための空間的隣接動きベクトル候補とアドバンスト動きベクトル予測(AMVP)モードとを示す概念図である。

【図6】本開示の1つまたは複数の技法による、イントラブロックコピー(BC)の例を示す概念図である。

【図7】本開示の1つまたは複数の技法による、イントラ $8 \times 8$ ブロックに対するターゲットブロックおよび参照サンプルの一例を示す概念図である。

【図8】本開示の1つまたは複数の技法による符号化技法を示すフロー図である。

【図9】本開示の1つまたは複数の技法による復号技法を示すフロー図である。

10

【発明を実施するための形態】

【0018】

いくつかの例では、本開示はスクリーンコンテンツコーディングに関し、高クロマサンプリングフォーマット4:4:4が使用される。いくつかの例では、本開示はまた、場合によつては高ビット深度(8ビット超)、高クロマサンプリングフォーマット4:4:4のサポートを含む範囲拡張(RCEx)に対して適用可能である。いくつかの例では、本開示はまた、クロマサンプリングフォーマット4:2:2など、他の色フォーマットに適用可能である。より具体的には、本開示では、色空間変換に関する多くの異なる技法について説明する。

【0019】

ビデオコーディング規格には、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、およびスケーラブルビデオコーディング(SVC)およびマルチビュービデオコーディング(MVC)拡張を含むITU-T H.264(ISO/IEC MPEG-4 AVCとも呼ばれる)が含まれる。

20

【0020】

新しいビデオコーディング規格、すなわち高効率ビデオコーディング(HEVC)の設計は、ITU-Tビデオコーディング専門家グループ(VCEG)およびISO/IEC動画専門家グループ(MPEG)のビデオコーディング共同研究部会(JCT-VC)によって確定された。

【0021】

HEVCでは、スライスの中の最大のコーディングユニットは、コーディングツリーブロック(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分木を含む。CTBのサイズは、HEVCメインプロファイルにおいて $16 \times 16 \sim 64 \times 64$ に及ぶことがあるが、 $8 \times 8$  CTBサイズがサポートされてよい。コーディングユニット(CU)は、CTBと同じサイズであつてよく、 $8 \times 8$ の大きさしかない。各CUは、1つのモードでコーディングされる。CUがインターコーディングされるとき、CUは、2つの予測単位(PU:prediction unit)にさらに区分されてよく、またはさらなる区分が適用されないときに单一のPUになる。1つのCUの中に2つのPUが存在するとき、各PUは、CUの半分のサイズの長方形であつてよく、またはCUの1/4もしくは3/4のサイズに等しいサイズの2つの長方形であつてよい。

30

【0022】

CUがインターコーディングされるとき、動き情報の1セットが各PUに対して存在する。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。HEVCでは、最小のPUサイズは、 $8 \times 4$ および $4 \times 8$ である。

40

【0023】

HEVCは、予測残差をコーディングするために $4 \times 4$ 、 $8 \times 8$ 、 $16 \times 16$ および $32 \times 32$ のサイズを有する4つの変換単位(TU:transform unit)を指定する。CTBは、4つ以上のTUに再帰的に区分されてよい。TUは、離散コサイン変換(DCT)関数に類似する整数ベースの関数を使用する。加えて、イントラコード化領域に属する $4 \times 4$ ルーマ変換ブロックは、離散サイン変換(DST)関数から導出される整数変換を使用して変換される。クロマは、ルーマと同じTUサイズを使用する。

【0024】

現在のHEVC規格では、各予測単位(PU)のルーマ成分に対して、図4に関して以下で説明

50

するように、イントラ予測方法が、33の角度予測モード(2から34までインデックス付けられる)、平均値モード(DC mode)(1でインデックス付けられる)、および平面モード(0でインデックス付けられる)とともに利用される。

【 0 0 2 5 】

上記の35のイントラモードに加えて、「I-PCM」という名称のもう1つのモードもまた、HEVCによって利用される。I-PCMモードでは、予測、変換、量子化、およびエントロピーコーディングがバイパスされる一方で、予測サンプルが所定の数のビットだけコーディングされる。I-PCMモードの主たる目的は、信号が他のモードによって効率的にコーディングされ得ない状況を処理することである。

【 0 0 2 6 】

現在のHEVC規格では、利用可能な2つのインター予測モードがある。1つのそのようなモードは、マージモード(マージの特別な場合としてスキップが考慮されている)であり、第2のそのようなモードは、予測単位(PU)に対するアドバンスト動きベクトル予測(AMVP)モードである。

【 0 0 2 7 】

AMVPモードとマージモードの両方において、動きベクトル(MV)候補リストが、複数の動きベクトル予測子のために維持される。現在のPUの動きベクトル、ならびにマージモードにおける参照インデックスは、MV候補リストから1つの候補を取ることによって生成される。

【 0 0 2 8 】

MV候補リストは、マージモードに対して5つまでの候補を含んでよく、AMVPモードに対して2つだけの候補を含んでよい。マージ候補は、動き情報のセット、たとえば参照ピクチャリスト(リスト0およびリスト1)と参照インデックスの両方に対応する動きベクトルを含んでよい。マージ候補がマージインデックスによって識別される場合、参照ピクチャが現在のブロックの予測のために使用されるだけでなく、関連する動きベクトルが決定される。しかしながら、AMVP候補は動きベクトルだけを含むので、リスト0またはリスト1のいずれかからの潜在的予測方向の各々に対するAMVPモードのもとで、参照インデックスは、MV候補リストに対するMVPインデックスとともに明示的にシグナリングされる必要がある。AMVPモードでは、予測された動きベクトルは、さらに改善され得る。

【 0 0 2 9 】

マージ候補は動き情報の全セットに対応する一方で、AMVP候補は特定の予測方向および参照インデックスに対する1つの動きベクトルだけを含んでよい。両モードに対する候補は、同じ空間的および時間的隣接ブロックから同様に導出される。

【 0 0 3 0 】

図5に関して以下で説明するように、空間的MV候補は、特定のPU(PU0)に対して図5に示す隣接ブロックから導出される。しかしながら、ブロックから候補を生成する方法は、マージモードおよびAMVPモードに対して異なる。

【 0 0 3 1 】

マージモードでは、4つまでの空間的MV候補が、図5に番号で示される順番に導出され得る。図5(a)に示すように、順番は次の通り、すなわち左(0)、上(1)、右上(2)、左下(3)、および左上(4)である。

【 0 0 3 2 】

図5(b)に示すように、AMVPモードでは、隣接ブロックは2つのグループ、すなわちブロック0および1を含む左のグループと、ブロック2、3および4を含む上のグループとに分割される。各グループに対して、シグナリングされた参照インデックスによって示されるものと同じ参照ピクチャを参照する隣接ブロック内の潜在的候補は、グループの最終候補を形成するために選択されるための最高の優先度を有する。すべての隣接ブロックが、同じ参照ピクチャを指す動きベクトルを含まないことが可能である。そのために、そのような候補を見出すことができない場合、第1の利用可能な候補が、最終候補を形成するようにスケーリングされることになり、したがって時間的距離の差が補償され得る。

10

20

30

40

50

## 【0033】

図6に関して以下で説明するように、イントラブロックコピー(BC)がSCCに含まれている。イントラBCの一例が図6に示されており、現在のCUは、現在のピクチャ/スライスのすでに復号されたブロックから予測される。現在のイントラBCブロックのサイズは、CUのサイズと同じ大きさであってよく、CUのサイズは8×8から64×64までに及ぶが、いくつかの適用例ではさらなる制約が適用されることがある。

## 【0034】

いくつかの例では、残差信号に対するループ内色空間変換プロセスは、4:4:4クロマフォーマット内のシーケンスに対して使用され得る。この方法は、RGB/YUVクロマフォーマット内の予測誤差信号を準最適色空間内の予測誤差信号に変換する。この追加のステップによって、色成分間の相関がさらに低減され得る。変換行列は、特異値分解(SVD)によって各コーディングユニットに対するピクセルサンプル値から導出される。色空間変換は、イントラモードとインターモードの両方の予測誤差に適用される。10

## 【0035】

いくつかの例では、色空間変換プロセスがインターモードに適用されるとき、残差が、最初に、導出された変換行列を用いて異なる領域に変換される。色空間変換の後、DCT/DST、量子化、およびエントロピーコーディングなど、従来のコーディングステップが、順番に実行される。

## 【0036】

いくつかの例では、色空間変換プロセスがイントラモードに適用されるとき、予測ブロックおよび現在のブロックが最初に、それぞれ、導出された変換行列を用いて異なる領域に変換される。色空間変換の後、現在のブロックとその予測子との間の残差が、さらに、DCT/DSTを用いて変換され、量子化され、かつエントロピーコーディングされる。20

## 【0037】

順方向演算では、色空間変換行列が、次のように3つの平面G、BおよびRに適用される。

## 【0038】

## 【数1】

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}$$

30

## 【0039】

いくつかの例では、得られた値は

## 【0040】

## 【数2】

$$\sqrt{3}$$

## 【0041】

倍まで拡大されるので、その値はHEVC仕様の範囲内にクリップされる。逆演算では、色空間変換行列は、次のように3つの成分P'、Q'およびR'に適用される。40

## 【0042】

## 【数3】

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}$$

## 【0043】

50

図7に関して以下で説明するように、変換行列は、参照サンプル値から導出されてよい。イントラの場合およびインターの場合に対して、異なる参照サンプルが利用され得る。イントラコード化ブロックの場合に対して、ターゲットブロックおよび参照サンプルを図7に示す。この例では、ターゲットブロックは、 $8 \times 8$ のクロスハッチのサンプルからなり、参照サンプルは、ストライプとドットのサンプルである。

## 【0044】

インターコード化ブロックの場合に対して、行列導出に対する参照サンプルは、動き補償に対する参照サンプルと同じである。シフト演算を実現するために、AMPブロック内の参照サンプルは、サンプルの数が2のべき乗になるようにサブサンプリングされる。たとえば、 $12 \times 16$ ブロック内の参照サンプルの数は、 $2/3$ に低減される。

10

## 【0045】

本開示の技法によれば、色空間変換プロセスは、各CUに適用されてよい。そのために、変換プロセスが起動されるか否かをシグナリングする必要はない。加えて、エンコーダ側とデコーダ側の両方が、変換行列をシグナリングするためのオーバーヘッドを回避するために同じ方法を用いて変換行列を導出する。

## 【0046】

本開示の技法によれば、色空間変換行列などの色空間変換プロセスが使用される。そのような行列の1つはYCbCr変換行列であり、それは、

順方向：

## 【0047】

20

## 【数4】

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

## 【0048】

逆方向：

## 【0049】

## 【数5】

30

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

## 【0050】

である。

## 【0051】

別のそのような行列はYCoCg変換行列であり、それは、

順方向：

## 【0052】

40

## 【数6】

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

## 【0053】

逆方向：

## 【0054】

50

## 【数7】

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

## 【0055】

である。

## 【0056】

別のそのような行列はYCoCg-R行列であり、それは、CoおよびCg成分を2倍にスケーリングするYCoCg行列の修正バージョンである。リフティング技法を使用することによって、順変換および逆変換が、次式によって達成され得る。

順方向：

$$C_o = R - B$$

$$t = B + [C_o / 2]$$

$$Cg = G - t$$

$$Y = t + [Cg / 2]$$

逆方向：

$$t = Y - [Cg / 2]$$

$$G = Cg + t$$

$$B = t - [C_o / 2]$$

$$R = B + C_o$$

## 【0057】

上記の式および行列では、順変換は、(たとえば、ビデオエンコーダによる)符号化プロセスの間に実行されてよく、逆変換は、(たとえば、ビデオデコーダによる)復号プロセスの中で実行されてよい。

## 【0058】

伝統的ビデオコーディングでは、画像は、連続階調を有し、空間的に平滑であることが仮定されている。これらの仮定に基づいて、自然なコンテンツを有するビデオに対して良好な性能を示している様々なツールが(たとえば、ロックベース変換、フィルタ処理、など)が開発されてきた。しかしながら、いくつかの適用例(たとえば、リモートデスクトップ、共同作業ディスプレイ、およびワイヤレスディスプレイ)では、コンピュータ生成スクリーンコンテンツが、圧縮されるべき主要コンテンツであることがある。このタイプのコンテンツは、離散階調を有し、かつ明確な線と高コントラストの対象境界を特徴づける傾向がある。連続階調および平滑性の仮定はもはや適用されず、したがって伝統的ビデオコーディング技法は効率的に働かないことがある。

## 【0059】

パレットモードコーディングが、上記の欠陥を克服するために使用されてよい。パレットコーディング技法の例は、2013年4月10日に出願された米国仮出願第61/810,649号に記載されている。各CUに対して、現在のCUの中で最も支配的なピクセル値を含むパレットが、導出されてよい。パレットのサイズおよび要素が、最初に送信される。送信後、CU中のピクセルが、一定の走査順序に従って符号化される。各ロケーションに対して、ピクセル値が、パレット内で(「実行モード」)であるか、または(「ピクセルモード」)でないかを示すために、フラグpalette\_flagなどの構文要素が最初に送信される。

## 【0060】

「実行モード」では、パレットインデックスがシグナリングされてよく、「run」が後続する。runは、現在コーディングされているピクセルと同じパレットインデックス値を有する、走査順序において連続するピクセルの数を示す構文要素である。走査順序において直後の複数のピクセルが、同じパレットインデックス値を有する場合、palette\_flagなどの構文要素によって、「実行モード」が示されてよい。現在のピクセルと同じパレット

10

20

30

40

50

インデックス値を有する、現在のピクセルに後続するピクセル数に等しいカウンタ値が決定されてよく、runがカウンタ値に等しく設定される。「run」によってカバーされる後続の位置はすべて、同じピクセル値を有するので、それらの位置に対して、palette\_flagもパレットインデックスも送信される必要はない。デコーダ側において、現在のピクセルに対する第1のパレットインデックス値だけが復号され、その結果が、「run」構文要素内に示されるピクセルの「run」内の各ピクセルに対して複製されることになる。

#### 【0061】

「ピクセルモード」では、ピクセルサンプル値が、この位置に対して送信される。palette\_flagなどの構文要素が「ピクセルモード」を示す場合、パレットインデックス値は、復号される1つのピクセルに対して決定されるのみである。

10

#### 【0062】

従来の技法は、様々な問題を被ることがある。たとえば、シーケンス特性および局所多様性を考慮に入れない色空間変換が、起動されることがある。そのために、コーディング性能が、準最適であることがある。別の例では、変換行列の導出がデコーダにおいて必要とされることがある、そのことが、デコーダの複雑さを著しく増大させる。さらに、変換行列は、空間的に再構成されたピクセルまたはインターフォーマットPUの予測子のいずれかを使用して導出されることがある。しかしながら、PUサイズが比較的小さいとき、予測があまり正確でないとき、または隣接ピクセルが利用できないときに、変換行列の効率が低減されることがある。本開示の技法は、これらの問題のうちの1つまたは複数を克服することができる。

20

#### 【0063】

ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、本開示で説明する技法を実行し得る。一般に、本開示は、色空間変換プロセスを使用してビデオブロックをコーディングするための技法を説明する。ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、ビデオデータを符号化するために色空間変換を使用するかどうかを決定し得る。ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオコーダは、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化し得、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。ビデオコーダは、さらに、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る。ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、ビデオコーダは、第1のQPの第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化し得る。ビデオデータが色空間変換を使用して符号化されなかったものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに第1のQPおよび第2のQPを使用してビデオデータを逆量子化し得る。

30

#### 【0064】

第1のQPおよび第2のQPは、色空間変換によらずにコーディングされたブロックと同じ導出方法に従って、第1および第2の色成分に対応して導出されたQPであり得る。たとえば、第1のQPおよび第2のQPに対するQPの値は、cu\_qp\_delta\_absなどのシグナリングされた構文要素から導出され得る。次いで、第1のオフセットおよび第2のオフセットは、それぞれ、ビデオデータが色空間変換を使用して符号化されたとき、第1のQPおよび第2のQPを取得するために、導出されたQPに加算されてよい。たとえば、QP(たとえば、QP<sub>i</sub>)に対して、それぞれのオフセット(たとえば、offset<sub>i</sub>)は、QPに加算されてよく、得られたQPの値を、元のQPプラスそれぞれのオフセットに等しくさせる(たとえば、QP<sub>finali</sub>=QP<sub>i</sub>+offset<sub>i</sub>)。

40

#### 【0065】

本開示は、ループ内色空間変換のコーディング性能を改善し、デコーダの複雑さを低減し得る技法を説明する。図1は、スクリーンコンテンツコーディングのための技法を利用

50

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り、高いクロマサンプリングフォーマットが使用される。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されたビデオデータを与えるソースデバイス12を含む。特に、ソースデバイス12は、ビデオデータをコンピュータ可読媒体16を介して宛先デバイス14に与える。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビ、カメラ、表示デバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、ビデオストリーミングデバイス、などを含む、広い範囲のデバイスのうちのいずれかを含んでもよい。いくつかの場合には、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てもよい。10

#### 【 0 0 6 6 】

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てもよい。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てよい。一例では、コンピュータ可読媒体16は、ソースデバイス12がリアルタイムで宛先デバイス14に直接符号化ビデオデータを送信することを可能にする通信媒体を備えてよい。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に送信されてもよい。通信媒体は、無線周波数(RF)スペクトルなどの任意のワイヤレスもしくはワイヤード通信媒体、または、1つもしくは複数の物理的伝送線を備えてよい。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てもよい。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ってもよい任意の他の機器を含んでもよい。20

#### 【 0 0 6 7 】

いくつかの例では、符号化データは、出力インターフェース22から記憶デバイスに出力されてもよい。同様に、符号化データは、入力インターフェースによって記憶デバイスからアクセスされてもよい。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、揮発性もしくは不揮発性メモリ、または、符号化ビデオデータを記憶するための任意の他の適切なデジタル記憶媒体などの、様々な分散されたまたは局所的にアクセスされるデータ記憶媒体のうちのいずれかを含んでもよい。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ビデオを記憶してもよいファイルサーバまたは別の中間記憶デバイスに対応してもよい。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てもよい。ファイルサーバは、符号化ビデオデータを記憶し、宛先デバイス14にその符号化ビデオデータを送信することが可能な任意のタイプのサーバであってもよい。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、ネットワーク接続ストレージ(NAS:network attached storage)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介して符号化ビデオデータにアクセスしてもよい。これは、ファイルサーバに記憶された符号化ビデオデータにアクセスするのに適した、ワイヤレスチャネル(たとえば、Wi-Fi接続)、ワイヤード接続(たとえば、DSL、ケーブルモ뎀など)、または両方の組合せを含んでもよい。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ってもよい。30

#### 【 0 0 6 8 】

本開示の技術は、必ずしもワイヤレス用途または設定に限定されない。技法は、オーバージエアテレビ放送、ケーブルテレビ送信、衛星テレビ送信、HTTP(DASH)を介する動的適応ストリーミングなどのインターネットストリーミングビデオ送信、データ記憶媒体上で40

符号化されるデジタルビデオ、データ記憶媒体に記憶されたデジタルビデオの復号、または他の用途など、様々なマルチメディア用途のうち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てもよい。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、ビデオ放送、および/またはビデオ電話などの用途をサポートするために、一方向または双方方向ビデオ送信をサポートするように構成されてもよい。

#### 【0069】

図1の例では、ソースデバイス12は、ビデオソース18と、ビデオエンコーダ20と、出力インターフェース22とを含む。宛先デバイス14は、入力インターフェース28と、ビデオデコーダ30と、表示デバイス32とを含む。本開示によれば、ソースデバイス12のビデオエンコーダ20は、色空間変換プロセスを使用してビデオブロックをコーディングするための技法を適用するように構成されてよい。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配置を含んでよい。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18からビデオデータを受信してもよい。同様に、宛先デバイス14は、一体化された表示デバイスを含むのではなく、外部表示デバイスとインターフェースしてもよい。

#### 【0070】

図1の例示的なシステム10は一例にすぎない。色空間変換プロセスを使用してビデオブロックをコーディングするための技法は、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によって実行されてよい。一般に、本開示の技法はビデオ符号化デバイスによって実行されるが、技法はまた、一般に「コーデック」と呼ばれるビデオエンコーダ/デコーダによって実行されてもよい。その上、本開示の技法はまた、ビデオプロセッサによって実行されてもよい。ソースデバイス12および宛先デバイス14は、ソースデバイス12が宛先デバイス14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単なる例である。いくつかの例では、デバイス12、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、デバイス12、14は、実質的に対称的に動作することができる。したがって、システム10は、たとえばビデオストリーミング、ビデオプレイバック、ビデオブロードキャスティング、またはビデオテレフォニーのために、ビデオデバイス12と14との間で一方向または双方方向ビデオ送信をサポートし得る。

#### 【0071】

ソースデバイス12のビデオソース18は、ビデオカメラ、前に捕捉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、ビデオ捕捉デバイスを含むことがある。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、またはライブビデオ、アーカイブビデオ、およびコンピュータ生成ビデオの組合せとして、コンピュータグラフィックスベースのデータを生成することがある。いくつかの場合には、ビデオソース18がビデオカメラである場合、ソースデバイス12および宛先デバイス14は、いわゆるカメラ付き電話またはテレビ電話を形成してもよい。しかしながら、上述のように、本開示で説明する技術は、一般にビデオコーディングに適用可能であってもよく、ワイヤレス用途および/またはワイヤード用途に適用されてもよい。各場合において、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によって生成されたビデオは、ビデオエンコーダ20によって符号化されてもよい。次いで、符号化されたビデオ情報は、出力インターフェース22によってコンピュータ可読媒体16に出力されてもよい。

#### 【0072】

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ワイヤード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、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）を含むことがある。いくつかの例では、ネットワークサーバ（図示せず）は、ソースデバイス12から符号化されたビデオデータを受信し、符号化されたビデオデータを宛先デバイス14に、たとえばネットワーク送信を介して与えることがある。同様に、ディスクスタンピング設備など、媒体製造設

10

20

30

40

50

備のコンピューティングデバイスは、ソースデバイス12から符号化されたビデオデータを受信し、符号化されたビデオデータを含むディスクを生産することがある。そのために、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。

【0073】

宛先デバイス14の入力インターフェース28は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ビデオエンコーダ20によって規定される構文情報を含んでよく、ビデオエンコーダ20はまた、ブロックおよび他のコード化ユニット、たとえばGOPの特性ならびに/または処理を記述する構文要素を含むビデオデコーダ30によって使用される。表示デバイス32は、復号されたビデオデータをユーザに表示し、陰極線管(CRT)、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表示デバイスなどの、様々な表示デバイスのうちのいずれかを備えてもよい。

【0074】

ビデオエンコーダ20およびビデオデコーダ30は、現在開発中の高効率ビデオコーディング(HEVC)規格などのビデオコーディング規格に従って動作してもよく、HEVC試験モデル(HM)に準拠してもよい。代替的には、ビデオエンコーダ20およびビデオデコーダ30は、代替的にMPEG-4、パート10、アドバンストビデオコーディング(AVC)と呼ばれるITU-T H.264規格、またはそのような規格の拡張など、他の所有権または工業規格に従って動作してもよい。本開示の技術は、しかしながら、任意の特定のコーディング規格に限定されない。ビデオコーディング規格の他の例は、MPEG-2とITU-T H.263とを含む。図1には示していないが、いくつかの態様では、ビデオエンコーダ20およびビデオデコーダ30は、各々、オーディオエンコーダおよびデコーダと一体化されてもよく、共通データストリームまたは別々のデータストリームにおけるオーディオとビデオの両方の符号化を処理するために、適切なMUX-DEMUXユニットまたは他のハードウェアおよびソフトウェアを含んでもよい。該当する場合、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:user datagram protocol)などの他のプロトコルに準拠してもよい。

【0075】

ITU-T H.264/MPEG-4(AVC)規格は、共同ビデオ部会(JVT)として知られている共同パートナーシップの製品として、ITU-Tビデオコーディング専門家グループ(VCEG)とISO/IEC動画専門家グループ(MPEG)とによって策定された。いくつかの態様では、本開示で説明する技法は、一般にH.264規格に準拠するデバイスに適用されてよい。H.264規格は、2005年3月の日付でITU-T研究委員会によって、ITU-T勧告H.264、ジェネリックオーディオビジュアルサービスのためのアドバンストビデオコーディングに記載されており、H.264規格は、本明細書では、H.264規格もしくはH.264仕様、またはH.264/AVC規格もしくは仕様と呼ばれることがある。共同ビデオ部会(JVT)は、H.264/MPEG-4 AVCへの拡張の作業を継続している。

【0076】

ビデオエンコーダ20およびビデオデコーダ30は、各々、1つもしく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)、ディスクリートロジック、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のうちのいずれかとして実装されてもよい。技術が部分的にソフトウェアにおいて実装されているとき、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のための命令を記憶してもよく、本開示の技術を実行するために、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してもよい。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダ内に含まれてもよく、そのいずれかは、それぞれのデバイス内に、組み合わされたエンコーダ/デコーダ(コーデック)の一部として組み込まれてもよい。

10

20

30

40

50

## 【0077】

JCT-VCは、HEVC規格の展開について作業中である。HEVC規格化の努力は、HEVCテストモデル(HM)と呼ばれるビデオコーダの進化モデルに基づいている。HMは、たとえば、ITU-T H.264/AVCによる既存のデバイスと比較してビデオコーダのいくつかの追加の能力を想定する。たとえば、H.264は、9つのイントラ予測符号化モードを提供する一方で、HMは、33ものイントラ予測符号化モードを提供してもよい。

## 【0078】

一般に、HMの作業中のモデルは、ビデオフレームまたはピクチャが、ルーマサンプルとクロマサンプルの両方を含む一連のツリーブロックまたは最大コーディング単位(LCU)に分割されてもよいことを記載している。ビットストリーム内の構文データは、ピクセル数に関して最大のコーディングユニットであるLCUに対するサイズを規定してよい。ライスは、コーディング順にいくつかの連続するツリーブロックを含む。ビデオフレームまたはピクチャは、1つまたは複数のライスに区分されてもよい。各ツリーブロックは、4分木に従ってコーディング単位(CU)に分割されてもよい。一般に、4分木データ構造は、CUごとに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つのサブCUに分割される場合、CUに対するノードは4つのリーフノードを含み、それらの各々はサブCUのうちの1つに対応する。

10

## 【0079】

4分木データ構造の各ノードは、対応するCUに対する構文データを与えることがある。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むことがある。CUに対する構文要素は、再帰的に規定されてよく、CUがサブCUに分割されるかどうかに応じて決められてよい。CUがそれ以上分割されない場合、それは、リーフCUと呼ばれる。本開示では、リーフCUのうちの4つのサブCUは、元のリーフCUの明示的な分割が存在しない場合でも、同様にリーフCUと呼ばれることになる。たとえば、 $16 \times 16$  サイズにおけるCUがそれ以上分割されない場合、4つの $8 \times 8$  サブCUは、 $16 \times 1$  6CUが決して分割されなくても、同様にリーフCUと呼ばれることになる。

20

## 【0080】

CUは、サイズの区別を持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)に分割されてもよく、各子ノードは、今度は親ノードになり、別の4つの子ノードに分割されてもよい。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。コード化ビットストリームと関連付けられた構文データは、最大CU深さと呼ばれる、ツリーブロックが分割され得る最大回数を規定してもよく、また、コーディングノードの最小サイズを規定してもよい。したがって、ビットストリームはまた、最小コーディングユニット(SCU)を規定してもよい。本開示は、HEVCのコンテキストにおけるCU、PU、もしくはTU、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、H.264/AVCにおけるマクロブロックおよびそれらのサブブロック)のうちのいずれかを指すために、「ブロック」という用語を使用する。

30

## 【0081】

CUは、コーディングノードと、コーディングノードに関連付けられた予測単位(PU)および変換単位(TU)とを含む。CUのサイズは、コーディングノードのサイズに対応し、形状において正方形でなければならない。CUのサイズは、 $8 \times 8$ ピクセルから、最大 $64 \times 64$ ピクセルまたはそれ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と1つまたは複数のTUとを含んでもよい。CUに関連付けられた構文データは、たとえば、1つまたは複数のPUへのCUの区分を記述してもよい。分割モードは、CUがスキップもしく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ってもよい。PUは、形状において非正方形であるように区分されてもよい。CUに関連付けられた構文データはまた、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、形状において正方形または非正方形(たとえば、長方形)であることができる。

40

50

## 【0082】

HEVC規格は、TUに従う変換を可能にし、それは、異なるCUのために異なってもよい。TUは、典型的には、区分されたLCUのために定義された所与の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、必ずしもそうでないことがある。TUは、典型的には、PUと同じサイズであり、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: residual quad tree)として知られる4分木構造を使用して、より小さい単位に細分されてもよい。RQTのリーフノードは、変換単位(TU)と呼ばれることがある。TUに関連付けられたピクセル差分値は、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。

## 【0083】

リーフCUは、1つまたは複数の予測単位(PU)を含んでよい。一般に、PUは、対応するCUの全部または一部に対応する空間エリアを表し、PUに対する参照サンプルを取り出すためのデータを含んでよい。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たとき、PUに対するデータは、PUに対応するTUに対するイントラ予測モードを記述するデータを含み得る残差4分木(RQT)の中に含まれてよい。別の例として、PUがインターモード符号化されたとき、PUは、PUのための1つまたは複数の動きベクトルを定義するデータを含んでもよい。PUのための動きベクトルを定義するデータは、たとえば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度(たとえば、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

## 【0084】

1つまたは複数のPUを有するリーフCUはまた、1つまたは複数の変換単位(TU)を含んでもよい。変換単位は、上記のように、RQT(TU4分木構造とも呼ばれる)を使用して指定されてよい。たとえば、分割フラグは、リーフCUが4つの変換単位に分割されるかどうかを示してよい。次いで、各変換単位は、さらなるサブTUにさらに分割されてよい。TUがそれ以上分割されないとき、それは、リーフTUと呼ばれることがある。一般に、イントラコーディングに対して、1つのリーフ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を共有する。すなわち、同じイントラ予測モードは、一般に、リーフCUのすべてのTUに対する予測値を計算するために適用される。イントラコーディングに対して、ビデオエンコーダは、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得る。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よりも大きくても小さくてもよい。イントラコーディングに対して、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

## 【0085】

その上、リーフCUのTUはまた、残差4分木(RQT)と呼ばれるそれぞれの4分木データ構造と関連付けられてよい。すなわち、リーフCUがTUに、どのように区分されているかを示す4分木を、リーフCUは含み得る。TUの4分木のルートノードは、概してリーフCUに対応する一方で、CUの4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。分割されないRQTのTUは、リーフTUと呼ばれる。一般に、本開示は、別段に記載されていない限り、リーフCUを指すために用語CUを、リーフTUを指すために用語TUを使用する。

30

## 【0086】

ビデオシーケンスは、典型的には、ビデオフレームまたはピクチャを含む。ピクチャのグループ(GOP:group of pictures)は、一般に、一連の1つまたは複数のビデオピクチャを備える。GOPは、GOPのヘッダ、1つまたは複数のピクチャのヘッダ、または他の場所に、GOP内に含まれるピクチャの数を記述する構文データを含んでもよい。ピクチャの各スライスは、それぞれのスライスのための符号化モードを記述するスライス構文データを含んでもよい。ビデオエンコーダ20は、典型的には、ビデオデータを符号化するために、個々のビデオスライス内のビデオブロックに対して動作する。ビデオブロックは、CU内のコーデ

40

50

イングノードに対応してもよい。ビデオブロックは、固定されたまたは変化するサイズを有してもよく、指定されたコーディング規格に従ってサイズが異なってもよい。

#### 【 0 0 8 7 】

一例として、HMは、様々なPUサイズにおける予測をサポートする。特定のCUのサイズが $2N \times 2N$ であると仮定すると、HMは、 $2N \times 2N$ または $N \times N$ のPUサイズにおけるイントラ予測と、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、または $N \times N$ の対称PUサイズにおけるインター予測とをサポートする。HMはまた、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ のPUサイズにおけるインター予測のための非対称区分をサポートする。非対称分割では、CUの1つの方向は、区分されず、他の方向は、25%および75%に区分される。25%区分に対応するCUの部分は、「上」、「下」、「左」、または「右」の指示が続く「n」によって示される。したがって、たとえば、「 $2N \times nU$ 」は、上に $2N \times 0.5N$ のPUおよび下に $2N \times 1.5N$ のPUで水平に区分された $2N \times 2N$ のCUを指す。  
10

#### 【 0 0 8 8 】

本開示では、「 $N \times N$ 」および「N掛けるN」は、垂直方向および水平方向の寸法に関するビデオブロックのピクセルの寸法、たとえば、 $16 \times 16$ ピクセル、または16掛ける16ピクセルを指すために、交換可能に使用されてもよい。一般に、 $16 \times 16$ ブロックは、垂直方向に $16$ ピクセル( $y=16$ )と水平方向に $16$ ピクセル( $x=16$ )とを有することになる。同様に、 $N \times N$ ブロックは、一般に、垂直方向に $N$ ピクセルと水平方向に $N$ ピクセルとを有し、ここで $N$ は、負ではない整数値を表す。ブロック内のピクセルは、行および列に配置されてもよい。さらに、ブロックは、必ずしも水平方向で垂直方向と同じ数のピクセルを有する必要はない。  
20 たとえば、ブロックは、 $N \times M$ ピクセルを備えてもよく、ここで $M$ は、 $N$ と必ずしも等しくない。

#### 【 0 0 8 9 】

CUのPUを使用するイントラ予測またはインター予測コーディングに続いて、ビデオエンコーダ20は、CUのTUのための残差データを計算してもよい。PUは、空間領域(ピクセル領域とも呼ばれる)における予測ピクセルデータを生成する方法またはモードを記述する構文データを備えてもよく、TUは、変換、たとえば離散コサイン変換(DCT)、整数変換、ウェーブレット変換、または概念的に同様の変換を残差ビデオデータに適用することに続いて、変換領域における係数を備えてもよい。残差データは、符号化されていないピクチャのピクセルと、PUに対応する予測値との間のピクセル差に対応してもよい。ビデオエンコーダ20は、CUのための残差データを含むTUを形成し、次いで、CUのための変換係数を生成するためにTUを変換してもよい。  
30

#### 【 0 0 9 0 】

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20は、変換係数の量子化を実行してもよい。量子化は、一般に、係数を表すために使用されるデータの量をできる限り低減するために変換係数が量子化され、さらなる圧縮を提供するプロセスを指す。量子化プロセスは、係数の一部またはすべてに関連付けられたビット深度を低減してもよい。たとえば、 $n$ ビット値は、量子化の間に $m$ ビット値に切り捨てられてもよく、ここで $n$ は、 $m$ よりも大きい。

#### 【 0 0 9 1 】

量子化に続いて、ビデオエンコーダは、変換係数を走査してよく、量子化された変換係数を含む2次元行列から1次元ベクトルを作成する。走査は、より高いエネルギー(それゆえより低い周波数)の係数をアレイの前方に置き、より低いエネルギー(それゆえより高い周波数)の係数をアレイの後方に置くように設計されてよい。いくつかの例では、ビデオエンコーダ20は、エントロピー符号化されることが可能なシリアル化ベクトルを生成するために、量子化された変換係数を走査するために事前定義された走査順を利用してもよい。他の例では、ビデオエンコーダ20は、適応走査を実行してもよい。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20は、たとえば、コンテキスト適応可変長コーディング(CAVLC:context adaptive variable length coding)、コンテキスト適応2値算術コーディング(CABAC:context adaptive binary arithmetic)  
40 50

coding)、構文ベースコンテキスト適応2値算術コーディング(SBAC:syntax-based context-adaptive binary arithmetic coding)、確率インターバル区分エントロピー(PIPE:Probability Interval Partitioning Entropy)コーディング、または別のエントロピー符号化方法に従って、1次元ベクトルをエントロピー符号化してもよい。ビデオエンコーダ20はまた、ビデオデータを復号する上でビデオデコーダ30によって使用するための符号化ビデオデータに関連付けられた構文要素をエントロピー符号化してもよい。

#### 【 0 0 9 2 】

CABACを実行するために、ビデオエンコーダ20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てもよい。コンテキストは、たとえば、シンボルの隣接値が非ゼロであるかどうかに関連してもよい。CAVLCを実行するために、ビデオエンコーダ20は、送信されるべきシンボルのための可変長コードを選択してもよい。VLCにおけるコードワードは、比較的より短いコードがより可能性が高いシンボルに対応し、より長いコードがより可能性が低いシンボルに対応するように構成されてもよい。このように、VLCの使用は、たとえば、送信されるべき各シンボルに等長コードワードを使用して、ビットの節約を達成してもよい。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いてもよい。

10

#### 【 0 0 9 3 】

本開示の技法によれば、ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、ビデオデータを符号化するために色空間変換を使用するかどうかを決定し得る。ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオコーダは、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化し得、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。ビデオコーダは、さらに、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る。ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、ビデオコーダは、第1のQPの第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化し、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化し得る。ビデオデータが色空間変換を使用して符号化されなかったものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに第1のQPおよび第2のQPを使用してビデオデータを逆量子化し得る。

20

#### 【 0 0 9 4 】

第1のQPおよび第2のQPは、色空間変換によらずにコーディングされたブロックと同じ導出方法に従って、第1および第2の色成分に対応して導出されたQPであり得る。たとえば、第1のQPおよび第2のQPに対するQPの値は、cu\_qp\_delta\_absなどのシグナリングされた構文要素から導出され得る。次いで、第1のオフセットおよび第2のオフセットは、それぞれ、ビデオデータが色空間変換を使用して符号化されたとき、第1のQPおよび第2のQPを取得するために、導出されたQPに加算されてよい。たとえば、QP(たとえば、 $QP_i$ )に対して、それぞれのオフセット(たとえば、 $offset_i$ )は、QPに加算されてよく、得られたQPの値を、元のQPプラスそれぞれのオフセットに等しくさせる(たとえば、 $QP_{final} = QP_i + offset_i$ )。

30

#### 【 0 0 9 5 】

ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、図1～図9に関して説明した技法のうちのいずれかを実行してよい。異なる色成分を量子化および逆量子化するために異なる量子化パラメータを使用することによって、ビデオデコーダは、色空間変換が使用されるときにビデオデータをより効率的に復号することができ、それによって、システムの総合的コーディング効率を向上させ、エネルギー消費を低減させる。

40

#### 【 0 0 9 6 】

多様な構文要素が、本開示の技法と整合して使用されることがある。これらの構文要素は、

50

【0097】

【表1A】

seq_parameter_set_rbsp( ) {	記述子
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level( sps_max_sub_layers_minus1 )	
...	10
<b>vui_parameters_present_flag</b>	u(1)
if( vui_parameters_present_flag )	
vui_parameters( )	
<b>sps_extension_present_flag</b>	u(1)
if( sps_extension_present_flag ) {	
for( i = 0; i < 1; i++ )	
<b>sps_extension_flag[ i ]</b>	u(1)
<b>sps_extension_7bits</b>	u(7)
if( sps_extension_flag[ 0 ] ) {	20
<b>transform_skip_rotation_enabled_flag</b>	u(1)
<b>transform_skip_context_enabled_flag</b>	u(1)
<b>intra_block_copy_enabled_flag</b>	u(1)
<b>implicit_rdpcm_enabled_flag</b>	u(1)
<b>explicit_rdpcm_enabled_flag</b>	u(1)
<b>extended_precision_processing_flag</b>	u(1)
<b>intra_smoothing_disabled_flag</b>	u(1)
<b>high_precision_offsets_enabled_flag</b>	u(1)
<b>fast_rice_adaptation_enabled_flag</b>	u(1)
<b>cabac_bypass_alignment_enabled_flag</b>	u(1)
<b>color_transform_enabled_flag (new)</b>	u(1)
}	
if( sps_extension_7bits )	
while( more_rbsp_data( ) )	
<b>sps_extension_data_flag</b>	u(1)
}	40
rbsp_trailing_bits( )	
}	

【0098】

【表 1 B】

coding_unit( x0, y0, log2CbSize ) {	記述子
if( transquant_bypass_enabled_flag )	
<b>cu_transquant_bypass_flag</b>	ae(v)
if( slice_type != I )	
<b>cu_skip_flag[ x0 ][ y0 ]</b>	ae(v)
nCbS = ( 1 << log2CbSize )	
if( cu_skip_flag[ x0 ][ y0 ] )	
prediction_unit( x0, y0, nCbS, nCbS )	
else {	
if( intra_block_copy_enabled_flag )	
<b>intra_bc_flag[ x0 ][ y0 ]</b>	ae(v)
if( slice_type != I && !intra_bc_flag[ x0 ][ y0 ] )	
<b>pred_mode_flag</b>	ae(v)
if( CuPredMode[ x0 ][ y0 ] != MODE_INTRA    intra_bc_flag[ x0 ][ y0 ] )	
log2CbSize == MinCbLog2SizeY )	
<b>part_mode</b>	ae(v)
if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {	
if( PartMode == PART_2Nx2N && pcm_enabled_flag &&	
!intra_bc_flag[ x0 ][ y0 ] &&	
log2CbSize >= Log2MinIpcmCbSizeY &&	
log2CbSize <= Log2MaxIpcmCbSizeY )	
<b>pcm_flag[ x0 ][ y0 ]</b>	ae(v)
if( pcm_flag[ x0 ][ y0 ] ) {	
while( !byte_aligned() )	
<b>pcm_alignment_zero_bit</b>	f(1)
pcm_sample( x0, y0, log2CbSize )	
} else if( intra_bc_flag[ x0 ][ y0 ] ) {	
mvd_coding( x0, y0, 2 )	
if( PartMode == PART_2NxN )	
mvd_coding( x0, y0 + ( nCbS / 2 ), 2 )	
else if( PartMode == PART_Nx2N )	
mvd_coding( x0 + ( nCbS / 2 ), y0, 2 )	
else if( PartMode == PART_NxN ) {	
mvd_coding( x0 + ( nCbS / 2 ), y0, 2 )	
mvd_coding( x0, y0 + ( nCbS / 2 ), 2 )	
mvd_coding( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), 2 )	
}	
} else {	

【0099】

【表 1 C】

pbOffset = ( PartMode == PART_NxN ) ? ( nCbS / 2 ) : nCbS		
for( j = 0; j < nCbS; j = j + pbOffset )		
for( i = 0; i < nCbS; i = i + pbOffset )		
<b>prev_intra_luma_pred_flag</b> [ x0 + i ][ y0 + j ]	ae(v)	
for( j = 0; j < nCbS; j = j + pbOffset )		
for( i = 0; i < nCbS; i = i + pbOffset )		
if( <b>prev_intra_luma_pred_flag</b> [ x0 + i ][ y0 + j ] )		
<b>mpm_idx</b> [ x0 + i ][ y0 + j ]	ae(v)	
else		
<b>rem_intra_luma_pred_mode</b> [ x0 + i ][ y0 + j ]	ae(v)	
if( ChromaArrayType == 3 )		
for( j = 0; j < nCbS; j = j + pbOffset )		
for( i = 0; i < nCbS; i = i + pbOffset )		
<b>intra_chroma_pred_mode</b> [ x0 + i ][ y0 + j ]	ae(v)	
else if( ChromaArrayType != 0 )		
<b>intra_chroma_pred_mode</b> [ x0 ][ y0 ]	ae(v)	
}		
}		
...		
}		
if( !pcm_flag[ x0 ][ y0 ] ) {		
if( CuPredMode[ x0 ][ y0 ] != MODE_INTRA &&		
!( PartMode == PART_2Nx2N && merge_flag[ x0 ][ y0 ] )		
( CuPredMode[ x0 ][ y0 ] == MODE_INTRA &&		
intra_bc_flag[ x0 ][ y0 ] )		
<b>rqt_root_cbf</b>	ae(v)	
if( rqt_root_cbf ) {		
if( color_transform_enabled_flag ) (new)		
<b>color_transform_flag</b> [ x0 ][ y0 ] (new)	ae(v)	
MaxTrafoDepth = ( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ?		
( max_transform_hierarchy_depth_intra +		
IntraSplitFlag ) :		
max_transform_hierarchy_depth_inter )		
transform_tree( x0, y0, x0, y0, log2CbSize, 0, 0 )		
}		
}		
}		
}		

Table 1

【0100】

を含み得る。

【0101】

構文要素**color\_transform\_enabled\_flag**が1に等しいとき、ループ内色空間変換プロセスは、復号プロセスの中で起動されてよい。構文要素**color\_transform\_enabled\_flag**が0に等しいとき、ループ内色空間変換プロセスは適用されなくてよい。構文要素が存在しな

いとき、構文要素color\_transform\_enabled\_flagの値は、0に等しいものと推測されてよい。

#### 【0102】

構文要素color\_transform\_flag[x0][y0]が1に等しいとき、現在のコーディングユニットは、色空間変換によってコーディングされてよい。構文要素color\_transform\_flag[x0][y0]が0に等しいとき、現在のコーディングユニットは、色空間変換によらないでコーディングされる。構文要素が存在しないとき、構文要素color\_transform\_flagの値は、0に等しいものと推測されてよい。アレイインデックスx0およびy0は、ピクチャの左上のルーマサンプルに対する考慮されるコーディングブロックの左上のルーマサンプルのロケーション(x0,y0)を指定し得る。

10

#### 【0103】

いくつかの例では、ビデオエンコーダ20またはビデオデコーダ30などのビデオコーダは、コーディングユニットに対して色空間変換を使用するかどうかを決定し得る。ビデオコーダは、色空間変換の使用を示すためにコーディングユニットの構文要素の値を設定し得る。ビデオコーダは、色空間変換行列をコーディングユニットの残差ブロックに適用し得る。ビデオコーダは、コーディングユニットの構文要素を復号し得、構文要素は、コーディングユニットが色空間変換を使用して符号化されたかどうかを示す。ビデオコーダは、コーディングユニットが色空間変換を使用して符号化されたことを構文要素の値が示すかどうかを決定し得る。ビデオコーダは、コーディングユニットが色空間変換を使用してコーディングされたことを構文要素が示すものと決定することに応答して、色空間逆変換行列をコーディングユニットの残差ブロックに適用し得る。

20

#### 【0104】

一例では、イントラコード化CUに対して、フラグが直接シグナリングされることがある。代替として、イントラコード化CUに対して、CUがI-PCMモードでないときだけ、フラグがシグナリングされることがある。別の例では、インターコード化CUおよび/またはイントラBCコード化CUに対して、現在のCU内に非ゼロ係数が存在するとき、すなわちrqt\_root\_cbfが1に等しいときだけ、フラグがシグナリングされてよい。別の例では、フラグは、CUがパレットモードでコーディングされるときはシグナリングされない。すべての上記の例では、復号された色変換フラグが0に等しいか、またはフラグが1つのCUに対するピットストリーム内に存在しないとき、色空間変換プロセスはスキップされてよい。

30

#### 【0105】

さらに、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)/スライスヘッダのいずれかにおける1つのフラグが、色空間変換の使用を制御するためにシグナリングされ得る。SPS/PPS/スライスヘッダ内のフラグが0に等しいとき、CUレベルフラグのシグナリングは、対応するシーケンス/ピクチャ/スライスそれぞれの中のすべてのCU内でスキップされてよい。代替として、色空間変換は、PUレベルまたはTUレベルで実行されてよく、PU/TUの各々は、色空間変換の使用を示すために1つのフラグを有する。

#### 【0106】

一例では、修正されたYCoCg変換は、符号化および復号プロセスにおいて適用されてよい。修正されたYCoCg変換は、次のように定義され得る。

40

順方向：

#### 【0107】

#### 【数8】

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

#### 【0108】

逆方向: temp=Y-Cg,

50

G=(Y+Cg+offset)>>2,  
 B=(temp-Co+offset)>>2,

R=(temp+Co+offset)>>2、ここでオフセットは2に等しい。

【 0 1 0 9 】

代替として、修正されたYCoCg変換は、次のように定義され得る。

順方向：

【 0 1 1 0 】

【 数 9 】

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

10

【 0 1 1 1 】

逆方向：temp=Y-Cg,

G=(Y+Cg+offset)>>1,

B=(temp-Co+offset)>>1,

R=(temp+Co+offset)>>1、ここでオフセットは1または0に等しい。

【 0 1 1 2 】

色変換によってコーディングされたCUまたはブロックによって使用される対応する量子化パラメータは、(dQP+2)に等しいものと推測されてよく、一方、色変換によらないCUまたはブロックによって使用される対応する量子化パラメータは、dQPに等しいものと推測されてよい。同時に、ピット深度は、量子化プロセスと変換プロセスの両方において、1だけさらに増加されてよい。

20

【 0 1 1 3 】

代替として、2ステップの順/逆変換は、元の不变のYCoCgに正規化処理を加えたものによって適用される。順変換に対して、元のYCoCg順変換が最初に適用される。次いで、各成分i、すなわちY、CoおよびCgに対して、成分は(i\*forward\_factor+offset)>>BIT\_PRECISIONにリセットされ、ここでBIT\_PRECISIONは符号なし整数であり、forward\_factorはBIT\_PRECISIONに応じて決まる。一例では、forward\_factorは、1/6\*(1<<BITS\_TRANS)+0.5に等しく、BIT\_PRECISIONは15に等しく、offsetは、(1<<(BIT\_PRECISION-1))に等しい。

30

【 0 1 1 4 】

逆変換に対して、元のYCoCg逆変換が最初に適用される。次いで、各成分i、すなわちY、CoおよびCgに対して、成分は(i\*backward\_factor+offset)>>BIT\_PRECISIONにリセットされ、ここでBIT\_PRECISIONは順変換において使用されるものと同じであり、backward\_factorはBIT\_PRECISIONに応じて決まる。一例では、backward\_factorは、6/4\*(1<<BITS\_TRANS)+0.5に等しく、offsetは、(1<<(BIT\_PRECISION-1))に等しい。

【 0 1 1 5 】

代替として/さらに、CUレベルフラグは、3つの色成分(すなわち、cbf\_luma、cbf\_cbおよびcbf\_cr)内のコード化ブロックフラグのうちの少なくとも1つが1に等しいときだけ、シグナリングされてよい。代替として、色空間変換によるCUまたはブロック内の残差の範囲が、色空間変換によらないCU/ブロック内の残差の範囲と同じであることを確実にするために、色空間変換の後の修正された残差は、さらに修正されてよい。一例では、クリップ動作が適用される。

40

【 0 1 1 6 】

色空間変換行列は、再構成されたピクセルとは無関係であってよい。代わりに、色空間変換行列は、非可逆コーディングモードまたは可逆コーディングモードに応じて決まることがある。一例では、CUが非可逆モードでコーディングされるとき、YCoCgが適用されてよく、一方、CUが可逆モードでコーディングされるとき、YCoCg-Rが適用される。その上、YCoCg-Rが適用されるとき、Co成分およびCg成分のピット深度が、1だけ増加されてよい

50

。別の例では、変換行列は、イントラモード、インター/イントラBCモードに応じて決まることがある。この場合、各モードに対する所定の行列は、エンコーダとデコーダの両方において指定されてよく、CUレベルフラグが1であるとき、コーディングモードに従って、同じ行列がエンコーダとデコーダの両方において使用されてよい。さらに、CUレベルフラグは、3つの色成分(すなわち、cbf\_luma、cbf\_cbおよびcbf\_cr)内のコード化ブロックフラグのうちの少なくとも1つが1に等しいときだけ、シグナリングされてよい。

#### 【0117】

本開示の技法はまた、たとえばフレーム内レベルにおける、変換行列導出およびシグナリング方法を含むことがある。一例では、各フレームに対して、変換行列は、元の画像内の色成分にわたって統計値に基づいて導出されてよい。変換行列はビットストリームにコーディングされ、デコーダに直接送信され得る。代替として、連続してコーディングされたフレームの間の変換行列予測が適用されてよい。たとえば、行列が前のフレーム内で使用された行列と同じであるかどうかを示すために、1つのフラグが送信されてよい。代替として、変換行列は、元のフレームのセットに基づいて導出され、ビットストリーム内でシグナリングされてよい。

10

#### 【0118】

別の例では、変換行列のセットは、PPSなどの高レベル構文内で導出され、シグナリングされる。一例では、画像は、特性に基づいて複数の領域に分類され、各領域に対して変換行列が導出される。導出された行列は、高レベル構文内でコーディングされ、シグナリングされ得る。選択された変換行列のインデックスは、CU/TU/PUレベルでさらにシグナリングされる。代替として、選択された変換行列のインデックスは、スライスヘッダ内で、または各タイル領域に対してシグナリングされる。代替として、変換行列は、スライスヘッダ内で、または各タイル領域に対して導出されてよい。

20

#### 【0119】

ルーマ成分およびクロマ成分のビット深度が異なるとき、すべての色成分のビット深度が、最初に、同じになるように修正され、次いで色変換が後で適用されることがある。代替として、ビットストリームは、ルーマ成分およびクロマ成分のビット深度がコード化ビットストリーム内で異なるとき、色変換は無効(すなわち、color\_transform\_enabled\_flagが0に等しい)にされるべきであるという制約に従うことがある。

30

#### 【0120】

異なるQPが、色変換によってまたはよらないでコーディングされたCU/ブロックに適用されてよく、ならびに/あるいは量子化プロセスおよび変換プロセスにおいて使用されるビット深度が、そのような正規化されない色変換のせいで修正されてよい。一例では、上記の修正されたYCoCg変換が使用されるとき、以下のステップ(すなわち、フィルタ処理をデブロックする前の、スケーリング、変換およびアレイ構成プロセス)が、color\_transform\_flagに基づいてさらに修正される。

#### 【0121】

一例では、上記で説明した、修正されたYCoCg変換が使用されるとき、それに応じて、対応する量子化パラメータ導出プロセスおよび逆変換プロセスが修正される。このプロセスへの入力には、現在のピクチャの左上のルーマサンプルに対する現在のルーマコーディングブロックの左上のサンプルを指定するルーマロケーション(xCb,yCb)が含まれてよい。このプロセスでは、変数QpY、ルーマ量子化パラメータQp'Y、およびクロマ量子化パラメータQp'CbおよびQp'Crが導出されてよい。ルーマロケーション(xQg,yQg)は、現在のピクチャの左上のルーマサンプルに対する現在の量子化グループの左上のルーマサンプルを指定してよい。水平位置xQgおよび垂直位置yQgは、それぞれ、xCb-(xCb&((1<<Log2MinCuQpDeltaSize)-1))およびyCb-(yCb&((1<<Log2MinCuQpDeltaSize)-1))に等しく設定されてよい。量子化グループのルーマサイズ、Log2MinCuQpDeltaSizeは、同じqPY\_PREDを共有するコーディングツリーブロック内部の最小エリアのルーマサイズを決定してよい。

40

#### 【0122】

変数QpYは、以下の

50

$QpY=((qPY\_PRED+CuQpDeltaVal+52+2*QpBdOffsetY)%(52+QpBdOffsetY))-QpBdOffsetY$   
のように導出され得る。

## 【0 1 2 3】

ルーマ量子化パラメータ  $Qp'Y$  は、以下の  
 $Qp'Y=QpY+QpBdOffsetY$   
のように導出され得る。

## 【0 1 2 4】

構文要素 ChromaArrayType が 0 に等しいとき、以下が適用されてよい。  
 変数  $qPiCb$  および  $qPiCr$  は、以下の

$qPiCb=Clip3(-QpBdOffsetC, 57, QpY+pps_cb_qp_offset+slice_cb_qp_offset+CuQpAdjValCb)$  10  
 $)$

$qPiCr=Clip3(-QpBdOffsetC, 57, QpY+pps_cr_qp_offset+slice_cr_qp_offset+CuQpAdjValCr)$   
 $)$

のように導出され得る。

## 【0 1 2 5】

構文要素 ChromaArrayType が 1 に等しい場合、変数  $qPCb$  および  $qPCr$  は、それぞれ  $qPiCb$  および  $qPiCr$  に等しいインデックス  $qPi$  に基づいて下表に指定されるように、 $QpC$  の値に等しく設定されてよい。そうでない場合、変数  $qPCb$  および  $qPCr$  は、それぞれ  $qPiCb$  および  $qPiCr$  に等しいインデックス  $qPi$  に基づいて  $Min(qPi, 51)$  に等しく設定される。

## 【0 1 2 6】

## 【表 2】

$qPi$	$< 30$	3	3	3	3	3	3	3	3	3	4	4	4	4	$> 43$
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	
$QpC$	$= qPi$	2	3	3	3	3	3	3	3	3	3	3	3	3	$= qPi - 6$
	9	0	1	2	3	3	4	4	5	5	6	6	7	7	

Table 2

## 【0 1 2 7】

$Cb$  成分および  $Cr$  成分に対するクロマ量子化パラメータ、 $Qp'Cb$  および  $Qp'Cr$  は、以下の  
 $Qp'Cb=qPCb+QpBdOffsetC$   
 $Qp'Cr=qPCr+QpBdOffsetC$   
のように導出され得る。

## 【0 1 2 8】

$color\_transform\_flag$  が 1 に等しいとき、以下が適用されてよい。

$Qp'Y=(Qp'Y+8)$   
 $ChromaArrayType 0$  のとき、 $Qp'Cb=(Qp'cb+8)$  および  $Qp'Cr=(Qp'cr+8)$   
 $BitDepthY=(BitDepthY+2)$   
 $BitDepthC=(BitDepthC+2)$

## 【0 1 2 9】

このプロセスへの入力は、現在のピクチャの左上のルーマサンプルに対する現在のルーマ変換ブロックの左上のサンプルを指定するルーマロケーション  $(xTbY, yTbY)$  と、現在の変換ブロックのサイズを指定する変数  $nTbS$  と、現在のブロックの色成分を指定する変数  $cl$   $dx$  と、量子化パラメータを指定する変数  $qP$  とを含んでよい。このプロセスの出力は、要素  $d[x][y]$  でスケーリングされた変換係数の  $(nTbS) \times (nTbS)$  アレイ  $d$  を含んでよい。

## 【0 1 3 0】

変数  $log2TransformRange$ 、 $bdShift$ 、 $coeffMin$  および  $coeffMax$  は、次のように導出されてよい。

$color\_transform\_flag=1$  のとき、以下が適用される。

20

30

40

50

```

CoeffCTMinY=-(1<<(extended_precision_processing_flag? Max(15,BitDepthY+6):15))
CoeffCTMinC=-(1<<(extended_precision_processing_flag? Max(15,BitDepthC+6):15))
CoeffCTMaxY=(1<<(extended_precision_processing_flag? Max(15,BitDepthY+6):15))-1
CoeffCTMaxC=(1<<(extended_precision_processing_flag? Max(15,BitDepthC+6):15))-1
cIdx=0の場合、
log2TransformRange=extended_precision_processing_flag? Max(15,BitDepthY+6):15
bdShift=BitDepthY+Log2(nTbS)+10-log2TransformRange
coeffMin=(color_transform_flag? CoeffCTMinY: CoeffMinY)
coeffMax=(color_transform_flag? CoeffCTMaxY: CoeffMaxY)

```

そうでない場合、

```

log2TransformRange=extended_precision_processing_flag? Max(15,BitDepthC+6):15
bdShift=BitDepthC+Log2(nTbS)+10-log2TransformRange
coeffMin=(color_transform_flag? CoeffCTMinC: CoeffMinC)
coeffMax=(color_transform_flag? CoeffCTMaxC: CoeffMaxY)

```

### 【 0 1 3 1 】

表levelScale[ ]は、levelScale[k]={40、45、51、57、64、72}、k=0.5として指定され  
てよい。

### 【 0 1 3 2 】

スケーリングされた変換係数d[x][y]、x=0...nTbS-1、y=0...nTbS-1の導出に対して、  
以下が適用されてよい。

スケーリング因子m[x][y]は、次のように導出され得る。

以下の条件のうちの1つまたは複数が真であるならば、m[x][y]は16に等しく設定される。

```

scaling_list_enabled_flag=0
transform_skip_flag[xTbY][yTbY]=1およびnTbS>4

```

そうでない場合、以下が適用される。

$m[x][y]=ScalingFactor[szId][matrixId][x][y]$  (8-283)

### 【 0 1 3 3 】

構文要素szIdが、(nTbS) × (nTbS)に等しい量子化行列のサイズに対して指定され、ma  
trixIdが、それぞれ、szId、CuPredMode[xTbY][yTbY]、およびcIdxに対してTable 7-4  
内に指定されるとき、スケーリングされた変換係数d[x][y]は、次のように導出され得る  
。

```

d[x][y]=Clip3(coeffMin,coeffMax,((TransCoeffLevel[xTbY][yTbY][cIdx][x][y]*m[x][y]
]*levelScale[qP%6]<<(qP/6)+(1<<(bdShift-1)))>>bdShift)

```

### 【 0 1 3 4 】

color\_transform\_flagが1に等しいとき、以下が適用されてよい。

BitDepthY=(BitDepthY-2)

BitDepthC=(BitDepthC-2)

### 【 0 1 3 5 】

一例では、色空間変換がイントラモードに適用されるとき、残差は、変換された色空間  
領域内で導出される。すなわち、イントラサンプル予測プロセスを起動する前に、現在の  
CUの隣接ピクセルが、最初に、修正された順YCoCg変換または修正された順YCoCg-R変換  
によって別の準最適色空間に変換されてよい。次いで、修正された隣接ピクセルは、現在の  
CUの予測子を導出するために使用されてよい。残差(すなわち、予測誤差)は、準最適領域  
内の現在のCUおよび現在のCUの隣接ピクセルから導出される。残差は、成分間残差予測、  
変換、量子化、およびエントロピーコーディングなど、従来のコーディングフローと同じ  
方法でコーディングされる。エントロピーコーディングの後、変換係数は、逆YCoCg変換  
または逆YCoCg-R変換によってさらに修正されてよい。次いで、修正された変換係数は、  
現在のCUの再構成されたピクセルを導出するために予測子に追加されてよい。再構成プロ  
セスが起動された後、逆YCoCg変換または逆YCoCg-R変換は、修正された隣接ピクセルに適  
用されてよい。そのために、デコーダ側において、再構成プロセスを起動する前に、逆色

10

20

30

40

50

空間変換が導出された変換係数に適用される1つの追加のプロセスが適用されてよい。

【0136】

色空間変換がイントラモードに適用されるとき、変換された色領域内の現在のブロックから予測子を引くことによって残差を生成する代わりに、残差は、色空間変換の前のピクセルを使用して生成されてよく、色空間変換がそれに続く。この場合、ルーマおよびクロマのイントラモードは、同等のモードに設定されてよい。代替として、クロマモードのシグナリングはスキップされてよい。代替として、色変換フラグは、ルーマおよびクロマのイントラモードと同じであるときだけシグナリングされてよい。

【0137】

上記で説明したように、いくつかの例では、可逆および非可逆のコーディング方法は、同じ行列、たとえばYCoCg-Rを共有してよい。追加または代替として、YCoCg-R変換が非可逆コーディングに適用されるとき、色変換によるブロックのクロマ成分のビット深度は、色変換によらないブロックと比較して1だけ増加されてよい。色変換によるブロックのルーマ成分QPは、色変換によらないブロックのルーマ成分QPマイナス4に等しくてよい。一例では、色変換によるブロックの2つのクロマ成分QPは、色変換によらないブロックのクロマ/ルーマ成分QPプラス2に等しくてよい。一例では、色変換によるブロックのCgクロマ成分QPは、色変換によらないブロックのルーマ/クロマ成分QPプラス2に等しくてよい。色変換によるブロックのCoクロマ成分QPは、色変換によらないブロックのルーマ/クロマ成分QPプラス3に等しくてよい。

【0138】

そのような例では、YCoCg-Rが非可逆モードと可逆モードの両方に使用されるとき、現在のブロックが非可逆コーディングされる場合、ルーマおよびクロマのビット深度は、適応色変換を有効にすることなくコーディングされたそれらのブロックに対して同じであってよい。この場合、非可逆コーディングが適用されるとき、YCoCg-Rの順変換の後、CoおよびCgの成分は、順YCoCg-R変換によって増加されたビット深度を低減して、ビット深度精度をY成分のビット深度精度と同じにするために、Nビット右シフトによってスケーリングされてよい。加えて、逆YCoCg-R変換を実行する前に、CoおよびCgの成分は、Nビット左シフトによって修正されてよい。一例では、Nは1に等しい。

【0139】

一例では、ルーマおよびクロマの成分の入力ビット深度は同じであることが仮定されてよい。非可逆コーディングモードに対して、シフト演算は、順変換の後で逆変換の前に、2つのクロマ成分(すなわち、Cg、Co)に適用されてよい。一例では、以下のプロセスが、順に適用される。

1. 順方向YCoCg-Rは不变である。

$$C_o = R - B$$

$$t = B + [C_o / 2]$$

$$Cg = G - t$$

$$Y = t + [Cg / 2]$$

2. 現在のブロックが非可逆モードでコーディングされる場合、以下がさらに適用される。

$$Cg = Cg >> 1$$

$$Co = Co >> 1$$

【0140】

別の例では、オフセットが、右シフトプロセスにおいて考慮されてよい。たとえば、上記の強調表示された式は、

$$Cg = (Cg + 1) >> 1$$

$$Co = (Co + 1) >> 1$$

によって置き換えられてよい。

【0141】

逆YCoCg-R変換を起動する前に、以下が適用されてよい。

10

20

30

40

50

現在のブロックが非可逆モードでコーディングされる場合、以下が適用される。

$Cg=Cg<<1$

$Co=Co<<1$

そして、逆方向YCoCg-Rは不变のままである。

$t=Y-[Cg/2]$

$G=Cg+t$

$B=t-[Co/2]$

$R=B+Co$

#### 【 0 1 4 2 】

そのような例では、適応色変換を使用する変換ブロックに対する残差修正プロセスは、  
ChromaArrayTypeが3に等しいときに起動されてよい。このプロセスへの入力には、ブロック  
サイズを指定する変数blkSize、要素 $rY[x][y]$ を有するルーマ残差サンプル $rY$ の $(blkSize) \times (blkSize)$ アレイ、要素 $rCb[x][y]$ を有するクロマ残差サンプル $rCb$ の $(blkSize) \times (blkSize)$ アレイ、および要素 $rCr[x][y]$ を有するクロマ残差サンプル $rCr$ の $(blkSize) \times (blkSize)$ アレイが含まれてよい。このプロセスへの出力には、ルーマ残差サンプルの修正された $(blkSize) \times (blkSize)$ アレイ $r_{cb}$ 、およびクロマ残差サンプルの修正された $(blkSize) \times (blkSize)$ アレイ $r_{cr}$ が含まれてよい。

#### 【 0 1 4 3 】

残差サンプル $rY$ 、 $r_{cb}$ および $r_{cr}$ の $(blkSize) \times (blkSize)$ アレイ、 $x=0 \dots blkSize-1$ 、 $y=0 \dots blkSize-1$ は、以下のように修正されてよい。

$cu\_transquant\_bypass\_flag=0$ ならば、 $r_{cb}[x][y]=r_{cb}[x][y]<<1$ および $r_{cr}[x][y]=r_{cr}[x][y]<<1$   
 $tmp=rY[x][y]-(r_{cb}[x][y]>>1)$   
 $rY[x][y]=tmp+r_{cb}[x][y]$   
 $r_{cb}[x][y]=tmp-(r_{cr}[x][y]>>1)$   
 $r_{cr}[x][y]=r_{cb}[x][y]+r_{cr}[x][y]$

#### 【 0 1 4 4 】

代替として、「 $cu\_transquant\_bypass\_flag$ が0に等しいならば、 $r_{cb}[x][y]=r_{cb}[x][y]<<1$ および $r_{cr}[x][y]=r_{cr}[x][y]<<1$ 」は、「 $r_{cb}[x][y]=r_{cb}[x][y]<<(1-cu\_transquant\_bypass\_flag)$ および $r_{cr}[x][y]=r_{cr}[x][y]<<(1-cu\_transquant\_bypass\_flag)$ 」で置き換えられてよい。

#### 【 0 1 4 5 】

色変換行列を共有する非可逆コーディング方法および可逆コーディング方法の別の例では、ルーマおよびクロマの成分の入力ビット深度は異なることがある。そのような例では、ルーマおよびクロマの成分の入力ビット深度が異なり、YCoCg-Rが非可逆コーディングモードと可逆コーディングモードの両方に使用されるとき、逆変換の前に、Y、Co、Cgの3つすべての成分に同じビット深度精度を持たせるために、Y、Co、Cgの値のうちの少なくとも1つは、最初にNビットシフトされてよい。加えて、もう1つのビットは、2つのクロマ成分(すなわち、CoおよびCg)にシフトされてよい。3つすべての成分の予測値のうちの少なくとも1つは、順YCoCg-R変換を起動する前に、同じビット深度精度を有するように修正されてよい。そして、順変換の後、2つのクロマ成分が、1ビット右シフトされてよい。可逆コーディングモードに対して、異なるビット深度は考慮されておらず、そのために、適応色空間変換は無効にされることがある。一例では、ビットストリームは、ルーマ成分およびクロマ成分のビット深度がコード化ビットストリーム内で異なるとき、色変換は無効(すなわち、 $color\_transform\_enabled\_flag$ が0に等しい)にされるべきであるという制約を含むことがある。

#### 【 0 1 4 6 】

この例では、適応色変換を使用する変換ブロックに対する残差修正プロセスは、Chroma  
ArrayTypeが3に等しいときに起動されてよい。このプロセスへの入力には、ブロックサイ

10

20

30

40

50

ズを指定する変数blkSize、要素rY[x][y]を有するルーマ残差サンプルrYの(blkSize) × (blkSize)アレイ、要素rCb[x][y]を有するクロマ残差サンプルrCbの(blkSize) × (blkSize)アレイ、および要素rCr[x][y]を有するクロマ残差サンプルrCrの(blkSize) × (blkSize)アレイが含まれてよい。このプロセスへの出力には、ルーマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Y</sub>、クロマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Cb</sub>、およびクロマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Cr</sub>が含まれてよい。

#### 【 0 1 4 7 】

変数deltaBD<sub>Y</sub>およびdeltaBD<sub>C</sub>は、以下のように導出され得る。

```
BitDepthMax=Max(BitDepthY, BitDepthC) 10
deltaBDY=cu_transquant_bypass_flag?0:BitDepthMax-BitDepthY
deltaBDC=cu_transquant_bypass_flag?0:BitDepthMax-BitDepthC
OY=cu_transquant_bypass_flag||(BitDepthMax==BitDepthY)?0:1<<(deltaBDY-1)
OC=cu_transquant_bypass_flag||(BitDepthMax==BitDepthC)?0:1<<(deltaBDC-1)
```

#### 【 0 1 4 8 】

残差サンプルr<sub>Y</sub>、r<sub>Cb</sub>およびr<sub>Cr</sub>の(blkSize) × (blkSize)アレイ、x=0...blkSize-1、y=0...blkSize-1は、以下のように修正されてよい。

残差サンプルrY[x][y]、rCb[x][y]およびrCr[x][y]は、以下のように修正されてよい。

```
rY[x][y]=rY[x][y]<<deltaBDY 20
rCb[x][y]=rCb[x][y]<<(deltaBDC+1-cu_transquant_bypass_flag)
rCr[x][y]=rCr[x][y]<<(deltaBDC+1-cu_transquant_bypass_flag)
tmp=rY[x][y]-(rCb[x][y]>>1)
rY[x][y]=tmp+rCb[x][y]
rCb[x][y]=tmp-(rCr[x][y]>>1)
rCr[x][y]=rCb[x][y]+rCr[x][y]
rY[x][y]=(rY[x][y]+OY)>>deltaBDY
rCb[x][y]=(rCb[x][y]+0c)>>deltaBDC
rCr[x][y]=(rCr[x][y]+0c)>>deltaBDC
```

#### 【 0 1 4 9 】

上記の例では、関数Maxは、2変数間のより大きい値を導出するために使用されてよい。代替として、丸めオフセット0<sub>Y</sub>および0<sub>C</sub>のうちの少なくとも一方が0に設定されてよい。

#### 【 0 1 5 0 】

代替として、可逆コーディングに対してさえも、ルーマおよびクロマの成分のビット深度が異なるとき、成分は、必要に応じて同じビット深度精度にシフトされることがある。エンコーダ側において、順変換の後、シフトして元のビット深度精度に戻す必要はない。言い換えば、3つの成分のコーディングされた残差は、それが可逆コーディングであることを確実にするために、同じビット深度であってよい。非可逆コーディングとは異なり、可逆コーディングのために逆YCoCg-R変換を起動する前に、入力は修正されない。出力が元の入力と同じビット深度であることを確実にするために、右シフトが、依然として必要であることがある。

#### 【 0 1 5 1 】

この例では、適応色変換を使用する変換ブロックに対する残差修正プロセスは、ChromaArrayTypeが3に等しいときに起動されてよい。このプロセスへの入力には、ブロックサイズを指定する変数blkSize、要素rY[x][y]を有するルーマ残差サンプルrYの(blkSize) × (blkSize)アレイ、要素rCb[x][y]を有するクロマ残差サンプルrCbの(blkSize) × (blkSize)アレイ、および要素rCr[x][y]を有するクロマ残差サンプルrCrの(blkSize) × (blkSize)アレイが含まれてよい。このプロセスへの出力には、ルーマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Y</sub>、クロマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Cb</sub>、およびクロマ残差サンプルの修正された(blkSize) × (blkSize)アレイr<sub>Cr</sub>が含まれてよい。

10

20

30

40

50

## 【 0 1 5 2 】

変数deltaBD<sub>Y</sub>およびdeltaBD<sub>C</sub>は、以下のように導出されてよい。

```
BitDepthMax=Max(BitDepthY, BitDepthC)
deltaBDY=BitDepthMax-BitDepthY
deltaBDC=BitDepthMax-BitDepthC
OY=(BitDepthMax==BitDepthY)?0:1<<(deltaBDY-1)
OC=(BitDepthMax==BitDepthC)?0:1<<(deltaBDC-1)
```

## 【 0 1 5 3 】

残差サンプルr<sub>Y</sub>、r<sub>Cb</sub>およびr<sub>Cr</sub>の(blkSize) × (blkSize)アレイ、x=0...blkSize-1、y=0...blkSize-1は、以下のように修正されてよい。

cu\_transquant\_bypass\_flagが0に等しいとき、残差サンプルrY[x][y]、rCb[x][y]およびrCr[x][y]は、以下のように修正されてよい。

```
rY[x][y]=rY[x][y]<<deltaBDY
rCb[x][y]=rCb[x][y]<<(deltaBDC+1-cu_transquant_bypass_flag)
rCr[x][y]=rCr[x][y]<<(deltaBDC+1-cu_transquant_bypass_flag)
tmp=rY[x][y]-(rCb[x][y]>>1)
rY[x][y]=tmp+rCb[x][y]
rCb[x][y]=tmp-(rCr[x][y]>>1)
rCr[x][y]=rCb[x][y]+rCr[x][y]
rY[x][y]=(rY[x][y]+OY)>>deltaBDY
rCb[x][y]=(rCb[x][y]+0c)>>deltaBDC
rCr[x][y]=(rCr[x][y]+0c)>>deltaBDC
```

## 【 0 1 5 4 】

代替として、O<sub>Y</sub>およびO<sub>C</sub>のうちの一方は、0に等しくてよい。

## 【 0 1 5 5 】

ビデオエンコーダ20は、さらに、ブロックベースの構文データ、フレームベースの構文データ、およびGOPベースの構文データなどの構文データを、たとえばフレームヘッダ、ブロックヘッダ、スライスヘッダ、またはGOPヘッダの中でビデオデコーダ30に送ることがある。GOP構文データは、それぞれのGOPの中のフレーム数を記述し、フレーム構文データは、対応するフレームを符号化するために使用される符号化/予測モードを示すことができる。

## 【 0 1 5 6 】

ビデオエンコーダ20およびビデオデコーダ30はそれぞれ、1つもしく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)、ディスクリートロジック回路、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組合せなど、適用可能な様々な適切なエンコーダもしくはデコーダの回路のうちのいずれかとして実装されてもよい。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダに含まれてもよく、これらのいずれもが、結合されたビデオエンコーダ/デコーダ(コードック)の一部として統合されてもよい。ビデオエンコーダ20および/またはビデオデコーダ30を含むデバイスは、集積回路、マイクロプロセッサ、および/またはセルラー電話などのワイヤレス通信デバイスを備えてよい。

## 【 0 1 5 7 】

図2は、色空間変換プロセスを使用してビデオブロックを符号化するための技法を実装し得るビデオエンコーダ20の一例を示すブロック図である。ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングとインターコーディングとを実行してもよい。イントラコーディングは、所与のビデオフレームまたはピクチャ内のビデオにおける空間的冗長性を低減または除去するために空間的予測に依存する。インターコーディングは、ビデオシーケンスの隣接フレームまたはピクチャ内のビデオにおける時間的冗長性を低減または除去するために時間的予測に依存する。イントラモード(1モード)は、

10

20

30

40

50

いくつかの空間ベースのコーディングモードのうちのいずれかを指してもよい。片方向予測(Pモード)または双方向予測(Bモード)などのインターモードは、いくつかの時間ベースのコーディングモードのうちのいずれかを指してもよい。

#### 【0158】

図2に示すように、ビデオエンコーダ20は、符号化されるべきビデオフレーム内の現在のビデオブロックを受信する。図2の例では、ビデオエンコーダ20は、モード選択ユニット40と、参照ピクチャメモリ64と、加算器50と、変換処理ユニット52と、量子化ユニット54と、エントロピー符号化ユニット56とを含む。モード選択ユニット40は、動き補償ユニット44と、動き推定ユニット42と、イントラ予測ユニット46と、区分ユニット48とを含む。ビデオブロック再構成に対して、ビデオエンコーダ20はまた、逆量子化ユニット58と、逆変換ユニット60と、加算器62とを含む。デブロッキングフィルタ(図2に示さず)がまた、ブロッキネスアーティファクトを再構成されたビデオから除去するために、フィルタブロック境界に含まれてよい。所望される場合、デブロッキングフィルタは、一般に、加算器62の出力をフィルタ処理する。追加のフィルタ(ループ内またはループ後)が、デブロッキングフィルタに加えて使用されてもよい。そのようなフィルタは、簡潔のために示さないが、所望される場合、(ループ内フィルタとして)加算器50の出力をフィルタ処理してよい。

10

#### 【0159】

符号化プロセスの間、ビデオエンコーダ20は、コーディングされるべきビデオフレームまたはスライスを受信する。フレームまたはスライスは、複数のビデオブロックに分割されてよい。動き推定ユニット42および動き補償ユニット44は、時間予測を提供するために、1つまたは複数の参照フレーム内の1つまたは複数のブロックに対する受信されたビデオブロックのインター予測コーディングを実行する。代替として、イントラ予測ユニット46は、空間予測を提供するために、コーディングされるべきブロックと同じフレームまたはスライス内の1つまたは複数の隣接ブロックに対する受信されたビデオブロックのイントラ予測コーディングを実行してもよい。ビデオエンコーダ20は、たとえば、ビデオデータの各ブロックに対する適切なコーディングモードを選択するために、複数のコーディングパスを実行してよい。

20

#### 【0160】

その上、区分ユニット48は、前のコーディングパス内の前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックに区分してよい。たとえば、区分ユニット48は、最初にフレームまたはスライスをLCUに区分し、レート-歪み分析(たとえば、レート-歪み最適化)に基づいてLCUの各々をサブCUに区分してよい。モード選択ユニット40は、LCUをサブCUに区分することを示す4分木データ構造をさらに作成してもよい。4分木のリーフノードCUは、1つまたは複数のPUと1つまたは複数のTUとを含んでもよい。

30

#### 【0161】

モード選択ユニット40は、たとえば誤差結果に基づいて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のうちの一方を選択し、得られたイントラコード化ブロックまたはインターコード化ブロックを、残差ブロックデータを生成するために加算器50に、および参照フレームとして使用するための符号化ブロックを再構成するために加算器62に供給してよい。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのような構文情報などの構文要素をエントロピー符号化ユニット56に供給する。

40

#### 【0162】

動き推定ユニット42および動き補償ユニット44は、高度に統合されてもよいが、概念的な目的のために別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックに関する動きを推定する動きベクトルを生成するプロセスである。たとえば、動きベクトルは、現在フレーム(または他のコード化ユニット)内でコーディングされる現在のブロックに対して、参照フレーム(または他のコード化ユニット)内の予測ブロックに対する現在のビデオフレームまたはピクチャ内のビデオブロックのPUの変位を示し

50

得る。予測ブロックは、差の絶対値の合計(SAD:sum of absolute difference)、自乗差の合計(SSD:sum of square difference)、または他の差メトリックによって決定されてもよいピクセル差の観点で、コーディングされるべきブロックと密接に一致することが見出されたブロックである。いくつかの例では、ビデオエンコーダ20は、参照ピクチャメモリ64内に記憶された参照ピクチャのサブ整数ピクセル位置のための値を計算してもよい。たとえば、ビデオエンコーダ20は、参照ピクチャの4分の1ピクセル位置の値、8分の1ピクセル位置の値、または他の分数ピクセル位置の値を補間してもよい。したがって、動き推定ユニット42は、フルピクセル位置および分数ピクセル位置に対する動き探索を実行し、分数ピクセル精度で動きベクトルを出力してもよい。

## 【0163】

10

動き推定ユニット42は、参照ピクチャの予測ブロックの位置とPUの位置とを比較することによって、インターフェース内のビデオブロックのPUのための動きベクトルを計算する。参照ピクチャは、その各々が参照ピクチャメモリ64内に記憶された1つまたは複数の参照ピクチャを識別する第1の参照ピクチャリスト(リスト0)または第2の参照ピクチャリスト(リスト1)から選択されてもよい。動き推定ユニット42は、エントロピー符号化ユニット56および動き補償ユニット44に計算された動きベクトルを送信する。

## 【0164】

動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定された動きベクトルに基づいて、予測ブロックを取得または生成することを伴ってもよい。再び、動き推定ユニット42および動き補償ユニット44は、いくつかの例では機能的に一体化されてよい。現在のビデオブロックのPUのための動きベクトルを受信すると、動き補償ユニット44は、参照ピクチャリストのうちの1つにおいて動きベクトルが指す予測ブロックの位置を特定してもよい。以下で説明するように、加算器50は、コーディングされている現在のビデオブロックのピクセル値から予測ブロックのピクセル値を減算し、ピクセル差分値を形成することによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42は、ルーマ成分に対する動き推定を実行し、動き補償ユニット44は、クロマ成分とルーマ成分の両方に対してルーマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40はまた、ビデオスライスのビデオブロックを復号する際にビデオエンコーダ30によって使用するための、ビデオブロックおよびビデオスライスに関連付けられた構文要素を生成してもよい。

20

## 【0165】

30

イントラ予測ユニット46は、上記で説明したように、動き推定ユニット42および動き補償ユニット44によって実行されるイントラ予測の代替として、現在のブロックをイントラ予測してもよい。具体的には、イントラ予測ユニット46は、現在のブロックを符号化するために使用するイントラ予測モードを決定してもよい。いくつかの例では、イントラ予測ユニット46は、たとえば、別々の符号化パスの間、様々なイントラ予測モードを使用して現在のブロックを符号化してもよく、イントラ予測ユニット46(または、いくつかの例ではモード選択ユニット40)は、試験されたモードから使用する適切なイントラ予測モードを選択してもよい。

## 【0166】

40

たとえば、イントラ予測ユニット46は、様々な試験されたイントラ予測モードに対してレート-歪み分析を使用してレート-歪み値を計算し、試験されたモードの中から最良のレート-歪み特性を有するイントラ予測モードを選択してもよい。レート-歪み分析は、一般に、符号化ブロックと、符号化ブロックを生成するために符号化された元の非符号化ブロックとの間の歪の量(または誤差)、ならびに、符号化ブロックを生成するために使用されたビットレート(すなわち、ビット数)を決定する。イントラ予測ユニット46は、どのイントラ予測モードがブロックのための最良のレート-歪み値を示すのかを決定するために、様々な符号化ブロックに関する歪みおよびレートから比を計算してもよい。

## 【0167】

ブロックのためのイントラ予測モードを選択した後、イントラ予測ユニット46は、エン

50

トロピー符号化ユニット56にブロックのための選択されたイントラ予測モードを示す情報を提供してもよい。エントロピー符号化ユニット56は、選択されたイントラ予測モードを示す情報を符号化してよい。ビデオエンコーダ20は、複数のイントラ予測モードインデックステーブルおよび複数の変更されたイントラ予測モードインデックステーブル(コードワードマッピングテーブルとも呼ばれる)を含んでもよい、送信されたビットストリーム構成データ内に、コンテキストの各々のために使用する、様々なブロックのための符号化コンテキストの定義と、最もあり得るイントラ予測モードの指示と、イントラ予測モードインデックステーブルと、変更されたイントラ予測モードインデックステーブルとを含んでもよい。

## 【0168】

10

ビデオエンコーダ20は、モード選択ユニット40からの予測データをコーディングされる元のビデオブロックから引くことによって、残差ビデオブロックを形成する。加算器50は、この減算演算を実行する1つまたは複数の構成要素を表す。変換処理ユニット52は、離散コサイン変換(DCT)または概念的に類似の変換などの変換を残差ブロックに適用して、残差変換係数値を含むビデオブロックを作成する。変換処理ユニット52は、DCTに概念的に類似する他の変換を実行してもよい。ウェーブレット変換、整数変換、サブバンド変換、またはさもなければ他のタイプの変換が使用されてもよい。いずれの場合も、変換処理ユニット52は、変換を残差ブロックに適用して、残差変換係数のブロックを作成する。変換は、残差情報を、ピクセル値領域から周波数領域などの変換領域に変換してもよい。変換処理ユニット52は、得られた変換係数を量子化ユニット54に送信してもよい。量子化ユニット54は、ビットレートをさらに低減するために、変換係数を量子化する。量子化プロセスは、係数の一部またはすべてに関連するビット深度を低減してもよい。量子化の程度は、量子化パラメータを調整することによって変更されてもよい。いくつかの例では、量子化ユニット54は、次いで、量子化変換係数を含む行列の走査を実行してもよい。代替的には、エントロピー符号化ユニット56は、走査を実行してもよい。

## 【0169】

20

量子化に続いて、エントロピー符号化ユニット56は、量子化変換係数をエントロピーコーディングする。たとえば、エントロピー符号化ユニット56は、コンテキスト適応可変長コーディング(CAVLC)、コンテキスト適応2値算術コーディング(CABAC)、構文ベースコンテキスト適応2値算術コーディング(SBAC)、確率インターバル区分エントロピー(PIPE)コーディング、または別のエントロピーコーディング技法を実行してもよい。コンテキストベースのエントロピーコーディングの場合、コンテキストは、隣接ブロックに基づくことがある。エントロピー符号化ユニット56によるエントロピーコーディングに続いて、符号化ビットストリームは、別のデバイス(たとえば、ビデオデコーダ30)に送信されてもよく、または、後の送信または検索のためにアーカイブされてもよい。

## 【0170】

30

本開示の技法によれば、ビデオデコーダ20のエントロピー符号化ユニット56は、本開示の1つまたは複数の技法を実行し得る。たとえば、ビデオエンコーダ20のエントロピー符号化ユニット56は、ビデオデータを符号化するために色空間変換を使用するかどうかを決定し得る。エントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化することを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

## 【0171】

40

逆量子化ユニット58および逆変換ユニット60は、たとえば参照ブロックとして後に使用するためのピクセル領域における残差ブロックを再構成するために、それぞれ、逆量子化および逆変換を適用する。動き補償ユニット44は、参照ピクチャメモリ64のフレームのうちの1つの予測ブロックに残差ブロックを加算することによって、参照ブロックを計算し

50

てもよい。動き補償ユニット44はまた、動き推定において使用するためのサブ整数ピクセル値を計算するために、再構成された残差ブロックに1つまたは複数の補間フィルタを適用してもよい。加算器62は、参照ピクチャメモリ64内に記憶するための再構成されたビデオブロックを生成するために、動き補償ユニット44によって生成された動き補償された予測ブロックに再構成された残差ブロックを加算する。再構成されたビデオブロックは、後続のビデオフレーム内のブロックをインターフォーマンスするために、参照ブロックとして動き推定ユニット42および動き補償ユニット44によって使用されてもよい。

#### 【 0 1 7 2 】

このようにして、ビデオエンコーダ20のエントロピー符号化ユニット56は、本開示の1つまたは複数の技法を実行し得る。たとえば、ビデオエンコーダ20のエントロピー符号化ユニット56は、ビデオデータを符号化するために色空間変換を使用するかどうかを決定し得る。エントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化することを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。

#### 【 0 1 7 3 】

第1のQPおよび第2のQPは、色空間変換によらずにコーディングされたブロックと同じ導出方法に従って、第1および第2の色成分に対応して導出されたQPであり得る。たとえば、第1のQPおよび第2のQPに対するQPの値は、cu\_qp\_delta\_absなどのシグナリングされた構文要素から導出され得る。次いで、第1のオフセットおよび第2のオフセットは、それぞれ、ビデオデータが色空間変換を使用して符号化されたとき、第1のQPおよび第2のQPを取得するために、導出されたQPに加算されてよい。たとえば、QP(たとえば、QP<sub>i</sub>)に対して、それぞれのオフセット(たとえば、offset<sub>i</sub>)は、QPに加算されてよく、得られたQPの値を、元のQPプラスそれぞれのオフセットに等しくさせる(たとえば、QP<sub>final i</sub>=QP<sub>i</sub>+offset<sub>i</sub>)。

#### 【 0 1 7 4 】

いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPに対する第1の値と第2のQPに対する第2の値とをさらに決定し得る。ビデオデータを符号化するために色空間変換を使用しないことを決定することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPの第1の値を使用して第1の色成分のデータを量子化し、第2のQPの第2の値を使用して第2の色成分のデータを量子化し得る。代替として、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPの第1の値に加えられた第1のオフセットを使用して第1の色成分のデータを量子化し、第2のQPの第2の値に加えられた第2のオフセットを使用して第2の色成分のデータを量子化し得、第1のオフセットは第2のオフセットと異なる。

#### 【 0 1 7 5 】

いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加し得、第1のオフセットは8に等しい。他の例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加し得、第1のオフセットは2に等しい。

#### 【 0 1 7 6 】

いくつかの例では、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、色空間変

10

20

30

40

50

換プロセスをコーディングユニットに適用してよい。いくつかの例では、色空間変換プロセスは、元のYCoCg-R変換行列を含む。いくつかの他の例では、色空間変換プロセスは、修正されたYCoCg変換行列を適用するステップを含む。そのような例では、修正されたYCoCg変換行列の係数は、丸めオフセットによって2ビット右シフトした後、元のYCoCg変換行列の係数に等しく設定されてよい。修正されたYCoCg変換行列は、0、1または2に等しいオフセットを有してよい。他の例では、色空間変換プロセスは、元のYCoCg行列を適用するステップを含む。そのような例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、正規化プロセスをビデオデータにさらに適用してよい。

#### 【0177】

いくつかの例では、第1の色成分は、ルーマ(Y)成分である。いくつかの例では、第1の色成分は、クロマ成分(Cb)である。さらに、いくつかの例では、第2の色成分は、クロマ(Cr)成分である。いくつかの例では、第1のオフセットおよび第2のオフセットのうちの少なくとも1つは、0に等しくない。

10

#### 【0178】

いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、ビデオデータを符号化するために色空間変換を使用することを決定し得る。そうする際に、ビデオデコーダ20のエントロピー符号化ユニット56は、コーディングユニットに対して色空間変換を使用することの決定に応答して、色空間変換の使用を示すためにビデオデータの構文要素の値を設定してよい。いくつかの例では、構文要素は、1ビットフラグを含む。いくつかの例では、構文要素は、コーディングユニットが、イントラパルス符号変調(IPCM)モード以外のモードを使用してコーディングされるときにシグナリングされる。他の例では、構文要素は、コーディングユニットの変換単位内に非ゼロ係数がないときだけシグナリングされる。いくつかの例では、構文要素に対する1の値は、コーディングユニットが色空間変換を使用して符号化されたことを示す。

20

#### 【0179】

いくつかの例では、構文要素はシグナリングされない。たとえば、構文要素は、コーディングユニットがイントラコーディングされるとき、およびコーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるときはシグナリングされないことがある。別の例では、構文要素は、コーディングユニットがパレットモードでコーディングされるときはシグナリングされない。

30

#### 【0180】

図3は、ビデオブロックを復号するための技法を実装し得るビデオデコーダ30の一例を示すブロック図であり、ビデオブロックのいくつかは、色空間変換プロセスを使用して符号化されている。図3の例では、ビデオデコーダ30は、エントロピー復号ユニット70と、動き補償ユニット72と、イントラ予測ユニット74と、逆量子化ユニット76と、逆変換ユニット78と、参照ピクチャメモリ82と、加算器80とを含む。ビデオデコーダ30は、いくつかの例では、ビデオエンコーダ20(図2)に関連して説明した符号化バスと全体的に相互的な復号バスを実行してもよい。動き補償ユニット72は、エントロピー復号ユニット70から受信された動きベクトルに基づいて予測データを生成してよく、一方、イントラ予測ユニット74は、エントロピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて予測データを生成してよい。

40

#### 【0181】

復号プロセスの間、ビデオデコーダ30は、ビデオエンコーダ20から、符号化ビデオストライスのビデオブロックと関連する構文要素とを表す符号化されたビデオビットストリームを受信する。ビデオデコーダ30のエントロピー復号ユニット70は、量子化係数と、動きベクトルもしくはイントラ予測モードインジケータと、他の構文要素とを生成するためにビットストリームをエントロピー復号する。エントロピー復号ユニット70は、動き補償ユニット72に動きベクトルと他の構文要素とを転送する。ビデオデコーダ30は、ビデオストライスレベルおよび/またはビデオブロックレベルにおいて構文要素を受信してもよい。

#### 【0182】

50

本開示の技法によれば、ビデオデコーダ30のエントロピー復号ユニット70は、本開示の1つまたは複数の技法を実行し得る。たとえば、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る。ビデオデコーダ30のエントロピー復号ユニット70は、さらに、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化することとを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。ビデオデータが色空間変換を使用して符号化されなかったものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに、第1の色成分に対する第1のQPと第2の色成分に対する第2のQPとを使用してビデオデータを逆量子化し得る。

#### 【 0 1 8 3 】

ビデオスライスがイントラコード化(I)スライスとしてコーディングされたとき、イントラ予測処理ユニット74は、シグナリングされたイントラ予測モードと、現在のフレームまたはピクチャの以前に復号されたブロックからのデータとに基づいて、現在のビデオスライスのビデオブロックのための予測データを生成してもよい。ビデオフレームがインターコード化(すなわち、B、P、またはGPB)スライスとしてコーディングされたとき、動き補償ユニット72は、エントロピー復号ユニット70から受信した動きベクトルと他の構文要素とに基づいて、現在のビデオスライスのビデオブロックのための予測ブロックを生成する。予測ブロックは、参照ピクチャリストのうちの1つ内の参照ピクチャのうちの1つから生成されてもよい。ビデオデコーダ30は、参照ピクチャメモリ82内に記憶された参照ピクチャに基づいて、デフォルトの構成技術を使用して、参照フレームリスト、リスト0およびリスト1を構成してもよい。動き補償ユニット72は、動きベクトルと他の構文要素とを構文解析することによって、現在のビデオスライスのビデオブロックのための予測情報を決定し、復号されている現在のビデオブロックのための予測ブロックを生成するために予測情報を使用する。たとえば、動き補償ユニット72は、ビデオスライスのビデオブロックをコーディングするために使用された予測モード(たとえば、イントラまたはインター予測)と、インター予測スライスタイプ(たとえば、Bスライス、Pスライス、またはGPBスライス)と、スライスのための参照ピクチャリストのうちの1つまたは複数のための構成情報と、スライスの各インター符号化ビデオブロックのための動きベクトルと、スライスの各インターコード化ビデオブロックのためのインター予測状態と、現在のビデオスライス内のビデオブロックを復号するための他の情報とを決定するために、受信した構文要素の一部を使用する。

#### 【 0 1 8 4 】

動き補償ユニット72はまた、補間フィルタに基づいて補間を実行してもよい。動き補償ユニット72は、参照ブロックのサブ整数ピクセルに関する補間値を計算するために、ビデオブロックの符号化の間にビデオエンコーダ20によって使用されるように補間フィルタを使用してもよい。この場合には、動き補償ユニット72は、受信した構文要素からビデオエンコーダ20によって使用された補間フィルタを決定し、予測ブロックを生成するために補間フィルタを使用してもよい。

#### 【 0 1 8 5 】

逆量子化ユニット76は、ビットストリーム内で提供され、エントロピー復号ユニット70によって復号された量子化変換係数を逆量子化する(inverse quantize)、すなわち逆量子化する(de-quantize)。逆量子化プロセスは、量子化の程度を決定し、同様に、適用されるべき逆量子化の程度を決定するために、ビデオスライス内の各ビデオブロックのための、ビデオエンコーダ30によって計算された量子化パラメータ $QP_Y$ の使用を含んでもよい。

#### 【 0 1 8 6 】

逆変換ユニット78は、ピクセル領域における残差ブロックを生成するために、変換係数に逆変換、たとえば、逆DCT、逆整数変換、または概念的に同様の逆変換プロセスを適用

10

20

30

40

50

する。

【0187】

動き補償ユニット72が、動きベクトルと他の構文要素に基づいて現在のビデオブロックのための予測ブロックを生成した後、ビデオデコーダ30は、逆変換ユニット78からの残差ブロックを、動き補償ユニット72によって生成された対応する予測ブロックと加算することによって、復号ビデオブロックを形成する。加算器80は、この加算演算を実行する1つまたは複数の構成要素を表す。所望される場合、デブロッキングフィルタは、ブロッキネスアーティファクトを除去するために、復号ブロックをフィルタ処理するために適用されてもよい。(コーディンググループの中またはコーディンググループの後のいずれかの)他のループフィルタはまた、ピクセル遷移をなめらかにするため、またはビデオ品質を他の方法で改善するために使用されてもよい。次いで、所与のフレームまたはピクチャ内の復号ビデオブロックは、後続の動き補償のために使用される参照ピクチャを記憶する参照ピクチャメモリ82内に記憶される。参照ピクチャメモリ82はまた、図1の表示デバイス32などの表示デバイス上で後に提示するための復号ビデオを記憶する。

【0188】

このようにして、ビデオデコーダ30のエントロピー復号ユニット70は、本開示の1つまたは複数の技法を実行し得る。たとえば、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る。ビデオデコーダ30のエントロピー復号ユニット70は、さらに、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを逆量子化することと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化することを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる。ビデオデータが色空間変換を使用して符号化されなかつたものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに第1のQPおよび第2のQPを使用してビデオデータを逆量子化し得る。

【0189】

第1のQPおよび第2のQPは、色空間変換によらずにコーディングされたブロックと同じ導出方法に従って、第1および第2の色成分に対応して導出されたQPであり得る。たとえば、第1のQPおよび第2のQPに対するQPの値は、cu\_qp\_delta\_absなどのシグナリングされた構文要素から導出され得る。次いで、第1のオフセットおよび第2のオフセットは、それぞれ、ビデオデータが色空間変換を使用して符号化されたとき、第1のQPおよび第2のQPを取得するために、導出されたくQPに加算されてよい。たとえば、QP(たとえば、QP<sub>i</sub>)に対して、それぞれのオフセット(たとえば、offset<sub>i</sub>)は、QPに加算されてよく、得られたQPの値を、元のQPプラスそれぞれのオフセットに等しくさせる(たとえば、QP<sub>final i</sub>=QP<sub>i</sub>+offset<sub>i</sub>)。

【0190】

いくつかの例では、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPに対する第1の値と第2のQPに対する第2の値とをさらに決定し得る。ビデオデータを符号化するために色空間変換を使用しないことを決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPの第1の値を使用して第1の色成分のデータを逆量子化し、第2のQPの第2の値を使用して第2の色成分のデータを逆量子化することができる。代替として、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPの第1の値に加えられた第1のオフセットを使用して第1の色成分のデータを逆量子化し、第2のQPの第2の値に加えられた第2のオフセットを使用して第2の色成分のデータを逆量子化することができ、第1のオフセットは第2のオフセットと異なる。

【0191】

いくつかの例では、第1のQPと第2のQPとの間の差は、ビデオデータを符号化するために使用される色空間変換プロセスのノルムによって決まる。いくつかの例では、ビデオデコ

10

20

30

40

50

ーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたとき、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加することができ、第1のオフセットは8に等しい。他の例では、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたとき、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加することができ、第1のオフセットは2に等しい。

#### 【0192】

いくつかの例では、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、色空間逆変換プロセスをコーディングユニットに適用してよい。いくつかの例では、色空間逆変換プロセスは、元のYCoCg-R逆変換行列を含む。いくつかの他の例では、色空間逆変換プロセスは、修正されたYCoCg逆変換行列を適用するステップ含んでよい。そのような例では、修正されたYCoCg逆変換行列の係数は、丸めオフセットによって2ビット右シフトした後、元のYCoCg逆変換行列の係数に等しく設定されてよい。修正されたYCoCg変換行列は、0、1または2に等しいオフセットを有してよい。他の例では、色空間逆変換プロセスは、元のYCoCg行列を適用するステップ含んでよい。そのような例では、ビデオデコーダ30のエントロピー復号ユニット70は、正規化プロセスをビデオデータにさらに適用してよい。

10

#### 【0193】

いくつかの例では、第1の色成分は、ルーマ(Y)成分である。いくつかの例では、第1の色成分は、クロマ成分(Cb)である。さらに、いくつかの例では、第2の色成分は、クロマ(Cr)成分である。いくつかの例では、第1のオフセットおよび第2のオフセットのうちの少なくとも1つは、0に等しくない。

20

#### 【0194】

いくつかの例では、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータのコーディングユニットの構文要素を復号してよい。構文要素は、コーディングユニットが色空間変換を使用して符号化されたかどうかを示し得る。いくつかの例では、構文要素は、1ビットフラグを含むことがある。いくつかの例では、コーディングユニットは、イントラパルス符号変調(IPCM)モード以外のモードでコーディングされ、構文要素は、IPCMモード以外のモードを使用するコーディングユニットに対してだけシグナリングされる。いくつかの例では、構文要素は、コーディングユニットの変換単位内に非ゼロ係数があるときに、コーディングユニットが色空間変換を使用して符号化されたことを示す。いくつかの例では、構文要素に対する1の値は、コーディングユニットが色空間変換を使用して符号化されたことを示す。ビデオデコーダ30のエントロピー復号ユニット70は、さらに、コーディングユニットが色空間変換を使用して符号化されたことを構文要素の値が示すかどうかを決定し得る。コーディングユニットが色空間変換を使用してコーディングされたことを構文要素が示すものと決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、色空間逆変換プロセスを適用してよい。

30

#### 【0195】

いくつかの例では、構文要素は、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。たとえば、構文要素は、コーディングユニットがイントラコーディングされるとき、およびコーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるとき、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。別の例では、構文要素は、コーディングユニットがパレットモードでコーディングされるとき、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。これらの例では、構文要素は、ビデオデータを含む受信されたビットストリーム内に存在せず、構文要素を復号するステップは、構文要素の値を推測するステップを含んでよい。

40

#### 【0196】

図4は、本開示の1つまたは複数の技法による35のHEVC予測モードを示す概念図である。現在のHEVCでは、各予測単位(PU)のルーマ成分に対して、図4に関して説明するように、

50

イントラ予測方法が、33の角度予測モード(2から34までインデックス付けられる)、平均値モード(1でインデックス付けられる)、および平面モード(0でインデックス付けられる)とともに利用される。

#### 【0197】

上記の35のイントラモードに加えて、「I-PCM」という名称のもう1つのモードもまた、HEVCによって利用される。I-PCMモードでは、予測、変換、量子化、およびエントロピーコーディングがバイパスされる一方で、予測サンプルが所定の数のビットだけコーディングされる。I-PCMモードの主たる目的は、信号が他のモードによって効率的にコーディングされ得ない状況を処理することである。

#### 【0198】

図5は、本開示の1つまたは複数の技法による、マージのための空間的隣接動きベクトル候補とアドバンスト動きベクトル予測(AMVP)モードとを示す概念図である。図5に関して説明するように、空間的MV候補が、特定のPU(PU0)に対して図5に示す隣接ブロックから導出されるが、ブロックから候補を生成する方法は、マージモードおよびAMVPモードに対して異なる。

#### 【0199】

図5(a)に示すように、マージモードでは、4つまでの空間的MV候補が、図5(a)に番号で示す順序で導出されてよく、順序は、以下の左(0)、上(1)、右上(2)、左下(3)、および左上(4)である。

#### 【0200】

図5(b)に示すように、AMVPモードでは、隣接ブロックは2つのグループ、すなわちブロック0および1からなる左のグループと、ブロック2、3および4からなる上のグループに分割される。各グループに対して、シグナリングされた参照インデックスによって示されるものと同じ参照ピクチャを参照する隣接ブロック内の潜在的候補は、グループの最終候補を形成するために選択されるための最高の優先度を有する。すべての隣接ブロックが、同じ参照ピクチャを指す動きベクトルを含まないことが可能である。そのために、そのような候補を見出すことができない場合、第1の利用可能な候補が、最終候補を形成するようにスケーリングされることになり、したがって時間的距離の差が補償され得る。

#### 【0201】

図6は、本開示の1つまたは複数の技法による、イントラブロックコピー(BC)の例を示す概念図である。図6に関して以下で説明するように、イントラブロックコピー(BC)が、RExt内に含まれている。イントラBCの一例が、図6におけるように示されており、現在のCUは、現在のピクチャ/スライスのすでに復号されたブロックから予測される。現在のイントラBCブロックのサイズは、CUのサイズと同じ大きさであってよく、CUのサイズは $8 \times 8$ から $64 \times 64$ までに及ぶが、いくつかの用途ではさらなる制約が追加で適用されることがある。

#### 【0202】

図7は、本開示の1つまたは複数の技法による、イントラ $8 \times 8$ ブロックに対するターゲットブロックおよび参照サンプルの一例を示す概念図である。図7に関して以下で説明するように、変換行列は、参照サンプル値から導出される。異なる参照サンプルは、イントラの場合およびインターの場合に利用される。イントラブロックの場合のターゲットブロックおよび参照サンプルを図7に示す。この図では、ターゲットブロックは、 $8 \times 8$ のクロスハッチのサンプルからなり、参照は、ストライプとドットのサンプルである。

#### 【0203】

インターブロックの場合、行列導出に対する参照サンプルは、動き補償に対する参照サンプルと同じである。シフト演算を実現するために、AMPブロック内の参照サンプルは、サンプルの数が2のべき乗になるようにサブサンプリングされる。たとえば、 $12 \times 16$ ブロック内の参照サンプルの数は、 $2/3$ に低減される。

#### 【0204】

本開示のいくつかの技法によれば、色空間変換プロセスが適用されてよい。そのような例では、変換プロセスが起動されるか否かをシグナリングする必要はない。加えて、エン

10

20

30

40

50

コーダ側とデコーダ側の両方が、変換行列をシグナリングするためのオーバーヘッドを回避するために同じ方法を用いて変換行列を導出することがある。

#### 【0205】

図8は、現在のブロックを符号化するための例示的な方法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または現在のCUの一部を備えてよい。ビデオエンコーダ20(図1および図2)に関して説明するが、他のデバイスが、図8の方法に類似する方法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。

#### 【0206】

この例では、ビデオエンコーダ20は、最初に、現在のブロックを予測する(150)。たとえば、ビデオエンコーダ20は、現在のブロックに対する1つまたは複数の予測単位(PU)を計算し得る。次いで、ビデオエンコーダ20は、たとえば変換単位(TU)を作成するために、現在のブロックに対する残差ブロックを計算し得る(152)。残差ブロックを計算するためには、ビデオエンコーダ20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間の差を計算し得る。ビデオエンコーダ20のエントロピー符号化ユニット56は、ビデオデータを符号化するために色空間変換を使用するかどうかを決定してよく、色空間変換が有効にされるとき、色空間変換が適用される(154)。次いで、ビデオエンコーダ20は、残差ブロックの係数を変換してよい(156)。エントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第1の色成分のデータを量子化する(158)ことと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを量子化することを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる(160)。次に、ビデオエンコーダ20は、残差ブロックの量子化された変換係数を走査し得る(162)。走査中、または走査に続いて、ビデオエンコーダ20は、係数をエントロピー符号化し得る(164)。たとえば、ビデオエンコーダ20は、CAVLCまたはCABACを使用して係数を符号化し得る。次いで、ビデオエンコーダ20は、ブロックのエントロピーコード化データを出力し得る(166)。

#### 【0207】

いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPに対する第1の値と第2のQPに対する第2の値とをさらに決定し得る。ビデオデータを符号化するために色空間変換を使用しないことを決定することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPの第1の値を使用して第1の色成分のデータを量子化し、第2のQPの第2の値を使用して第2の色成分のデータを量子化することができる。代替として、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、第1のQPの第1の値に加えられた第1のオフセットを使用して第1の色成分のデータを量子化し、第2のQPの第2の値に加えられた第2のオフセットを使用して第2の色成分のデータを量子化し得、第1のオフセットは第2のオフセットと異なる。

#### 【0208】

いくつかの例では、第1のQPと第2のQPとの間の差は、ビデオデータを符号化するために使用される色空間変換プロセスのノルムによって決まる。いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加することができ、第1のオフセットは8に等しい。他の例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、さらに、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加することができ、第1のオフセットは2に等しい。

#### 【0209】

いくつかの例では、ビデオデータを符号化するために色空間変換を使用することを決定

10

20

30

40

50

することに応答して、ビデオエンコーダ20のエントロピー符号化ユニット56は、色空間変換プロセスをコーディングユニットに適用してよい。いくつかの例では、色空間変換プロセスは、元のYCoCg-R変換行列を含む。いくつかの他の例では、色空間変換プロセスは、修正されたYCoCg変換行列を適用するステップを含む。そのような例では、修正されたYCoCg変換行列の係数は、丸めオフセットによって2ビット右シフトした後、元のYCoCg変換行列の係数に等しく設定されてよい。修正されたYCoCg変換行列は、0、1または2に等しいオフセットを有してよい。他の例では、色空間変換プロセスは、元のYCoCg行列を適用するステップを含む。そのような例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、正規化プロセスをビデオデータにさらに適用してよい。

## 【0210】

10

いくつかの例では、第1の色成分は、ルーマ(Y)成分である。いくつかの例では、第1の色成分は、クロマ成分(Cb)である。さらに、いくつかの例では、第2の色成分は、クロマ(Cr)成分である。いくつかの例では、第1のオフセットおよび第2のオフセットのうちの少なくとも1つは、0に等しくない。

## 【0211】

いくつかの例では、ビデオエンコーダ20のエントロピー符号化ユニット56は、ビデオデータを符号化するために色空間変換を使用することを決定し得る。そうする際に、ビデオエンコーダ20のエントロピー符号化ユニット56は、コーディングユニットに対して色空間変換を使用することの決定に応答して、色空間変換の使用を示すためにビデオデータの構文要素の値を設定してよい。いくつかの例では、構文要素は、1ビットフラグを含む。いくつかの例では、構文要素は、コーディングユニットが、イントラパルス符号変調(IPCM)モード以外のモードを使用してコーディングされるときにシグナリングされる。他の例では、構文要素は、コーディングユニットの変換単位内に非ゼロ係数がないときだけシグナリングされる。いくつかの例では、構文要素に対する1の値は、コーディングユニットが色空間変換を使用して符号化されたことを示す。

20

## 【0212】

いくつかの例では、構文要素はシグナリングされない。たとえば、構文要素は、コーディングユニットがイントラコーディングされるとき、およびコーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるときはシグナリングされない。別の例では、構文要素は、コーディングユニットがパレットモードでコーディングされるときはシグナリングされない。これらの例では、構文要素は、ビデオデータを含む受信されたビットストリーム内に存在せず、構文要素を復号するステップは、構文要素の値を推測するステップを含んでよい。

30

## 【0213】

図9は、ビデオデータの現在のブロックを復号するための例示的な方法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または現在のCUの一部を備えてよい。ビデオデコーダ30(図1および図3)に関して説明するが、他のデバイスが、図9の方法に類似する方法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。

## 【0214】

40

ビデオデコーダ30は、たとえば、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イントラ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(200)。ビデオデコーダ30はまた、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ド化データなど、現在のブロックに対するエントロピーコード化データを受信し得る(202)。ビデオデコーダ30は、残差ブロックの係数を復元するために、エントロピーコード化データをエントロピー復号し得る(204)。次いで、ビデオデコーダ30は、量子化された変換係数のブロックを生成するために、復元された係数を逆走査し得る(206)。ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたかどうかを決定し得る(208)。ビデオデコーダ30のエントロピー復号ユニット70は、さらに、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、第1の量子化パラメータ(QP)の第1のオフセットを使用してビデオデータの第

50

1の色成分のデータを逆量子化する(210)ことと、第2のQPの第2のオフセットを使用してビデオデータの第2の色成分のデータを逆量子化することを行ってよく、第2の色成分は第1の色成分と異なり、第2のQPの第2のオフセットは第1のQPの第1のオフセットと異なる(212)。次いで、ビデオデコーダ30は、係数を逆変換してよく(214)、色空間変換が有効にされるとき、色空間逆変換がさらに適用されてよい。ビデオコーダ30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(216)。ビデオデータが色空間変換を使用して符号化されなかったものと決定することに応答して、ビデオコーダは、第1および第2のオフセットなしに、第1の色成分に対する第1のQPと第2の色成分に対する第2のQPとを使用してビデオデータを逆量子化し得る。

## 【0215】

10

いくつかの例では、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPに対する第1の値と第2のQPに対する第2の値とをさらに決定し得る。ビデオデータを符号化するために色空間変換を使用しないことを決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPの第1の値を使用して第1の色成分のデータを逆量子化し、第2のQPの第2の値を使用して第2の色成分のデータを逆量子化することができる。代替として、ビデオデータを符号化するために色空間変換を使用することを決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、第1のQPの第1の値に加えられた第1のオフセットを使用して第1の色成分のデータを逆量子化し、第2のQPの第2の値に加えられた第2のオフセットを使用して第2の色成分のデータを逆量子化することができ、第1のオフセットは第2のオフセットと異なる。

## 【0216】

20

いくつかの例では、第1のQPと第2のQPとの間の差は、ビデオデータを符号化するために使用される色空間変換プロセスのノルムによって決まる。いくつかの例では、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたとき、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を2だけ増加することができ、第1のオフセットは8に等しい。他の例では、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータが色空間変換を使用して符号化されたとき、ビデオデータに対する量子化プロセスおよび変換プロセスのために使用されるビット深度を1だけ増加することができ、第1のオフセットは2に等しい。

## 【0217】

30

いくつかの例では、ビデオデータが色空間変換を使用して符号化されたものと決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、色空間逆変換プロセスをコーディングユニットに適用してよい。いくつかの例では、色空間逆変換プロセスは、元のYCoCg-R逆変換行列を含む。いくつかの他の例では、色空間逆変換プロセスは、修正されたYCoCg逆変換行列を適用するステップ含んでよい。そのような例では、修正されたYCoCg逆変換行列の係数は、丸めオフセットによって2ビット右シフトした後、元のYCoCg逆変換行列の係数に等しく設定されてよい。修正されたYCoCg変換行列は、0、1または2に等しいオフセットを有してよい。他の例では、色空間逆変換プロセスは、元のYCoCg行列を適用するステップ含んでよい。そのような例では、ビデオデコーダ30のエントロピー復号ユニット70は、正規化プロセスをビデオデータにさらに適用してよい。

## 【0218】

40

いくつかの例では、第1の色成分は、ルーマ(Y)成分である。いくつかの例では、第1の色成分は、クロマ成分(Cb)である。さらに、いくつかの例では、第2の色成分は、クロマ(Cr)成分である。いくつかの例では、第1のオフセットおよび第2のオフセットのうちの少なくとも1つは、0に等しくない。

## 【0219】

いくつかの例では、ビデオデコーダ30のエントロピー復号ユニット70は、ビデオデータのコーディングユニットの構文要素を復号してよい。構文要素は、コーディングユニットが色空間変換を使用して符号化されたかどうかを示し得る。いくつかの例では、構文要素は、1ビットフラグを含むことがある。いくつかの例では、コーディングユニットは、イ

50

ントラパルス符号変調(IPCM)モード以外のモードでコーディングされ、構文要素は、IPCMモード以外のモードを使用するコーディングユニットに対してだけシグナリングされる。いくつかの例では、構文要素は、コーディングユニットの変換単位内に非ゼロ係数があるときに、コーディングユニットが色空間変換を使用して符号化されたことを示す。いくつかの例では、構文要素に対する1の値は、コーディングユニットが色空間変換を使用して符号化されたことを示す。ビデオデコーダ30のエントロピー復号ユニット70は、さらに、コーディングユニットが色空間変換を使用して符号化されたことを構文要素の値が示すかどうかを決定し得る。コーディングユニットが色空間変換を使用して符号化されたことを構文要素が示すものと決定することに応答して、ビデオデコーダ30のエントロピー復号ユニット70は、色空間逆変換プロセスを適用してよい。

10

#### 【0220】

いくつかの例では、構文要素は、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。たとえば、構文要素は、コーディングユニットがインストラコーディングされるとき、およびコーディングユニット内の予測単位のルーマ予測モードおよびクロマ予測モードが異なるとき、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。別の例では、構文要素は、コーディングユニットがパレットモードでコーディングされるとき、色空間変換がコーディングユニットを符号化するために使用されなかったことを示し得る。これらの例では、受信されたビットストリームは、構文要素を含まなくてよい。

#### 【0221】

例に応じて、本明細書で説明される技法のいずれかのいくつかの行為またはイベントは異なる順序で実行されてもよく、一緒に追加され、統合され、または省略されてもよい(たとえば、説明される行為またはイベントのすべてが技法の実施のために必要とは限らない)ことを認識されたい。さらに、いくつかの例では、行為またはイベントは、順次的ではなく、たとえばマルチスレッド処理、割り込み処理またはマルチプロセッサを通じて同時に実行され得る。

20

#### 【0222】

1つまたは複数の例において、説明される機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せで実装され得る。ソフトウェアで実装される場合、機能は、1つもしく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、またはコンピュータ可読媒体を介して送信されてよく、かつハードウェアに基づく処理ユニットによって実行されてよい。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に対応するコンピュータ可読記憶媒体を含むこと、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可能にする任意の媒体を含む通信媒体を含むことがある。このようにして、コンピュータ可読媒体は、概して、(1)非一時的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、本開示で説明された技法を実装するための命令、コード、および/またはデータ構造を取り出すために1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスされ得る任意の利用可能な媒体であってよい。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。

30

#### 【0223】

限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、フラッシュメモリ、または、命令もしくはデータ構造の形式の所望のプログラムコードを記憶するために使用され、コンピュータによってアクセスされ得る任意の他の媒体を含み得る。また、任意の接続も厳密にはコンピュータ可読媒体と呼ばれる。たとえば、同軸ケーブル、光ファイバケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから命令が送信される場合、同軸ケーブル、

40

50

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は、接続、搬送波、信号、または他の一時的な媒体を含まず、代わりに非一時的な有形記憶媒体を指すことを理解されたい。ディスク(disk)およびディスク(disc)は、本明細書で使用される場合、コンパクトディスク(CD)、レーザディスク、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびBlu-ray(登録商標)ディスクを含み、ディスク(disk)は通常、磁気的にデータを再生し、ディスク(disc)は、レーザを用いて光学的にデータを再生する。上記の組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。

【0224】

10

命令は、1つまたは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、または他の等価の集積論理回路もしくはディスクリート論理回路のような、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、または本明細書で説明される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、本明細書で説明される機能は、符号化および復号のために構成された専用のハードウェアモジュールおよび/またはソフトウェアモジュール内に与えられてよく、あるい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。

【0225】

20

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。本開示では、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、様々なコンポーネント、モジュール、またはユニットが説明されたが、これらのコンポーネント、モジュール、またはユニットは、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。そうではなくて、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結合されてよく、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、前述のような1つもしくは複数のプロセッサを含む、相互動作可能なハードウェアユニットの集合によって提供されてよい。

【0226】

30

本開示の様々な例を説明してきた。説明したシステム、動作、または機能の任意の組合せが考えられる。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】

【0227】

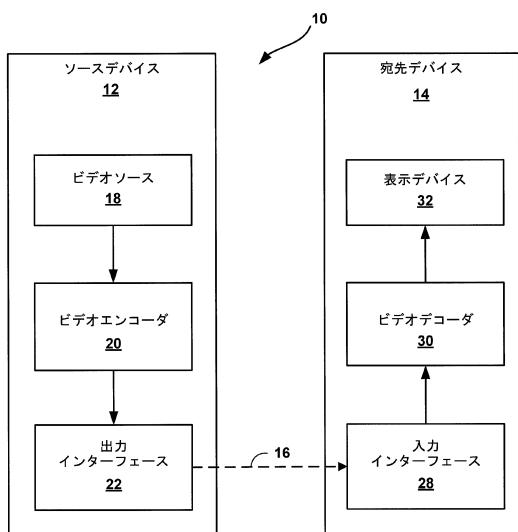
- 10 ビデオ符号化および復号システム
- 12 ソースデバイス
- 14 宛先デバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 20 ビデオエンコーダ
- 22 出力インターフェース
- 28 入力インターフェース
- 30 ビデオデコーダ
- 32 表示デバイス
- 40 モード選択ユニット
- 42 動き推定ユニット
- 44 動き補償ユニット
- 46 イントラ予測ユニット
- 48 区分ユニット
- 50 加算器

40

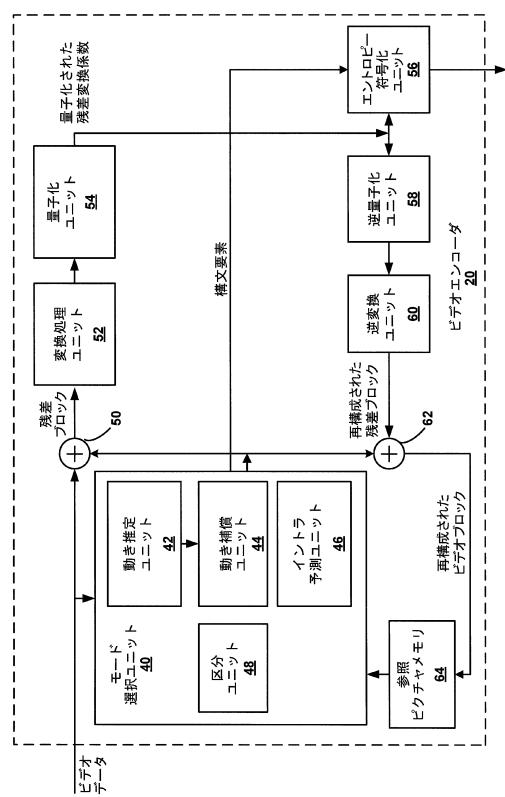
50

- |    |               |
|----|---------------|
| 52 | 変換処理ユニット      |
| 54 | 量子化ユニット       |
| 56 | エントロピー符号化ユニット |
| 58 | 逆量子化ユニット      |
| 60 | 逆変換ユニット       |
| 62 | 加算器           |
| 64 | 参照ピクチャメモリ     |
| 70 | エントロピー復号ユニット  |
| 72 | 動き補償ユニット      |
| 74 | イントラ予測ユニット    |
| 76 | 逆量子化ユニット      |
| 78 | 逆変換ユニット       |
| 80 | 加算器           |
| 82 | 参照ピクチャメモリ     |

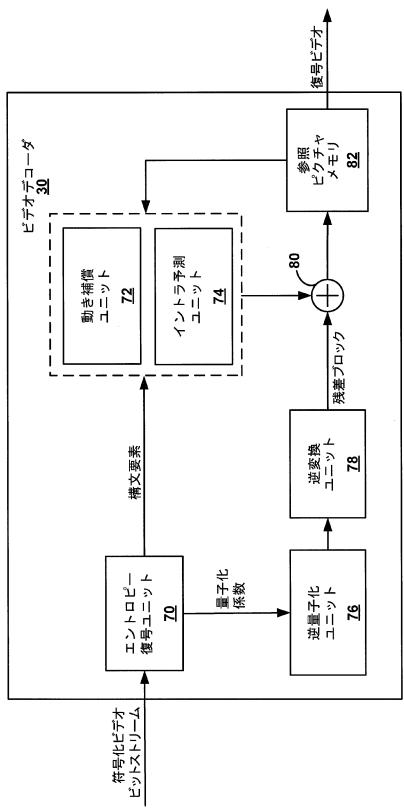
【 図 1 】



【 図 2 】



【図3】



【図4】

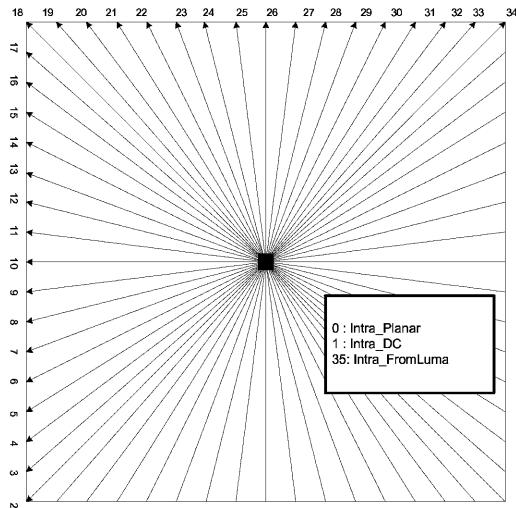
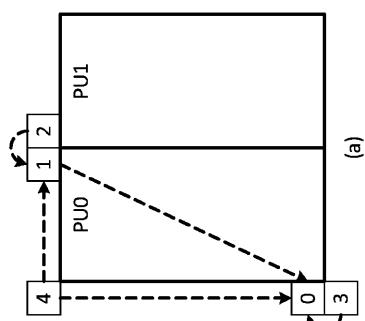


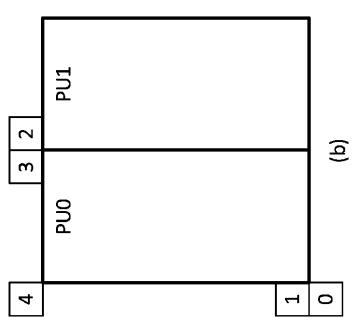
FIG. 4

【図5 (a)】



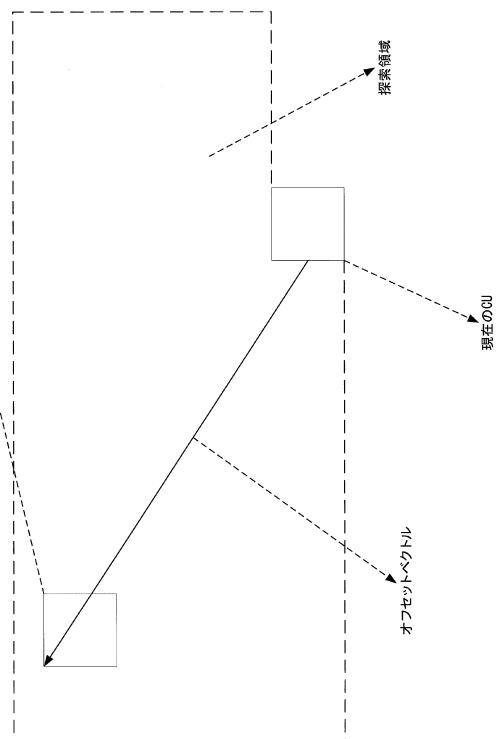
(a)

【図5 (b)】



(b)

【図6】



【図7】

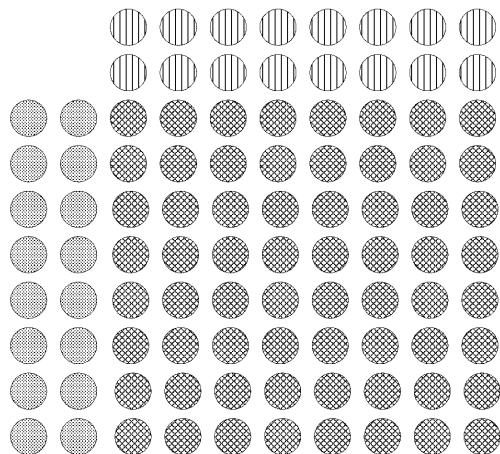
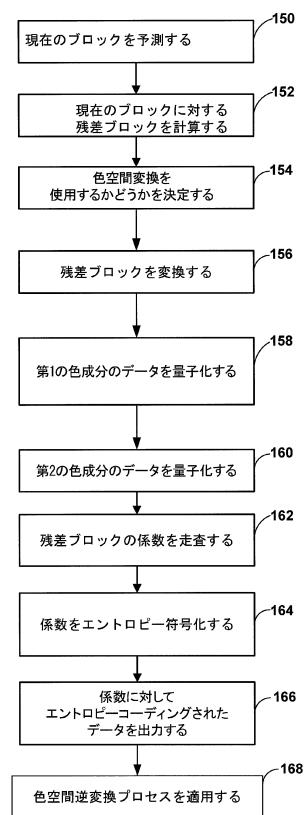
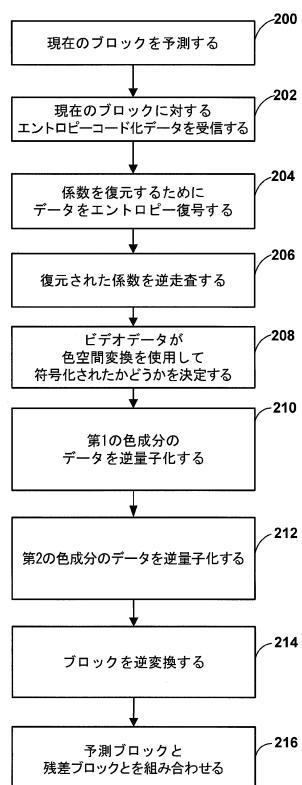


FIG. 7

【図8】



【図9】



---

フロントページの続き

(31)優先権主張番号 62/062,637

(32)優先日 平成26年10月10日(2014.10.10)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 14/656,482

(32)優先日 平成27年3月12日(2015.3.12)

(33)優先権主張国・地域又は機関

米国(US)

(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・92121-1714・サン・ディエゴ・モアハウス・ドライ  
ヴ・5775

(72)発明者 マータ・カルチェヴィチ

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ  
ヴ・5775

(72)発明者 ジョエル・ソール・ロジャルス

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ  
ヴ・5775

(72)発明者 ウ・シク・キム

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ  
ヴ・5775

審査官 堀井 啓明

(56)参考文献 米国特許出願公開第2005/0259730(US, A1)

Woo-Shik Kim Multimedia Lab. Samsung AIT P.O.Box 111, Suwon 440-600 Korea, Residue Col  
or Transform[online], JVT-L JVT-L025, インターネット<URL:[http://wftp3.itu.int/av-arch/jvt-site/2004\\_07\\_Redmond/JVT-L025r2.doc](http://wftp3.itu.int/av-arch/jvt-site/2004_07_Redmond/JVT-L025r2.doc)>, 2012年12月 3日

(58)調査した分野(Int.Cl., DB名)

H04N19/00-19/98