

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号

特表2022-544413

(P2022-544413A)

(43)公表日 令和4年10月18日(2022.10.18)

(51)国際特許分類	F I	テーマコード(参考)
H 0 4 N 19/91 (2014.01)	H 0 4 N 19/91	5 C 1 5 9
H 0 4 N 19/70 (2014.01)	H 0 4 N 19/70	
H 0 4 N 19/186 (2014.01)	H 0 4 N 19/186	

審査請求 有 予備審査請求 未請求 (全204頁)

(21)出願番号	特願2022-509093(P2022-509093)	(71)出願人	520477474
(86)(22)出願日	令和2年8月15日(2020.8.15)		バイトダンス インコーポレイテッド
(85)翻訳文提出日	令和4年3月2日(2022.3.2)		BYTEDANCE INC.
(86)国際出願番号	PCT/US2020/046581		アメリカ合衆国 カリフォルニア州 9 0
(87)国際公開番号	WO2021/030788		0 6 6 , ロサンジェルス, ウェスト ジ
(87)国際公開日	令和3年2月18日(2021.2.18)		ェファーソン ブールヴァード 1 2 6 5
(31)優先権主張番号	PCT/CN2019/100850		5 , シックス フロアー, スイート ナ
(32)優先日	令和1年8月15日(2019.8.15)		ンバー・1 3 7
(33)優先権主張国・地域又は機関	中国(CN)		1 2 6 5 5 West Jefferso
(31)優先権主張番号	PCT/CN2019/106700		n Boulevard, Sixth
(32)優先日	令和1年9月19日(2019.9.19)		Floor, Suite No. 137
(33)優先権主張国・地域又は機関	中国(CN)		Los Angeles, Califo
(31)優先権主張番号	PCT/CN2019/107494	(74)代理人	100107766
	最終頁に続く		最終頁に続く

(54)【発明の名称】 パレット・エスケープ・シンボルのためのエントロピーコーディング

(57)【要約】

パレット・モード符号化および復号におけるパレット・エスケープ・シンボルのエントロピー符号化を実行するための方法、システム、および装置が記載される。ビデオ処理のための例示的な方法が、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

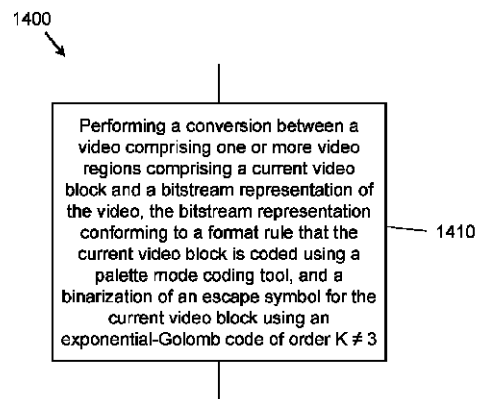


FIG. 14

## 【特許請求の範囲】

## 【請求項 1】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数 $K$ の指数ゴロム(EG)符号を使用し、

$K$ は、3と等しくない負でない整数であり、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、

10

方法。

## 【請求項 2】

$K = 0$ である、請求項 1 に記載の方法。

## 【請求項 3】

$K = 1$ である、請求項 1 に記載の方法。

## 【請求項 4】

$K = 2$ である、請求項 1 に記載の方法。

## 【請求項 5】

20

ビデオ処理の方法であって：

一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化され、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される、

30

方法。

## 【請求項 6】

前記固定長バイナリ化は、 $N$ ビットを使用し、 $N$ は1より大きい整数である、請求項 5 に記載の方法。

## 【請求項 7】

$N$ が、内部ビット深さに基づく、請求項 6 に記載の方法。

## 【請求項 8】

$N$ の値が、スライス・サブピクチャー、タイル、ピクチャー、またはビデオにおいて信号伝達される、請求項 6 に記載の方法。

## 【請求項 9】

40

$N$ が量子化パラメータに基づいている、請求項 6 に記載の方法。

## 【請求項 10】

$N$ が、 $f(Qp)$ として示される量子化パラメータ( $Qp$ )の関数( $f()$ )に基づく、請求項 9 に記載の方法。

## 【請求項 11】

$N$ が( $ibd - \max(16, (Qp - 4)/6)$ )に設定され、ここで、 $ibd$ が内部ビット深さである、請求項 9 に記載の方法。

## 【請求項 12】

$N$ が( $ibd - \max(QpPrimeTsMin, (Qp - 4)/6)$ )に設定され、ここで、 $ibd$ は内部ビット深さであり、 $QpPrimeTsMin$ は変換スキップ・モードのための最小の許容される量

50

子化パラメータである、請求項 9 に記載の方法。

【請求項 13】

$N$ が $\max(A, (\text{ibd} - \max(16, (\text{QpPrimeTsMin} - 4)/6)))$ に設定され、ここで、 $\text{ibd}$ は内部ビット深さであり、 $\text{QpPrimeTsMin}$ は変換スキップ・モードのための最小の許容される量子化パラメータであり、 $A$ は負でない整数である、請求項 9 に記載の方法。

【請求項 14】

$A = 0$ または $A = 1$ である、請求項 13 に記載の方法。

【請求項 15】

前記量子化パラメータが、前記ビデオのスライスの量子化パラメータと定数値との和であり、前記定数値が整数値である、請求項 9 ないし 14 のうちいずれか一項に記載の方法

10

【請求項 16】

ビデオ処理の方法であって、

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長符号化を使用し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、

20

方法。

【請求項 17】

前記可変長符号化が次数3の指数ゴロム符号を除外する、請求項 16 に記載の方法。

【請求項 18】

前記可変長符号化が、入力パラメータ $K$ をもつ打ち切りされたバイナリ(TB)符号であり、 $K$ は整数である、請求項 16 に記載の方法。

【請求項 19】

$K$ は、(a)シーケンスパラメータセット(SPS)、ビデオパラメータセット(VPS)、ピクチャーパラメータセット(PPS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイトル・グループ・ヘッダ、最大符号化単位(LCU)行、LCUのグループ、またはブリック、(b)内部ビット深さ、(c)入力ビット深さ、(d)内部ビット深さと入力ビット深さとの間の相違、(e)現在ビデオ・ブロックの寸法、(f)現在ビデオ・ブロックの現在の量子化パラメータ、(g)前記ビデオのカラー・フォーマットの指示、(h)符号化ツリー構造、または(i)前記ビデオの色成分に基づいている、請求項 18 に記載の方法。

30

【請求項 20】

前記エスケープ・シンボルの複数の値が、複数のバイナリ化方法を使用して信号伝達される、請求項 5 に記載の方法。

【請求項 21】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、

前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、

前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、

方法。

40

50

## 【請求項 2 2】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、  
方法。

10

## 【請求項 2 3】

前記量子化が右ビットシフトに対応する、請求項 2 2 に記載の方法。

## 【請求項 2 4】

前記エスケープ・シンボルが  $f(p, Q_p)$  として符号化され、 $f()$  は関数であり、 $p$  は入力シンボル値であり、 $Q_p$  は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、請求項 2 2 に記載の方法。

## 【請求項 2 5】

$f$  が  $p$  と  $g(Q_p)$  として定義される、請求項 2 4 に記載の方法。

## 【請求項 2 6】

$f$  が、 $(p + (1 - (g(Q_p) - 1)))$  と  $g(Q_p)$  として定義される、請求項 2 4 に記載の方法。

20

## 【請求項 2 7】

$f$  が、

$\text{clip}(0, (1 - bd) - 1, (p + (1 - (g(Q_p) - 1)))$  と  $g(Q_p)$

として定義され、 $\text{clip}(x, \text{min}, \text{max})$  はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $\text{max}$  は整数である、請求項 2 4 に記載の方法。

## 【請求項 2 8】

前記エスケープ・シンボルは  $h(p)$  として符号化され、 $h()$  は関数であり、 $p$  は入力値シンボルである、請求項 2 2 に記載の方法。

30

## 【請求項 2 9】

$h$  が  $p$  と  $N$  として定義され、 $N$  は負でない整数である、請求項 2 8 に記載の方法。

## 【請求項 3 0】

$h$  が、 $(p + (1 - (N - 1)))$  と  $N$  として定義され、 $N$  は負でない整数である、請求項 2 8 に記載の方法。

## 【請求項 3 1】

$\text{cu\_transquant\_bypass\_flag} = 1$  のとき  $N = 0$  である、請求項 2 9 または 3 0 に記載の方法。

## 【請求項 3 2】

$\text{cu\_transquant\_bypass\_flag} = 1$  のとき  $N = (bd - ibd)$  であり、 $bd$  は内部ビット深さであり、 $ibd$  は入力ビット深さである、請求項 2 9 または 3 0 に記載の方法。

40

## 【請求項 3 3】

$h$  は、 $\text{clip}(0, (1 - (bd - N) - 1, p + N)$  として定義され、ここで、 $bd$  は、現在のビデオ・ブロックの現在の色成分の内部ビット深さであり、 $N$  は負でない整数であり、 $\text{clip}(x, \text{min}, \text{max})$  は、クリッピング関数であり、 $x$ 、 $\text{min}$ 、および  $\text{max}$  は整数である、請求項 2 8 に記載の方法。

## 【請求項 3 4】

$h$  は、 $\text{clip}(0, (1 - (bd - N) - 1, (p + (1 - (N - 1)))$  と  $N$  として定義され、ここで、 $bd$  は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $N$  は、負でない整数であり、 $\text{clip}(x, \text{min}, \text{max})$  はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $m$

50

axは整数である、請求項28に記載の方法。

【請求項35】

Nが $[0, (bd - 1)]$ の範囲内であり、bdは、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、請求項29ないし34のうちいずれか一項に記載の方法。

【請求項36】

前記脱量子化は、左ビットシフトに対応する、請求項22に記載の方法。

【請求項37】

前記エスケープ・シンボルは、 $f(p, Qp)$ として脱量子化され、ここで、 $f()$ は関数であり、pはデコードされたエスケープ・シンボルであり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、請求項36に記載の方法。

10

【請求項38】

fが $p$   $g(Qp)$ として定義される、請求項37に記載の方法。

【請求項39】

前記エスケープ・シンボルは $f(p, Qp)$ として再構成され、ここで、 $f()$ は関数であり、pはデコードされたエスケープ・シンボルであり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、請求項36に記載の方法。

【請求項40】

20

fは、 $\text{clip}(0, (1 - bd) - 1, p - g(Qp))$ として定義され、bdは、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $\text{clip}(x, \text{min}, \text{max})$ はクリッピング関数であり、x、min、maxは整数である、請求項39に記載の方法。

【請求項41】

クリッピング関数 $\text{clip}(x, \text{min}, \text{max})$ は

【数3】

$$\text{clip}(x, \text{min}, \text{max}) = \begin{cases} \text{min} & x < \text{min} \\ x & \text{min} \leq x \leq \text{max} \\ \text{max} & x > \text{max} \end{cases}$$

30

として定義される、請求項27、33、34または40に記載の方法。

【請求項42】

前記エスケープ・シンボルは $h(p)$ として再構成され、 $h()$ は関数であり、pはデコードされたエスケープ・シンボルである、請求項36に記載の方法。

【請求項43】

hは $p - N$ として定義され、Nは負でない整数である、請求項42に記載の方法。

【請求項44】

$\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = 0$ である、請求項42または43に記載の方法。

40

【請求項45】

$\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = (bd - ibd)$ であり、bdは内部ビット深さであり、ibdは入力ビット深さである、請求項42または43に記載の方法。

【請求項46】

$N = (\max(QpPrimeTsMin, qP) - 4) / 6$ であって、qPはデコードされた量子化パラメータであり、 $QpPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータである、請求項42または43に記載の方法。

【請求項47】

Nがさらに $\text{min}(bd - 1, N)$ としてクリップされ、bdが現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、請求項43ないし46のうちいずれか一項に

50

記載の方法。

【請求項 48】

$N$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ は現在のビデオ・ブロックの現在の色成分の内部ビット深さである、請求項 43 ないし 47 のうちいずれか一項に記載の方法。

【請求項 49】

前記エスケープ・シンボルの再構成オフセットは、ビット深さ情報に基づく、請求項 36 に記載の方法。

【請求項 50】

前記ビット深さ情報は、内部ビット深さと入力ビット深さとの間の差 ( $B_D$ と記される) を含む、請求項 49 に記載の方法。

10

【請求項 51】

$K$   $B_D$ である場合、再構成されたオフセットは  $p$   $K$ に等しく、ここで、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ は、整数である、請求項 50 に記載の方法。

【請求項 52】

前記再構成されたオフセットは、 $K$   $T_0$ の場合、 $p$   $K$ に等しく、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ および $T_0$ は整数である、請求項 49 に記載の方法。

【請求項 53】

$T_0 = 2$ である、請求項 50 に記載の方法。

【請求項 54】

再構成されたオフセットは、  
 $(p - K) + ((1 - (K - 1)) \cdot B_D - B_D)$   
 に等しく、  
 $p$ はデコードされたエスケープ・シンボルであり、 $K$ は整数である、  
 請求項 50 に記載の方法。

20

【請求項 55】

$B_D$ が、シーケンス・レベル、ピクチャー・レベル、スライス・レベル、タイル・レベル、ブリック・レベル、またはサブピクチャー・レベルでビットストリーム表現において信号伝達される、請求項 50 に記載の方法。

【請求項 56】

前記エスケープ・シンボルがコンテキスト符号化される、請求項 22 ないし 55 のうちいずれか一項に記載の方法。

30

【請求項 57】

前記エスケープ・シンボルがバイパス符号化される、請求項 22 ないし 55 のうちいずれか一項に記載の方法。

【請求項 58】

$g(Q_p)$ が $(Q_p - 4)/6$ として定義される、請求項 25 ないし 27、38 または 40 のうちいずれか一項に記載の方法。

【請求項 59】

$g(Q_p)$ が $(\max(M, Q_p) - 4)/6$ として定義され、 $M$ は整数である、請求項 25 ないし 27、39 または 40 のうちいずれか一項に記載の方法。

40

【請求項 60】

$M$ が、シーケンスパラメータセット (SPS) において信号伝達される、請求項 59 に記載の方法。

【請求項 61】

$g(Q_p)$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ が、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、請求項 58 ないし 60 のうちいずれか一項に記載の方法。

【請求項 62】

ビデオ処理の方法であって：

50

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずに符号化され、

前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、  
方法。

【請求項 6 3】

前記一つまたは複数のパレット・インデックスのバイナリ化が、バイナリ化入力パラメータとして最大パレット・インデックスを有する打ち切りされたバイナリ (TB) 符号である、請求項 6 2 に記載の方法。

10

【請求項 6 4】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化され、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、  
方法。

20

【請求項 6 5】

前記非エスケープ・シンボルのインデックスから前記エスケープ・シンボルのインデックスを導出することが許されない、請求項 6 4 に記載の方法。

【請求項 6 6】

前記エスケープ・シンボルのインデックスから前記非エスケープ・シンボルのインデックスを導出することが許されない、請求項 6 4 に記載の方法。

【請求項 6 7】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの導出されるパレット・インデックスは最大値をもち、

前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、  
方法。

30

【請求項 6 8】

前記最大値は、現在のパレット・テーブル・サイズである、請求項 6 7 に記載の方法。

40

【請求項 6 9】

前記最大値は、一つまたは複数のエスケープ・シンボルについてのインデックスを除外する現在のパレット・テーブル・サイズであり、前記一つまたは複数のエスケープ・シンボルのうちのあるエスケープ・シンボルが、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、請求項 6 7 に記載の方法。

【請求項 7 0】

ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現

50

在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、

前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【請求項 7 1】

前記インデックスが `MaxPaletteIndex` に等しい、請求項 7 0 に記載の方法。

【請求項 7 2】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいて符号化される構文要素を使用して表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

【請求項 7 3】

前記現在インデックスと前記参照インデックスとの間の差が符号化される、請求項 7 2 に記載の方法。

【請求項 7 4】

前記差の符号化された表現は、ゼロ値の差を除外する、請求項 7 3 に記載の方法。

【請求項 7 5】

前記現在インデックスと前記参照インデックスとの間の差のモジュロが符号化される、請求項 7 2 に記載の方法。

【請求項 7 6】

前記モジュロが  $I = \text{modulo}(C - R, \text{MaxPaletteIndex})$  として表され、 $C$  は前記現在インデックスであり、 $R$  は前記参照インデックスであり、`MaxPaletteIndex` は所定の負でない整数である、請求項 7 5 に記載の方法。

【請求項 7 7】

前記パレット・モード符号化ツールのパレット・ブロックの始めにおいて、前記参照インデックスが  $-1$  に設定される、請求項 7 2 に記載の方法。

【請求項 7 8】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、予測的に符号化されるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【請求項 7 9】

前記エスケープ・シンボルは、以前に符号化されたエスケープ・シンボルに基づいて予測的に符号化される、請求項 7 8 に記載の方法。

【請求項 8 0】

前記ビデオのある色成分における前記エスケープ・シンボルは、同じ色成分における値に基づいて予測的に符号化される、請求項 7 8 に記載の方法。

10

20

30

40

50



## 【請求項 8 1】

前記ビデオの第1の色成分における前記エスケープ・シンボルは、前記第1の色成分とは異なる前記ビデオの第2の色成分における値に基づいて予測的に符号化される、請求項78に記載の方法。

## 【請求項 8 2】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングス符号化される構文要素を用いて表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

## 【請求項 8 3】

長さ要素のプレフィックスについてのコンテキストは、デコーダにおけるインデックス調整プロセスの後のパレット・インデックスに基づく、請求項82に記載の方法。

## 【請求項 8 4】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

## 【請求項 8 5】

前記前の諸パレット・インデックスを使用することは、現在のビデオ・ブロックが一つまたは複数のエスケープ・シンボルを含むかどうかに基づき、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのためにエスケープ・シンボルが使用される、請求項84に記載の方法。

## 【請求項 8 6】

ビデオ処理の方法であって：

整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ・ブロックの符号化モードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップと、

該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップとを含む、方法。

## 【請求項 8 7】

前記第1の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、請求項86に記載の方法。

## 【請求項 8 8】

前記第2の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、請求項86または87に記載の方法。

## 【請求項 8 9】

前記符号化モードが、現在のビデオ・ブロックについての最確モード (most proba

10

20

30

40

50

ble mode、MPM)を含む、請求項 86 ないし 88 のうちいずれか一項に記載の方法。

【請求項 90】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ(QP)差を含む構文要素を使用して表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、

10

【請求項 91】

前記QP差が、閾値よりも大きい幅を有するパレット・ブロックについて符号化される、請求項 90 に記載の方法。

【請求項 92】

前記QP差が、閾値よりも高い高さを有するパレット・ブロックについて符号化される、請求項 90 に記載の方法。

【請求項 93】

20

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数の符号化ブロック・フラグ(CBF)を含む構文要素を使用して表現されるというフォーマット規則に適合し、

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、

【請求項 94】

30

前記CBFのそれぞれが1に等しく設定される、請求項 93 に記載の方法。

【請求項 95】

前記一つまたは複数のCBFの値は、前記現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかに基づいており、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される前記現在のビデオ・ブロックのサンプルのために使用される、請求項 93 に記載の方法。

【請求項 96】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数( $Num_{pit\_idx}$ )は、K以上であり、

40

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、Kは正の整数である、

【請求項 97】

Kは、現在のパレット・サイズ(S)、エスケープ・フラグ(E)、または現在のビデオ・ブロックのサイズ( $B_{lks}$ )に基づく、請求項 96 に記載の方法。

50

## 【請求項 98】

$K = S + E$ である、請求項 97 に記載の方法。

## 【請求項 99】

$K$ が、パレット・インデックスの最大値 (MaxPaletteIndex) に 1 を加えたものに等しい、請求項 96 に記載の方法。

## 【請求項 100】

前記構文要素の 1 つが NumPltIdx -  $K$  を含む、請求項 96 に記載の方法。

## 【請求項 101】

(NumPltIdx -  $K$ ) の値のバイナリ化が、打ち切りされたバイナリ符号である、請求項 100 に記載の方法。

10

## 【請求項 102】

(NumPltIdx -  $K$ ) の値のバイナリ化が打ち切りされた単進符号である、請求項 100 に記載の方法。

## 【請求項 103】

(BlkS -  $K$ ) がバイナリ化入力パラメータであり、BlkS が現在のビデオ・ブロックのサイズである、請求項 101 または 102 に記載の方法。

## 【請求項 104】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ (QP) に基づく構文要素を使用して表現されるというフォーマット規則に適合し、

20

前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

## 【請求項 105】

無損失モードが適用されたこと、QP が閾値より大きいこと、または変換スキップが適用されたことを判別したとき、現在のブロックのためのパレットのサイズは、現在のビデオ・ブロックのサイズに等しいと推定される、請求項 104 に記載の方法。

30

## 【請求項 106】

前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット (DPS)、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、適応パラメータセット (APS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大符号化単位 (LCU)、符号化単位 (CU)、LCU 行、LCU のグループ、変換単位 (TU)、予測単位 (PU) ブロック、またはビデオ符号化単位において信号伝達されるメッセージ、前記ビデオのカラー・フォーマットの指示、符号化ツリー構造、時間的 ID レイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、請求項 105 のうちいずれか一項に記載の方法。

40

## 【請求項 107】

ビデオ処理の方法であって：

現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調 (BDPCM) モードで符号化され、複数の変換ブロックまたはサブブロックに分割されることを決定するステップと、

前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることとを実行するステップとを含む、

50

方法。

【請求項 108】

ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ (CG) パレット・モードを使用して符号化されるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各符号化単位 (CU) の複数のセグメントを表す、方法。

【請求項 109】

前記ビットストリーム表現は、各係数グループについてエスケープ・サンプルが存在するかどうかの指示を含み、エスケープ・サンプルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、請求項 108 に記載の方法。

【請求項 110】

前記ビットストリーム表現は、コンテキスト符号化されない上のインデックスをコピーの使用の指示を含む、請求項 108 に記載の方法。

【請求項 111】

前記指示がバイパス符号化される、請求項 110 に記載の方法。

【請求項 112】

一つまたは複数のコピー・フラグ、一つまたは複数のラン・タイプ、上のインデックスをコピーすること使用の一つまたは複数の指示、およびエスケープ値が、前記ビットストリーム表現において、インターリーブされた仕方で信号伝達される、請求項 108 に記載の方法。

【請求項 113】

前記ライン・ベースのCGパレット・モードは、前記現在のビデオ・ブロックのサイズが閾値 (Th) 以下であると判定されると、無効にされる、請求項 108 に記載の方法。

【請求項 114】

前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット (DPS)、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、適応パラメータセット (APS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大符号化単位 (LCU)、符号化単位 (CU)、LCU行、LCUのグループ、変換単位 (TU)、予測単位 (PU) ブロック、またはビデオ符号化単位において信号伝達されるメッセージ、前記ビデオのカラー・フォーマットの指示、符号化ツリー構造、時間的IDレイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、請求項 107 ないし 113 のうちいずれか一項に記載の方法。

【請求項 115】

前記変換を実行することが、前記一つまたは複数のビデオ領域から前記ビットストリーム表現を生成することを含む、請求項 1 ないし 114 のうちいずれか一項に記載の方法。

【請求項 116】

前記変換を実行することが、前記ビットストリーム表現から前記一つまたは複数のビデオ領域を生成することを含む、請求項 1 ないし 114 のうちいずれか一項に記載の方法。

【請求項 117】

プロセッサと、命令を有する非一時メモリとを備えるビデオ・システム内の装置であって、前記命令は、前記プロセッサによる実行の際に、前記プロセッサに、請求項 1 ないし 116 のうち一つまたは複数に記載の方法を実施させる、装置。

【請求項 118】

非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ・プログラム・プロダクトであって、請求項 1 ないし 116 のうち一つまたは複数に記載の方法を実行する

10

20

30

40

50

ためのプログラム・コードを含む、コンピュータ・プログラム・プロダクト。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願への相互参照

パリ条約による適用可能な特許法および/または規則のもと、この出願は、2019年8月15日に提出された国際特許出願第PCT/CN2019/100850号、2019年9月29日に提出された同PCT/CN2019/106700号、2019年9月24日に提出された同PCT/CN2019/107494号、2019年9月27日に提出された同PCT/CN2019/108736号、2019年10月1日に提出された同PCT/CN2019/109793号、2019年10月29日に提出された同PCT/CN2019/113931号および2020年1月9日に提出された同PCT/CN2020/071221の優先権および利益を適時に主張してなされる。法のもとのすべての目的のために、前述の出願の開示全体は、参照により本出願の開示の一部として援用される。

10

【0002】

技術分野

本稿は、ビデオおよび画像の符号化・復号技術に関する。

【背景技術】

【0003】

デジタル・ビデオは、インターネットおよび他のデジタル通信ネットワークにおける最大の帯域幅使用を占める。ビデオを受信および表示できる接続されたユーザー装置の数が増加するにつれて、デジタル・ビデオの利用のための帯域幅需要は増加し続けることが予想される。

20

【発明の概要】

【課題を解決するための手段】

【0004】

開示される技法は、パレット・モード符号化および復号においてパレット・エスケープ・シンボルのエントロピー符号化を実行するために、ビデオまたは画像のデコーダまたはエンコーダの実施形態によって使用されうる。

【0005】

ある例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

30

【0006】

別の例示的側面では、ビデオ処理の方法が開示される。本方法は、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化され、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モード符号化ツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

40

【0007】

さらに別の例では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロッ

50

クを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長符号化を使用し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

**【 0 0 0 8 】**

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

10

**【 0 0 0 9 】**

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

20

**【 0 0 1 0 】**

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずに符号化され、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

30

**【 0 0 1 1 】**

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化され、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

40

**【 0 0 1 2 】**

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの導出されるパレット・インデックス

50

は最大値をもち、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

【0013】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

10

【0014】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいて符号化される構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

20

【0015】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、予測的に符号化されるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

【0016】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングス符号化される構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

30

【0017】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

40

【0018】

さらに別の例示的側面では、ビデオ処理の方法が開示される。この方法は、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ

50

・ブロックの符号化モードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップと、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップとを含む。

【0019】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ(QP)差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

10

【0020】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数の符号化ブロック・フラグ(CBF)を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

20

【0021】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数(NumPltIdx)は、K以上であり、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、Kは正の整数である。

30

【0022】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ(QP)に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0023】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調(BDPCM)モードで符号化され、複数の変換ブロックまたはサブブロックに分割されることを決定するステップと、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることを実行するステップとを含む。

40

【0024】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ

50



・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ（CG）パレット・モードを使用して符号化されるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各符号化単位（CU）の複数のセグメントを表す。

【0025】

さらに別の例示的側面では、上述の方法は、プロセッサを備えるビデオ・エンコーダ装置によって実装されてもよい。

【0026】

さらに別の例示的側面では、上述の方法は、プロセッサを備えるビデオ・デコーダ装置によって実装されてもよい。

【0027】

さらに別の例示的側面では、これらの方法は、プロセッサ実行可能命令の形で具現され、コンピュータ読み取り可能なプログラム媒体に記憶されてもよい。

【0028】

これらおよび他の側面は、本稿においてさらに説明される。

【図面の簡単な説明】

【0029】

【図1】パレット・モードで符号化されるブロックの例を示す図である。

【0030】

【図2】パレット項目を信号伝達するためのパレット予測子の使用例を示す。

【0031】

【図3】水平および垂直横断走査の例を示す。

【0032】

【図4】パレット・インデックスの例示的な符号化を示す。

【0033】

【図5】AおよびBは、最小クロマ・イントラ予測単位（SCIPU）の例を示す。

【0034】

【図6】ビデオ処理におけるループ内フィルタリングの一例のブロック図を示す。

【0035】

【図7】ローカル・デュアル・ツリーの場合における反復されるパレット項目の例を示す。

【0036】

【図8】コンテキスト導出のプロセスにおける左および上のブロックの例を示す。

【0037】

【図9】本稿に記載される技法を実装するために使用されるハードウェア・プラットフォームの例のブロック図である。

【0038】

【図10】開示された技法が実装される例示的なビデオ処理システムのブロック図である。

【0039】

【図11】本開示のいくつかの実施形態によるビデオ符号化システムを示すブロック図である。

【0040】

【図12】本開示のいくつかの実施形態によるエンコーダを示すブロック図である。

【0041】

【図13】本開示のいくつかの実施形態によるデコーダを示すブロック図である。

【0042】

【図14】ビデオ処理の例示的方法のフローチャートを示す。

10

20

30

40

50

- 【図 1 5】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 1 6】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 1 7】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 1 8】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 1 9】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 0】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 1】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 2】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 3】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 4】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 5】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 6】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 7】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 8】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 2 9】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 3 0】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 3 1】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 3 2】ビデオ処理の例示的方法のフローチャートを示す。
- 【図 3 3】ビデオ処理の例示的方法のフローチャートを示す。
- 【発明を実施するための形態】

10

20

#### 【0043】

本稿は、圧縮解除または復号されたデジタル・ビデオまたは画像の品質を改善するために、画像またはビデオ・ビットストリームのデコードによって使用されることのできるさまざまな技法を提供する。簡潔のため、本明細書では、用語「ビデオ」は、ピクチャーのシーケンス（伝統的にビデオと呼ばれる）および個々の画像の両方を含むために使用される。さらに、ビデオ・エンコードも、さらなるエンコードのために使用されるデコードされたフレームを再構成するために、エンコードのプロセス中にこれらの技法を実装しうる。

#### 【0044】

セクション見出しは、理解の容易のために本稿で使用されており、実施形態および技法を対応するセクションに限定するものではない。よって、あるセクションからの実施形態は、他のセクションからの実施形態と組み合わせることができる。

30

#### 【0045】

##### 1. 概要

本稿は、ビデオ符号化技術に関する。具体的には、パレット符号化におけるインデックスおよびエスケープ・シンボル符号化に関する。これは、HEVCのような既存のビデオ符号化規格、またはこれから最終化される規格（多用途ビデオ符号化）に適用されてもよい。これは、将来のビデオ符号化規格またはビデオ・コーデックにも適用可能でありうる。

#### 【0046】

##### 2. 背景

ビデオ符号化規格は、主に、周知のITU-TおよびISO/IEC規格の開発を通じて発展してきた。ITU-TはH.261とH.263を生み出し、ISO/IECはMPEG-1とMPEG-4 Visualを生み出し、両組織は合同して、H.262/MPEG-2 VideoとH.264/MPEG-4 Advanced Video Coding (AVC)とH.265/HEVC規格とを生み出した。H.262以来、ビデオ符号化規格は、時間的予測と変換符号化が利用されるハイブリッドビデオ符号化構造に基づいている。HEVCを越えた将来のビデオ符号化技術を探求するため、2015年にVCEGとMPEGが合同で合同ビデオ探査チーム（JVET）を設立した。それ以来、JVETによって多くの新しい方法が採用され、合同探査モデル（Joint Exploration Model、JEM）と名付けられた参照ソフトウェアに入れられた。2018年4月には、VCEG（Q6/16）とISO/IEC JTC1 SC29/WG11（MPEG）の間の合同ビデオエキスパートチ

40

50

ーム (JVET) が発足し、HEVCに比べ50%のビットレート削減を目指すVVC規格に取り組んでいる。

【0047】

VVCドラフトの最新バージョン、すなわち、Versatile Video Coding (Draft 6) は下記の場所に見出せる：

[http://phenix.it-sudparis.eu/jvet/doc\\_end\\_user/documents/15\\_Gothenburg/wg11/JVET-O2001-v14.zip](http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v14.zip)

【0048】

VVCの最新の参照ソフトウェア (VTM) は、下記の場所に見出せる：

[https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/tags/VTM-5.0](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-5.0)

10

【0049】

## 2.1 HEVCスクリーン・コンテンツ符号化拡張 (HEVC-SCC) におけるパレット・モード

### 2.1.1 パレット・モードの概念

【0050】

パレット・モードの背後にある基本的な発想は、CU内のピクセルが代表的な色値の小さな集合によって表現されるということである。この集合はパレットと呼ばれる。エスケープ・シンボルおよびそれに続く (可能性としては量子化された) 成分値を信号伝達することによって、パレットの外側にあるサンプルを示すことも可能である。この種のピクセルはエスケープ・ピクセルと呼ばれる。パレット・モードは図1に示されている。図1に示されるように、3つの色成分 (ルーマおよび2つのクロマ成分) を有する各ピクセルについて、パレットへのインデックスが設立され、ブロックは、パレットにおける設立された値に基づいて再構成されることができる。

20

【0051】

### 2.1.2 パレット項目の符号化

パレット項目の符号化のために、パレット予測子が維持される。パレットの最大サイズとパレット予測子は、SPSにおいて信号伝達される。HEVC-SCCでは、palette\_predictor\_initializer\_present\_flagがPPSにおいて導入される。このフラグが1である場合、パレット予測子を初期化するための項目が、ビットストリームにおいて信号伝達される。パレット予測子は、各CTU行、各スライスおよび各タイルの先頭において初期化される。palette\_predictor\_initializer\_present\_flagの値に依存して、パレット予測子は0にリセットされるか、PPSにおいて信号伝達されたパレット予測子初期化子項目 (palette predictor initializer entries) を使用して初期化される。HEVC-SCCでは、サイズ0のパレット予測子初期化子により、PPSレベルでパレット予測子初期化の明示的な無効化を許容することができるようになった。

30

【0052】

パレット予測子における各項目について、再利用フラグは、それが現在のパレットの一部であるかどうかを示すために信号伝達される。これは図2に示される。再利用フラグは、ゼロのランレングス符号化を使用して送信される。その後、新しいパレット項目の数、次数0の指数ゴロム (Exponential Golomb, EG) 符号、すなわちEG-0を使用して信号伝達される。最後に、新しいパレット項目のための成分値が信号伝達される。

40

【0053】

### 2.1.3 パレット・インデックスの符号化

パレット・インデックスは、図3に示されるように水平および垂直横断走査を用いて符号化される。走査順序は、palette\_transpose\_flagを使用してビットストリームにおいて明示的に信号伝達される。サブセクションの残りについては、走査は水平であると想定される。

【0054】

パレット・インデックスは、'COPY\_LEFT' [左をコピー] と 'COPY\_ABOVE' [上をコピー] という2つのパレット・サンプル・モードを使用して符号化される。'COPY\_LEFT'モードでは、パレット・インデックスはデコードされたインデックスに割り当てられ

50

る。'COPY\_ABOVE'モードでは、上の行のサンプルのパレット・インデックスがコピーされる。'COPY\_LEFT'モードと'COPY\_ABOVE'モードの両方について、やはり同じモードで符号化される後続のサンプルの数を指定するラン値が信号伝達される。

【0055】

パレット・モードでは、エスケープ・シンボルについてのインデックスの値は、パレット項目の数である。また、エスケープ・シンボルが'COPY\_LEFT'または'COPY\_ABOVE'モードにおけるランの一部である場合、エスケープ成分値は、各エスケープ・シンボルについて信号伝達される。パレット・インデックスの符号化が図4に示される。

【0056】

この構文順序 (syntax order) は、次のように達成される。まず、CUについてのインデックス値の数が信号伝達される。これに続いて、打ち切りされたバイナリ符号化を使用して、CU全体についての実際のインデックス値が信号伝達される。インデックスの数とインデックス値の両方がバイパス・モードで符号化される。これは、インデックスに関連したバイパス・ピンをグループ化する。次いで、パレット・サンプル・モード (必要な場合) およびランが、インターリーブされた仕方でも信号伝達される。最後に、CU全体についての諸エスケープ・シンボルのための成分エスケープ値と一緒にグループ化され、バイパス・モードで符号化される。エスケープ・シンボルのバイナリ化は、3次のEG符号化、すなわちEG-3である。

10

【0057】

追加的な構文要素last\_run\_type\_flagが、インデックス値を信号伝達した後に信号伝達される。この構文要素は、インデックスの数とともに、ブロックにおける最後のランに対応するラン値を信号伝達する必要をなくす。

20

【0058】

HEVC-SCCでは、パレット・モードは4:2:2、4:2:0、およびモノクロ・クロマ・フォーマットについても有効にされる。パレット項目とパレット・インデックスの信号伝達は、すべてのクロマ・フォーマットについてほぼ同一である。非モノクロ・フォーマットの場合、各パレット項目は3つの成分で構成される。モノクロ・フォーマットの場合、各パレット項目は単一の成分で構成される。サブサンプリングされたクロマ方向については、クロマ・サンプルは2で割れるルーマ・サンプル・インデックスと関連付けられる。CUのためのパレット・インデックスを再構成した後、サンプルが、それに関連する単一の成分をもつだけである場合、パレット項目の最初の成分のみが使用される。信号伝達における唯一の違いは、エスケープ成分値についてである。各エスケープ・シンボルについて、信号伝達されるエスケープ成分値の数は、そのシンボルに関連する成分の数に依存して異なる可能性がある。

30

【0059】

さらに、パレット・インデックス符号化にはインデックス調整プロセスがある。パレット・インデックスを信号伝達するとき、左近傍インデックスまたは上近傍インデックスは、現在インデックスとは異なるはずである。よって、1つの可能性を除去することにより、現在のパレット・インデックスの範囲を1だけ減らすことができる。その後、インデックスは、打ち切りされたバイナリ (truncated binary、TB) バイナリ化で信号伝達される。

40

【0060】

この部分に関連するテキストは次のように示される。ここで、CurrPaletteIndexは現在のパレット・インデックスであり、adjustedRefPaletteIndexは予測インデックスである。

【0061】

変数PaletteIndexMap[xC][yC]は、パレット・インデックスを指定する。これは、CurrentPaletteEntriesによって表現される配列へのインデックスである。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置 (xC,yC) を指定する。Palette

50

IndexMap[xC][yC]の値は、0からMaxPaletteIndexの範囲である（両端含む）。変数adjustedRefPaletteIndexは次のように導出される：

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if ( PaletteScanPos == 0 ) {
    xcPrev = x0 + TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos - 1][0]
    ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos - 1][1]
    if ( CopyAboveIndicesFlag[xcPrev][ycPrev]==0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[xcPrev][ycPrev]{
(7-157)
    }
    else{
        if ( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[xC][yC - 1]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[xC - 1][yC]
    }
}

```

【 0 0 6 2 】

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++

```

【 0 0 6 3 】

さらに、パレット・モードにおけるランレングス要素はコンテキスト符号化される。J VET-O2011-vEにおいて記載される関連するコンテキスト導出プロセスは、以下のように示される。

構文要素palette\_run\_prefixについてのctxIncの導出プロセス

このプロセスへの入力は、ピン・インデックスbinIdxならびに構文要素copy\_above\_palette\_indices\_flagおよびpalette\_idx\_idcである。

このプロセスの出力は変数ctxIncである。

変数ctxIncは次のように導出される：

・copy\_above\_palette\_indices\_flagが0に等しく、binIdxが0に等しい場合、ctxIncは以下のように導出される：

$$ctxInc = ( palette\_idx\_idc == 1 ) ? 0 : ( ( palette\_idx\_idc == 3 ) ? 1 : 2 ) \quad (9-69)$$

・それ以外の場合、ctxIncは表1によって与えられる：

【表 1】

表1 -ctxIdxMap[ copy\_above\_palette\_indices\_flag ][ binIdx ]の指定

binIdx	0	1	2	3	4	> 4
copy_above_palette_indices_flag == 1	5	6	6	7	7	バイパス
copy_above_palette_indices_flag == 0	0, 1, 2	3	3	4	4	バイパス

【 0 0 6 4 】

2.2 VVCにおけるパレット・モード

2.2.1 デュアル・ツリーにおけるパレット

VVCでは、デュアル・ツリー符号化構造がイントラスライスの符号化に使用されるため、ルーマ成分と2つのクロマ成分は、異なるパレットおよびパレット・インデックスをもつことがある。さらに、2つのクロマ成分は、同じパレットおよびパレット・インデッ

10

20

30

40

50

クスを共有する。

【0065】

### 2.2.2 別個のモードとしてのパレット

JVET-N0258および現在のVTMでは、符号化単位についての予測モードはMODE\_INTRA、MODE\_INTER、MODE\_IBC、MODE\_PLTとすることができる。予測モードのバイナリ化はそれに応じて変更される。

【0066】

IBCがオフにされるときは、1タイル上で、最初の1ピンが、現在の予測モードがMODE\_PLTであるか否かを示すために使用される。P/Bタイル上では、最初のピンは現在の予測モードがMODE\_INTRAであるか否かを示すために使用される。そうでない場合、現在の予測モードがMODE\_PLTまたはMODE\_INTERであることを示すために、1つの追加的なピンが使用される。IBCがオンにされるときは、1タイル上で、最初のピンが、現在の予測モードがMODE\_IBCであるか否かを示すために使用される。もしそうでなければ、現在の予測モードがMODE\_PLTかMODE\_INTRAかを示すために第2のピンが使われる。P/Bタイル上では、最初のピンは現在の予測モードがMODE\_INTRAであるか否かを示すために使用される。イントラモードである場合、現在の予測モードがMODE\_PLTまたはMODE\_INTRAであることを示すために第2のピンが使用される。もしそうでなければ、現在の予測モードがMODE\_IBCまたはMODE\_INTERであることを示すために第2のピンが使用される。

10

【0067】

JVET-O2001-vEにおける関連するテキストを以下に示す。

20

【0068】

符号化単位構文

30

40

50

【表 2】

coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {	記述子
chType = treeType == DUAL_TREE_CHROMA? 1 : 0	
if( slice_type != I    sps_abc_enabled_flag    sps_palette_enabled_flag ) {	
if( treeType != DUAL_TREE_CHROMA && !( ( cbWidth == 4 && cbHeight == 4 )    modeType == MODE_TYPE_INTRA ) && !sps_abc_enabled_flag )	10
<b>cu_skip_flag[ x0 ][ y0 ]</b>	ae(v)
if( cu_skip_flag[ x0 ][ y0 ] == 0 && slice_type != I && !( cbWidth == 4 && cbHeight == 4 ) && modeType == MODE_TYPE_ALL )	
<b>pred_mode_flag</b>	ae(v)
if( ( ( slice_type == I && cu_skip_flag[ x0 ][ y0 ] == 0 )    ( slice_type != I && ( CuPredMode[ chType ][ x0 ][ y0 ] != MODE_INTRA    ( cbWidth == 4 && cbHeight == 4 && cu_skip_flag[ x0 ][ y0 ] == 0 ) ) ) ) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_abc_enabled_flag && treeType != DUAL_TREE_CHROMA )	20
<b>pred_mode_abc_flag</b>	ae(v)
if( ( ( slice_type == I    ( cbWidth == 4 && cbHeight == 4 )    sps_abc_enabled_flag ) && CuPredMode[ x0 ][ y0 ] == MODE_INTRA )    ( slice_type != I && !( cbWidth == 4 && cbHeight == 4 ) && !sps_abc_enabled_flag && CuPredMode[ x0 ][ y0 ] != MODE_INTRA ) ) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[ x0 ][ y0 ] == 0 && modeType != MODE_TYPE_INTER )	30
<b>pred_mode_plt_flag</b>	ae(v)
}	
...	
}	40

## 【 0 0 6 9 】

## 2.2.3 パレット・モード構文

【表 3】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
palettePredictionFinished = 0		
NumPredictedPaletteEntries = 0		
for( predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[ startComp ] && !palettePredictionFinished && NumPredictedPaletteEntries[ startComp ] < palette_max_size; predictorEntryIdx++ ) {		10
<b>palette_predictor_run</b>	ae(v)	
if( palette_predictor_run != 1 ) {		
if( palette_predictor_run > 1 )		
predictorEntryIdx += palette_predictor_run - 1		
PalettePredictorEntryReuseFlags[ predictorEntryIdx ] = 1		
NumPredictedPaletteEntries++		
} else		20
palettePredictionFinished = 1		
}		
if( NumPredictedPaletteEntries < palette_max_size )		
<b>num_signalled_palette_entries</b>	ae(v)	
for( cIdx = startComp; cIdx < ( startComp + numComps); cIdx++ )		
for( i = 0; i < num_signalled_palette_entries; i++ )		
<b>new_palette_entries[ cIdx ][ i ]</b>	ae(v)	30
if( CurrentPaletteSize[ startComp ] > 0 )		
<b>palette_escape_val_present_flag</b>	ae(v)	
if( MaxPaletteIndex > 0 ) {		
<b>num_palette_indices_minus1</b>	ae(v)	
adjust = 0		
for( i = 0; i <= num_palette_indices_minus1; i++ ) {		
if( MaxPaletteIndex - adjust > 0 ) {		40



<b>palette_idx_idc</b>	ae(v)	
PaletteIndexIdc[ i ] = palette_idx_idc		
}		
adjust = 1		
}		
<b>copy_above_indices_for_final_run_flag</b>	ae(v)	
<b>palette_transpose_flag</b>	ae(v)	10
}		
if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {		
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {		
<b>cu_qp_delta_abs</b>	ae(v)	
if( cu_qp_delta_abs )		
<b>cu_qp_delta_sign_flag</b>	ae(v)	
}		20
}		
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {		
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {		
<b>cu_chroma_qp_offset_flag</b>	ae(v)	
if( cu_chroma_qp_offset_flag )		
<b>cu_chroma_qp_offset_idx</b>	ae(v)	
}		
}		30
remainingNumIndices = num_palette_indices_minus1 + 1		
PaletteScanPos = 0		
log2CbWidth = Log2( cbWidth )		
log2CbHeight = Log2( cbHeight )		
while( PaletteScanPos < cbWidth*cbHeight ) {		
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]		40

10

20

30

40

50

if( PaletteScanPos > 0 ) {		
xcPrev	=	
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev	=	
y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
PaletteRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1		10
RunToEnd = 1		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
if( MaxPaletteIndex > 0 )		
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC > 0 ) )		
&& CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 )		
if( remainingNumIndices > 0 && PaletteScanPos < cbWidth * cbHeight - 1 ) {		
<b>copy_above_palette_indices_flag</b>	ae(v)	
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag		20
} else {		
if( PaletteScanPos == cbWidth * cbHeight - 1 && remainingNumIndices > 0 )		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
else		
CopyAboveIndicesFlag[ xC ][ yC ] = 1		
}		
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {		30
currNumIndices = num_palette_indices_minus1 + 1 - remainingNumIndices		
PaletteIndexMap[ xC ][ yC ] = PaletteIndexIdx[ currNumIndices ]		
}		
if( MaxPaletteIndex > 0 ) {		
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 )		
remainingNumIndices - = 1		
if( remainingNumIndices > 0    CopyAboveIndicesFlag[ xC ][ yC ] !=		
copy_above_indices_for_final_run_flag ) {		40

PaletteMaxRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1 - remainingNumIndices - copy_above_indices_for_final_run_flag	
RunToEnd = 0	
if( PaletteMaxRunMinus1 > 0 ) {	
<b>palette_run_prefix</b>	ae(v)
if( ( palette_run_prefix > 1 ) && ( PaletteMaxRunMinus1 != ( 1 << ( palette_run_prefix - 1 ) ) ) )	
<b>palette_run_suffix</b>	ae(v)
}	
}	
}	
runPos = 0	
while ( runPos <= PaletteRunMinus1 ) {	
xR = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
yR = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	
if( CopyAboveIndicesFlag[ xR ][ yR ] == 0 ) {	
CopyAboveIndicesFlag[ xR ][ yR ] = 0	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xC ][ yC ]	
} else {	
CopyAboveIndicesFlag[ xR ][ yR ] = 1	
if ( !palette_transpose_flag )	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xR ][ yR - 1 ]	
else	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xR - 1 ][ yR ]	
}	
runPos++	
PaletteScanPos ++	
}	
}	

10

20

30

40

50

if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )	
for( sPos = 0; sPos < cbWidth* cbHeight; sPos++ ) {	
xC = x0 + TraverseScanOrder[ log2CbWidth][ log2CbHeight ][ sPos ][ 0 ]	
yC = y0 + TraverseScanOrder[ log2CbWidth][ log2CbHeight ][ sPos ][ 1 ]	
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {	
<b>palette_escape_val</b>	ae(v)
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val	
}	
}	
}	
}	

10

20

## 【 0 0 7 0 】

## 2.2.4 パレット・モードのセマンティクス

以下のセマンティクスでは、配列インデックスx0、y0は、ピクチャーの左上のルーマ・サンプルに対する、考慮される符号化ブロックの左上のルーマ・サンプルの位置(x0, y0)を指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置(xC, yC)を指定する。配列インデックスstartCompは、現在のパレット・テーブルの最初の色成分を指定する。0に等しいstartCompはY成分を示し、1に等しいstartCompはCb成分を示し、2に等しいstartCompはCr成分を示す。numCompsは、現在のパレット・テーブルにおける色成分の数を指定する。

## 【 0 0 7 1 】

予測子パレットは、現在のパレット内の項目を予測するために使用される、以前の符号化単位からのパレット項目で構成される。

変数PredictorPaletteSize[startComp]は、現在のパレット・テーブルの最初の色成分startCompについての予測子パレットのサイズを指定する。PredictorPaletteSizeは、節8.4.5.3において規定されるように導出される。

変数PalettePredictorEntryReuseFlags[i]が1に等しいことは、予測子パレットにおけるi番目の項目が現在のパレットにおいて再利用されることを指定する。PalettePredictorEntryReuseFlags[i]が0に等しいことは、予測子パレットのi番目の項目が現在のパレットにおける項目ではないことを指定する。配列PalettePredictorEntryReuseFlags[i]のすべての要素が0に初期化される。

30

40

## 【 0 0 7 2 】

palette\_predictor\_runは、配列PalettePredictorEntryReuseFlagsにおけるゼロでない項目の前にあるゼロの数を決定するために使用される。

palette\_predictor\_runの値は、0から(PredictorPaletteSize - predictorEntryIdx)の範囲内(両端含む)であることがビットストリーム適合性の要件である。ここで、predictorEntryIdxは、配列PalettePredictorEntryReuseFlagsにおける現在位置に対応する。変数NumPredictedPaletteEntriesは、予測子パレットから再利用される現在のパレットにおける項目数を指定する。NumPredictedPaletteEntriesの値は、0からpalette\_max\_sizeの範囲内(両端含む)である。

## 【 0 0 7 3 】

50

num\_signalled\_palette\_entriesは、現在のパレット・テーブルの最初の色成分startCompについて明示的に信号伝達される、現在のパレットにおける項目の数を指定する。num\_signalled\_palette\_entriesが存在しない場合、0に等しいと推定される。

変数CurrentPaletteSize[startComp]は、現在のパレット・テーブルの最初の色成分startCompについての現在のパレットのサイズを指定し、次のように導出される：

```
CurrentPaletteSize[startComp] = NumPredictedPaletteEntries + num_signalled_palette_entries (7-155)
```

CurrentPaletteSize[startComp]の値は、0からpalette\_max\_sizeの範囲内（両端含む）である。

【0074】

new\_palette\_entries[cIdx][i]は、色成分cIdxについてのi番目の信号伝達されるパレット項目についての値を指定する。

変数PredictorPaletteEntries[cIdx][i]は、色成分cIdxについての予測子パレットにおけるi番目の要素を指定する。

変数CurrentPaletteEntries[cIdx][i]は、色成分cIdxについての現在のパレットにおけるi番目の要素を指定し、次のように導出される：

【表4】

```
numPredictedPaletteEntries = 0
```

```
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
```

```
    if( PalettePredictorEntryReuseFlags[ i ] ) {
```

```
        for( cIdx =startComp; cIdx < ( startComp + numComps ); cIdx++ )
```

```
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
```

```
PredictorPaletteEntries[ cIdx ][ i ]
```

```
            numPredictedPaletteEntries++
```

```
        }
```

```
for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ ) (7-156)
```

```
    for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
```

```
        CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = new_palette_entries[ cIdx ][ i ]
```

【0075】

palette\_escape\_val\_present\_flagが1に等しいことは、現在の符号化単位が少なくとも1つのエスケープ符号化サンプルを含むことを指定する。escape\_val\_present\_flagが0に等しいことは、現在の符号化単位においてエスケープ符号化されたサンプルがないことを指定する。存在しない場合、palette\_escape\_val\_present\_flagの値は1に等しいと推定される。

変数MaxPaletteIndexは、現在の符号化単位についてのパレット・インデックスの可能な最大値を指定する。MaxPaletteIndexの値は

CurrentPaletteSize[startComp] - 1 + palette\_escape\_val\_present\_flagに等しく設定される。

【0076】

num\_palette\_indices\_minus1に1を加えたものは、現在のブロックについて明示的に信号伝達されるまたは推定されるパレット・インデックスの数である。num\_palette\_indices\_minus1が存在しない場合、0に等しいと推定される。

palette\_idx\_idcは、パレット・テーブル、CurrentPaletteEntriesへのインデック

10

20

30

40

50

スの指示である。palette\_idx\_idcの値は、ブロック内の最初のインデックスについては0からMaxPaletteIndexの範囲内（両端含む）であり、ブロック内の残りのインデックスについては0から（MaxPaletteIndex - 1）の範囲内（両端含む）である。

palette\_idx\_idcが存在しない場合、0に等しいと推定される。

変数PaletteIndexIdc[i]は、明示的に信号伝達されるまたは推定されるi番目のpalette\_idx\_idcを格納する。配列PaletteIndexIdc[i]のすべての要素は0に初期化される。

【0077】

copy\_above\_indices\_for\_final\_run\_flagが1に等しいことは、符号化単位の最後の諸位置のパレット・インデックスが、水平横断走査が使用される場合は上の行におけるパレット・インデックスから、垂直横断走査が使用される場合は左の列におけるパレット・インデックスから、コピーされることを指定する。 10

copy\_above\_indices\_for\_final\_run\_flagが0に等しいことは、符号化単位における最後の諸位置のパレット・インデックスがPaletteIndexIdc[num\_palette\_indices\_minus1]からコピーされることを指定する。

copy\_above\_indices\_for\_final\_run\_flagが存在しない場合、0に等しいと推定される。

【0078】

palette\_transpose\_flagが1に等しいことは、現在の符号化単位内のサンプルについてのインデックスを走査するために垂直横断走査が適用されることを指定する。palette\_transpose\_flagが0に等しいことは、現在の符号化単位内のサンプルについてのインデックスを走査するために水平横断走査が適用されることを指定する。存在しない場合、palette\_transpose\_flagの値は0に等しいと推定される。 20

配列TraverseScanOrderは、パレット符号化のための走査順序配列を指定する。palette\_transpose\_flagが0に等しい場合、TraverseScanOrderは水平走査順序HorTravScanOrderを割り当てられ、palette\_transpose\_flagが1に等しい場合、TraverseScanOrderは垂直走査順序VerTravScanOrderを割り当てられる。

【0079】

copy\_above\_palette\_indices\_flagが1に等しいことは、パレット・インデックスが、水平横断走査が使用される場合には、上の行における同じ位置、または垂直横断走査が使用される場合には、左の列における同じ位置にあるパレット・インデックスに等しいことを指定する。copy\_above\_palette\_indices\_flagが0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいて符号化されるか、または推定されることを指定する。 30

【0080】

変数CopyAboveIndicesFlag[xC][yC]が1に等しいことは、パレット・インデックスが上の行（水平走査）または左の列（垂直走査）におけるパレット・インデックスからコピーされることを指定する。

CopyAboveIndicesFlag[xC][yC]が0に等しいことは、パレット・インデックスがビットストリームにおいて明示的に符号化されるか、推定されることを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置（xC,yC）を指定する。PaletteIndexMap[xC][yC]の値は、0から（MaxPaletteIndex - 1）の範囲内（両端含む）とする。 40

【0081】

変数PaletteIndexMap[xC][yC]は、CurrentPaletteEntriesによって表される配列へのインデックスであるパレット・インデックスを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置（xC,yC）を指定する。PaletteIndexMap[xC][yC]の値は、0からMaxPaletteIndexの範囲内（両端含む）である。

【0082】

変数adjustedRefPaletteIndexは次のように導出される：

【表 5】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++ (7-158)

```

## 【0083】

palette\_run\_prefixは、存在する場合、PaletteRunMinus1のバイナリ化におけるプレフィックス部分を指定する。

## 【0084】

palette\_run\_suffixは、変数PaletteRunMinus1の導出において使用される。存在しない場合、palette\_run\_suffixの値は0に等しいと推定される。

## 【0085】

RunToEndが0に等しい場合、変数PaletteRunMinus1は次のように導出される：

- ・ PaletteMaxRunMinus1が0に等しい場合、PaletteRunMinus1は0に等しく設定される。

- ・ それ以外の場合（PaletteMaxRunMinus1が0より大きい場合）、次が適用される：

- ・ palette\_run\_prefixが2より小さい場合、次が適用される：
 

```

PaletteRunMinus1 = palette_run_prefix (7-159)

```

- ・ それ以外の場合（palette\_run\_prefixが2以上）、次が適用される：

```

PrefixOffset = 1 (palette_run_prefix - 1)
PaletteRunMinus1 = PrefixOffset + palette_run_suffix (7-160)

```

## 【0086】

変数PaletteRunMinus1は次のように使用される：

- ・ CopyAboveIndicesFlag[xC][yC]が0に等しい場合、PaletteRunMinus1は、同じパレット・インデックスをもつ連続する位置の数から1を引いた数を指定する。

- ・ それ以外の場合、palette\_transpose\_flagが0に等しい場合、PaletteRunMinus1は、上の行における対応する位置で使用されるのと同じパレット・インデックスをもつ、連続する位置の数から1を引いた数を指定する。

- ・ それ以外の場合、PaletteRunMinus1は、左の列における対応する位置で使用される

のと同じパレット・インデックスをもつ、連続する位置の数から1を引いた数を指定する。

RunToEndが0に等しい場合、変数PaletteMaxRunMinus1はPaletteRunMinus1について可能な最大値を表し、PaletteMaxRunMinus1の値が0以上であることはビットストリーム適合性の要件である。

【0087】

palette\_escape\_valは、ある成分についての量子化されたエスケープ符号化されたサンプル値を指定する。

変数PaletteEscapeVal[cldx][xC][yC]は、PaletteIndexMap[xC][yC]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しいサンプルのエスケープ値を指定する。配列インデックスcldxは、色成分を指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。

PaletteEscapeVal[cldx][xC][yC]が、0に等しいcldxについては0から(1 - (BitDepthY + 1) - 1)の範囲内(両端含む)であり、0に等しくないcldxについては0から(1 - ((BitDepthC + 1)) - 1)の範囲内(両端含む)であることがビットストリーム適合性の要件である。

【0088】

#### 1.1.1 ライン・ベースのCGパレット・モード

ライン・ベースのCGパレット・モードがVVCに採用された。この方法では、パレット・モードの各CUは、横断走査モードに基づいて複数のサンプル・セグメント(この試験では)に分割される。各セグメントにおけるパレット・ラン符号化のためのエンコード順序は、次のとおり: 各ピクセルについて、1つのコンテキスト符号化されたピンrun\_copy\_flag = 0が信号伝達されて、ピクセルが前のピクセルと同じモードであるかどうか、すなわち前の走査されたピクセルと現在のピクセルが両方ともラン・タイプCOPY\_ABOVEであるかどうか、または前の走査されたピクセルと現在のピクセルが両方ともラン・タイプINDEXであり、同じインデックス値であるかどうかを示す。そうでない場合、run\_copy\_flag = 1が信号伝達される。ピクセルと前のピクセルが異なるモードである場合、1つのコンテキスト符号化されたピンcopy\_above\_palette\_indices\_flagが信号伝達されて、ピクセルのラン・タイプ、すなわち、INDEXまたはCOPY\_ABOVEを示す。VTM 6.0におけるパレット・モードと同じく、サンプルが最初の行(水平横断走査)または最初の列(垂直横断走査)にある場合、デフォルトでINDEXモードが使用されるため、デコーダはラン・タイプをパースする必要はない。また、前にパースされたラン・タイプがCOPY\_ABOVEである場合も、デコーダはラン・タイプをパースする必要はない。1つのセグメントにおけるピクセルのパレット・ラン符号化の後、インデックス値(INDEXモードについて)および量子化されたエスケープ色は、各ラインCG内のスループットを改善するために、コンテキスト符号化されたピンのエンコード/パースとは別に、バイパス符号化およびグループ化される。インデックス値は、VTMのようにパレット・ラン符号化の前に処理されるのではなく、今やラン符号化の後に符号化/パースされるので、エンコーダは、インデックス値の数num\_palette\_indices\_minus1と最後のラン・タイプcopy\_above\_indices\_for\_final\_run\_flagを信号伝達する必要はない。

JVET-P0077におけるライン・ベースのCGパレット・モードのテキストを以下に示す。

【0089】

#### パレット符号化構文

10

20

30

40



【表 6】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
palettePredictionFinished = 0		
NumPredictedPaletteEntries = 0		
for( predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[ startComp ] && !palettePredictionFinished && NumPredictedPaletteEntries[ startComp ] < palette_max_size; predictorEntryIdx++ ) {		10
<b>palette_predictor_run</b>	ae(v)	
if( palette_predictor_run != 1 ) {		
if( palette_predictor_run > 1 )		
predictorEntryIdx += palette_predictor_run - 1		
PalettePredictorEntryReuseFlags[ predictorEntryIdx ] = 1		
NumPredictedPaletteEntries++		
} else		20
palettePredictionFinished = 1		
}		
if( NumPredictedPaletteEntries < palette_max_size )		
<b>num_signalled_palette_entries</b>	ae(v)	
for( cIdx = startComp; cIdx < ( startComp + numComps); cIdx++ )		
for( i = 0; i < num_signalled_palette_entries; i++ )		
<b>new_palette_entries</b> [ cIdx ][ i ]	ae(v)	30
if( CurrentPaletteSize[ startComp ] > 0 )		
<b>palette_escape_val_present_flag</b>	ae(v)	
if( MaxPaletteIndex > 0 ) {		
adjust = 0		
<b>palette_transpose_flag</b>	ae(v)	
}		40

if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {	
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {	
<b>cu_qp_delta_abs</b>	ae(v)
if( cu_qp_delta_abs )	
<b>cu_qp_delta_sign_flag</b>	ae(v)
}	
}	
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {	
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {	
<b>cu_chroma_qp_offset_flag</b>	ae(v)
if( cu_chroma_qp_offset_flag )	
<b>cu_chroma_qp_offset_idx</b>	ae(v)
}	
}	
PreviousRunTypePosition = 0	
PreviousRunType = 0	
for( subSetId = 0; subSetId <= (cbWidth* cbHeight - 1) >> 4; subSetId++) {	
minSubPos = subSetId << 4	
if( minSubPos + 16 > cbWidth * cbHeight)	
maxSubPos = cbWidth * cbHeight	
else	
maxSubPos = minSubPos + 16	
RunCopyMap[ 0 ][ 0 ] = 0	
log2CbWidth = Log2( cbWidth )	
log2CbHeight = Log2( cbHeight )	
PaletteScanPos = minSubPos	
while( PaletteScanPos < maxSubPos ) {	
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	

10

20

30

40

50

if( PaletteScanPos > 0 ) {		
xcPrev =		
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev =		
y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
if( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {		10
<b>run_copy_flag</b>	ae(v)	
RunCopyMap[ xC ][ yC ] = run_copy_flag		
}		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
if( MaxPaletteIndex > 0 && ! RunCopyMap[startComp][xC][yC] ) {		
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC >		
0 ) )		20
&& CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {		
<b>copy_above_palette_indices_flag</b>	ae(v)	
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag		
}		
PreviousRunType = CopyAboveIndicesFlag[ xC ][ yC ]		
PreviousRunTypePosition = curPos		
} else {		
CopyAboveIndicesFlag[ xC ][ yC ] = CopyAboveIndicesFlag[xcPrev][ ycPrev]		30
}		
}		
PaletteScanPos ++		
}		
PaletteScanPos = minSubPos		
while( PaletteScanPos < maxSubPos ) {		
xC =		
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		40

<pre> yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ] </pre>		
<pre> if( PaletteScanPos &gt; 0 ) { </pre>		
<pre>     xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ] </pre>		
<pre>     ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ] </pre>		10
<pre> } </pre>		
<pre> if( MaxPaletteIndex &gt; 0 ) { </pre>		
<pre>     if( ! RunCopyMap [ xC ][ yC ] &amp;&amp; CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>         if( MaxPaletteIndex - adjust &gt; 0 ) { </pre>		
<pre>             <b>palette_idx_idc</b> </pre>	ae(v)	
<pre>         } </pre>		
<pre>         adjust = 1 </pre>		
<pre>     } </pre>		20
<pre> } </pre>		
<pre> if( ! RunCopyMap [ xC ][ yC ] &amp;&amp; CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>     CurrPaletteIndex = palette_idx_idc </pre>		
<pre>     if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>         PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex </pre>		
<pre>     } else { </pre>		
<pre>         if( !palette_transpose_flag ) </pre>		
<pre>             PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ] </pre>		30
<pre>         else </pre>		
<pre>             PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ] </pre>		
<pre>     } </pre>		
<pre> } </pre>		
<pre> if( palette_escape_val_present_flag ) { </pre>		
<pre>     for( cIdx = startComp; cIdx &lt; ( startComp + numComps ); cIdx++ ) </pre>		
<pre>         for( sPos = minSubPos; sPos &lt; maxSubPos; sPos++ ) { </pre>		40

$xC = x0 + \text{TraverseScanOrder}[\log_2CbWidth][\log_2CbHeight][sPos][0]$	
$yC = y0 + \text{TraverseScanOrder}[\log_2CbWidth][\log_2CbHeight][sPos][1]$	
$\text{if}(\text{PaletteIndexMap}[cIdx][xC][yC] == \text{MaxPaletteIndex}) \{$	
<b>palette_escape_val</b>	ae(v)
$\text{PaletteEscapeVal}[cIdx][xC][yC] = \text{palette\_escape\_val}$	
$\}$	
$\}$	
$\}$	
$\}$	
$\}$	

10

## 【0090】

## 7.4.9.6 パレット符号化のセマンティクス

以下のセマンティクスでは、配列インデックス $x0$ 、 $y0$ は、ピクチャーの左上のルーマ・サンプルに対する、考慮される符号化ブロックの左上のルーマ・サンプルの位置( $x0, y0$ )を指定する。配列インデックス $xC$ 、 $yC$ は、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置( $xC, yC$ )を指定する。配列インデックス $startComp$ は、現在のパレット・テーブルの最初の色成分を指定する。0に等しい $startComp$ はY成分を示し、1に等しい $startComp$ はCb成分を示し、2に等しい $startComp$ はCr成分を示す。 $numComps$ は、現在のパレット・テーブルにおける色成分の数を指定する。

20

予測子パレットは、現在のパレット内の項目を予測するために使用される、以前の符号化単位からのパレット項目で構成される。

変数 $\text{PredictorPaletteSize}[startComp]$ は、現在のパレット・テーブルの最初の色成分 $startComp$ についての予測子パレットのサイズを指定する。 $\text{PredictorPaletteSize}$ は、節8.4.5.3において規定されるように導出される。

30

変数 $\text{PalettePredictorEntryReuseFlags}[i]$ が1に等しいことは、予測子パレットにおける $i$ 番目の項目が現在のパレットにおいて再利用されることを指定する。 $\text{PalettePredictorEntryReuseFlags}[i]$ が0に等しいことは、予測子パレットの $i$ 番目の項目が現在のパレットにおける項目ではないことを指定する。配列 $\text{PalettePredictorEntryReuseFlags}[i]$ のすべての要素が0に初期化される。

## 【0091】

$palette\_predictor\_run$ は、配列 $\text{PalettePredictorEntryReuseFlags}$ におけるゼロでない項目の前にあるゼロの数を決定するために使用される。

$palette\_predictor\_run$ の値は、0から( $\text{PredictorPaletteSize} - \text{predictorEntryIdx}$ )の範囲内(両端含む)であることがビットストリーム適合性の要件である。ここで、 $predictorEntryIdx$ は、配列 $\text{PalettePredictorEntryReuseFlags}$ における現在位置に対応する。変数 $\text{NumPredictedPaletteEntries}$ は、予測子パレットから再利用される現在のパレットにおける項目数を指定する。 $\text{NumPredictedPaletteEntries}$ の値は、0から $palette\_max\_size$ の範囲内(両端含む)である。

40

## 【0092】

$num\_signalled\_palette\_entries$ は、現在のパレット・テーブルの最初の色成分 $startComp$ について明示的に信号伝達される、現在のパレットにおける項目の数を指定する。 $num\_signalled\_palette\_entries$ が存在しない場合、0に等しいと推定される。

変数 $\text{CurrentPaletteSize}[startComp]$ は、現在のパレット・テーブルの最初の色成分

50

startCompについての現在のパレットのサイズを指定し、次のように導出される：

```
CurrentPaletteSize[startComp] = NumPredictedPaletteEntries + num_signalled_palette_entries (7-155)
```

CurrentPaletteSize[startComp]の値は、0からpalette\_max\_sizeの範囲内（両端含む）である。

【0093】

new\_palette\_entries[cIdx][i]は、色成分cIdxについてのi番目の信号伝達されるパレット項目についての値を指定する。

変数PredictorPaletteEntries[cIdx][i]は、色成分cIdxについての予測子パレットにおけるi番目の要素を指定する。

10

変数CurrentPaletteEntries[cIdx][i]は、色成分cIdxについての現在のパレットにおけるi番目の要素を指定し、次のように導出される：

【表7】

```
numPredictedPaletteEntries = 0
```

```
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
```

```
  if( PalettePredictorEntryReuseFlags[ i ] ) {
```

```
    for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )
```

```
      CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
```

20

```
      PredictorPaletteEntries[ cIdx ][ i ]
```

```
      numPredictedPaletteEntries++
```

```
    }
```

```
for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ ) (7-156)
```

```
  for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
```

```
    CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = new_palette_entries[ cIdx ][ i ]
```

30

【0094】

palette\_escape\_val\_present\_flagが1に等しいことは、現在の符号化単位が少なくとも1つのエスケープ符号化サンプルを含むことを指定する。escape\_val\_present\_flagが0に等しいことは、現在の符号化単位においてエスケープ符号化されたサンプルがないことを指定する。存在しない場合、palette\_escape\_val\_present\_flagの値は1に等しいと推定される。

変数MaxPaletteIndexは、現在の符号化単位についてのパレット・インデックスの可能な最大値を指定する。MaxPaletteIndexの値は

```
CurrentPaletteSize[startComp] - 1 + palette_escape_val_present_flag
```

に等しく設定される。

40

【0095】

palette\_idx\_idcは、パレット・テーブル、CurrentPaletteEntriesへのインデックスの指示である。palette\_idx\_idcの値は、ブロック内の最初のインデックスについては0からMaxPaletteIndexの範囲内（両端含む）であり、ブロック内の残りのインデックスについては0から（MaxPaletteIndex - 1）の範囲内（両端含む）である。

palette\_idx\_idcが存在しない場合、0に等しいと推定される。

【0096】

palette\_transpose\_flagが1に等しいことは、現在の符号化単位内のサンプルについてのインデックスを走査するために垂直横断走査が適用されることを指定する。palette\_transpose\_flagが0に等しいことは、現在の符号化単位内のサンプルについてのイン

50

デックスを走査するために水平横断走査が適用されることを指定する。存在しない場合、`palette_transpose_flag`の値は0に等しいと推定される。

配列`TraverseScanOrder`は、パレット符号化のための走査順序配列を指定する。`palette_transpose_flag`が0に等しい場合、`TraverseScanOrder`は水平走査順序`HorTravScanOrder`を割り当てられ、`palette_transpose_flag`が1に等しい場合、`TraverseScanOrder`は垂直走査順序`VerTravScanOrder`を割り当てられる。

【0097】

`run_copy_flag`が1に等しいことは、`copy_above_palette_indices_flag`が0に等しい場合、パレット・ラン・タイプが前に走査された位置でのラン・タイプと同じであり、パレット・ラン・インデックスが前の位置でのインデックスと同じであることを指定する。それ以外の場合は、`run_copy_flag`は0に等しい。

10

【0098】

`copy_above_palette_indices_flag`が1に等しいことは、パレット・インデックスが、水平横断走査が使用される場合には、上の行における同じ位置、または垂直横断走査が使用される場合には、左の列における同じ位置にあるパレット・インデックスに等しいことを指定する。`copy_above_palette_indices_flag`が0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいて符号化されるか、または推定されることを指定する。

【0099】

変数`CopyAboveIndicesFlag[xC][yC]`が1に等しいことは、パレット・インデックスが上の行（水平走査）または左の列（垂直走査）におけるパレット・インデックスからコピーされることを指定する。

20

`CopyAboveIndicesFlag[xC][yC]`が0に等しいことは、パレット・インデックスがビットストリームにおいて明示的に符号化されるか、推定されることを指定する。配列インデックス`xC`、`yC`は、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置（`xC`,`yC`）を指定する。

【0100】

変数`PaletteIndexMap[xC][yC]`は、`CurrentPaletteEntries`によって表される配列へのインデックスであるパレット・インデックスを指定する。配列インデックス`xC`、`yC`は、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置（`xC`,`yC`）を指定する。`PaletteIndexMap[xC][yC]`の値は、0から`MaxPaletteIndex`の範囲内（両端含む）である。

30

【0101】

変数`adjustedRefPaletteIndex`は次のように導出される：

40

50

## 【表 8】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

10

20

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++ (7-158)

```

## 【0 1 0 2】

palette\_escape\_valは、ある成分についての量子化されたエスケープ符号化されたサンプル値を指定する。

変数PaletteEscapeVal[cldx][xC][yC]は、PaletteIndexMap[xC][yC]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しいサンプルのエスケープ値を指定する。配列インデックスcldxは、色成分を指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。

30

PaletteEscapeVal[cldx][xC][yC]が、0に等しいcldxについては0から(1 - (BitDepth<sub>c</sub> + 1) - 1)の範囲内(両端含む)であり、0に等しくないcldxについては0から(1 - ((BitDepth<sub>c</sub> + 1)) - 1)の範囲内(両端含む)であることがビットストリーム適合性の要件である。

## 【0 1 0 3】

## 2.3 VVCにおけるローカル・デュアル・ツリー

典型的なハードウェア・ビデオ・エンコーダおよびデコーダでは、ピクチャーがより多くの小さなイントラブロックを有する場合、近傍のイントラブロック間のサンプル処理データ依存性のため、処理スループットが低下する。イントラブロックの予測子生成は、近傍ブロックからの上部および左の境界の再構成されたサンプルを必要とする。よって、イントラ予測はブロックごとに順次処理されなければならない。

40

## 【0 1 0 4】

HEVCでは、最小のCUは8×8ルーマ・サンプルである。最小のイントラCUのルーマ成分は、4つの4×4ルーマ・イントラ予測単位(PU)にさらに分割できるが、最小のイントラCUのクロマ成分はそれ以上分割できない。したがって、最悪ケースのハードウェア処理スループットは、4×4クロマ・イントラブロックまたは4×4ルーマ・イントラブロックが処理されるときに発生する。

50



## 【0105】

VTM5.0では、単一符号化ツリーでは、クロマ・パーティションは常にルーマに従い、最小のイントラCUは $4 \times 4$ ルーマ・サンプルであるため、最小のクロマ・イントラCBは $2 \times 2$ である。したがって、VTM5.0では、単一符号化ツリーにおける最小のクロマ・イントラCBは $2 \times 2$ である。VVC復号のための最悪ケースのハードウェア処理スループットは、HEVC復号の場合の $1/4$ だけである。さらに、クロマ・イントラCBの再構成プロセスは、成分横断線形モデル (cross-component linear model、CCLM)、4タップ補間フィルタ、位置依存イントラ予測結合 (position-dependent intra prediction combination、PDPC)、および復号インター・イントラ予測 (combined inter intra prediction、CIIP) を含むツールを採用した後、HEVCの場合よりはるかに複雑になる。ハードウェア・デコーダにおいて高い処理スループットを達成することは困難である。本節では、最悪ケースのハードウェア処理スループットを改善する方法が提案される。

10

## 【0106】

この方法の目標は、クロマ・イントラCBのパーティション分割を制約することにより、 $16$ クロマ・サンプルより小さいクロマ・イントラCBを禁止することである。

## 【0107】

単一符号化ツリーでは、SCIPUは、クロマ・ブロック・サイズがTHクロマ・サンプル以上であり、 $4$ THルーマ・サンプルより小さい少なくとも1つの子ルーマ・ブロックをもつ符号化ツリー・ノードとして定義される。ここで、THはこの寄稿では $16$ に設定される。各SCIPUでは、すべてのCBがインターであるか、またはすべてのCBが非インターである、すなわちイントラまたはIBCのいずれかであることが要求される。非インターSCIPUの場合、さらに、非インターSCIPUのクロマがそれ以上分割されず、SCIPUのルーマがさらに分割されることが許されることが要求される。このようにして、最小のクロマ・イントラCBサイズは $16$ クロマ・サンプルであり、 $2 \times 2$ 、 $2 \times 4$ 、および $4 \times 2$ クロマCBが除去される。さらに、クロマ・スケーリングは、非インターSCIPUの場合には適用されない。さらに、ルーマ・ブロックがさらに分割され、クロマ・ブロックが分割されない場合、ローカルのデュアル・ツリー符号化構造が構築される。

20

## 【0108】

2つのSCIPUの例を図5のAおよびBに示されている。図5のAにおいては、 $8 \times 4$ クロマ・サンプルの1つのクロマCBと3つのルーマCB ( $4 \times 8$ 、 $8 \times 8$ 、 $4 \times 8$ のルーマCB) が1つのSCIPUを形成するが、これは、 $8 \times 4$ クロマ・サンプルからの三分木 (TT) 分割は、 $16$ クロマ・サンプルより小さなクロマCBを生じるためである。図5のBでは、 $4 \times 4$ クロマ・サンプルの一方のクロマCB ( $8 \times 4$ クロマ・サンプルの左側) と3つのルーマCB ( $8 \times 4$ 、 $4 \times 4$ 、 $4 \times 4$ のルーマCB) が1つのSCIPUを形成し、 $4 \times 4$ サンプルの他方のクロマCB ( $8 \times 4$ クロマ・サンプルの右側) と2つのルーマCB ( $8 \times 4$ 、 $8 \times 4$ のルーマCB) が1つのSCIPUを形成する。これは、 $4 \times 4$ クロマ・サンプルからの二分木 (BT) 分割は、 $16$ のクロマ・サンプルよりも小さなクロマCBを生じるためである。

30

## 【0109】

提案される方法では、現在のスライスが1スライスであるか、または現在のSCIPUがさらに1回分割した後に $4 \times 4$ ルーマ・パーティションをその中に有する場合、SCIPUのタイプは非インターであると推定される (VVCではインター $4 \times 4$ は許可されないため) ; それ以外の場合、SCIPUのタイプ (インターまたは非インター) は、SCIPUにおいてCUをパースする前に、1つの信号伝達されるフラグによって示される。

40

## 【0110】

上記の方法を適用することにより、 $2 \times 2$ のクロマ・ブロックの代わりに $4 \times 4$ 、 $2 \times 8$ 、または $8 \times 2$ のクロマ・ブロックが処理する場合に、最悪ケースのハードウェア処理スループットが発生する。最悪ケースのハードウェア処理スループットはHEVCの場合と同じであり、VTM5.0の場合の4倍である。

## 【0111】

50

## 2.4 変換スキップ (TS)

HEVCの場合と同様に、ブロックの残差は変換スキップ・モードで符号化されることができる。構文符号化の冗長性を避けるために、CUレベルMTS\_CU\_flagがゼロに等しくない場合、変換スキップ・フラグは信号伝達されない。変換スキップのためのブロック・サイズ制限は、JEM4におけるMTSについての場合と同じである。このことは、ブロック幅と高さの両方が32以下の場合に、変換スキップがCUに適用できることを示している。現在のCUについてLFNSTまたはMIPがアクティブ化される場合、暗黙的なMTS変換がDCT2に設定されることに注意されたい。また、暗黙的なMTSは、インター符号化されたブロックについてMTSが有効にされている場合でも、有効にされることができる。

【0112】

また、変換スキップ・ブロックについて、最小の許容される量子化パラメータ (QP) は  $6 * (\text{internalBitDepth} - \text{inputBitDepth}) + 4$  と定義される。

【0113】

## 2.5 代替的なルーマ半画素補間フィルタ

JVET-N0309では、代替的な半画素補間フィルタが提案されている。

【0114】

半画素ルーマ補間フィルタのスイッチングは、動きベクトル精度に依存して行われる。既存の1/4画素、フル画素、および4画素AMVRモードに加えて、新しい半画素精度AMVRモードが導入される。半画素動きベクトル精度の場合にのみ、代替的な半画素ルーマ補間フィルタが選択されることができる。

【0115】

半画素動きベクトル精度 (すなわち、半画素AMVRモード) を使用する非アフィン非マージ・インター符号化されるCUについて、HEVC/VVC半画素ルーマ補間フィルタと一つまたは複数の代替的な半画素補間との間の切り換えが、新しい構文要素hpellfidxの値に基づいて行われる。構文要素hpellfidxは、半画素AMVRモードの場合にのみ信号伝達される。空間的マージ候補を使用するスキップ/マージ・モードの場合、構文要素hpellfidxの値は近傍ブロックから継承される。

【0116】

## 2.6 適応色変換 (Adaptive Color Transform, ACT)

図6は、ACTが適用される復号フローチャートを示す。図6に示されるように、色空間変換は、残差ドメインで実行される。具体的には、1つの追加的な復号モジュール、すなわち逆ACTが、YCgCoドメインからの残差をもとのドメインに戻す変換をするための逆変換の後に、導入される。

【0117】

VVCでは、最大変換サイズが1符号化単位 (CU) の幅または高さより小さいのでない限り、1つのCUリーフノードも、変換処理の単位として使用される。よって、提案される実装では、ACTフラグは、残差を符号化するための色空間を選択するために、1つのCUについて信号伝達される。さらに、HEVC ACT設計に従い、インターおよびIBC CUについて、ACTは、CU内に少なくとも1つのゼロでない係数がある場合にのみ有効にされる。イントラCUについては、ACTは、クロマ成分がルーマ成分の同じイントラ予測モード、すなわちDMモードを選択する場合にのみ、有効にされる。

色空間変換のために使用されるコア変換は、HEVCのために使用されるものと同じに保たれる。具体的には、以下の順方向および逆方向YCgCo色変換行列が適用される。

【数1】

$$\begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} / 4 \quad \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \end{bmatrix}$$

10

20

30

40

50

## 【 0 1 1 8 】

さらに、色変換前後の残差信号のダイナミックレンジ変化を補償するために、( - 5, - 5, - 3 ) のQP調整が変換残差に適用される。

他方、順色変換と逆色変換は、3つの成分すべての残差にアクセスする必要がある。対応して、提案される実装では、ACTは、3つの成分のすべての残差が利用可能ではない以下の2つのシナリオでは無効にされる。

1. 別個木パーティション ( separate-tree partition ) : 別個の木が適用される場合、1つのCTU内のルーマおよびクロマ・サンプルは、異なる構造によって分割される。その結果、ルーマ・ツリー内のCUはルーマ成分のみを含み、クロマ・ツリー内のCUは2つのクロマ成分のみを含むことになる。イントラ・サブパーティション予測 ( Intra sub-partition prediction、ISP ) : ISPサブパーティションはルーマに適用されるだけであり、一方、クロマ信号は分割なしに符号化される。現在のISP設計では、最後の諸ISPサブパーティションを除いて、他のサブパーティションはルーマ成分だけを含む。

10

## 【 0 1 1 9 】

3. 本明細書に記載される技術的解決策および実施形態によって解決される技術的問題

1. エスケープ・シンボルの現在のバイナリ化は固定長ではなく、このことは、一様分布をもつ源に好適でありうる。

2. 現在のパレット符号化設計は、可能性のある冗長性を除去するためにインデックス調整プロセスを実行し、これは、たとえばエスケープ値インデックスが誤って導出された場合に、パース依存性 ( parsing dependency ) を導入する可能性がある。

20

3. 現在インデックスを導出するために使用される参照インデックスは、エンコーダ制約条件を必要とすることがあり、これは、現在の設計では考慮されず、コーデック設計のために望ましくない。

4. ローカル・デュアル・ツリーが有効にされる場合、前のブロックと現在のブロックのパレット項目は、異なる数の色成分をもつ可能性がある。そのような場合の対処方法は不明である。

5. ローカル・デュアル・ツリーとPLTは、単一ツリー領域からデュアル・ツリー領域に符号化する際に、いくつかのパレット項目が繰り返される可能性があるため、同時に適用できなかった。一例が図7に示される。

6. joint\_cbrモードのためのクロマQPテーブルが制約される可能性がある。

30

7. ある種の条件下では、エスケープ・サンプルが冗長になることがある。

8. ライン・ベースのCGモードは、高スループットで処理できなかった。

## 【 0 1 2 0 】

4. 実施形態および解決策のリスト

以下のリストは、一般的な概念を説明するための例として考慮されるべきである。これらの項目は狭義に解釈すべきではない。さらに、これらの項目は、任意の仕方で組み合わせることができる。

## 【 0 1 2 1 】

以下の例は、VVCにおけるパレット方式および他のすべてのパレット関連方式に適用されうる。

40

## 【 0 1 2 2 】

以下の項目では、Qpは、JVET-P2001-vEのセクション8.4.5.3におけるqPを示しうる。

## 【 0 1 2 3 】

以下の項目では、QpPrimeTsMinは、変換スキップ・モードのための許容される最小量子化パラメータである。

## 【 0 1 2 4 】

Modulo(x, M)は、xが正の整数の場合は ( x % M ) として定義され、そうでない場合は M - ( - x ) % M として定義される。

## 【 0 1 2 5 】

50

以下では、無損失モードで符号化されたブロックは、1に等しいtranquant\_bypass\_flagで符号化されたブロック、または所与の閾値より大きくないQPおよび1に等しいtransform\_skip\_flagで符号化されたブロックを意味しうる。

【0126】

以下の例は、VVCにおけるパレット方式および他のすべてのパレット関連方式に適用されうる。

【0127】

1. 固定長符号化が、エスケープ・シンボルを符号化するために適用されてもよい。
  - a. 一例では、エスケープ・シンボルは、固定長バイナリ化で信号伝達されてもよい。
  - b. 一例では、エスケープ・シンボルは、Nビットを使用して固定長バイナリ化で信号伝達されてもよい。
  - c. 一例では、エスケープ・シンボルを信号伝達するための符号長（たとえば、項目1.bで言及されているN）は、内部ビット深さに依存しうる。
    - i. あるいはまた、エスケープ・シンボルを信号伝達するための符号長は、入力ビット深さに依存してもよい。
    - ii. あるいはまた、エスケープ・シンボルを信号伝達するための符号長は、内部ビット深さと入力ビット深さととの間の差に依存してもよい。
    - iii. 一例では、Nは入力/内部ビット深さに等しく設定される。
  - d. 一例では、固定長符号化の長さは、ビデオ処理単位レベル、たとえば、スライス・サブピクチャー、タイル、ピクチャー、ビデオにおいて信号伝達されてもよい。
  - e. 一例では、エスケープ・シンボルを信号伝達するための符号長（たとえば、項目1.bで言及されているN）は、量子化パラメータ、すなわちQpに依存してもよい。
    - i. 一例では、エスケープ・シンボルを信号伝達するための符号長は、 $f(Qp)$ で示されるような量子化パラメータの関数であってもよい。
      1. 一例では、関数 $f$ は（内部ビット深さ -  $g(Qp)$ ）と定義されてもよい。
      2. 一例では、Nは（内部ビット深さ -  $\max(16, (Qp - 4)/6)$ ）に設定されてもよい。
      3. 一例では、Nは（内部ビット深さ -  $\max(QpPrimeTsMin, (Qp - 4)/6)$ ）に設定されてもよい。ここで、 $qP$ は、デコードされた量子化パラメータであり、 $QpPrimeTsMin$ は、変換スキップ・モードのための最小の許容される量子化パラメータである。
      4. あるいはまた、さらに、符号長Nは、 $\max(A, \text{内部ビット深さ} - (\max(QpPrimeTsMin, Qp) - 4)/6)$ に設定されてもよく、ここで、Aは、0または1のような負でない整数値である。
        - ii. 上記サブ項目において言及された $Qp$ は、スライスQPを指してもよい。
          1. あるいはまた、 $Qp$ は、スライスQPに一定値を加えたものを指してもよい。
    - f. 上記の諸例において、Nは、0以上であってもよい。

【0128】

2. エスケープ・シンボルのための脱量子化 $Qp$ は、スライス/ピクチャー/PPSレベル $Qp$ に基づいていてもよい。
  - a. 一例では、エスケープ・シンボルのための脱量子化 $Qp$ は、スライス/ピクチャー/PPSレベル $Qp$ に所与のオフセットを加えたものに基づいていてもよい。
    - i. オフセットは定数であってもよい。
    - ii. オフセットは、暗黙的にまたは明示的に、ビットストリームにおいて示されてもよい。
  - b. 一例では、ブロック・レベルの $Qp$ 差はビットストリームにおいてスキップされてもよい。
    - i. 一例では、 $cbf$ は、0と推定されてもよい。

【0129】

3. エスケープ・シンボルについて、脱量子化の前に左シフトが適用されてもよい。
  - a. 一例では、Nビットの左シフト（ $N = 0$ ）が、脱量子化の前に適用されてもよい。

i . 一例では、 $N$ は $\text{Min}(\text{bitDepth} - 1, (\text{QpPrimeTsMin} - 4) / 6)$ に等しくてもよい。ここで、 $\text{bitDepth}$ は内部ビット深さであり、ここで、 $\text{bitDepth}$ は内部ビット深さである。

i i . あるいはまた、 $N$ は、 $\text{bitDepth} - \text{inputBD}$ に等しくてもよく、ここで、 $\text{inputBD}$ は、入力ビット深さである。

1 . 一例では、 $\text{inputBD}$ がビットストリームにおいて示されてもよい。

i i i . あるいはまた、 $N$ は、 $\text{deltaBD}$ と等しくてもよく、ここで、 $\text{deltaBD}$ は、ビットストリームにおいて示される。

【0130】

4 . エスケープ・シンボルの脱量子化は $(\text{Qp} - \text{QpPrimeTsMin})$ に依存してもよい。 10

a . 一例では、 $(\text{Qp} - \text{QpPrimeTsMin} + 4)$ は、脱量子化 $\text{Qp}$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

b . 一例では、 $\text{Min}(\text{Qp} - \text{QpPrimeTsMin} + 4, 63 + \text{QpBdOffset})$ は、脱量子化 $\text{Qp}$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

【0131】

5 . エスケープ・シンボルの脱量子化は $(\text{Qp} - N * 6)$ に依存してもよい。

a . 一例では、 $N$ は、項目3.aにおける左方シフトの数を指してもよい。

b . 一例では、 $\text{Max}(0, \text{Qp} - N * 6)$ が脱量子化 $\text{Qp}$ として適用されてもよい。

【0132】

6 . エスケープ・シンボルの脱量子化は、 $\text{deltaBD}$ 、すなわち、内部ビット深さと入力ビット深さの差に依存してもよい。 20

a . 一例では、 $(\text{Qp} - \text{deltaBD} * 6)$ は、脱量子化 $\text{Qp}$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

b . 一例では、 $\text{Min}(\text{Max}(0, \text{Qp} - \text{deltaBD} * 6), 63 + \text{QpBdOffset})$ は、脱量子化 $\text{Qp}$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

【0133】

7 . 一つのビデオ単位（たとえば、CU）におけるエスケープ・シンボルの使用を無効にすることが提案される。

a . あるいはまた、さらに、エスケープ・シンボルの存在 の指示を信号伝達することはスキップされる。 30

b . 一例では、エスケープ・シンボルの使用を有効 / 無効にするかどうかは、量子化パラメータおよび / またはビット深さに依存してもよい。

i . 一例では、 $(\text{内部ビット深さ} - (\text{Max}(\text{QpPrimeTsMin}, \text{Qp}) - 4) / 6)$ が0より大きくない場合、エスケープ・シンボルの使用は無効にされてもよい。

【0134】

8 . 3次のEGを除く可変長符号化が、エスケープ・シンボルの符号化のために適用されてもよい。

a . 一例では、エスケープ・シンボルのバイナリ化は、入力パラメータ $K$ をもつ打ち切りされたバイナリ（truncated binary、TB）であってもよい。

b . 一例では、エスケープ・シンボルのバイナリ化は、 $K$ 次のEGであってもよい。ここで、 $K$ は3に等しくない。 40

i . 一例では、エスケープ・シンボルのバイナリ化は、0次のEGであってもよい。

1 . あるいはまた、一例では、エスケープ・シンボルのバイナリ化は、1次のEGであってもよい。

2 . あるいはまた、一例では、エスケープ・シンボルのバイナリ化は、2次のEGであってもよい。

c . 上記の諸例では、 $K$ は整数であってもよく、以下に依存してもよい。

i . SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ。

i i . 内部ビット深さ

- i i i . 入力ビット深さ
- i v . 内部ビット深さと入力深さの差
- v . 現在ブロックのブロック寸法
- v i . 現在ブロックの現在の量子化パラメータ
- v i i . カラー・フォーマットの指示 (4:2:0、4:4:4、RGBまたはYUVなど)
- v i i i . 符号化構造 (単一ツリーまたはデュアル・ツリーなど)
- i x . 色成分 (ルーマ成分および/またはクロマ成分など)

## 【0135】

9 . エスケープ・シンボルを符号化するための複数のバイナリ化方法が、ビデオ単位 (たとえば、シーケンス/ピクチャー/スライス/タイル/ブリック/サブピクチャー/CTU行/CTU/CTB/CB/CU/ピクチャー内のサブ領域) および/またはエスケープ・シンボルの一つまたは複数の値に対して適用されてもよい。 10

a . 一例では、複数のバイナリ化方法のうちの一つをどのようにして選択するかが、ビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について信号伝達されてもよい。

b . 一例では、複数のバイナリ化方法のうちの一つをどのようにして選択するかが、ビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について導出されてもよい。

c . 一例では、一つのビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について、2つ以上のバイナリ化方法が適用されてもよい。 20

i . 一例では、インデックスまたはフラグが、選択されたバイナリ化方法を知らせるためにエンコード/デコードされてもよい。

## 【0136】

以下の項目では、pは色成分のシンボル値を表してもよく、bdはビット深さ(たとえば、内部ビット深さまたは入力ビット深さ)を表してもよく、ibdは入力ビット深さを表してもよく、Qpは変換スキップ・ブロックまたは変換ブロックについての量子化パラメータを表してもよい。さらに、ルーマ成分とクロマ成分についてのQPは、異なっても、同じであってもよい。ビット深さは、所与の色成分に関連付けられていてもよい。

## 【0137】

10 . 量子化および/または逆量子化プロセスをどのように適用するかは、ブロックがパレット・モードで符号化されるか否かに依存してもよい。 30

a . 一例では、エスケープ・シンボルについての量子化および/または逆量子化プロセスは、量子化が適用された通常のイントラ/インター符号化されたブロックについて使用されるものとは異なってもよい。

## 【0138】

11 . エスケープ・シンボルについての量子化および/または逆量子化プロセスは、ビットシフトを使用してもよい。

a . 一例では、エスケープ・シンボルを量子化するために右ビットシフトが使用されてもよい。

i . 一例では、エスケープ・シンボルは、 $f(p, Q_p)$ として信号伝達されてもよく、ここで、pは入力シンボル値(たとえば、入力ルーマ/クロマ・サンプル値)であり、 $Q_p$ は、対応する色成分についての導出された量子化パラメータである。 40

1 . 一例では、関数fは $p \cdot g(Q_p)$ として定義されてもよい。

2 . 一例では、関数fは $(p + (1 - g(Q_p))) \cdot g(Q_p)$ として定義されてもよい。

3 . 一例では、関数fは、 $(0, (1 - bd) - 1, (p + (1 - g(Q_p)))) \cdot g(Q_p)$ として定義されてもよい。

i i . 一例では、エスケープ・シンボルは、 $h(p)$ として信号伝達されてもよい。

1 . 一例では、関数hは、 $p \cdot N$ として定義されてもよい。

2 . 一例では、関数hは、 $(p + (1 - (N - 1))) \cdot N$ として定義されてもよい。 50

3 . 一例では、cu\_transquant\_bypass\_flagが1に等しい場合、Nは0に設定されてもよい。

4 . 一例では、cu\_transquant\_bypass\_flagが1に等しい場合、Nは $(bd - ibd)$ に等しくてもよい。ここで、bdは内部ビット深さであり、ibdは入力ビット深さである。

5 . 一例では、関数hは $\text{clip}(0, (1 - (bd - N) - 1), p - N)$ として定義されてもよく、ここで、bdは現在の色成分についての内部ビット深さである。

6 . 一例では、関数hは $\text{clip}(0, (1 - (bd - N) - 1), (p + (1 - (N - 1)))) - N$ として定義されてもよく、ここで、bdは現在の色成分についての内部ビット深さである。

7 . 上記の例において、Nは $[0, (bd - 1)]$ の範囲であってもよい。

10

b . 一例では、左ビットシフトは、エスケープ・シンボルを逆量子化するために使用されてもよい。

i . 一例では、エスケープ・シンボルは、 $f(p, Qp)$ として脱量子化されてもよい。ここで、pは、復号されたエスケープ・シンボルであり、Qpは、対応する色成分についての導出された量子化パラメータである。

1 . 一例では、fは $p - g(Qp)$ として定義されてもよい。

2 . 一例では、fは $(p - g(Qp)) + (1 - (g(Qp) - 1))$ として定義されてもよい。

ii . 一例では、エスケープ・シンボルは、 $f(p, Qp)$ として再構成されてもよい。ここで、pは、復号されたエスケープ・シンボルである。

1 . 一例では、fは $\text{clip}(0, (1 - bd) - 1, p - g(Qp))$ として定義されてもよい。

20

2 . 一例では、fは、 $\text{clip}(0, (1 - bd) - 1, (p - g(Qp)) + (1 - (g(Qp) - 1)))$ として定義されてもよい。

iii . 一例では、エスケープ・シンボルは $h(p)$ として再構成されてもよい。

1 . 一例では、関数hは $p - N$ として定義されてもよい。

2 . 一例では、関数hは、 $(p - N) + (1 - (N - 1))$ として定義されてもよい。

3 . 一例では、cu\_transquant\_bypass\_flagが1に等しい場合、Nは0に設定されてもよい。

4 . 一例では、cu\_transquant\_bypass\_flagが1に等しい場合、Nは $(bd - ibd)$ に等しくてもよく、ここで、bdは内部ビット深さであり、ibdは入力ビット深さである。

30

5 . 一例では、Nは $(\max(QpPrimeTsMin, qP) - 4) / 6$ に設定され、ここで、qPは復号された量子化パラメータであり、QpPrimeTsMinは変換スキップ・モードのための最小の許容される量子化パラメータである。

a ) 上記の例では、ルーマとクロマの両方が変換スキップ・モードをもつ場合、変換スキップ・モードのための異なる最小の許容される量子化パラメータが、異なる色成分について適用されてもよい。

6 . あるいはまた、上記の諸例について、Nは、 $\min(bd - 1, N)$ のように、さらにクリップされてもよい。

7 . 上記の例では、Nは $[0, (bd - 1)]$ の範囲であってもよい。

#### 【 0 1 3 9 】

40

1 2 . 脱量子化として左シフトを適用する場合、エスケープ・シンボルpの再構成オフセットはビット深さ情報に依存してもよい。

a . 一例では、それは、内部ビット深さと入力ビット深さの間の差、すなわち、 $\text{deltaBD} = \text{内部ビット深さ} - \text{入力ビット深さ}$ に依存してもよい。

b . Kが $\text{deltaBD}$ 以下である場合、再構成された値は $p - K$ であってもよい。

c . Kが $\text{deltaBD}$ より大きい場合、再構成値は $(p - K) + (1 - (K - 1))$ であってもよい。

d . Kが $T0$ 以下である場合（たとえば、 $T0 = 2$ ）、再構成された値は $p - K$ であってもよい。

e . Kが $T1$ より大きい場合（たとえば、 $T1 = 2$ ）、再構成値は $(p - K) + (1 - (K -$

50

1))であってもよい。

f . 一例では、項目dおよびeにおけるT0およびT1は、シーケンス/ピクチャー/スライス/タイル/ブリック/サブピクチャー・レベルなどで、ビットストリームにおいて信号伝達されてもよい。

g . 一例では、再構成値は、 $(p \quad K) + ((1 \quad (K - 1) \quad \text{deltaBD} \quad \text{deltaBD}))$ であってもよい。

h . 一例では、再構成値は、 $((p \quad (K + 1)) + (1 \quad K)) \quad (\text{deltaBD} + 1) \quad \text{del}$   
 $\text{taBD})$ であってもよい。

i . 一例では、deltaBDは、シーケンス/ピクチャー/スライス/タイル/ブリック/サブピクチャー・レベルなどでビットストリームにおいて信号伝達されてもよい。

10

j . 一例では、どの再構成値が使用されるか(たとえば、項目bないしe)は、現在のブロックの量子化パラメータに依存してもよい。

k . 一例では、どの再構成値が使用されるか(たとえば、項目bないしe)は、deltaBDの値に依存してもよい。

l . 一例では、Kは、g(Qp)に設定されてもよい。

【 0 1 4 0 】

1 3 . 上記の諸例では、以下が適用されてもよい :

a . 一例では、エスケープ・シンボルは、コンテキスト符号化されてもよい。

b . 一例では、エスケープ・シンボルはバイパス符号化されてもよい。

c . 一例では、g(Qp)は、 $(Qp - 4)/6$ または $QP/8$ として定義されてもよい。

20

i . あるいはまた、g(Qp)は $Qp/6$ または $QP/8$ として定義されてもよい。

i i . あるいはまた、g(Qp)は $\max(16, Qp/6)$ として定義されてもよい。

i i i . あるいはまた、g(Qp)は $\max(16, (Qp - 4)/6)$ として定義されてもよい。

i v . あるいはまた、g(Qp)は $\max((bd - ibd) * 6 + 4, (Qp - 4)/6)$ として定義されてもよい。

v . あるいはまた、g(Qp)は $\max(M, (Qp - 4)/6)$ として定義されてもよい。

1 . 一例では、Mはデコーダに信号伝達されてもよい。

v i . あるいはまた、g(Qp)は $\max((M, Qp) - 4)/6$ として定義されてもよい。

1 . 一例では、MはSPSにおいて示されてもよい。

2 . 一例では、ルーマ成分およびクロマ成分に同じMまたは異なるMが適用される。

30

3 . 一例では、Mは $(bd - ibd) * 6 + 4$ に等しくてもよい。

v i i . あるいはまた、g(Qp)は $Qp/6$ または $QP/8$ として定義されてもよい。

v i i i . あるいはまた、g(Qp)は $(\max(16, Qp)/6)$ として定義されてもよい。

i x . あるいはまた、g(Qp)は $(\max(16, Qp) - 4)/6$ として定義されてもよい。

d . 一例では、g(Qp)の値は $[0, (bd - 1)]$ の範囲であってもよい。

e . 一例では、最大関数 $\max(a, i)$ は $(i \quad =a?a:i)$ として定義されてもよい。

i . あるいはまた、一例では、最大関数 $\max(a, i)$ は $(i \quad a?a:i)$ として定義されてもよい。

f . 一例では、Nは整数(たとえば、8または10)であってもよく、以下に依存してもよい。

40

i . SPS/VPS/PPS/ピクチャー・ヘッダ/スライス・ヘッダ/タイル・グループ・ヘッダ/LCU行/LCUのグループ/ブリックにおいて信号伝達されるメッセージ

i i . 内部ビット深さ

i i i . 入力ビット深さ

i v . 内部ビット深さと入力深さの差

v . 現在ブロックのブロック寸法

v i . 現在ブロックの現在の量子化パラメータ

v i i . カラー・フォーマットの指示(4:2:0、4:4:4、RGB、またはYUVなど)

v i i i . 符号化構造(単一ツリーまたはデュアル・ツリーなど)

50



i x . 色成分 ( ルーマ成分および / またはクロマ成分など )

x . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

g . 一例では、Nはデコーダに信号伝達されてもよい。

【 0 1 4 1 】

1 4 . エスケープ値についてのQpはクリップされてもよい。

a . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus } 4$ と等しくてもよい。

b . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus } 4$ に関係してもよい。

i . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus } 4 + 4$ と等しくてもよい。 10

c . 一例では、各色成分についての最低のQpは、SPS / PPS / VPD / DPS / タイル / スライス・ヘッダにおいて示されてもよい。

d . 一例では、エスケープ値に適用される最低のQpは、 $(bd - ibd) * 6 + 4$ であってもよい。ここで、bdは、内部ビット深さであり、ibdは、ある色成分についての入力ビット深さを示す。

e . 一例では、上記の諸例は、ある色成分に適用されてもよい。

【 0 1 4 2 】

1 5 . 上記の諸例では、エスケープ値についてのクロマQpは、マッピングの前 / 後にQpを使用してもよい。 20

【 0 1 4 3 】

1 6 . パレット・モードで現在のパレット・インデックスを導出する際に参照インデックスを使用しないことが提案される。

a . 一例では、パレット・インデックスは、参照インデックス (たとえば、adjusted RefPaletteIndex) の可能性を排除することなく、直接的に信号伝達されてもよい。

i . あるいはまた、一例では、エンコーダは、参照インデックスが常に現在インデックスと異なることを可能にするように制約されてもよい。そのような場合、パレット・インデックスは、参照インデックスの可能性を排除することによって、信号伝達されてもよい。

b . 一例では、パレット・インデックスのバイナリ化は、バイナリ化入力パラメータとして最大パレット・インデックスを用いることを伴う打ち切りされたバイナリ (TB) であってもよい。 30

c . 一例では、パレット・インデックスのバイナリ化は固定長であってもよい。

d . 一例では、パレット・インデックスのバイナリ化は、K次のEGであってもよい。

i . 一例では、Kは整数 (たとえば、1、2または3) であってもよく、以下に依存してもよい

1 . SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ

2 . 内部ビット深さ

3 . 入力ビット深さ

4 . 内部ビット深さと入力深さの差

5 . 現在ブロックのブロック寸法

6 . 現在ブロックの現在の量子化パラメータ

7 . カラー・フォーマットの指示 (4:2:0、4:4:4、RGB、またはYUVなど)

8 . 符号化構造 (単一ツリーまたはデュアル・ツリーなど)

9 . 色成分 (ルーマ成分および / またはクロマ成分など)

e . 一例では、上記の諸例は、現在のブロックが少なくとも、1つのエスケープ・サンプルを有する場合にのみ適用されうる。 40

【 0 1 4 4 】

1 7 . 現在のパレット・インデックスは、以前の諸パレット・インデックスとは独立して 50

信号伝達されてもよい。

a . 一例では、以前の諸パレット・インデックスを使用するかどうか、および/または、どのように使用するかは、現在のブロックにおいてエスケープ・サンプル（単数または複数）があるかどうか依存してもよい。

【0145】

18 . エスケープ・シンボルについてのインデックスから非エスケープ・シンボルについてのインデックスへの導出は許可されなくてもよい。

a . 一例では、エスケープ・シンボルが適用され、パレット・インデックスがエスケープ・シンボルについてのインデックスと等しくない場合、それらのシンボルをエスケープ・シンボルとしてデコードすることは許されなくてもよい。

10

【0146】

19 . 非エスケープ・シンボルについてのインデックスからエスケープ・シンボルについてのインデックスへの導出は、許されなくてもよい。

a . 一例では、エスケープ・シンボルが適用され、パレット・インデックスがエスケープ・シンボルについてのインデックスと等しい場合、それらのシンボルを非エスケープ・シンボルとしてデコードすることは許されなくてもよい。

【0147】

20 . 導出されたパレット・インデックスは、現在のパレット・テーブル・サイズによってキャップされてもよい。

a . 一例では、パレット・インデックスが  $\text{MaxPaletteIndex}$  より大きい場合、 $\text{MaxPaletteIndex}$  と等しくなるように修正されてもよい。

20

【0148】

21 . 導出されたパレット・インデックスは、エスケープ・シンボルについてのインデックスを除き、現在のパレット・テーブル・サイズによってキャップされてもよい。

a . 一例では、エスケープ・シンボルが適用されず、パレット・インデックスが  $\text{MaxPaletteIndex}$  より大きい場合、 $\text{MaxPaletteIndex}$  と等しくなるように修正されてもよい。

b . 一例では、エスケープ・シンボルが適用され、パレット・インデックスが  $(\text{MaxPaletteIndex} - 1)$  より大きい場合、 $(\text{MaxPaletteIndex} - 1)$  と等しくなるように修正されてもよい。

30

【0149】

22 . エスケープ・シンボルを示すインデックスは、修正されることが許されなくてもよい。

a . 一例では、インデックスが  $\text{MaxPaletteIndex}$  と等しいことは、現在のブロックにエスケープ・シンボルが存在する場合、常に、エスケープ・シンボルを示してもよい。

b . 一例では、 $\text{MaxPaletteIndex}$  と等しくないインデックスは、エスケープ・シンボルを示すインデックスとしてデコードされることはできない。

【0150】

23 . 参照インデックスと現在のインデックスとの間の差を符号化することが提案される。

40

a . 一例では、0に等しい差は、符号化されることが許されなくてもよい。

b . あるいはまた、パレット符号化されたブロック内の最初のインデックスについて、インデックスは直接符号化されてもよい。

【0151】

24 . 参照インデックス（Rと記される）と現在のインデックス（Cと記される）との間の差のモジュロを符号化することが提案される。

a . 一例では、 $l = \text{Modulo}(C - R, \text{MaxPaletteIndex})$  が符号化されてもよい。

i . 一例では、インデックスは、 $\text{Modulo}(l + R, \text{MaxPaletteIndex})$  として再構成されてもよい。

i i . 一例では、0に等しい  $\text{Modulo}(C - R, \text{MaxPaletteIndex})$  は、ビットストリ

50

ームにおいて許されなくてもよい。

i i i . 一例では、 $cMax = MaxPaletteIndex$ である打ち切りされたバイナリ符号が、前記値を符号化するために使用されてもよい。

i v . あるいはまた、パレット符号化されたブロック内の最初のインデックスについては、インデックスは直接符号化されてもよい。

b . 一例では、 $l = Modulo(C - R, MaxPaletteIndex) - 1$ が符号化されてもよい。

i . 一例では、インデックスは、 $Modulo(l + 1 + R, MaxPaletteIndex)$ として再構成されてもよい。

i i . 一例では、0より小さい $Modulo(C - R, MaxPaletteIndex) - 1$ は、ビットストリームにおいて許されなくてもよい。

i i i . 一例では、 $cMax = (MaxPaletteIndex - 1)$ である打ち切りされたバイナリ符号が、値 $l$ を符号化するために使用されてもよい。

i v . あるいはまた、パレット符号化されたブロック内の最初のインデックスについて、 $Modulo(C - R, MaxPaletteIndex)$ が符号化されてもよい。

v . あるいはまた、パレット符号化されたブロック内の最初のインデックスについては、インデックスは直接符号化されてもよい。

【0152】

25 . パレット・ブロックのデコードの開始時には、参照インデックス $R$ は $-1$ に等しく設定されてもよい。

a . あるいはまた、参照インデックス $R$ は $0$ に等しく設定されてもよい。

【0153】

26 . パレット・モードとローカル・デュアル・ツリーを排他的に有効にすることが提案される。

a . 一例では、パレット・モードが有効にされている場合、ローカル・デュアル・ツリーは許容されなくてもよい。

i . あるいはまた、一例では、ローカル・デュアル・ツリーが有効にされている場合、パレット・モードは許容されなくてもよい。

b . 一例では、ローカル・デュアル・ツリーは、4:4:4のような特定のカラー・フォーマットでは有効にされない。

c . 一例では、パレット・モードは、符号化ツリーが $MODE\_TYPE\_INTRA$ の場合には、許されなくてもよい。

d . ローカル・デュアル・ツリーの使用に基づいてパレット予測子をリセットすることが提案される。

i . 一例では、パレット予測子は、単一ツリーがローカル・デュアル・ツリーに切り換えられるときにリセットされてもよい。

i i . 一例では、パレット予測子は、ローカル・デュアル・ツリーが単一ツリーに切り換えられるときにリセットされてもよい。

i i i . あるいはまた、さらに、パレット予測子における諸項目の使用(たとえば、 $palette\_predictor\_run$ )を信号伝達するかどうかは、ツリー・タイプに依存してもよい。

1 . 一例では、ローカル・デュアル・ツリーと単一ツリーとの間の切り換えに出会うとき、パレット予測子における諸項目の使用(たとえば、 $palette\_predictor\_run$ )の信号伝達は省略される。

【0154】

27 . ローカル・デュアル・ツリーが適用されるとき、パレット予測テーブルにおける反復されるパレット項目を除去することが提案される。

a . 一例では、パレット予測テーブルは、ローカル・デュアル・ツリーが適用される場合にリセットされてもよい。

i . あるいはまた、一例では、デコーダは、ローカル・デュアル・ツリーが適用されるときに、予測テーブル内のすべてのパレット項目をチェックし、反復される項目を除去

10

20

30

40

50

してもよい。

i i . あるいはまた、一例では、デコーダは、ローカル・デュアル・ツリーが適用されるときに、予測テーブル内の部分的なパレット項目をチェックし、反復された項目を除去してもよい。

i i i . 一例では、パレット項目をチェックする際に、フル剪定 (full pruning) または部分剪定 (partial pruning) が適用されてもよい。

1 . 一例では、選択された項目の集合がチェックされてもよい (たとえば、該集合は、パレット予測子における全部のパレット項目または部分的なパレット項目を含む)。

a ) 一例では、フル剪定または部分剪定は、選択された項目に対して適用されてもよい。

2 . 一例では、フル剪定は、1つの項目が、追加されうるすべての項目と比較されることを表してもよい。

3 . 一例では、部分剪定は、1つの項目が、追加されうる部分的な項目と比較されることを表してもよい。

i v . 一例では、2つのパレット項目が同じであるかどうかは、それらのルーマ成分値が同じであるかどうかだけに基づいていてもよい。

1 . あるいはまた、一例では、2つのパレット項目が同じであるかどうかは、それらのクロマ成分値が同じであるかどうかのみに基づいていてもよい。

2 . あるいはまた、一例では、2つのパレット項目が同じであるかどうかは、それらのルーマとクロマ成分値の両方が同じであるかどうかに基づいていてもよい。

v . 一例では、上記の方法は、ローカル・デュアル・ツリーがルーマ成分を処理し始めるときにのみ、ルーマ・ブロックに対して適用されてもよい。

1 . あるいはまた、一例では、上記の方法は、ローカル・デュアル・ツリーがクロマ成分を処理し始めるときにのみ、クロマ・ブロックに適用されてもよい。

v i . あるいはまた、一例では、エンコーダは、項目の3つの成分が異なる場合に、2つのパレット項目を異なると考える制約条件を追加してもよい。

【 0 1 5 5 】

2 8 . 現在のパレット項目が、パレット予測テーブルの項目と異なる数の色成分をもつ場合、パレット予測テーブルは使用されることが許されなくてもよい。

a . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグ (reused flag) が真であるとマークされることがありうるが、現在のパレット項目が予測とは異なる数の色成分を有する場合には、現在のブロックについては使用されなくてもよい。

b . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグは、現在のパレット項目が予測とは異なる数の色成分を有する場合には、偽としてマークされてもよい。

【 0 1 5 6 】

2 9 . 予測テーブルと現在のパレット・テーブルが異なる色成分 (単数または複数) をもつ場合、パレット予測テーブルは使用されることが許されなくてもよい。

a . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグが真であるとマークされることがありうるが、予測テーブルと現在のパレット・テーブルが異なる色成分を有する場合には、現在のブロックについては使用されなくてもよい。

b . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグは、予測テーブルと現在のパレット・テーブルが異なる色成分を有する場合には、偽としてマークされてもよい。

【 0 1 5 7 】

3 0 . エスケープ・シンボルは、たとえば以前に符号化されたエスケープ・シンボルに基づいて、予測的に符号化されてもよい。

a . 一例では、ある成分のエスケープ・シンボルは、同じ色成分における符号化された値によって予測されてもよい。

10

20

30

40

50

i . 一例では、エスケープ・シンボルは、同じ成分における前に符号化されたあるエスケープ・シンボルを予測子として使用してもよく、それらの間の残差が信号伝達されてもよい。

i i . 代替的に、エスケープ・シンボルは、同じ成分における前に符号化されたK番目のエスケープ・シンボルを予測子として使用してもよく、それらの間の残差が信号伝達されてもよい。

i i i . あるいはまた、エスケープ・シンボルは、同じ成分における複数の（たとえば、K個の）符号化されたエスケープ・シンボルから予測されてもよい。

1 . 一例では、Kは整数（たとえば、1、2または3）であってもよく、以下に依存してもよい

a ) SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ

b ) 内部ビット深さ

c ) 入力ビット深さ

d ) 内部ビット深さと入力深さの差

e ) 現在ブロックのブロック寸法

f ) 現在ブロックの現在の量子化パラメータ

g ) カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

h ) 符号化構造（単一ツリーまたはデュアル・ツリーなど）

i ) 色成分（ルーマ成分および/またはクロマ成分など）

b . 一例では、ある成分のエスケープ・シンボルは、別の成分の符号化された値によって予測されてもよい。

c . 一例では、ピクセルは複数の色成分を有してもよく、ピクセルがエスケープ・シンボルとして扱われる場合、ある成分の値は、他の成分のサンプルの値によって予測されてもよい。

i . 一例では、エスケープ・シンボルのU成分が、そのシンボルのV成分によって予測されてもよい。

d . 一例では、上記の諸方法は、ある色成分に対して（たとえば、ルーマ成分またはクロマ成分に対して）、または符号化された情報などに基づくある種の条件のもとで適用されるだけであってもよい。

【 0 1 5 8 】

3 1 . パレットに関連する構文要素の信号伝達は、パレットの最大サイズ、および/またはブロック寸法、および/または無損失モードの使用および/または量子化パラメータ（QP）に依存してもよい。

a . 一例では、無損失符号ブロックの場合および/またはQPが閾値以下である場合および/または変換スキップが適用される場合、ブロックのパレット・サイズはブロック寸法と等しいと推定される。

i . あるいはまた、無損失符号ブロックの場合および/またはQPが閾値以下である場合、ブロックのパレット・サイズは $\min(\text{ブロック寸法}, \text{最大パレット・サイズ})$ に等しいと推定される。

b . ブロック内のエスケープ・サンプルの使用を信号伝達するかどうかは、ブロック寸法および/または無損失符号化モードの使用（たとえば、QPが所与の値（たとえば4）に等しいか否か；および/または`transform_skip_flag`が1に等しいか；または`transform_bypass_flag`が`true`に等しいか否か）および/またはQPに依存してもよい。

i . あるいはまた、さらに、エスケープ・サンプルの使用を信号伝達するかどうかは、現在のブロックのブロック寸法と現在のパレット・サイズとの間の関係に依存してもよい。

1 . 一例では、それを信号伝達するかどうかは、ブロック寸法が現在のパレット・サイズに等しいかどうかに依存してもよい。

a ) あるいはまた、さらに、ブロック寸法が現在のパレット・サイズに等しい場

10

20

30

40

50

合、それは信号伝達されず、偽であると推定される。

2. あるいはまた、それを信号伝達するかどうかは、ブロック寸法が現在のパレット・サイズより小さくないかどうか依存してもよい。

a) あるいはまた、さらに、ブロック寸法が現在のパレット・サイズよりも小さくない場合、それは信号伝達されず、偽であると推定される。

i i . あるいはまた、さらに、エスケープ・サンプルの使用を信号伝達するかどうかは、ブロック寸法、パレットの最大サイズ、および/または無損失モードの関係に依存してもよい。

1. 一例では、あるブロックが無損失モードで符号化され、かつブロック寸法がパレットの最大サイズより小さい場合、エスケープ・サンプルの使用の信号伝達は省略されてもよく、それは偽であると推定される。

2. 一例では、あるブロックが閾値より大きくないQPで符号化され、かつブロック寸法がパレットの最大サイズより小さい場合、エスケープ・サンプルの使用の信号伝達は省略されてもよく、それは偽であると推定される。

i i i . エスケープ・サンプルの使用の指示(たとえば、`palette_escape_val_present_flag`)は、ある種の条件下では推定されてもよい。

1. 一例では、現在のブロック・サイズが最大許容されるパレット・サイズ(たとえば、`palette_max_size`)以下である場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

a) あるいはまた、一例では、現在のブロック・サイズが最大許容されるパレット・サイズより大きい場合、エスケープ・サンプルの使用の指示が信号伝達されてもよい。

b) あるいはまた、一例では、現在のブロック・サイズが最大許容されるパレット・サイズより大きい場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

2. 一例では、上記の諸方法は、無損失符号化条件の下で適用されうる。

3. 一例では、上記の諸方法は、無損失符号化されるCUに適用されうる。

4. 一例では、現在のブロック・サイズが現在のブロックのパレット・サイズ以下である場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

5. 一例では、エスケープ・サンプルの使用フラグが推定されるとき、対応する構文要素、たとえば`palette_escape_val_present_flag`はビットストリームにおいてスキップされてもよい。

【0159】

32. パレット・モードにおけるランレングス符号化のためのコンテキストは、パレット項目をインデックス付けするためのパレット・インデックスに依存してもよい。

a. 一例では、デコーダでのインデックス調整プロセス(セクション2.1.3において述べた)の後のパレット・インデックスが、長さ要素のプレフィックス(たとえば、`palette_run_prefix`)のためのコンテキストを導出するために用いられてもよい。

b. あるいはまた、一例では、項目13で定義されたIは、長さ要素のプレフィックス(たとえば、`palette_run_prefix`)のためのコンテキストを導出するために、パレット・インデックスを置き換えてもよい。

【0160】

33. 量子化パラメータ予測子のための導出プロセスにおいて使用される左近傍ブロックおよび/または上近傍ブロックの位置を、モード/MV(たとえばMPM)導出において使用される近傍左ブロックおよび/または上近傍ブロックの位置と整列させることが提案される。

a. 量子化パラメータのための導出プロセスで使用される左近傍ブロックおよび/または上近傍ブロックの位置は、マージ/AMVP候補リスト導出プロセスにおいて使用される位置と整列されてもよい。

b. 一例では、量子化パラメータのための導出プロセスにおいて使用される近傍左ブ

10

20

30

40

50

ックおよび/または上ブロックの位置は、図8に示される左/上の隣接ブロックであってもよい。

【0161】

34. ブロック・レベルのQP差は、エスケープ・サンプルが現在のブロックに存在するかどうかに関係なく、送信されてもよい。

a. 一例では、ブロック・レベルのQP差を送信するかどうか、および/またはどのように送信するかは、パレット以外のモードで符号化されたブロックに従うことができる。

b. 一例では、ブロック・レベルのQP差は、あるパレット・ブロックについて常に送信されないのでもよい。

c. 一例では、ブロック幅が閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

d. 一例では、ブロック高さが閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

e. 一例では、ブロック・サイズが閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

f. 一例では、上記の諸例はルーマ・ブロックまたはクロマ・ブロックにのみ適用される。

【0162】

35. パレット・ブロックについての符号化ブロック・フラグ(coded block flag、CBF)(たとえば、cbf\_luma、cbf\_cb、cbf\_cr)のうちの一つまたは複数に1に設定されてもよい。

a. 一例では、パレット・ブロックについてのCBFは、常に1に等しく設定されてもよい。

b. パレット・ブロックについてのCBFのうちの一つまたは複数は、エスケープ・ピクセルが現在のブロックに存在するかどうか依存してもよい。

i. 一例では、パレット・ブロックがエスケープ・サンプルを有する場合、そのcbfを1に設定されてもよい。

ii. あるいはまた、パレット・ブロックがエスケープ・サンプルを有しない場合、そのcbfは0に設定されてもよい。

c. あるいはまた、近傍のパレット符号化されたブロックにアクセスするとき、それは、1に等しいCBFを有するイントラ符号化されたブロックとして扱われてもよい。

【0163】

36. パレット・ブロックに適用されるルーマおよび/またはクロマQPと、そのブロックについて導出されるQP(たとえば、JVET-O2001-vE仕様における $Q_{pY}$ または $Q_{p'Y}$ )との間の差は、諸パレット・ブロックのための固定された値に等しく設定されてもよい。

a. 一例では、ルーマおよび/またはクロマQPオフセットが0に設定されてもよい。

b. 一例では、CbとCrについてのクロマQPオフセットが異なってもよい。

c. 一例では、ルーマQPオフセットとクロマQPオフセットは異なってもよい。

d. 一例では、クロマQPオフセット(単数または複数)がDPS/VPS/SPS/PPS/スライス/ブリック/タイル・ヘッダにおいて示されてもよい。

【0164】

37. NumPltIdxで示される、現在のブロックについて明示的に信号伝達されるまたは推測されるパレット・インデックスの数(たとえば、num\_palette\_indices\_minus1+1)は、K以上に制約されてもよい。

a. 一例では、Kは、現在のパレット・サイズ、エスケープ・フラグ、および/またはパレット符号化されたブロックの他の情報に基づいて決定されてもよい。Sを現在のブロックの現在のパレット・サイズ、Eをエスケープ存在フラグ(たとえば、palette\_escape\_val\_present\_flag)の値とする。BlkSを現在のブロック・サイズとする。

i. 一例では、KはSに等しく設定されてもよい。

10

20

30

40

50

i i . あるいはまた、一例では、Kは、 $S + E$ に等しく設定されてもよい。

i i i . あるいはまた、一例では、Kは（予測されたパレット項目の数 + 信号伝達されたパレット項目の数 + `palette_escape_val_present_flag`）（たとえば、`NumPredictedPaletteEntries + num_signalled_palette_entries + palette_escape_val_present_flag`）に等しく設定されてもよい。

i v . あるいはまた、一例では、Kは、（パレット・インデックスの最大値（たとえば、`MaxPaletteIndex`） + 1）に等しく設定されてもよい。

v . あるいはまた、一例では、Kはデコーダに信号伝達されてもよい。

i . 一例では、Kは固定された整数値であってもよい。

i i . 一例では、Kは整数であり、以下に基づいて決定されてもよい

10

- 1 . 以前に符号化されたブロック / 現在のブロックのデコードされた情報
- 2 . 現在ブロック / 近傍（隣接または非隣接）ブロックの量子化パラメータ
- 3 . ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）

4 . DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大符号化単位（`Largest coding unit`、LCU） / 符号化単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオ符号化単位において信号伝達されるメッセージ

5 . CU / PU / TU / ブロック / ビデオ符号化単位の位置

6 . 現在ブロックおよび / またはその近傍ブロックのブロック寸法

20

7 . 現在ブロックおよび / またはその近傍ブロックのブロック形状

8 . カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

9 . 符号化ツリー構造（デュアル・ツリーまたは単一ツリーなど）

10 . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

11 . 色成分（たとえば、ルーマ成分および / またはクロマ成分に対してのみ適用されうる）

12 . 時間的レイヤーID

13 . 規格のプロファイル / レベル / 層（Tier）

b . 一例では、`num_palette_indices_minus1`の代わりに（`NumPltIdx - K`）が信号伝達 / パースされてもよい。

30

i . あるいはまた、さらに、それは、（ $S + E$ ）が1より小さくない場合にのみ、信号伝達されてもよい。

i i . 一例では、（`NumPltIdx - K`）の値は、バイナリ化されたピン・ストリングがプレフィックス（たとえば、打ち切りされた単進（`truncated unary`））および / または第m EG符号をもつサフィックスを有していてもよいというバイナリ化方法で信号伝達されてもよい。

i i i . 一例では、（`NumPltIdx - K`）の値は、打ち切りされたバイナリ・バイナリ化法（`truncated binary binarization method`）で信号伝達されてもよい。

i v . 一例では、（`NumPltIdx - K`）の値は、打ち切りされた単進バイナリ化法（`truncated unary binarization method`）で信号伝達されてもよい。

40

v . 一例では、（`NumPltIdx - K`）の値は、第m EGバイナリ化法（`m-th EG binarization method`）で信号伝達されてもよい。

v i . 一例では、`BlkS - K`の値は、打ち切りされた単進 / 打ち切りされたバイナリ・バイナリ化方法に対する最大値として使用されるなど、上記の諸バイナリ化方法における入力パラメータ（たとえば、`cMax`）として使用されてもよい。

c . 一例では、適合ビットストリームは、`NumPltIdx`がK以上であることを満たす。

d . 一例では、適合ビットストリームは、`NumPltIdx`がK'以下であることを満たす。

i . 一例では、K'は（ブロック幅 \* ブロック高さ）に設定される。

i i . 一例では、K'は（ブロック幅 \* ブロック高さ - K）に設定される。

【 0 1 6 5 】

50



38. 上記の諸方法を適用するかどうか、および/または、どのように適用するかは、下記に基づいていてもよい:

a. ビデオ・コンテンツ(たとえば、スクリーン・コンテンツまたは自然コンテンツ)  
 b. DPS/SPS/VPS/PPS/APS/ピクチャー・ヘッダ/スライス・ヘッダ/タイトル・グループ・ヘッダ/最大符号化単位(Largest coding unit、LCU)/符号化単位(CU)/LCU行/LCUのグループ/TU/PUブロック/ビデオ符号化単位において信号伝達されるメッセージ

c. CU/PU/TU/ブロック/ビデオ符号化単位的位置

d. 現在ブロックおよび/またはその近傍ブロックのブロック寸法

e. 現在ブロックおよび/またはその近傍ブロックのブロック形状

f. カラー・フォーマットの指示(4:2:0、4:4:4、RGB、またはYUVなど)

g. 符号化ツリー構造(デュアル・ツリーまたは単一ツリーなど)

h. スライス/タイトル・グループ・タイプおよび/またはピクチャー・タイプ

i. 色成分(たとえば、ルーマ成分および/またはクロマ成分に対してのみ適用される)

j. 時間的レイヤーID

k. 規格のプロファイル/レベル/層(Tier)

l. 現在のブロックが1つのエスケープ・サンプルをもつか否か

i. 一例では、上記の諸方法は、現在のブロックが少なくとも、1つのエスケープ・サンプルを有する場合にのみ適用される。

m. 現在のブロックが無損失モードで符号化されるか否か(たとえば、`cu_transquant_bypass_flag`)

i. 一例では、上記の諸方法は、現在のブロックが無損失モードで符号化されない場合にのみ適用される。

n. 無損失符号化が有効にされているか否か(たとえば、`transquant_bypass_enabled`、`cu_transquant_bypass_flag`)

i. 一例では、上記の諸方法は、無損失符号化が無効にされている場合にのみ適用される。

【0166】

ライン・ベースのCGパレット・モード関連

39. これらがエスケープ・サンプルであるかどうかは、各CGについて示されてもよい。

a. 一例では、各CGについて、構文要素、たとえば`palette_escape_val_present_flag`が、エスケープ・サンプルが存在するか否かを示すためにビットストリームにおいて送られてもよい。

i. 一例では、`palette_escape_val_present_flag`は、CGサイズ、現在のブロック中のデコードされたサンプルの数、および/または現在のブロックのパレット・サイズに基づいて推定される、または信号伝達されるのもよい。

b. 一例では、現在のCGについて、エスケープ・サンプルが存在しない場合、インデックス調整が適用されてもよい。

c. 一例では、現在のCGについて、エスケープ・サンプルが存在する場合、インデックス調整は適用されるべきではない。

d. あるいはまた、上記の諸方法は、現在のブロックがエスケープ・サンプルを含む場合にのみ適用される。

【0167】

40. ライン・ベースのCGパレット・モードでは、上のインデックスをコピーの使用の指示(たとえば`copy_above_palette_indices_flag`)は、コンテキスト符号化されなくてもよい。

e. あるいはまた、一例では、上のインデックスをコピーの使用の指示(たとえば、`copy_above_palette_indices_flag`)は、いかなるコンテキストも使用することなくバイパス符号化されてもよい。

10

20

30

40

50

i . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）および現在のセグメントにおけるコピー・フラグ（たとえば、run\_copy\_flag）は、インターリーブされて信号伝達されてもよい。

f . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）は、現在のセグメントにおけるすべてのコピー・フラグ（たとえば、run\_copy\_flag）の後に符号化されてもよい。

g . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）および信号伝達されるインデックスは、インターリーブされて符号化されてもよい。

h . 上記の諸方法は、他のパレット・ベースの符号化モードにも適用されうる。

10

#### 【0168】

4 1 . コピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値は、インターリーブされて信号伝達されてもよい。

i . 一例では、第1のコピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値が順に符号化されてもよく；続いて第2のコピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値が符号化されてもよい。

j . あるいはまた、さらに、所与のCGについて、上記の方法が適用されてもよい。

#### 【0169】

4 2 . ライン・ベースのCGパレット・モードは、Thと記される所与の閾値以下のサイズのブロックについては無効にされてもよい。

20

k . 一例では、Thは、ライン・ベースのCGパレット・モードにおけるセグメントのサンプル数に等しい。

l . 一例では、Thは固定した数（たとえば16）であり、以下に基づいてもよい

i . ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）

i i . DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大符号化単位（Largest coding unit、LCU） / 符号化単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオ符号化単位において信号伝達されるメッセージ

30

i i i . CU / PU / TU / ブロック / ビデオ符号化単位の位置

i v . 現在ブロックおよび / またはその近傍ブロックのブロック寸法

v . 現在ブロックおよび / またはその近傍ブロックのブロック形状

v i . カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

v i i . 符号化ツリー構造（デュアル・ツリーまたは単一ツリーなど）

v i i i . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

i x . 色成分（たとえば、ルーマ成分および / またはクロマ成分に対してのみ適用されうる）

x . 時間的レイヤーID

x i . 規格のプロファイル / レベル / 層（Tier）

40

x i i . 現在のブロックの量子化パラメータ

x i i i . 現在のブロックが1つのエスケープ・サンプルをもつか否か

x i v . 無損失符号化が有効にされているか否か（たとえば、transquant\_bypass\_enabled、cu\_transquant\_bypass\_flag）

#### 【0170】

#### BDPCM関連

4 3 . 1つのブロックがBDPCMで符号化され、それが複数の変換ブロックまたはサブブロックに分割される場合、残差予測はブロック・レベルで行われてもよく、残差の信号伝達はサブブロック / 変換ブロック・レベルで行われる。

a . あるいはまた、さらに、あるサブブロックの再構成は、別のサブブロックの再構成

50

プロセスにおいては許可されない。

b. あるいはまた、残差予測および残差の信号伝達は、サブブロック / 変換ブロック・レベルで行われる。

i. このようにして、あるサブブロックの再構成は、別のサブブロックの再構成プロセスにおいて利用されうる。

【0171】

#### クロマQPテーブル関連

44. 所与のインデックスについて、joint\_cb\_crモードについてのクロマQPテーブルの値は、CbについてのクロマQPテーブルの値とCrについてのクロマQPテーブルの値の両方によって制約されうる。

c. 一例では、joint\_cb\_crモードについてのクロマQPテーブルの値の値は、CbについてのクロマQPテーブルの値とCrについてのクロマQPテーブルの値との間（両端含む）に制約されうる。

【0172】

#### ブロッキング解除関連

45. ブロッキング解除におけるMVの比較は、代替的な半画素補間フィルタが使用されるかどうか（たとえば、JVET-O2001-vE仕様におけるhpellIdxによって示される）に依存してもよい。

d. 一例では、異なる補間フィルタを使用するブロックは、異なるMVを有するものとして扱われてもよい。

e. 一例では、代替的な半画素補間フィルタが関与する場合、ブロッキング解除比較のためにMV差に一定のオフセットが加算されてもよい。

【0173】

#### 一般的なクレーム

46. 上記の諸方法を適用するかどうか、および / または、どのように適用するかは下記に基づいていてもよい：

a. ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）

b. DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大符号化単位（Largest coding unit、LCU） / 符号化単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオ符号化単位において信号伝達されるメッセージ

c. CU / PU / TU / ブロック / ビデオ符号化単位の位置

d. 現在ブロックおよび / またはその近傍ブロックのブロック寸法

e. 現在ブロックおよび / またはその近傍ブロックのブロック形状

f. 現在ブロックの量子化パラメータ

g. カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

h. 符号化ツリー構造（デュアル・ツリーまたは単一ツリーなど）

i. スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

j. 色成分（たとえば、ルーマ成分および / またはクロマ成分に対してのみ適用される）

k. 時間的レイヤーID

l. 規格のプロファイル / レベル / 層（Tier）

m. 現在のブロックが1つのエスケープ・サンプルをもつか否か

i. 一例では、上記の諸方法は、現在のブロックが少なくとも、1つのエスケープ・サンプルを有する場合にのみ適用されうる。

n. 現在のブロックが無損失モードで符号化されるか否か（たとえば、cu\_transquant\_bypass\_flag）

ii. 一例では、上記の諸方法は、現在のブロックが無損失モードで符号化されない場合にのみ適用されうる。

o. 無損失符号化が有効にされているか否か（たとえば、transquant\_bypass\_ena

10

20

30

40

50

bled、cu\_transquant\_bypass\_flag)

【0174】

## 5. 実施形態

実施形態は、JVET-O2001-vEに基づく。新しく追加されたテキストは、二重括弧で囲まれる。たとえば、{{a}}は、"a"が追加されたことを示す。削除されるテキストは、二重ブラケットで囲まれる。たとえば、[[b]]は、"b"が削除されることを示す。

【0175】

### 5.1 実施形態#1

#### パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (xCb,yCb)
- ・パレット・テーブルにおける最初の色成分を指定する変数startComp
- ・現在のブロックの色成分を指定する変数cldx、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数nCbwとnCbh

このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列recSamples[x][y]であり、 $x = 0..nCbw - 1$ 、 $y = 0..nCbh - 1$

【0176】

cldxの値に依存して、変数nSubWidthおよびnSubHeightは次のように導出される：

- ・cldxが0に等しい場合、nSubWidthは1に設定され、nSubHeightは1に設定される。
- ・それ以外の場合、nSubWidthはSubWidthCに設定され、nSubHeightはSubHeightCに設定される。

【0177】

位置 (xCb,yCb) における再構成されたサンプル配列recSamplesの (nCbw x nCbh) ブロックは、recSamples[x][y] ( $x = 0..nCtbw - 1$  および  $y = 0..nCbh - 1$ ) によって表され、0からnCtbw - 1の範囲 (両端含む) における各xおよび0からnCbh - 1の範囲 (両端含む) における各yについてのrecSamples[x][y]の値は、以下のように導出される：

- ・変数xLおよびyLは次のように導出される：

$xL = \text{palette\_transpose\_flag} ? x * nSubHeight : x * nSubWidth$  (8-268)

$yL = \text{palette\_transpose\_flag} ? y * nSubWidth : y * nSubHeight$  (8-269)

- ・変数blsEscapeSampleは、次のように導出される：

・PaletteIndexMap[xCb + xL][yCb + yL]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しい場合、blsEscapeSampleは1に等しく設定される。

- ・それ以外の場合、blsEscapeSampleは0に等しく設定される。

- ・blsEscapeSampleが0に等しい場合、以下が適用される：

$\text{recSamples}[x][y] = \text{CurrentPaletteEntries}[cldx][\text{PaletteIndexMap}[xCb + xL][yCb + yL]]$  (8-270)

- ・それ以外の場合、cu\_transquant\_bypass\_flagが1に等しい場合、以下が適用される：

$\text{recSamples}[x][y] = \text{PaletteEscapeVal}[cldx][xCb + xL][yCb + yL]$  (8-271)

- ・それ以外の場合 (blsEscapeSampleが1に等しく、cu\_transquant\_bypass\_flagが0に等しい)、次の順序付けられたステップが適用される：

【0178】

1. 量子化パラメータqPは次のように導出される：

- ・cldxが0に等しい場合、

$qP = \text{Max}(0, Qp'Y)$  (8-272)

- ・それ以外で、cldxが1に等しい場合、

$qP = \text{Max}(0, Qp'Cb)$  (8-273)

10

20

30

40

50

・ それ以外の場合 (  $cIdx$  が 2 に 等しい )、

$$qP = \text{Max}(0, Qp'Cr) \quad (8-274)$$

2 . 変数  $bitDepth$  は 次の よう に 導 出 さ れ る :

$$bitDepth = (cIdx == 0) ? BitDepth_y : BitDepth_c \quad (8-275)$$

3 . [[ リスト  $levelScale$  ]] は  $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$  と して 指 定 さ れ る。こ こ で、 $k = 0..5$ ]]

4 . 下 記 が 適 用 さ れ る :

$$[[ tmpVal = ( PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] * levelScale[qP \% 6] ) \quad (qP / 6) + 32 ) \quad 6 \quad (8-276) ]]$$

{{  $T$  は、成分  $cIdx$  について (  $internal\_bit\_depth - input\_bit\_depth$  ) [ 内部ビット深さ - 入力ビット深さ ] に 等 しく 設 定 さ れ る。 10

$$Nbits = \text{max}(T, (qP - 4) / 6)$$

・  $Nbits$  が  $T$  に 等 しい 場 合、

$$recSamples[x][y] = PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] \quad Nbits$$

・ それ 以 外 の 場 合、

$$recSamples[x][y] = ( PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] \quad Nbits ) + ( 1 \quad (Nbits - 1) )$$

$$[[ recSamples[x][y] = Clip3( 0, ( 1 \quad bitDepth ) - 1, tmpVal ) \quad (8-277) ]]$$

【 0 1 7 9 】

以 下 の 条 件 :

・  $cIdx$  が 0 に 等 しく、 $numComps$  が 1 に 等 しい ;

・  $cIdx$  が 2 に 等 しい

の い ず れ が 真 で あ る 場 合、

変数  $PredictorPaletteSize[startComp]$  と 配 列  $PredictorPaletteEntries$  は 次 の よう に 導 出 ま た は 修 正 さ れ る :

【 表 9 】

```
for( i = 0; i < CurrentPaletteSize[ startComp ]; i++ )
```

```
  for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ )
```

```
    newPredictorPaletteEntries[ cIdx ][ i ] = CurrentPaletteEntries[ cIdx ][ i ]
```

```
  newPredictorPaletteSize = CurrentPaletteSize[ startComp ]
```

```
  for( i = 0; i < PredictorPaletteSize && newPredictorPaletteSize <
```

```
    PaletteMaxPredictorSize; i++ )
```

```
    if( !PalettePredictorEntryReuseFlags[ i ] ) {
```

```
      for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ )
```

```
        newPredictorPaletteEntries[ cIdx ][ newPredictorPaletteSize ] =
```

```
          PredictorPaletteEntries[ cIdx ][ i ]
```

```
        newPredictorPaletteSize++
```

```
      }
```

```
  for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )
```

```
    for( i = 0; i < newPredictorPaletteSize; i++ )
```

```
      PredictorPaletteEntries[ cIdx ][ i ] = newPredictorPaletteEntries[ cIdx ][ i ]
```

```
  PredictorPaletteSize[ startComp ] = newPredictorPaletteSize
```

20

30

40

50

PredictorPaletteSize[startComp]の値が0からPaletteMaxPredictorSizeまでの範囲（両端含む）であることはビットストリーム適合性の要件である。

【0180】

### 5.2 実施形態#2

この実施形態は、パレット・インデックス導出を記載する。

#### パレット符号化のセマンティクス

[[変数adjustedRefPaletteIndex]は次のように導出される：

【表10】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev =
x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev =
y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] {      (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++

```

【0181】

#### palette\_idx\_idcについてのバイナリ化プロセス

このプロセスへの入力は、構文要素palette\_idx\_idcについてのバイナリ化の要求と変数MaxPaletteIndexである。

このプロセスの出力は、前記構文要素のバイナリ化である。

変数cMaxは次のように導出される：

- ・[[このプロセスが現在のブロックについて初めて呼び出される場合、]]cMaxはMaxPaletteIndexに等しく設定される。
- ・[[それ以外の場合（このプロセスが現在のブロックについて初めて呼び出されたのでない）、cMaxはMaxPaletteIndexから1を引いたものに等しく設定される。]]

palette\_idx\_idcについてのバイナリ化は、cMaxで節9.3.3.4で規定されているTBバイナリ化プロセスを呼び出すことによって導出される。

【0182】

### 5.3 実施形態#3

10

20

30

40

50

【表 1 1】

テーブル 9-77 – 構文要素および関連するバイナリ化

構文要素	入力パラメータ	
	プロセス	
palette_escape_val	[[EG3]] {{FL}}	{{bitDepth - (Max(QpPrimeTsMin, Qp'Y) - 4)/6}}

10

## 【 0 1 8 3】

## 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力には以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (xCb, yCb)
- ・パレット・テーブルにおける最初の色成分を指定する変数 startComp
- ・現在のブロックの色成分を指定する変数 cldx、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数 nCbW と nCbH

20

このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列 recSamples[x][y] であり、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$

## 【 0 1 8 4】

cldx の値に依存して、変数 nSubWidth および nSubHeight は次のように導出される：

....

- ・それ以外の場合 (blsEscapeSample が 1 に等しく、cu\_transquant\_bypass\_flag が 0 に等しい)、次の順序付けられたステップが適用される：

## 【 0 1 8 5】

5 . 量子化パラメータ qP は次のように導出される：

- ・ cldx が 0 に等しい場合、
- qP = Max(0, Qp'Y) (8-272)
- ・ それ以外で、cldx が 1 に等しい場合、
- qP = Max(0, Qp'Cb) (8-273)
- ・ それ以外の場合 (cldx が 2 に等しい)、
- qP = Max(0, Qp'Cr) (8-274)

30

6 . 変数 bitDepth は次のように導出される：

$$\text{bitDepth} = (\text{cldx} == 0) ? \text{BitDepth}_Y : \text{BitDepth}_C \quad (8-275)$$

3 . [[リスト levelScale[] は levelScale[k] = {40, 45, 51, 57, 64, 72} として指定される。ここで、k = 0..5]]

4 . 下記が適用される：

40

$$\text{tmpVal} = (\text{PaletteEscapeVal}[\text{cldx}][\text{xCb} + \text{xL}][\text{yCb} + \text{yL}] * \text{levelScale}[\text{qP} \% 6]) \quad (\text{qP} / 6) + 32 \quad 6 \quad (8-276)]]$$

$$\{\{\text{shift} = (\text{max}(\text{QpPrimeTsMin}, \text{qP}) - 4) / 6$$

$$\text{tmpVal} = (\text{PaletteEscapeVal}[\text{cldx}][\text{xCb} + \text{xL}][\text{yCb} + \text{yL}] \quad \text{shift}) \}}$$

$$\text{recSamples}[\text{x}][\text{y}] = \text{Clip3}(0, (1 \quad \text{bitDepth}) - 1, \text{tmpVal}) \quad (8-277)$$

## 【 0 1 8 6】

## 5.4 実施形態 #4

copy\_above\_palette\_indices\_flag が 1 に等しいことは、パレット・インデックスが、水平横断走査が使用される場合は上の行における同じ位置、または垂直横断走査が使用される場合は左の列における同じ位置にあるパレット・インデックスに等しいことを指

50

定する。copy\_above\_palette\_indices\_flagが0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいて符号化されるか、または推定されることを指定する。

… …

変数adjustedRefPaletteIndexは次のように導出される：

【表 1 2】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 {{&& !palette_escape_val_present_flag}}) {
  xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
  ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
  if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
    adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
  }
  else {
    if( !palette_transpose_flag )
      adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
    else
      adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
  }
}

```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
  CurrPaletteIndex++ (7-158)
【 0 1 8 7 】

```

5.5 実施形態#5

【表 1 3】

テーブル 9-77 – 構文要素およびその関連するバイナリ化

構文要素	入力パラメータ	
	プロセス	
palette_escape_val	[[EG3]] {{FL}}	{{max(1, bitDepth - (Max(QpPrimeTsMin, Qp^Y) - 4)/6)}}

8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力には以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (xCb, yCb)
- ・パレット・テーブルにおける最初の色成分を指定する変数startComp
- ・現在のブロックの色成分を指定する変数cIdx、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数nCbWとnCbH



このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列  $recSamples[x][y]$  であり、 $x = 0..nCbw - 1$ 、 $y = 0..nCbh - 1$

【 0 1 8 8 】

$cIdx$ の値に依存して、変数 $nSubWidth$ および $nSubHeight$ は次のように導出される：

....

・それ以外の場合 ( $blsEscapeSample$ が1に等しく、 $cu\_transquant\_bypass\_flag$ が0に等しい)、次の順序付けられたステップが適用される：

【 0 1 8 9 】

9 . 量子化パラメータ $qP$ は次のように導出される：

・  $cIdx$ が0に等しい場合、

$$qP = \text{Max}(0, Qp'Y) \quad (8-272)$$

・ それ以外で、 $cIdx$ が1に等しい場合、

$$qP = \text{Max}(0, Qp'Cb) \quad (8-273)$$

・ それ以外の場合 ( $cIdx$ が2に等しい)、

$$qP = \text{Max}(0, Qp'Cr) \quad (8-274)$$

10 . 変数 $bitDepth$ は次のように導出される：

$$bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C \quad (8-275)$$

11 . [[リスト $levelScale[]$ は $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$ として指定される。ここで、 $k = 0..5$ ]]

12 . 下記が適用される：

[[ $tmpVal = (\text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL] * levelScale[qP \% 6]) / ((qP / 6) + 32) \quad (8-276)$ ]]

[[ $shift = \min(bitDepth - 1, (\max(QpPrimeTsMin, qP) - 4) / 6)$ ]]

$tmpVal = (\text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL] \ll shift)$ ]]

$recSamples[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, tmpVal) \quad (8-277)$

【 0 1 9 0 】

## 5.6 実施形態#6

この実施形態は、変換スキップのための変換シフトをスキップするための設計を示しており、JVET-O2001-vEに基づいている。

【 0 1 9 1 】

### 8.7.2 スケーリングおよび変換プロセス

このプロセスへの入力には以下の通りである：

・現在のピクチャーの左上のルーマ・サンプルに対して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位置 ( $xTbY, yTbY$ )

・現在のブロックの色成分を指定する変数 $cIdx$

・変換ブロック幅を指定する変数 $nTbW$

・変換ブロック高さを指定する変数 $nTbH$ 。

このプロセスの出力は、 $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプルの  $(nTbW) \times (nTbH)$  配列  $resSamples[x][y]$  である。

変数 $bitDepth$ 、 $bdShift$ および $tsShift$ は、次のように導出される：

$$bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C \quad (8-942)$$

$$bdShift = \text{Max}(20 - bitDepth, 0) \quad (8-943)$$

[[ $tsShift = 5 + ((\text{Log2}(nTbW) + \text{Log2}(nTbH)) / 2) \quad (8-944)$ ]]

変数 $codedCIdx$ は次のように導出される：

・ $cIdx$ が0に等しい、または $TuCResMode[xTbY][yTbY]$ が0に等しい場合、 $codedCIdx$ は $cIdx$ に等しく設定される。

・それ以外で、 $TuCResMode[xTbY][yTbY]$ が1または2に等しい場合、 $codedCIdx$ は1に等しく設定される。

・それ以外の場合、 $codedCIdx$ は2に等しく設定される。

変数 $cSign$ は  $(1 - 2 * slice\_joint\_cbcr\_sign\_flag)$  に等しく設定される。

残差サンプルの $(nTbW) \times (nTbH)$ 配列resSampleは以下のように導出される：

【 0 1 9 2 】

1. 節8.7.3に規定される変換係数についてのスケーリング・プロセスは、変換ブロック位置 $(xTbY, yTbY)$ 、変換ブロック幅 $nTbW$ 、変換ブロック高さ $nTbH$ 、codedCIdxに等しく設定された色成分変数cIdx、および現在の色成分ビット深さbitDepthを入力として呼び出され、出力はスケーリングされた変換係数 $d$ の $(nTbW) \times (nTbH)$ 配列である。

2. 残差サンプル $r$ の $(nTbW) \times (nTbH)$ 配列は以下のように導出される：

・ [[transform\_skip\_flag[xTbY][yTbY]が1に等しく、cIdxが0に等しい場合、 $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプル配列値 $r[x][y]$ は次のように導出される：

10

$r[x][y] = d[x][y] \quad \text{tsShift} \quad (8-945)$

・ [[それ以外の場合 (transform\_skip\_flag[xTbY][yTbY]が0に等しい、またはおよびcIdxが0に等しくない場合)]節8.7.4.1で規定されたスケーリングされた変換係数のための変換プロセスが、変換ブロック位置 $(xTbY, yTbY)$ 、変換ブロック幅 $nTbW$ および変換ブロック高さ $nTbH$ 、色成分変数cIdxならびにスケーリングされた変換係数 $d$ の $(nTbW) \times (nTbH)$ 配列を入力として呼び出され、出力は、残差サンプル $r$ の $(nTbW) \times (nTbH)$ 配列である。

3.  $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、中間残差サンプル $res[x][y]$ は、次のように導出される：

20

・ {{transform\_skip\_flag[xTbY][yTbY]が1に等しく、かつcIdxが0に等しい場合、次が適用される：

$res[x][y] = d[x][y]}$

・ {{それ以外の場合 (transform\_skip\_flag[xTbY][yTbY]が0に等しい、またはおよびcIdxが0に等しくない)、次が適用される：}}

$res[x][y] = (r[x][y] + (1 \quad (bdShift - 1))) \quad bdShift \quad (8-946)$

4.  $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプル $resSamples[x][y]$ は次のように導出される：

・ cIdxがcodedCIdxに等しい場合、次が適用される：

$resSamples[x][y] = res[x][y] \quad (8-947)$

30

・ それ以外で、TuCResMode[xTbY][yTbY]が2に等しい場合、次が適用される：

$resSamples[x][y] = cSign * res[x][y] \quad (8-948)$

・ それ以外の場合、次が適用される：

$resSamples[x][y] = (cSign * res[x][y]) \quad 1$

【 0 1 9 3 】

### 8.7.3 変換係数のためのスケーリング・プロセス

....

変数rectNonTsFlagは次のように導出される：

$rect[[NonTs]]Flag = ((Log2(nTbW) + Log2(nTbH)) \& 1) == 1 [[\&\&]]$

(8-955)

40

$[[transform\_skip\_flag[xTbY][yTbY]=]] = 0)$

変数bdShift、rectNormおよびbdOffsetは次のように導出される：

・ {{transform\_skip\_flag[xTbY][yTbY]が1に等しく、かつcIdxが0に等しい場合、次が適用される：

$bdShift = 10}}$

・ {{それ以外の場合、次が適用される：}}

$bdShift = bitDepth + (rect[[NonTs]]Flag ? 1 : 0) +$

$(Log2(nTbW) + Log2(nTbH)) / 2) - 5 + dep\_quant\_enabled\_flag \quad (8-956)$

$dOffset = (1 \quad bdShift) \quad 1 \quad (8-957)$

リストlevelScale[][]は、levelScale[j][k] = {{40, 45, 51, 57, 64, 72}, {57, 64, 72,

50

80,90,102}として指定される。ここで、 $j = 0..1$ 、 $k = 0..5$ 。

$(nTbW) \times (nTbH)$  配列  $dz$  は、 $(nTbW) \times (nTbH)$  配列  $TransCoeffLevel[xTbY][yTbY][cIdx]$  に等しく設定される。

$x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$  として、スケーリングされた変換係数  $d[x][y]$  の導出については、以下が適用される：

- ・ 中間スケーリング因子  $m[x][y]$  は以下のように導出される：
  - ・ 次の条件のうちの一つまたは複数が真である場合、 $m[x][y]$  は 16 に等しく設定される：

- ・  $sps\_scaling\_list\_enabled\_flag$  が 0 に等しい。
- ・  $transform\_skip\_flag[xTbY][yTbY]$  が 1 に等しい。

10

- ・ それ以外の場合、次が適用される：

$$m[x][y] = ScalingFactor[Log2(nTbW)][Log2(nTbH)][matrixId][x][y]$$

ここで、 $matrixId$  はテーブル 7 において指定される (8-958)

- ・ スケーリング因子  $ls[x][y]$  は次のように導出される：

- ・  $dep\_quant\_enabled\_flag$  が 1 に等しい場合、次が適用される：

$$ls[x][y] = (m[x][y] * levelScale[rect[[NonTs]]Flag][(qP + 1) \% 6]) \quad ((qP + 1) / 6) \quad (8-960)$$

- ・ それ以外の場合 ( $dep\_quant\_enabled\_flag$  が 0 に等しい)、次が適用される：

$$ls[x][y] = (m[x][y] * levelScale[rect[[NonTs]]Flag][qP \% 6]) \quad (qP / 6) \quad (8-961)$$

20

- ・  $BdpcmFlag[xTbY][yYbY]$  が 1 に等しい場合、 $dz[x][y]$  は次のように修正される：

- ・  $BdpcmDir[xTbY][yYbY]$  が 0 に等しく、 $x$  が 0 より大きい場合、次が適用される：

$$dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x - 1][y] + dz[x][y]) \quad (8-961)$$

- ・ それ以外で、 $BdpcmDir[xTbY][yYbY]$  が 1 に等しく、 $y$  が 0 より大きい場合、次が適用される：

$$dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x][y - 1] + dz[x][y]) \quad (8-962)$$

- ・ 値  $dnc[x][y]$  は次のように導出される：

$$dnc[x][y] = (dz[x][y] * ls[x][y] + bdOffset) \quad bdShift \quad (8-963)$$

- ・ 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る：

$$d[x][y] = Clip3(CoeffMin, CoeffMax, dnc[x][y]) \quad (8-964)$$

30

【 0 1 9 4 】

### 5.7 実施形態 #7

この実施形態は、パレット・インデックスの数を信号伝達するための設計を示す。

【 0 1 9 5 】

#### 7.3.8.6 パレット符号化構文

40

50

【表 1 4】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
...		
if( MaxPaletteIndex > 0 ) {		
<b>num_palette_indices</b> {_{diff}}[_minus1]	ae(v)	
adjust = 0		
for( i = 0; i <= [[num_palette_indices_minus1]][NumPaletteIndices]; i++ ) {		10
if( MaxPaletteIndex - adjust > 0 ) {		
<b>palette_idx_idc</b>	ae(v)	
PaletteIndexIdc[ i ] = palette_idx_idc		
}		
adjust = 1		
}		
<b>copy_above_indices_for_final_run_flag</b>	ae(v)	
<b>palette_transpose_flag</b>	ae(v)	20
}		
if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {		
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {		
<b>cu_qp_delta_abs</b>	ae(v)	
if( cu_qp_delta_abs )		
<b>cu_qp_delta_sign_flag</b>	ae(v)	
}		30
}		
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {		
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {		
<b>cu_chroma_qp_offset_flag</b>	ae(v)	
if( cu_chroma_qp_offset_flag )		
<b>cu_chroma_qp_offset_idx</b>	ae(v)	
}		40

<pre> } remainingNumIndices = [[num_palette_indices_minus1 + 1]][[NumPaletteIndices]] ... if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {     currNumIndices = [[num_palette_indices_minus1 + 1]][[NumPaletteIndices]] - remainingNumIndices     PaletteIndexMap[ xC ][ yC ] = PaletteIndexIdc[ currNumIndices ] } ... } </pre>	
---	--

10

num\_palette\_indices{{\_diff}}[\_minus1]に[[1]] ( {{MaxPaletteIndex + 1}} ) を加えたものは、現在のブロックについて明示的に信号伝達されるまたは推定されるパレット・インデックスの数である。

20

{{NumPaletteIndices}}は ( num\_palette\_indices\_diff + MaxPaletteIndex + 1 ) に設定される。}}

num\_palette\_indices{{\_diff}}[\_minus1]が存在しない場合は、0に等しいと推定される。

{{num\_palette\_indices\_diffの値は、0からcbWidth\*cbHeight - ( MaxPaletteIndex + 1 ) ( 両端含む ) の範囲とする。}}

copy\_above\_indices\_for\_final\_run\_flagが1に等しいことは、符号化単位における最後の諸位置のパレット・インデックスが、水平横断走査が使用される場合は上の行におけるパレット・インデックスから、垂直横断走査が使用される場合は左の列におけるパレット・インデックスからコピーされることを指定する。

30

copy\_above\_indices\_for\_final\_run\_flagが0に等しいことは、符号化単位における最後の諸位置のパレット・インデックスがPaletteIndexIdc[ [[num\_palette\_indices\_minus1]][[NumPaletteIndices - 1]] ]からコピーされることを指定する。

【 0 1 9 6 】

### 9.5.3.13 num\_palette\_indices{{\_diff}}[\_minus1]についてのバイナリ化プロセス

このプロセスへの入力は、構文要素num\_palette\_indices{{\_diff}}[\_minus1]についてのバイナリ化の要求およびMaxPaletteIndexである。

このプロセスの出力は、前記構文要素のバイナリ化である。

40

変数cRiceParamは次のように導出される：

$$cRiceParam = 3 + ( ( MaxPaletteIndex + 1 ) - 3 ) \quad (9-26)$$

変数cMaxは、cRiceParamから次のように導出される：

$$cMax = 4 - cRiceParam \quad (9-27)$$

構文要素num\_palette\_indices{{\_diff}}[\_minus1]のバイナリ化は、プレフィックスのピン・ストリングと ( 存在する場合は ) サフィックスのピン・ストリングの連結である。

プレフィックス・ピン・ストリングの導出のためには、次が適用される：

・ num\_palette\_indices{{\_diff}}[\_minus1]のプレフィックス値prefixValは次のように導出される：

50

$prefixVal = \text{Min}(cMax, num\_palette\_indices\{\{\_diff\}\}[\[_minus1\]])$  (9-28)

・プレフィックス・ビン・ストリングは、 $prefixVal$ について節9.3.3.3で規定されるTRバイナリ化プロセスを、変数 $cMax$ と $cRiceParam$ を入力として呼び出すことによって指定される。

プレフィックス・ビン・ストリングが長さ4ですべてのビットが1に等しいビット・ストリングに等しい場合、サフィックス・ビン・ストリングが存在し、それは次のように導出される：

・ $num\_palette\_indices\{\{\_diff\}\}[\[_minus1\]]$ のサフィックス値 $suffixVal$ は次のように導出される：

$suffixVal = num\_palette\_indices\{\{\_diff\}\}[\[_minus1\]] - cMax$  (9-29)

・サフィックス・ビン・ストリングは、指数ゴロム次数 $k$ を $cRiceParam + 1$ に等しく設定して、 $suffixVal$ のバイナリ化について節9.3.3.5で規定された $k$ 次のEGkバイナリ化プロセスを呼び出すことによって指定される。

#### 【表15】

テーブル9-77 – 構文要素および関連するバイナリ化

構文構造	構文要素	バイナリ化	
		プロセス	入力パラメータ
<code>palette_coding()</code>	$num\_palette\_indices\{\{\_diff\}\}[\[_minus1\]]$	9.5.3.13	MaxPaletteIndex

テーブル9-82 –  $ctxInc$  のコンテキスト符号化されたビンを用いた構文要素への割り当て

構文要素	binIdx					
	0	1	2	3	4	>= 5
$num\_palette\_indices\{\{\_diff\}\}[\[_minus1\]]$	バイパス	バイパス	バイパス	バイパス	バイパス	バイパス

#### 【0197】

#### 5.8 実施形態#8

この実施形態は、ライン・ベースのCGパレット・モードにおけるインターリーブされた信号伝達の設計を示す。本実施形態は、JVET-P2001-v4で提供されるドラフトに基づく。

10

20

30

40

50

【表 1 6】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子
palettePredictionFinished = 0	
...	
PreviousRunPosition = 0	
PreviousRunType = 0	
for( subSetId = 0; subSetId <= ( cbWidth * cbHeight - 1 ) / 16; subSetId++ ) {	10
minSubPos = subSetId * 16	
if( minSubPos + 16 > cbWidth * cbHeight)	
maxSubPos = cbWidth * cbHeight	
else	
maxSubPos = minSubPos + 16	
RunCopyMap[ x0 ][ y0 ] = 0	
PaletteScanPos = minSubPos	20
log2CbWidth = Log2( cbWidth )	
log2CbHeight = Log2( cbHeight )	
while( PaletteScanPos < maxSubPos ) {	
x <sub>C</sub> = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
y <sub>C</sub> = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	
if( PaletteScanPos > 0 ) {	
x <sub>C</sub> Prev = x0 +	30
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]	
y <sub>C</sub> Prev = y0 +	
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]	
}	
if( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {	
<b>run_copy_flag</b>	ac(v)
	40

RunCopyMap[ xC ][ yC ] = run_copy_flag	
}	
CopyAboveIndicesFlag[ xC ][ yC ] = 0	
if( MaxPaletteIndex > 0 && !RunCopyMap[ xC ][ yC ] ) {	
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC > 0 ) ) && CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {	10
<b>copy_above_palette_indices_flag</b>	ae(v)
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag	
}	
PreviousRunType = CopyAboveIndicesFlag[ xC ][ yC ]	
PreviousRunPosition = curPos	
} else {	
CopyAboveIndicesFlag[ xC ][ yC ] = CopyAboveIndicesFlag[ xcPrev ][ ycPrev ]	20
}	
{{ if( MaxPaletteIndex > 0 ) {	
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
if( MaxPaletteIndex - adjust > 0 ) {	
palette_idx_idc	{{ae(v)}}}
}	
adjust = 1	
}	30
}	
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
CurrPaletteIndex = palette_idx_idc	
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex	
} else {	
if( !palette_transpose_flag )	40

10

20

30

40

50



PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ]		
else		
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ]		
}		
}		
if( palette_escape_val_present_flag ) {		
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++){		10
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]		
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {		
<b>palette_escape_val</b>		ac(v)
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val		
}		
}		
}}		20
PaletteScanPos ++		
}		
[[PaletteScanPos = minSubPos		
while( PaletteScanPos < maxSubPos ) {		
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]		
if( PaletteScanPos > 0 ) {		30
xcPrev	=x0 +	
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev	=	y0 +
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
if( MaxPaletteIndex > 0 ) {		40

if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
if( MaxPaletteIndex - adjust > 0 ) {	
<b>palette_idx_idc</b>	[[ac(v)]]
}	
adjust = 1	
}	
}	
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
CurrPaletteIndex = palette_idx_idc	
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex	
} else {	
if( !palette_transpose_flag )	
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ]	
else	
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ]	
}	
}	
if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )	
for( sPos = minSubPos; sPos < maxSubPos; sPos++ ) {	
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]	
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]	
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {	
<b>palette_escape_val</b>	[[ac(v)]]
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val	
}	
}	
}	
}	

10

20

30

40

}]]	
}	

【 0 1 9 8 】

## 5.9 実施形態#9

変更はJVET-P2001-vEに基づく。

【 0 1 9 9 】

50

### 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在の符号化ブロックの左上のサンプルを指定する位置 (xCbComp, yCbComp)
- ・単一ツリーが使用されるかデュアル・ツリーが使用されるかを指定する変数treeType。デュアル・ツリーが使用される場合、これは、現在のツリーがルーマ成分に対応するかクロマ成分に対応するかを指定する。
- ・現在のブロックの色成分を指定する変数cldx、
- ・現在の符号化ブロックの幅および高さをそれぞれ指定する2つの変数nCbwおよびnCbH

10

このプロセスの出力は、 $x = 0..nCbw - 1$ 、 $y = 0..nCbH - 1$ として、ブロックについての再構成されたサンプル値を指定する配列recSamples[x][y]である。

#### 【0200】

treeTypeの値に依存して、変数startCompおよびnumCompsは次のように導出される：

- ・treeTypeがSINGLE\_TREEに等しい場合：
  - startComp = 0 (444)
  - numComps = 3 (445)
- ・それ以外の場合、treeTypeはDUAL\_TREE\_LUMAに等しい：
  - startComp = 0 (446)
  - numComps = 1 (447)
- ・それ以外の場合、treeTypeはDUAL\_TREE\_CHROMAに等しい：
  - startComp = 1 (448)
  - numComps = 2 (449)

20

#### 【0201】

cldxの値に依存して、変数nSubWidthおよびnSubHeightは次のように導出される：

・cldxが0より大きく、startCompが0に等しい場合、nSubWidthはSubWidthCに設定され、nSubHeightはSubHeightCに設定される。

・それ以外の場合は、nSubWidthは1に設定され、nSubHeightは1に設定される。

位置 (xCbComp, yCbComp) における再構成されたサンプル配列recSamplesの(nCbW × nCbH)ブロックは、recSamples[x][y] (ここで、 $x = 0..nCbW - 1$ および $y = 0..nCbH - 1$ ) によって表され、0からnCbW - 1の範囲 (両端含む) における各xおよび0からnCbH - 1の範囲 (両端含む) における各yについてのrecSamples[x][y]の値は、次のように導出される：

30

・変数xL、yL、xCbL、yCbLは次のように導出される：

$$xL = x * nSubWidth \quad (450)$$

$$yL = y * nSubHeight \quad (451)$$

$$xCbL = xCbComp * nSubWidth \quad (452)$$

$$yCbL = yCbComp * nSubHeight \quad (453)$$

・変数blsEscapeSampleは、次のように導出される：

40

・PaletteIndexMap[xCbL + xL][yCbL + yL]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しい場合、blsEscapeSampleは1に等しく設定される。

#### 【0202】

・それ以外の場合、blsEscapeSampleは0に設定される。

・blsEscapeSampleが0に等しい場合、以下が適用される：

recSamples[x][y] = CurrentPaletteEntries[cldx][PaletteIndexMap[xCbL + xL][yCbL + yL]] (454)

・それ以外の場合 (blsEscapeSampleが1に等しい)、以下の順序付けられたステップが適用される：

50

1. 量子化パラメータ $qP$ は次のように導出される：

- ・  $cldx$ が0に等しい場合、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Y)$  (455)

- ・ それ以外で、 $cldx$ が1に等しい場合、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Cb)$  (456)

- ・ それ以外の場合 ( $cldx$ は2に等しい)、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Cr)$  (457)

2. リスト $levelScale[]$ は、 $k = 0..5$ として、 $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$ として指定される。

3. 次が適用される：

```

{{shift = Min(bitDepth - 1, (QpPrimeTsMin - 4) / 6)}}
[[tmpVal = (PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] *
  levelScale[qP%6]) (qP/6) + 32) 6 (458)]]
{{tmpVal = ((PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] shift) *
  levelScale[(qP - QpPrimeTsMin + 4) % 6]) ((qP - QpPrimeTsMin +
  4) / 6) + 32) 6 (458)}}
recSamples[x][y] = Clip3(0, (1 - BitDepth) - 1, tmpVal) (459)

```

【0203】

#### 5.10 実施形態#10

変更はJVET-P2001-vEに基づく。

【0204】

#### 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・ 現在のピクチャーの左上のルーマ・サンプルに対して現在の符号化ブロックの左上のサンプルを指定する位置 ( $xCbComp, yCbComp$ )

- ・ 単一ツリーが使用されるかデュアル・ツリーが使用されるかを指定する変数 $treeType$ 。デュアル・ツリーが使用される場合、これは、現在のツリーがルーマ成分に対応するかクロマ成分に対応するかを指定する。

- ・ 現在のブロックの色成分を指定する変数 $cldx$ 、

- ・ 現在の符号化ブロックの幅および高さをそれぞれ指定する2つの変数 $nCbW$ および $nCbH$

このプロセスの出力は、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$ として、ブロックについての再構成されたサンプル値を指定する配列 $recSamples[x][y]$ である。

【0205】

$treeType$ の値に依存して、変数 $startComp$ および $numComps$ は次のように導出される：

- ・  $treeType$ が $SINGLE\_TREE$ に等しい場合：

$startComp = 0$  (444)

$numComps = 3$  (445)

- ・ それ以外の場合、 $treeType$ は $DUAL\_TREE\_LUMA$ に等しい：

$startComp = 0$  (446)

$numComps = 1$  (447)

- ・ それ以外の場合、 $treeType$ は $DUAL\_TREE\_CHROMA$ に等しい：

$startComp = 1$  (448)

$numComps = 2$  (449)

【0206】

$cldx$ の値に依存して、変数 $nSubWidth$ および $nSubHeight$ は次のように導出される：

- ・  $cldx$ が0より大きく、 $startComp$ が0に等しい場合、 $nSubWidth$ は $SubWidthC$ に設定され、 $nSubHeight$ は $SubHeightC$ に設定される。

- ・ それ以外の場合、 $nSubWidth$ は1に設定され、 $nSubHeight$ は1に設定される。

10

20

30

40

50

位置 (xCbComp, yCbComp) における再構成されたサンプル配列 recSamples の (nCbw × nCbH) ブロックは、recSamples[x][y] (ここで、x = 0..nCtbW - 1 および y = 0..nCbH - 1) によって表され、0 から nCtbW - 1 の範囲 (両端含む) における各 x および 0 から nCbH - 1 の範囲 (両端含む) における各 y についての recSamples[x][y] の値は、次のように導出される：

- ・変数 xL、yL、xCbL、yCbL は次のように導出される：

$$xL = x * nSubWidth \quad (450)$$

$$yL = y * nSubHeight \quad (451)$$

$$xCbL = xCbComp * nSubWidth \quad (452)$$

$$yCbL = yCbComp * nSubHeight \quad (453)$$

10

- ・変数 blsEscapeSample は、次のように導出される：

・PaletteIndexMap[xCbL + xL][yCbL + yL] が MaxPaletteIndex に等しく、palette\_escape\_val\_present\_flag が 1 に等しい場合、blsEscapeSample は 1 に等しく設定される。

【0207】

- ・それ以外の場合、blsEscapeSample は 0 に設定される。

- ・blsEscapeSample が 0 に等しい場合、以下が適用される：

recSamples[x][y] = CurrentPaletteEntries[cldx][PaletteIndexMap[xCbL + xL][yCbL + yL]] (454)

・それ以外の場合 (blsEscapeSample が 1 に等しい)、以下の順序付けられたステップが適用される： 20

【0208】

4. 量子化パラメータ qP は次のように導出される：

- ・cldx が 0 に等しい場合、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Y) \quad (455)$$

- ・それ以外で、cldx が 1 に等しい場合、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Cb) \quad (456)$$

- ・それ以外の場合 (cldx は 2 に等しい)、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Cr) \quad (457)$$

5. リスト levelScale[] は、k = 0..5 として、levelScale[k] = {40, 45, 51, 57, 64, 72} 30 } として指定される。

6. 次が適用される：

{{shift = Min(bitDepth - 1, (QpPrimeTsMin - 4) / 6)}}

[[tmpVal = (PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] \* levelScale[qP%6]) (qP/6) + 32) 6 (458)]]

{{qP' = Max(0, qP - 6 \* shift)}}

tmpVal = ((PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] \* shift) \* levelScale[qP'%6]) (qP'/6) + 32) 6 (458)}}

recSamples[x][y] = Clip3(0, (1 - BitDepth) - 1, tmpVal) (459)

【0209】

40

図9は、ビデオ処理装置900のブロック図である。装置900は、本明細書に記載される方法のうちの一つまたは複数を実装するために使用されてもよい。装置900は、スマートフォン、タブレット、コンピュータ、モノのインターネット (IoT) 受信機などにおいて具現されてもよい。装置900は、一つまたは複数のプロセッサ902、一つまたは複数のメモリ904、およびビデオ処理ハードウェア906を含んでいてもよい。プロセッサ902は、本稿に記載される一つまたは複数の方法を実装するように構成されてもよい。メモリ (単数または複数) 904は、本明細書に記載される方法および技法を実装するために使用されるデータおよびコードを記憶するために使用されてもよい。ビデオ処理ハードウェア906は、ハードウェア回路において、本稿に記載されるいくつかの技法を実装するために使用されてもよい。いくつかの実施形態では、ハードウェア906は、プロセッサ9 50

02、たとえばグラフィックスコプロセッサに対して少なくとも部分的に内部であってもよい。

【0210】

開示された技術のいくつかの実施形態は、ビデオ処理ツールまたはモードを有効にする判断または決定を行うことを含む。一例では、ビデオ処理ツールまたはモードが有効にされている場合、エンコーダは、ビデオのブロックの処理においてそのツールまたはモードを使用または実装するが、必ずしもそのツールまたはモードの使用に基づく、結果として生じるビットストリームを修正しなくてもよい。すなわち、ビデオのブロックからビデオのビットストリーム表現への変換は、前記判断または決定に基づいて有効にされているとき、そのビデオ処理ツールまたはモードを使用する。別の例では、ビデオ処理ツールまたはモードが有効にされている場合、デコーダは、ビットストリームがそのビデオ処理ツールまたはモードに基づいて修正されているという知識を用いて、そのビットストリームを処理する。すなわち、ビデオのビットストリーム表現からビデオのブロックへの変換は、前記判断または決定に基づいて有効にされたビデオ処理ツールまたはモードを使用して実行される。

10

【0211】

開示された技術のいくつかの実施形態は、ビデオ処理ツールまたはモードを無効にする判断または決定を行うことを含む。一例では、ビデオ処理ツールまたはモードが無効にされている場合、エンコーダは、ビデオのブロックをビデオのビットストリーム表現に変換する際に、そのツールまたはモードを使用しない。別の例では、ビデオ処理ツールまたはモードが無効にされている場合、デコーダは、ビットストリームが、判断または決定に基づいて有効にされたビデオ処理ツールまたはモードを使用して修正されていないという知識を用いて、そのビットストリームを処理する。

20

【0212】

図10は、本明細書に開示されるさまざまな技法が実装されうる例示的なビデオ処理システム1000を示すブロック図である。さまざまな実装は、システム1000の構成要素の一部または全部を含んでいてもよい。システム1000は、ビデオ・コンテンツを受信するための入力1002を含んでいてもよい。ビデオ・コンテンツは、生または非圧縮フォーマット、たとえば、8または10ビットの多成分ピクセル値の形で受領されてもよく、または圧縮またはエンコードされたフォーマットで受領されてもよい。入力1002は、ネットワークインターフェース、周辺バスインターフェース、または記憶インターフェースを表すことができる。ネットワークインターフェースの例は、イーサネット、受動光ネットワーク(PON)などの有線インターフェース、およびWi-Fiまたはセルラーインターフェースなどの無線インターフェースを含む。

30

【0213】

システム1000は、本稿に記載されるさまざまな符号化またはエンコード方法を実装しうる符号化コンポーネント1004を含んでいてもよい。符号化コンポーネント1004は、入力1002から符号化コンポーネント1004の出力へ、ビデオの平均ビットレートを低減して、ビデオの符号化表現を生成してもよい。よって、符号化技法は、時に、ビデオ圧縮またはビデオトランスコード技法と呼ばれる。符号化コンポーネント1004の出力は、コンポーネント1006によって表されるように接続された通信を介して、記憶されるか、または送信されてもよい。入力1002で受信されたビデオの記憶されたまたは通信されたビットストリーム(または符号化された)表現は、ディスプレイインターフェース1010に送られるピクセル値または表示可能なビデオを生成するために、コンポーネント1008によって使用されうる。ビットストリーム表現からユーザーが見ることができるビデオを生成するプロセスは、時に、ビデオ圧縮解除と呼ばれる。さらに、ある種のビデオ処理動作は、「符号化」操作またはツールと称されるが、符号化ツールまたは操作は、エンコーダにおいて使用され、符号化の結果を反転させる対応する復号ツールまたは操作がデコーダにおいて実行されることが理解されるであろう。

40

【0214】

50

周辺バスインターフェースまたはディスプレイインターフェースの例は、ユニバーサルシリアルバス（USB）または高精細度マルチメディアインターフェース（MDMI）またはディスプレイポートなどを含む。記憶インターフェースの例は、SATA（シリアルアドバンスドテクノロジータッチメント）、PCI、IDEインターフェースなどを含む。本稿に記載される技法は、携帯電話、ラップトップ、スマートフォン、またはデジタルデータ処理および/またはビデオ表示を実行することができる他の装置のようなさまざまな電子装置において具現されうる。

【0215】

図11は、本開示の技表を利用することができる例示的なビデオ符号化システム100を示すブロック図である。

10

【0216】

図11に示されるように、ビデオ符号化システム100は、源装置110および宛先装置120を含んでいてもよい。源装置110は、ビデオ・エンコード装置と称されてもよく、エンコードされたビデオ・データを生成する。宛先装置120は、ビデオ・デコード装置と称されてもよく、源装置110によって生成されたエンコードされたビデオ・データをデコードしてもよい。

【0217】

源装置110は、ビデオ源112、ビデオ・エンコーダ114、および入出力（I/O）インターフェース116を含むことができる。

【0218】

ビデオ源112は、ビデオ捕捉装置のような源、ビデオ・コンテンツ・プロバイダーからビデオ・データを受信するためのインターフェース、および/またはビデオ・データを生成するためのコンピュータ・グラフィックス・システム、またはそのような源の組み合わせを含んでいてもよい。ビデオ・データは、一つまたは複数のピクチャーを含んでいてもよい。ビデオ・エンコーダ114は、ビデオ源112からのビデオ・データをエンコードしてビットストリームを生成する。ビットストリームは、ビデオ・データの符号化表現を形成するビットのシーケンスを含んでいてもよい。ビットストリームは、符号化されたピクチャーおよび関連するデータを含んでいてもよい。符号化ピクチャーは、ピクチャーの符号化表現である。関連するデータは、シーケンスパラメータセット、ピクチャーパラメータセット、および他の構文構造を含んでいてもよい。I/Oインターフェース116は、変調器/復調器（モデム）および/または送信器を含んでいてもよい。エンコードされたビデオ・データは、ネットワーク130aを通じてI/Oインターフェース116を介して宛先装置120に直接送信されてもよい。エンコードされたビデオ・データはまた、宛先装置120によるアクセスのために記憶媒体/サーバー130b上に記憶されてもよい。

20

30

【0219】

宛先装置120は、I/Oインターフェース126、ビデオ・デコーダ124、および表示装置122を含んでいてもよい。

【0220】

I/Oインターフェース126は、受信機および/またはモデムを含んでいてもよい。I/Oインターフェース126は、源装置110または記憶媒体/サーバー130bから、エンコードされたビデオ・データを取得してもよい。ビデオ・デコーダ124は、エンコードされたビデオ・データをデコードしてもよい。表示装置122は、デコードされたビデオ・データをユーザーに対して表示してもよい。表示装置122は、宛先装置120と一体化されてもよく、または外部表示装置とインターフェースするように構成された宛先装置120の外部にあってもよい。

40

【0221】

ビデオ・エンコーダ114およびビデオ・デコーダ124は、高効率ビデオ符号化（HEVC）標準、多用途ビデオ符号化（VVC）標準、およびその他の現行のおよび/またはさらなる標準のようなビデオ圧縮標準に従って動作してもよい。

【0222】

50

図12は、図11に示されるシステム100内のビデオ・エンコーダ114であってもよいビデオ・エンコーダ200の一例を示すブロック図である。

【0223】

ビデオ・エンコーダ200は、本開示の技法のいずれかまたはすべてを実行するように構成されてもよい。図12の例では、ビデオ・エンコーダ200は、複数の機能コンポーネントを含む。本開示に記載される技法は、ビデオ・エンコーダ200のさまざまなコンポーネント間で共有されてもよい。いくつかの例では、プロセッサが、本開示に記載される技法のいずれかまたはすべてを実行するように構成されてもよい。

【0224】

ビデオ・エンコーダ200の機能コンポーネントは、分割部201と、モード選択部203、動き推定部204、動き補償部205、およびイントラ予測部206を含みうる予測部202と、残差生成部207と、変換部208と、量子化部209と、逆量子化部210と、逆変換部211と、再構成部212と、バッファ213と、エン트로ピー符号化部214とを含んでいてもよい。

10

【0225】

他の例では、ビデオ・エンコーダ200は、より多くの、より少ない、または異なる機能コンポーネントを含んでいてもよい。一例では、予測部202は、イントラブロックコピー（IBC）部を含んでいてもよい。IBC部は、少なくとも1つの参照ピクチャーが現在のビデオ・ブロックが位置するピクチャーであるIBCモードで予測を実行することができる。

20

【0226】

さらに、動き推定部204や動き補償部205のようないくつかのコンポーネントは、高度に統合されてもよいが、説明のために図12の例では別個に表現されている。

【0227】

分割部201は、ピクチャーを一つまたは複数のビデオ・ブロックにパーティション分割することができる。ビデオ・エンコーダ200およびビデオ・デコーダ300は、さまざまなビデオ・ブロック・サイズをサポートすることができる。

【0228】

モード選択部203は、たとえば誤差結果に基づいて、符号化モードのうちの1つ、イントラまたはインターを選択し、結果として得られるイントラまたはインター符号化されたブロックを、残差ブロック・データを生成する残差生成部207に、また、参照ピクチャーとして使用するためにエンコードされたブロックを再構成する再構成部212に提供することができる。いくつかの例では、モード選択部203は、予測がインター予測信号およびイントラ予測信号に基づく、イントラおよびインター予測の組み合わせ（CIIP）モードを選択してもよい。モード選択部203は、また、インター予測の場合、ブロックについての動きベクトルの分解能（たとえば、サブピクセルまたは整数ピクセル精度）を選択してもよい。

30

【0229】

現在のビデオ・ブロックに対してインター予測を実行するために、動き推定部204は、バッファ213からの一つまたは複数の参照フレームを現在のビデオ・ブロックと比較することによって、現在のビデオ・ブロックについての動き情報を生成してもよい。動き補償部205は、現在のビデオ・ブロックに関連するピクチャー以外のバッファ213からのピクチャーのデコードされたサンプルと前記動き情報とに基づいて、現在のビデオ・ブロックについて予測されるビデオ・ブロックを決定することができる。

40

【0230】

動き推定部204と動き補償部205は、たとえば、現在のビデオ・ブロックがIスライス内にあるか、Pスライス内にあるか、Bスライス内にあるかによって、現在のビデオ・ブロックについて異なる動作を実行してもよい。

【0231】

いくつかの例では、動き推定部204は、現在のビデオ・ブロックについて一方向の予

50



測を実行してもよく、動き推定部204は、現在のビデオ・ブロックについての参照ビデオ・ブロックを求めて、リスト0またはリスト1の参照ピクチャーを探索してもよい。次いで、動き推定部204は、参照ビデオ・ブロックを含むリスト0またはリスト1内の参照ピクチャーを示す参照インデックスと、現在のビデオ・ブロックと参照ビデオ・ブロックとの間の空間的変位を示す動きベクトルとを生成することができる。動き推定部204は、参照インデックス、予測方向指示子、および動きベクトルを現在のビデオ・ブロックの動き情報として出力することができる。動き補償部205は、現在のビデオ・ブロックの動き情報によって示される参照ビデオ・ブロックに基づいて、現在のブロックの予測されたビデオ・ブロックを生成することができる。

#### 【0232】

10

他の例では、動き推定部204は、現在のビデオ・ブロックについて双方向予測を実行してもよく、動き推定部204は、現在のビデオ・ブロックについての参照ビデオ・ブロックを求めてリスト0内の参照ピクチャーを探索してもよく、現在のビデオ・ブロックについての別の参照ビデオ・ブロックを求めてリスト1内の参照ピクチャーをも探索してもよい。次いで、動き推定部204は、参照ビデオ・ブロックを含むリスト0およびリスト1における参照ピクチャーを示す参照インデックスと、参照ビデオ・ブロックと現在のビデオ・ブロックとの間の空間的変位を示す動きベクトルとを生成してもよい。動き推定部204は、現在のビデオ・ブロックの参照インデックスおよび動きベクトルを、現在のビデオ・ブロックの動き情報として、出力することができる。動き補償部205は、現在のビデオ・ブロックの動き情報によって示される参照ビデオ・ブロックに基づいて、現在のビデオ・ブロックの予測ビデオ・ブロックを生成することができる。

20

#### 【0233】

いくつかの例では、動き推定部204は、デコーダのデコード処理のための動き情報のフルセットを出力してもよい。

#### 【0234】

いくつかの例では、動き推定部204は、現在のビデオについての動き情報のフルセットは出力しなくてもよい。むしろ、動き推定部204は、別のビデオ・ブロックの動き情報を参照して現在のビデオ・ブロックの動き情報を信号伝達してもよい。たとえば、動き推定部204は、現在のビデオ・ブロックの動き情報が近傍のビデオ・ブロックの動き情報と十分に類似していると判断することができる。

30

#### 【0235】

一例では、動き推定部204は、現在のビデオ・ブロックに関連する構文構造において、現在のビデオ・ブロックが別のビデオ・ブロックと同じ動き情報を有することをビデオ・デコーダ300に対して示す値を示してもよい。

#### 【0236】

別の例では、動き推定部204は、現在のビデオ・ブロックに関連する構文構造において、別のビデオ・ブロックおよび動きベクトル差(MVD)を同定してもよい。動きベクトル差は、現在のビデオ・ブロックの動きベクトルと示されたビデオ・ブロックの動きベクトルとの間の差を示す。ビデオ・デコーダ300は、示されたビデオ・ブロックの動きベクトルと動きベクトル差とを使用して、現在のビデオ・ブロックの動きベクトルを決定することができる。

40

#### 【0237】

上述のように、ビデオ・エンコーダ200は、動きベクトルを予測的に信号伝達してもよい。ビデオ・エンコーダ200によって実装されうる予測的な信号伝達技法の2つの例は、先進動きベクトル予測(AMVP)およびマージ・モード信号伝達を含む。

#### 【0238】

イントラ予測部206は、現在のビデオ・ブロックに対してイントラ予測を実行することができる。イントラ予測部206が現在のビデオ・ブロックに対してイントラ予測を実行するとき、イントラ予測部206は、同じピクチャー内の他のビデオ・ブロックのデコードされたサンプルに基づいて、現在のビデオ・ブロックのための予測データを生成する

50

ことができる。現在のビデオ・ブロックのための予測データは、予測されるビデオ・ブロックおよびさまざまな構文要素を含んでいてもよい。

【0239】

残差生成部207は、現在のビデオ・ブロックから現在のビデオ・ブロックの予測ビデオ・ブロックを差し引くことによって（たとえば、マイナス記号によって示される）、現在のビデオ・ブロックのための残差データを生成することができる。現在のビデオ・ブロックの残差データは、現在のビデオ・ブロック内のサンプルの異なるサンプル成分に対応する残差ビデオ・ブロックを含んでいてもよい。

【0240】

他の例では、たとえばスキップ・モードにおいて、現在のビデオ・ブロックのための現在のビデオ・ブロックのための残差データが存在しないことがあり、残差生成部207は減算演算を実行しないことがある。 10

【0241】

変換処理部208は、現在のビデオ・ブロックに関連する残差ビデオ・ブロックに一つまたは複数の変換を適用することによって、現在のビデオ・ブロックのための一つまたは複数の変換係数ビデオ・ブロックを生成することができる。

【0242】

変換処理部208が現在のビデオ・ブロックに関連する変換係数ビデオ・ブロックを生成した後、量子化部209は、現在のビデオ・ブロックに関連する一つまたは複数の量子化パラメータ（QP）値に基づいて、現在のビデオ・ブロックに関連する変換係数ビデオ・ブロックを量子化することができる。 20

【0243】

逆量子化部210および逆変換部211は、変換係数ビデオ・ブロックから残差ビデオ・ブロックを再構成するために、変換係数ビデオ・ブロックに対して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。再構成部212は、再構成された残差ビデオ・ブロックを、予測部202によって生成された一つまたは複数の予測ビデオ・ブロックからの対応するサンプルに加算して、バッファ213に記憶するために現在のブロックに関連する再構成されたビデオ・ブロックを生成する。

【0244】

再構成部212がビデオ・ブロックを再構成した後、ビデオ・ブロック内のビデオ・ブロックリング・アーチファクトを低減するためにループ・フィルタリング動作が実行されてもよい。 30

【0245】

エントロピー符号化部214は、ビデオ・エンコーダ200の他の機能コンポーネントからデータを受領することができる。エントロピー符号化部214がデータを受領すると、エントロピー符号化部214は、一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、エントロピー符号化データを含むビットストリームを出力することができる。

【0246】

図13は、図11に示されるシステム100内のビデオ・デコーダ114であってもよいビデオ・デコーダ300の例を示すブロック図である。 40

【0247】

ビデオ・デコーダ300は、本開示の技法のいずれかまたはすべてを実行するように構成されうる。図13の例では、ビデオ・デコーダ300は、複数の機能的コンポーネントを含む。本開示に記載される技法は、ビデオ・デコーダ300のさまざまなコンポーネント間で共有されてもよい。いくつかの例では、プロセッサが、本開示に記載される技法のいずれかまたはすべてを実行するように構成されてもよい。

【0248】

図13の例では、ビデオ・デコーダ300は、エントロピー復号部301、動き補償部302、イントラ予測部303、逆量子化部304、逆変換部305、再構成部306、バッファ307 50

を含む。ビデオ・デコーダ300は、いくつかの例では、ビデオ・エンコーダ200（図12）に関して説明したエンコード・パス（pass）とおおむね逆のデコード・パスを実行することができる。

【0249】

エントロピー復号部301は、エンコードされたビットストリームを取り出すことができる。エンコードされたビットストリームはエントロピー符号化されたビデオ・データ（たとえば、ビデオ・データのエンコードされたブロック）を含むことができる。エントロピー復号部301はエントロピー符号化されたビデオ・データを復号することができ、エントロピー復号されたビデオ・データから、動き補償部302は、動きベクトル、動きベクトル精度、参照ピクチャー・リスト・インデックス、および他の動き情報を含む動き情報を決定することができる。動き補償部302は、たとえば、AMVPおよびマージ・モードを実行することによって、そのような情報を決定することができる。

【0250】

動き補償部302は、可能性としては補間フィルタに基づく補間を実行して、動き補償されたブロックを生成することができる。サブピクセル精度で使用される補間フィルタのための識別子が、構文要素に含まれてもよい。

【0251】

動き補償部302は、ビデオ・ブロックのエンコード中にビデオ・エンコーダ200によって使用される補間フィルタを使用して、参照ブロックの整数未満ピクセルについての補間された値を計算することができる。動き補償部302は、受領された構文情報に従ってビデオ・エンコーダ200によって使用される補間フィルタを決定し、該補間フィルタを使用して予測ブロックを生成することができる。

【0252】

動き補償部302は、構文情報の一部を使用して、エンコードされたビデオ・シーケンスのフレームおよび/またはスライスをエンコードするために使用されるブロックのサイズ、エンコードされたビデオ・シーケンスのピクチャーの各マクロブロックがどのようにパーティション分割されるかを記述する分割情報、各パーティションがどのようにエンコードされるかを示すモード、各インター符号化されたブロックについての一つまたは複数の参照フレーム（および参照フレーム・リスト）、およびエンコードされたビデオ・シーケンスをデコードするための他の情報を決定することができる。

【0253】

イントラ予測部303は、たとえばビットストリームにおいて受領されたイントラ予測モードを使用して、空間的に隣接するブロックから予測ブロックを形成してもよい。逆量子化部303は、ビットストリーム内で提供され、エントロピー復号部301によって復号された量子化ビデオ・ブロック係数を逆量子化する、すなわち、脱量子化する。逆変換部303は、逆変換を適用する。

【0254】

再構成部306は、残差ブロックを、動き補償部202またはイントラ予測部303によって生成された対応する予測ブロックと加算して、デコードされたブロックを形成することができる。所望であれば、ブロック性アーチファクトを除去するために、デコードされたブロックをフィルタリングするようブロック解除フィルタも適用されてもよい。次いで、デコードされたビデオ・ブロックはバッファ307に格納され、バッファ307は、その後の動き補償/イントラ予測のための参照ブロックを提供し、表示装置上に提示するためのデコードされたビデオをも生成する。

【0255】

いくつかの実施形態では、以下の方法は、上記に列挙した例および実施形態のリストに基づく。一例では、これらの方法は、図9～図13に示される実装を使用して実施できるが、これに限定されない。

【0256】

図14は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるよ

うに、方法1400は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1410)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

【0257】

10

図15は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1500は、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1510)を含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化され、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モード符号化ツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される

20

図16は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1600は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1610)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長符号化を使用し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

【0258】

30

図17は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1700は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1710)を含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0259】

40

図18は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1800は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行すること(1810)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

【0260】

50

図19は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1900は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1910)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずに符号化され、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

【0261】

図20は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2000は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2010)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化され、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

10

【0262】

図21は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2100は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2110)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの導出されるパレット・インデックスは最大値をもち、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

20

【0263】

図22は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2200は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2210)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

30

【0264】

図23は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2300は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2310)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいて符号化される構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

40

【0265】

図24は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2400は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2410)を含

50

み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、予測的に符号化されるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

【0266】

図25は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2500は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行すること(2510)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングス符号化される構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

10

【0267】

図26は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2600は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2610)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

20

【0268】

図27は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2700は、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ・ブロックの符号化モードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップ(2710)と、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップ(2720)とを含む。

30

【0269】

図28は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2800は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行すること(2810)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ(QP)差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される。

40

【0270】

図29は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2900は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2910)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数の符号化ブロック・フラグ(CBF)を含む構文要素を使用して表現されるというフォーマット規則に

50

適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0271】

図30は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3000は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(3010)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数(NumPltIdx)は、K以上であり、前記パレット・モード符号化ツ

10

【0272】

図31は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3100は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(3110)を含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ(QP)に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード

20

【0273】

図32は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3200は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調(BDPCM)モードで符号化され、複数の変換ブロックまたはサブブロックに分割されることを決定するステップ(3210)と、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることとを実行するステップ(3220)とを含む。

30

【0274】

図33は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3300は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行すること(3310)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ(CG)パレット・モードを使用して符号化されるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各符号化単位(CU)の複数のセグメントを表す。

【0275】

40

以下の解決策は、いくつかの実施形態の好ましい特徴として、前のセクションに列挙されている項目(たとえば、項目1)に記載されている追加の技法とともに実装されてもよい。

【0276】

1. ビデオ処理の方法であって：現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モード符号化ツ

50

ルは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【0277】

2.  $K = 0$ である、解決策1の方法。

【0278】

3.  $K = 1$ である、解決策1の方法。

【0279】

4.  $K = 2$ である、解決策1の方法。

【0280】

ビデオ処理の方法であって、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化され、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モード符号化ツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

10

【0281】

6. 前記固定長バイナリ化は、 $N$ ビットを使用し、 $N$ は1より大きい整数である、解決策5の方法。

20

【0282】

7.  $N$ が、内部ビット深さに基づく、解決策6の方法。

【0283】

8.  $N$ の値が、スライス・サブピクチャー、タイル、ピクチャー、またはビデオにおいて信号伝達される、解決策6の方法。

【0284】

9.  $N$ が量子化パラメータに基づいている、解決策6の方法。

【0285】

10.  $N$ が、 $f(Qp)$ として示される量子化パラメータ( $Qp$ )の関数( $f()$ )に基づく、解決策9の方法。

30

【0286】

11.  $N$ が $(ibd - \max(16, (Qp - 4)/6))$ に設定され、ここで、 $ibd$ が内部ビット深さである、解決策9の方法。

【0287】

12.  $N$ が $(ibd - \max(QpPrimeTsMin, (Qp - 4)/6))$ に設定され、ここで、 $ibd$ は内部ビット深さであり、 $QpPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータである、解決策9の方法。

【0288】

13.  $N$ が $\max(A, (ibd - \max(16, (QpPrimeTsMin - 4)/6)))$ に設定され、ここで、 $ibd$ は内部ビット深さであり、 $QpPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータであり、 $A$ は負でない整数である、解決策9の方法。

40

【0289】

14.  $A = 0$ または $A = 1$ である、解決策13の方法。

【0290】

15. 前記量子化パラメータが、前記ビデオのスライスの量子化パラメータと定数値との和であり、前記定数値が整数値である、解決策9ないし14のうちいずれか一項の方法。

【0291】

50



16. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長符号化を使用し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【0292】

17. 前記可変長符号化が次数3の指数ゴロム符号を除外する、解決策16の方法。

10

【0293】

18. 前記可変長符号化が、入力パラメータKをもつ打ち切りされたバイナリ(TB)符号であり、Kは整数である、解決策16の方法。

【0294】

19. Kは、(a)シーケンスパラメータセット(SPS)、ビデオパラメータセット(VPS)、ピクチャーパラメータセット(PPS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大符号化単位(LCU)行、LCUのグループ、またはブリック、(b)内部ビット深さ、(c)入力ビット深さ、(d)内部ビット深さと入力ビット深さとの間の相違、(e)現在ビデオ・ブロックの寸法、(f)現在ビデオ・ブロックの現在の量子化パラメータ、(g)前記ビデオのカラー・フォーマットの指示、(h)符号化ツリー構造、または(i)前記ビデオの色成分に基づいている、解決策18の方法。

20

【0295】

20. 前記エスケープ・シンボルの複数の値が、複数のバイナリ化方法を使用して信号伝達される、解決策5の方法。

【0296】

21. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化されるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

30

【0297】

ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

40

【0298】

23. 前記量子化が右ビットシフトに対応する、解決策22の方法。

【0299】

24. 前記エスケープ・シンボルが $f(p, Qp)$ として符号化され、 $f()$ は関数であり、 $p$ は入力シンボル値であり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、解決策22の方法。

【0300】

25.  $f$ が $p$   $g(Qp)$ として定義される、解決策24の方法。

50

- 【0301】  
26.  $f$ が、 $(p + (1 - (g(QP) - 1))) \cdot g(Qp)$ として定義される、解決策24の方法。
- 【0302】  
27.  $f$ は、 $\text{clip}(0, (1 - bd) - 1, (p + (1 - (g(QP) - 1))) \cdot g(Qp))$ として定義され、 $\text{clip}(x, \min, \max)$ はクリッピング関数であり、 $x$ 、 $\min$ 、 $\max$ は整数である、解決策24の方法。
- 【0303】  
28. 前記エスケープ・シンボルは $h(p)$ として符号化され、 $h()$ は関数であり、 $p$ は入力値シンボルである、解決策22の方法。 10
- 【0304】  
29.  $h$ が $p \cdot N$ として定義され、 $N$ は負でない整数である、解決策28の方法。
- 【0305】  
30.  $h$ は、 $(p + (1 - (N - 1))) \cdot N$ として定義され、 $N$ は負でない整数である、解決策28の方法。
- 【0306】  
31.  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = 0$ である、解決策29または30の方法。
- 【0307】  
32.  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = (bd - ibd)$ であり、 $bd$ は内部ビット深さであり、 $ibd$ は入力ビット深さである、解決策29または30の方法。 20
- 【0308】  
33.  $h$ は、 $\text{clip}(0, (1 - (bd - N) - 1, p \cdot N)$ として定義され、ここで、 $bd$ は、現在のビデオ・ブロックの現在の色成分の内部ビット深さであり、 $N$ は負でない整数であり、 $\text{clip}(x, \min, \max)$ は、クリッピング関数であり、 $x$ 、 $\min$ 、および $\max$ は整数である、解決策28の方法。
- 【0309】  
34.  $h$ は、 $\text{clip}(0, (1 - (bd - N) - 1, (p + (1 - (N - 1))) \cdot N)$ として定義され、ここで、 $bd$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $N$ は、負でない整数であり、 $\text{clip}(x, \min, \max)$ はクリッピング関数であり、 $x$ 、 $\min$ 、 $\max$ は整数である、解決策28の方法。 30
- 【0310】  
35.  $N$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策29ないし34のうちいずれか一項の方法。
- 【0311】  
36. 前記脱量子化は、左ビットシフトに対応する、解決策22の方法。
- 【0312】  
37. 前記エスケープ・シンボルは、 $f(p, Qp)$ として脱量子化され、ここで、 $f()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルであり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、解決策36の方法。 40
- 【0313】  
38.  $f$ が $p \cdot g(Qp)$ として定義される、解決策37の方法。
- 【0314】  
39. 前記エスケープ・シンボルは $f(p, Qp)$ として再構成され、ここで、 $f()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルであり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、解決策36の方法。
- 【0315】 50

40.  $f$ は、 $\text{clip}(0, (1 - \text{bd}) - 1, p - g(Qp))$ として定義され、 $\text{bd}$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $\text{clip}(x, \text{min}, \text{max})$ はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $\text{max}$ は整数である、解決策39の方法。

【0316】

41. クリッピング関数 $\text{clip}(x, \text{min}, \text{max})$ は

【数2】

$$\text{clip}(x, \text{min}, \text{max}) = \begin{cases} \text{min} & x < \text{min} \\ x & \text{min} \leq x \leq \text{max} \\ \text{max} & x > \text{max} \end{cases}$$

10

として定義される、解決策27、33、34または40の方法。

【0317】

42. 前記エスケープ・シンボルは $h(p)$ として再構成され、 $h()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルである、解決策36の方法。

【0318】

43.  $h$ は $p - N$ として定義され、 $N$ は負でない整数である、解決策42の方法。

【0319】

44.  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = 0$ である、解決策42または43の方法。

20

【0320】

45.  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = (\text{bd} - \text{ibd})$ であり、 $\text{bd}$ は内部ビット深さであり、 $\text{ibd}$ は入力ビット深さである、解決策42または43の方法。

【0321】

46.  $N = (\max(\text{QpPrimeTsMin}, \text{qP}) - 4) / 6$ であって、 $\text{qP}$ はデコードされた量子化パラメータであり、 $\text{QpPrimeTsMin}$ は変換スキップ・モードのための最小の許容される量子化パラメータである、解決策42または43の方法。

【0322】

47.  $N$ がさらに $\min(\text{bd} - 1, N)$ としてクリップされ、 $\text{bd}$ が現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策43ないし46のうちいずれか一項の方法。

30

【0323】

48.  $N$ が $[0, (\text{bd} - 1)]$ の範囲内であり、 $\text{bd}$ は現在のビデオ・ブロックの現在の色成分の内部ビット深さである、解決策43ないし47のうちいずれか一項の方法。

【0324】

49. 前記エスケープ・シンボルの再構成オフセットは、ビット深さ情報に基づく、解決策36の方法。

【0325】

50. 前記ビット深さ情報は、内部ビット深さと入力ビット深さとの間の差 ( $\text{BD}$ と記される) を含む、解決策49の方法。

40

【0326】

51.  $K - \text{BD}$ である場合、再構成されたオフセットは $p - K$ に等しく、ここで、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ は、整数である、解決策50の方法。

【0327】

52. 前記再構成されたオフセットは、 $K - T0$ の場合、 $p - K$ に等しく、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ および $T0$ は整数である、解決策49の方法。

【0328】

53.  $T0 = 2$ である、解決策50の方法。

【0329】

50

54. 再構成されたオフセットは、 $(p - K) + ((1 - (K - 1)) \cdot B_D - B_D)$ に等しく、 $p$ はデコードされたエスケープ・シンボルであり、 $K$ は整数である、解決策50の方法。

【0330】

55.  $B_D$ が、シーケンス・レベル、ピクチャー・レベル、スライス・レベル、タイル・レベル、ブリック・レベル、またはサブピクチャー・レベルでビットストリーム表現において信号伝達される、解決策50の方法。

【0331】

56. 前記エスケープ・シンボルがコンテキスト符号化される、解決策22ないし55のうちいずれか一項の方法。

【0332】

57. 前記エスケープ・シンボルがバイパス符号化される、解決策22ないし55のうちいずれか一項の方法。

【0333】

58.  $g(Q_p)$ が $(Q_p - 4)/6$ として定義される、解決策25ないし27、38または40のうちいずれか一項の方法。

【0334】

59.  $g(Q_p)$ が $(\max(M, Q_p) - 4)/6$ として定義され、 $M$ は整数である、解決策25ないし27、39または40のうちいずれか一項の方法。

【0335】

60.  $M$ が、シーケンスパラメータセット (SPS) において信号伝達される、解決策59の方法。

【0336】

61.  $g(Q_p)$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ が、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策58ないし60のうちいずれか一項の方法。

【0337】

62. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずに符号化され、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

【0338】

63. 前記一つまたは複数のパレット・インデックスのバイナリ化が、バイナリ化入力パラメータとして最大パレット・インデックスを有する打ち切りされたバイナリ (TB) 符号である、解決策62の方法。

【0339】

64. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを使用して符号化され、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【0340】

65. 前記非エスケープ・シンボルのインデックスから前記エスケープ・シンボルのインデックスを導出することが許されない、解決策64の方法。

10

20

30

40

50

## 【 0 3 4 1 】

66．前記エスケープ・シンボルのインデックスから前記非エスケープ・シンボルのインデックスを導出することが許されない、解決策 64 の方法。

## 【 0 3 4 2 】

67．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モード符号化ツールを用いて符号化されるというフォーマット規則に適合し、前記パレット・モード符号化ツールの導出されるパレット・インデックスは最大値をもち、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

10

## 【 0 3 4 3 】

68．前記最大値は、現在のパレット・テーブル・サイズである、解決策 67 の方法。

## 【 0 3 4 4 】

69．前記最大値は、一つまたは複数のエスケープ・シンボルについてのインデックスを除外する現在のパレット・テーブル・サイズであり、前記一つまたは複数のエスケープ・シンボルのうちのあるエスケープ・シンボルが、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、解決策 67 の方法。

## 【 0 3 4 5 】

70．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

20

## 【 0 3 4 6 】

71．前記インデックスが  $MaxPaletteIndex$  に等しい、解決策 70 の方法。

30

## 【 0 3 4 7 】

72．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいて符号化される構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

## 【 0 3 4 8 】

73．前記現在インデックスと前記参照インデックスとの間の差が符号化される、解決策 72 の方法。

40

## 【 0 3 4 9 】

74．前記差の符号化された表現は、ゼロ値の差を除外する、解決策 73 の方法。

## 【 0 3 5 0 】

75．前記現在インデックスと前記参照インデックスとの間の差のモジュロが符号化される、解決策 72 の方法。

## 【 0 3 5 1 】

76．前記モジュロが  $l = \text{modulo}(C - R, MaxPaletteIndex)$  として表され、 $C$  は前記現在インデックスであり、 $R$  は前記参照インデックスであり、 $MaxPaletteIndex$  は所定の負でない整数である、解決策 75 の方法。

50

## 【 0 3 5 2 】

77. 前記パレット・モード符号化ツールのパレット・ブロックの始めにおいて、前記参照インデックスが - 1 に設定される、解決策 7 2 の方法。

## 【 0 3 5 3 】

78. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、予測的に符号化されるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

10

## 【 0 3 5 4 】

79. 前記エスケープ・シンボルは、以前に符号化されたエスケープ・シンボルに基づいて予測的に符号化される、解決策 7 8 の方法。

## 【 0 3 5 5 】

80. 前記ビデオのある色成分における前記エスケープ・シンボルは、同じ色成分における値に基づいて予測的に符号化される、解決策 7 8 の方法。

## 【 0 3 5 6 】

81. 前記ビデオの第 1 の色成分における前記エスケープ・シンボルは、前記第 1 の色成分とは異なる前記ビデオの第 2 の色成分における値に基づいて予測的に符号化される、解決策 7 8 の方法。

20

## 【 0 3 5 7 】

82. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを用いて符号化される現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングス符号化される構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

30

## 【 0 3 5 8 】

83. 長さ要素のプレフィックスについてのコンテキストは、デコーダにおけるインデックス調整プロセスの後のパレット・インデックスに基づく、解決策 8 2 の方法。

## 【 0 3 5 9 】

84. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

40

## 【 0 3 6 0 】

85. 前記前の諸パレット・インデックスを使用することは、現在のビデオ・ブロックが一つまたは複数のエスケープ・シンボルを含むかどうかに基づき、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのためにエスケープ・シンボルが使用される、解決策 8 4 の方法。

## 【 0 3 6 1 】

86. ビデオ処理の方法であって、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第 1 の近傍ビデオ・ブロックと、現在のビデオ・ブロックの符号化モードを予測的に決

50

定するために使用される第2の近傍ビデオ・ブロックとを決定するステップと、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップとを含む、方法。

【0362】

87. 前記第1の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、解決策86の方法。

【0363】

88. 前記第2の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、解決策86または87の方法。

【0364】

89. 前記符号化モードが、現在のビデオ・ブロックについての最確モード (most probable mode、MPM) を含む、解決策86ないし88のうちいずれか一項の方法。

【0365】

90. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ (QP) 差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、方法。

【0366】

91. 前記QP差が、閾値よりも大きい幅を有するパレット・ブロックについて符号化される、解決策90の方法。

【0367】

92. 前記QP差が、閾値よりも高い高さを有するパレット・ブロックについて符号化される、解決策90の方法。

【0368】

93. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数の符号化ブロック・フラグ (CBF) を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

【0369】

94. 前記CBFのそれぞれが1に等しく設定される、解決策93の方法。

【0370】

95. 前記一つまたは複数のCBFの値は、前記現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかに基づいており、前記エスケープ・シンボルは、前記代表的な色値を使用せずに符号化される前記現在のビデオ・ブロックのサンプルのために使用される、解決策93の方法。

【0371】

96. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数 (NumPltIdx) は、K以上であり、前記パレット・モード符号

10

20

30

40

50

化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、 $K$ は正の整数である、方法。

【0372】

97.  $K$ は、現在のパレット・サイズ ( $S$ )、エスケープ・フラグ ( $E$ )、または現在のビデオ・ブロックのサイズ ( $BkS$ ) に基づく、解決策 96 の方法。

【0373】

98.  $K = S + E$  である、解決策 97 の方法。

【0374】

99.  $K$ が、パレット・インデックスの最大値 ( $MaxPaletteIndex$ ) に1を加えたものに等しい、解決策 96 の方法。

10

【0375】

100. 前記構文要素の1つが  $NumPltIdx - K$  を含む、解決策 96 の方法。

【0376】

101. ( $NumPltIdx - K$ ) の値のバイナリ化が、打ち切りされたバイナリ符号である、解決策 100 の方法。

【0377】

102. ( $NumPltIdx - K$ ) の値のバイナリ化が打ち切りされた単進符号である、解決策 100 の方法。

【0378】

103. ( $BkS - K$ ) がバイナリ化入力パラメータであり、 $BkS$ が現在のビデオ・ブロックのサイズである、解決策 101 または 102 の方法。

20

【0379】

104. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モード符号化ツールを使用して符号化される現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ ( $QP$ ) に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モード符号化ツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

30

【0380】

105. 無損失モードが適用されたこと、 $QP$ が閾値より大きいこと、または変換スキップが適用されたことを判別したとき、現在のブロックのためのパレットのサイズは、現在のビデオ・ブロックのサイズに等しいと推定される、解決策 104 の方法。

【0381】

106. 前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット ( $DPS$ )、シーケンスパラメータセット ( $SPS$ )、ビデオパラメータセット ( $VPS$ )、ピクチャーパラメータセット ( $PPS$ )、適応パラメータセット ( $APS$ )、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大符号化単位 ( $LCU$ )、符号化単位 ( $CU$ )、 $LCU$ 行、 $LCU$ のグループ、変換単位 ( $TU$ )、予測単位 ( $PU$ ) ブロック、またはビデオ符号化単位において信号伝達されるメッセージ、前記ビデオのカラー・フォーマットの指示、符号化ツリー構造、時間的IDレイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、解決策 104 または 105 のうちいずれか一項の方法。

40

【0382】

107. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調 ( $BDPCM$ ) モードで符号化され、複数の変換ブロックまたはサブブロックに分割されることを決定するステップと、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に

50



基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることを実行するステップとを含む、方法。

【0383】

108. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ (CG) パレット・モードを使用して符号化されるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各符号化単位 (CU) の複数のセグメントを表す、方法。

10

【0384】

109. 前記ビットストリーム表現は、各係数グループについてエスケープ・サンプルが存在するかどうかの指示を含み、エスケープ・サンプルは、前記代表的な色値を使用せずに符号化される現在のビデオ・ブロックのサンプルのために使用される、解決策108の方法。

【0385】

110. 前記ビットストリーム表現は、コンテキスト符号化されない上のインデックスをコピーの使用の指示を含む、解決策108の方法。

【0386】

111. 前記指示がバイパス符号化される、解決策110の方法。

20

【0387】

112. 一つまたは複数のコピー・フラグ、一つまたは複数のラン・タイプ、上のインデックスをコピーすること使用の一つまたは複数の指示、およびエスケープ値が、前記ビットストリーム表現において、インターリーブされた仕方では信号伝達される、解決策108の方法。

【0388】

113. 前記ライン・ベースのCGパレット・モードは、前記現在のビデオ・ブロックのサイズが閾値 (Th) 以下であると判定されると、無効にされる、解決策108の方法。

【0389】

114. 前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット (DPS)、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、適応パラメータセット (APS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大符号化単位 (LCU)、符号化単位 (CU)、LCU行、LCUのグループ、変換単位 (TU)、予測単位 (PU) ブロック、またはビデオ符号化単位において信号伝達されるメッセージ、前記ビデオのカラー・フォーマットの指示、符号化ツリー構造、時間的IDレイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、解決策107ないし113のうちいずれか一項の方法。

30

【0390】

115. 前記変換を実行することが、前記一つまたは複数のビデオ領域から前記ビットストリーム表現を生成することを含む、解決策1ないし114のうちいずれか一項の方法。

40

【0391】

116. 前記変換を実行することが、前記ビットストリーム表現から前記一つまたは複数のビデオ領域を生成することを含む、解決策1ないし114のうちいずれか一項の方法。

【0392】

117. プロセッサと、命令を有する非一時メモリとを備えるビデオ・システム内の装置であって、前記命令は、前記プロセッサによる実行の際に、前記プロセッサに、解決策1ないし116のうちいずれか一項の方法を実施させる、装置。

50

## 【0393】

118. 非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ・プログラム・プロダクトであって、解決策1ないし116のうちいずれか一項の方法を実行するためのプログラム・コードを含む、コンピュータ・プログラム・プロダクト。

## 【0394】

本明細書に記載された、開示されたおよびその他の解決策、例、実施形態、モジュール、および機能的な動作は、デジタル電子回路において、または本明細書に開示された構造およびそれらの構造的等価物を含むコンピュータ・ソフトウェア、ファームウェア、またはハードウェアにおいて、またはそれらの一つまたは複数の組み合わせにおいて実装されることができる。開示されたおよびその他の実施形態は、一つまたは複数のコンピュータ・プログラム・プロダクト、すなわち、データ処理装置による実行のため、またはデータ処理装置の動作を制御するために、コンピュータ読み取り可能媒体上にエンコードされたコンピュータ・プログラム命令の一つまたは複数のモジュールとして実装されることができる。コンピュータ読み取り可能媒体は、機械読み取り可能な記憶デバイス、機械読み取り可能な記憶基板、メモリ装置、機械読み取り可能な伝搬信号を実現する物質の組成、またはそれらの一つまたは複数の組み合わせでありうる。用語「データ処理装置」は、たとえば、プログラム可能なプロセッサ、コンピュータ、または複数のプロセッサまたはコンピュータを含む、データを処理するためのすべての装置、デバイス、および機械を包含する。装置は、ハードウェアに加えて、問題のコンピュータ・プログラムのための実行環境を生成するコード、たとえば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはそれらの一つまたは複数の組み合わせを構成するコードを含むことができる。伝搬信号は、人工的に生成された信号、たとえば、好適な受信器装置に送信するための情報をエンコードするために生成される機械生成された電気信号、光信号、または電磁信号である。

## 【0395】

コンピュータ・プログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、またはコードとしても知られる）は、コンパイルされるまたはインタープリットされる言語を含む、任意の形のプログラミング言語で書くことができ、それは、スタンドアロン・プログラムとして、またはコンピューティング環境での使用に好適なモジュール、コンポーネント、サブルーチン、または他のユニットとして展開することを含め、任意の形で展開できる。コンピュータ・プログラムは、必ずしもファイルシステム内のファイルに対応するものではない。プログラムは、他のプログラムまたはデータを保持するファイルの一部に（たとえば、マークアップ言語文書に格納される一つまたは複数のスクリプト）、問題のプログラム専用の単一のファイルに、または複数の協調させられるファイル（たとえば、一つまたは複数のモジュール、サブプログラム、またはコード部分を格納するファイル）に格納されることができる。コンピュータ・プログラムは、1つのコンピュータ上で、または1つのサイトに位置するか、または複数のサイトに分散され、通信ネットワークによって相互接続される複数のコンピュータ上で実行されるように展開されることができる。

## 【0396】

本稿に記載されるプロセスおよび論理フローは、一つまたは複数のコンピュータ・プログラムを実行する一つまたは複数のプログラム可能なプロセッサによって実行されて、入力データに対して作用し、出力を生成することによって機能を実行することができる。プロセスおよび論理フローは、特殊目的の論理回路、たとえばFPGA（フィールドプログラマブルゲートアレイ）またはASIC（特定用途向け集積回路）によって実行されることもでき、装置もそのようなものとして実装されることができる。

## 【0397】

コンピュータ・プログラムの実行に好適なプロセッサは、たとえば、汎用および専用マイクロプロセッサの両方、および任意の種類のデジタル・コンピュータの任意の一つまたは複数のプロセッサを含む。一般に、プロセッサは、読み出し専用メモリまたはランダム

10

20

30

40

50

アクセスメモリまたはその両方から命令およびデータを受領する。コンピュータの必須の要素は、命令を実行するためのプロセッサと、命令およびデータを記憶するための一つまたは複数のメモリデバイスである。一般に、コンピュータはまた、データを記憶するための一つまたは複数の大容量記憶装置、たとえば、磁気ディスク、磁気光ディスク、または光ディスクをも含む、またはそれらからデータを受領したり、またはそれにデータを転送したり、またはその両方をしたりするよう、動作上結合される。しかしながら、コンピュータは、そのような装置を有する必要はない。コンピュータ・プログラム命令およびデータを記憶するのに好適なコンピュータ読み取り可能媒体は、あらゆる形の不揮発性メモリ、媒体およびメモリデバイスを含み、たとえば、半導体メモリデバイス、たとえばEPROM、EEPROM、およびフラッシュメモリデバイス；磁気ディスク、たとえば、内蔵ハードディスクまたはリムーバブルディスク；光磁気ディスク；ならびにCD-ROMおよびDVD-ROMディスクを含む。プロセッサおよびメモリは、特殊目的論理回路によって補足されるか、またはそれに組み込まれることができる。

10

**【0398】**

この特許文献は多くの個別的事項を含んでいるが、これらは、いずれかの主題の範囲または特許請求されうるものの範囲に対する限定として解釈されるべきではなく、むしろ特定の技法の特定の実施形態に特有でありうる特徴の記述と解釈されるべきである。この特許文献において別個の実施形態の文脈で記載されているある種の特徴は、単一の実施形態において組み合わせで実施されることもできる。逆に、単一の実施形態の文脈において記載されるさまざまな特徴が、複数の実施形態において別個に、または任意の好適なサブコンビネーションにおいて実施されることもできる。さらに、特徴は、ある種の組み合わせにおいて作用するものとして上述され、さらには当初はそのようなものとして請求項に記載されることがあるが、請求項に記載された組み合わせからの一つまたは複数の特徴が、場合によっては、組み合わせから削除されてもよく、請求項に記載された組み合わせは、サブコンビネーションまたはサブコンビネーションの変形に向けられてもよい。

20

**【0399】**

同様に、図面には特定の順序で動作が示されているが、これは、所望の結果を達成するために、そのような動作が示されている特定の順序で、あるいは逐次順に実行されること、あるいはすべての示されている動作が実行されることを要求すると解釈されるべきではない。さらに、この特許文献に記載されている実施形態におけるさまざまなシステムコンポーネントの分離は、すべての実施形態においてそのような分離を必要とするものとして理解されるべきではない。

30

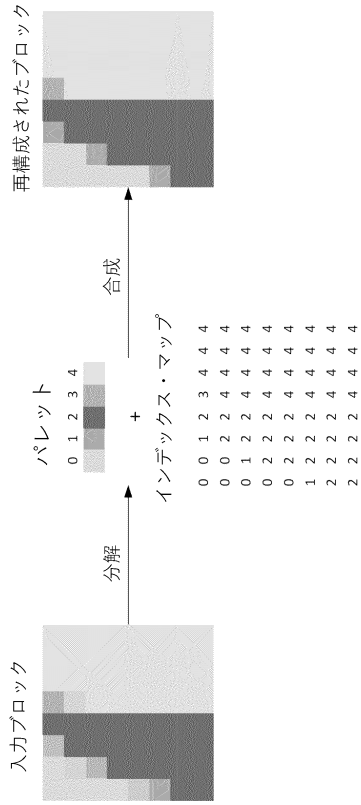
**【0400】**

少数の実装および例のみが記述されており、この特許文献に記載され、説明されている内容に基づいて、他の実装、向上および変形がなされてもよい。

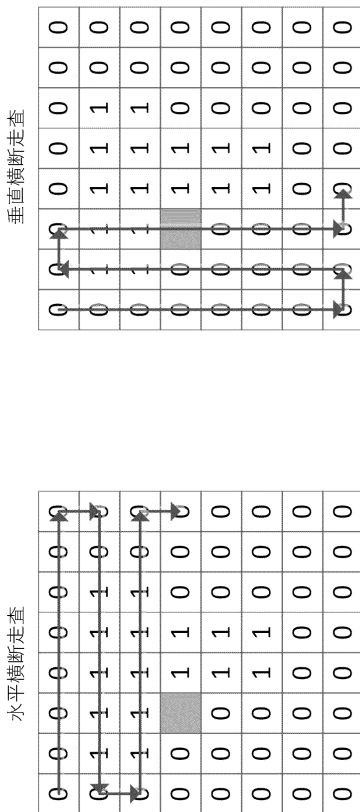
40

50

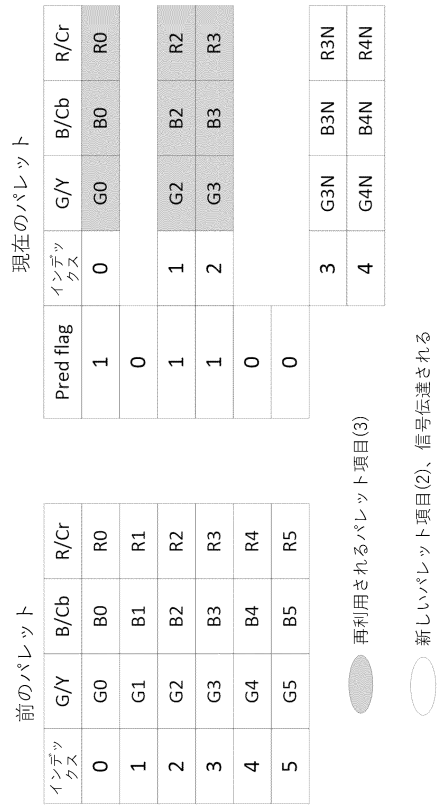
【 図 1 】



【 図 3 】



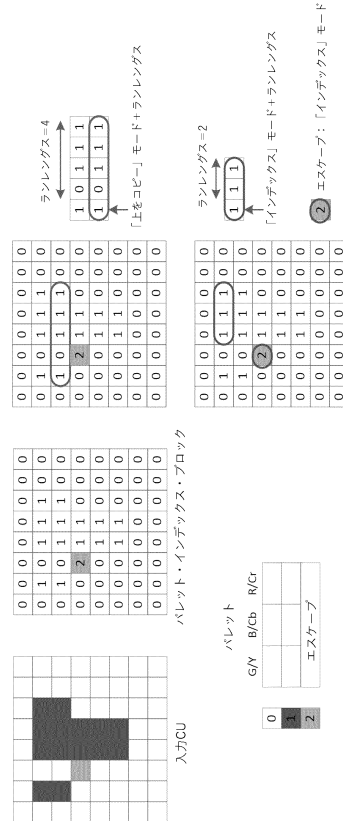
【 図 2 】



10

20

【 図 4 】

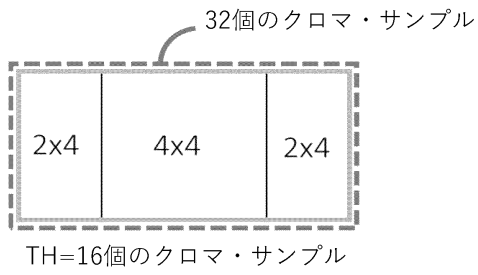


30

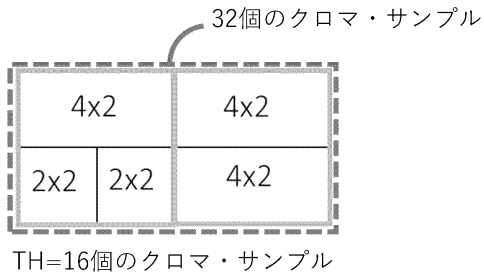
40

50

【 図 5 】

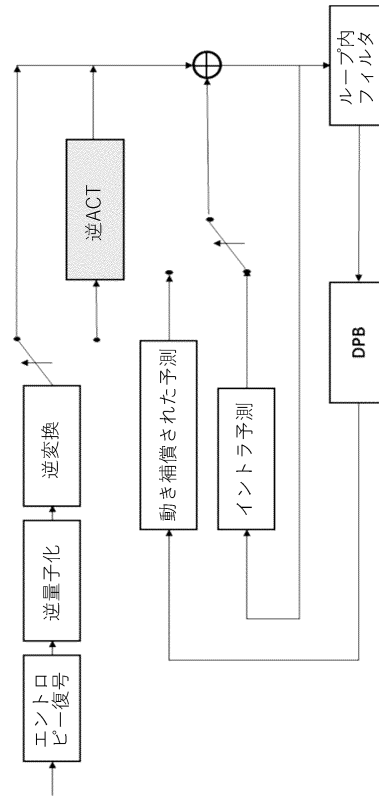


A



B

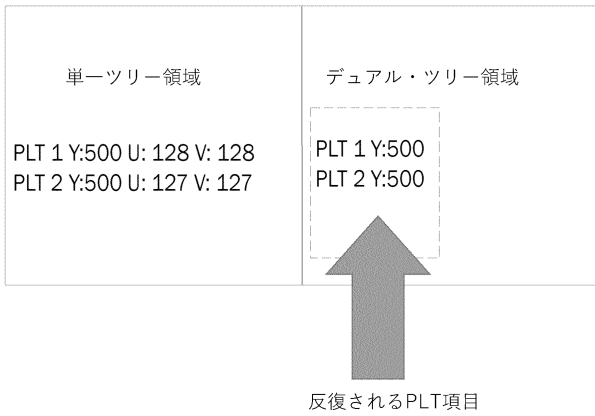
【 図 6 】



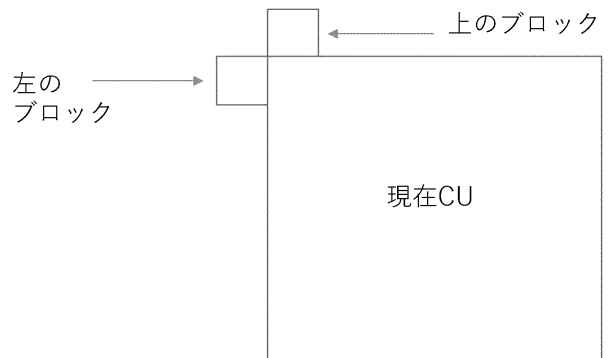
10

20

【 図 7 】



【 図 8 】

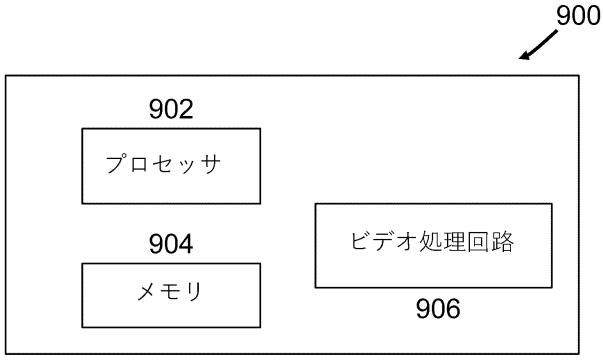


30

40

50

【 図 9 】



【 図 10 】

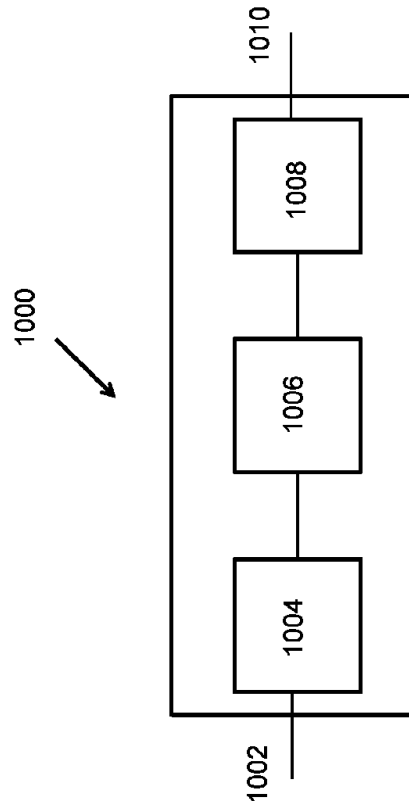
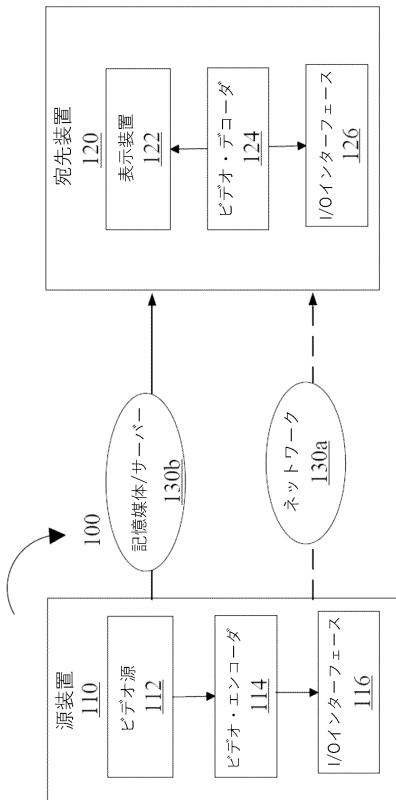


FIG. 10

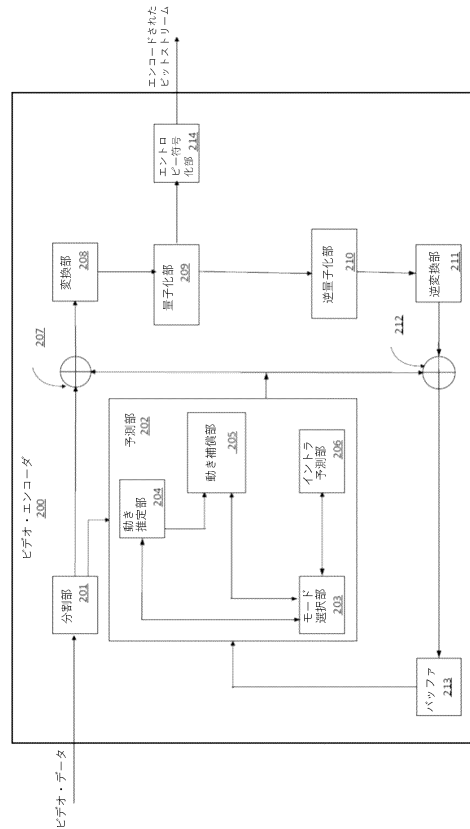
10

20

【 図 11 】



【 図 12 】

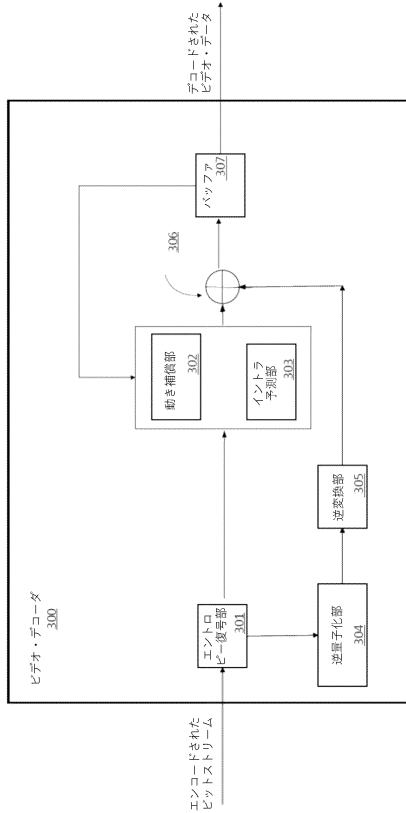


30

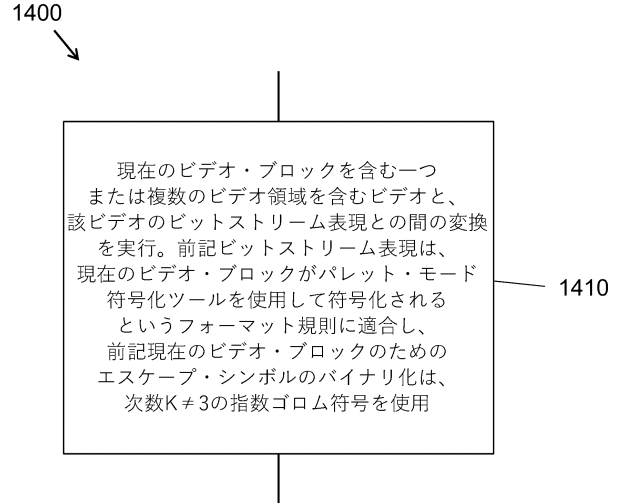
40

50

【図 1 3】



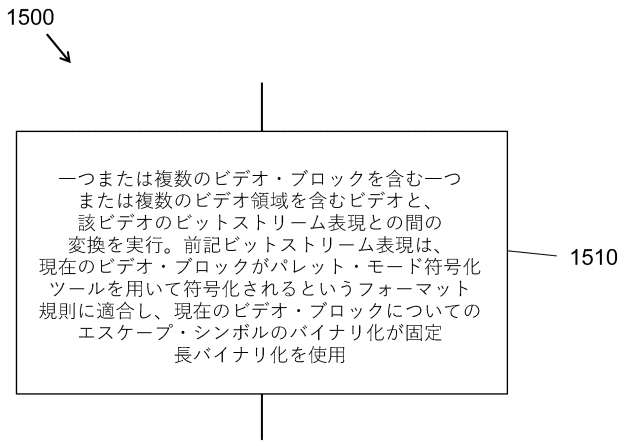
【図 1 4】



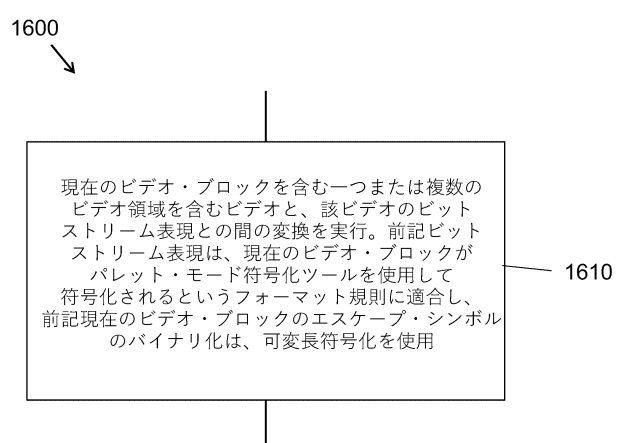
10

20

【図 1 5】



【図 1 6】

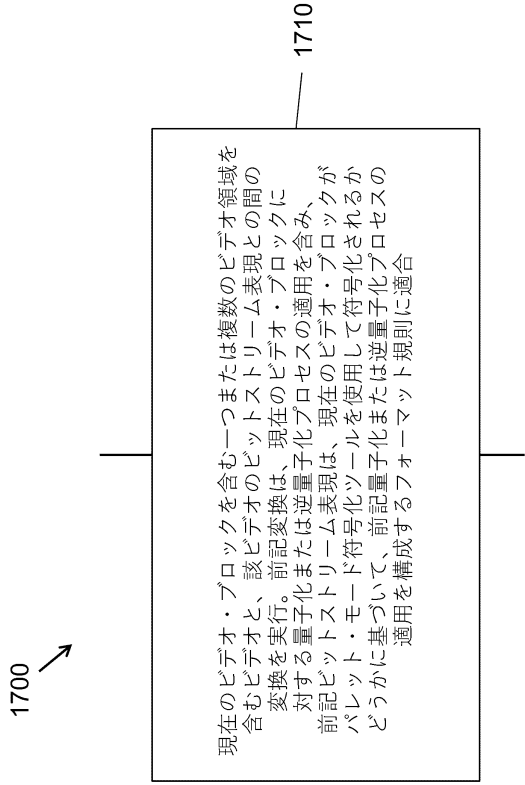


30

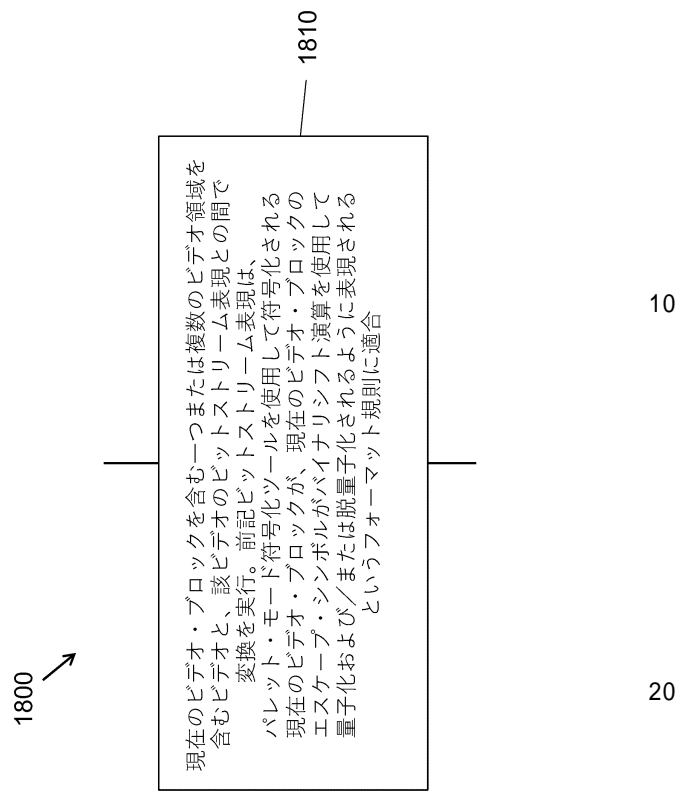
40

50

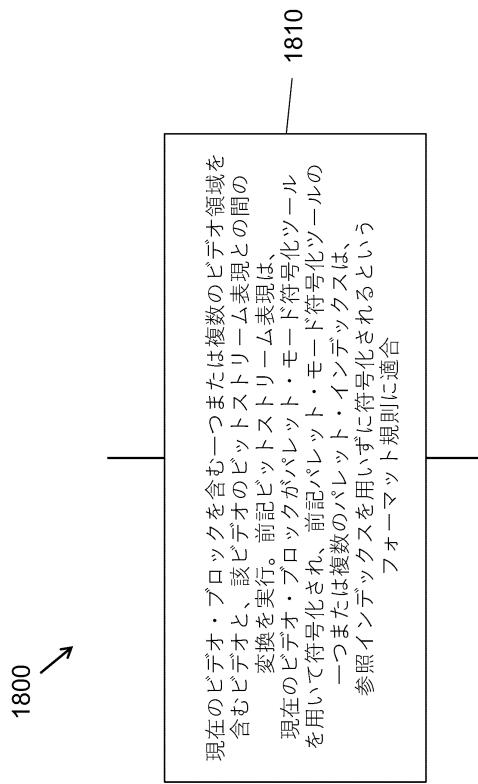
【 図 1 7 】



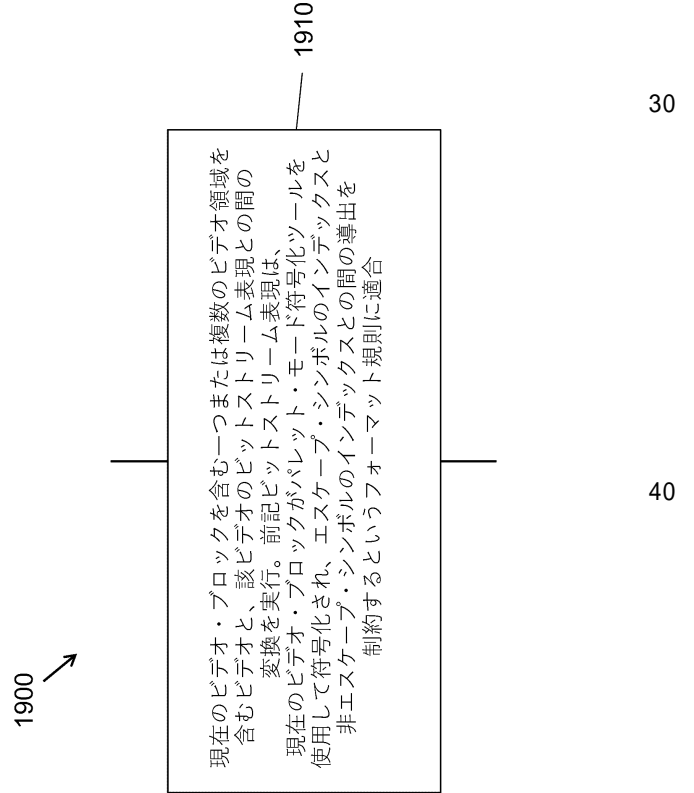
【 図 1 8 】



【 図 1 9 】

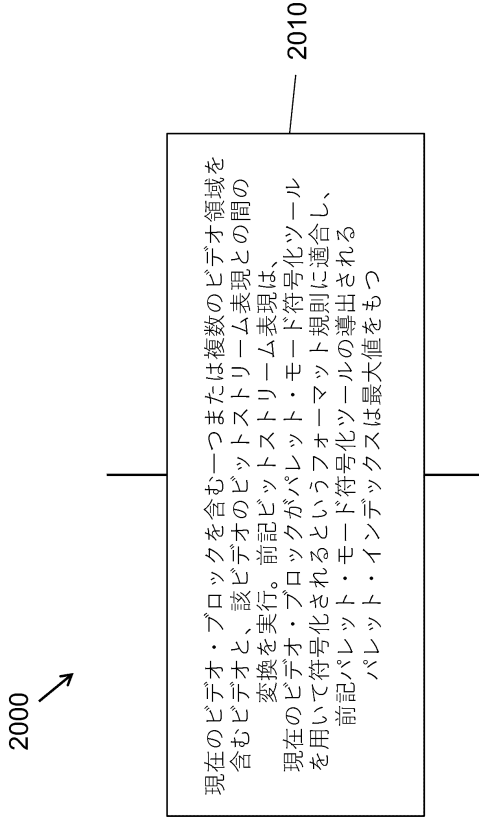


【 図 2 0 】

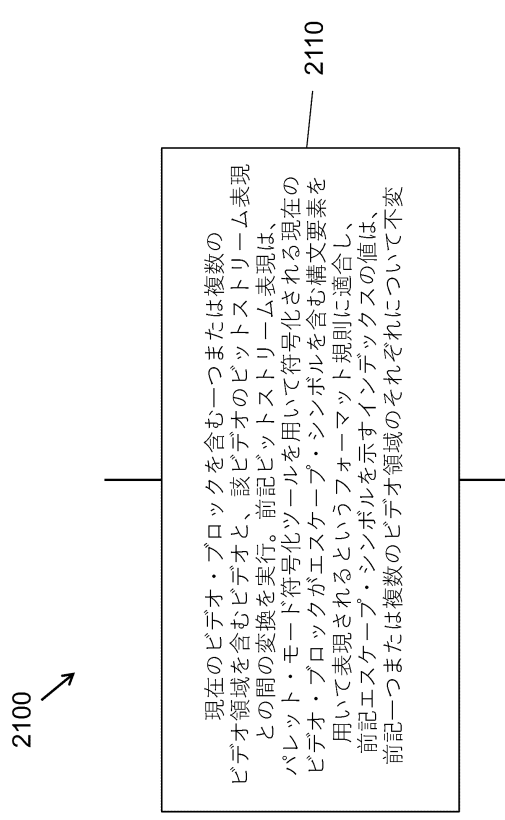




【 図 2 1 】

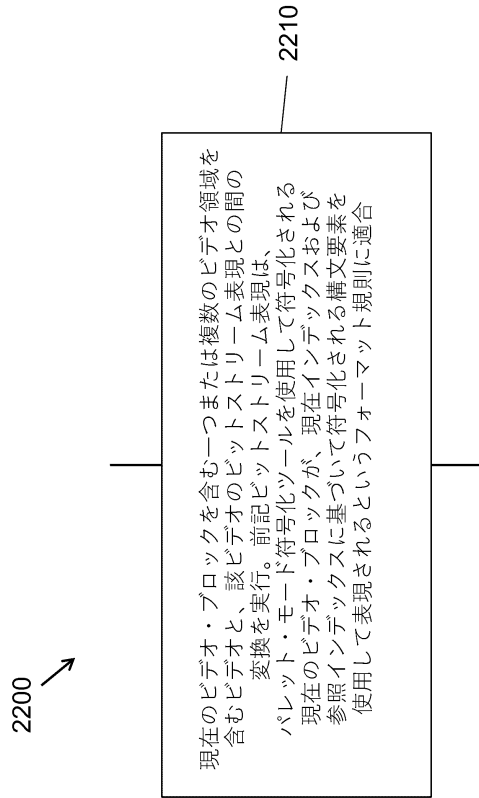


【 図 2 2 】

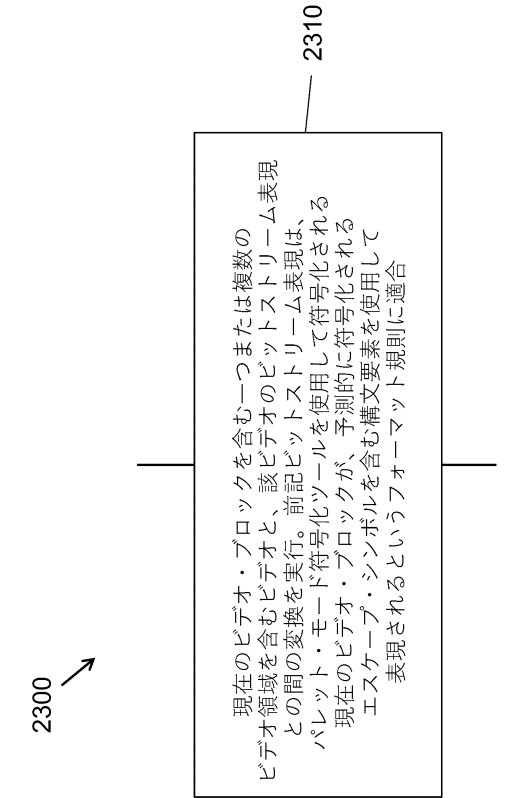


10  
20

【 図 2 3 】

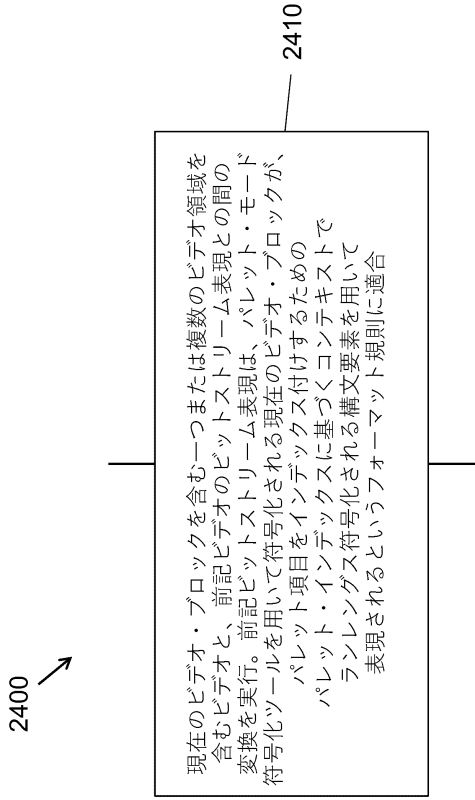


【 図 2 4 】

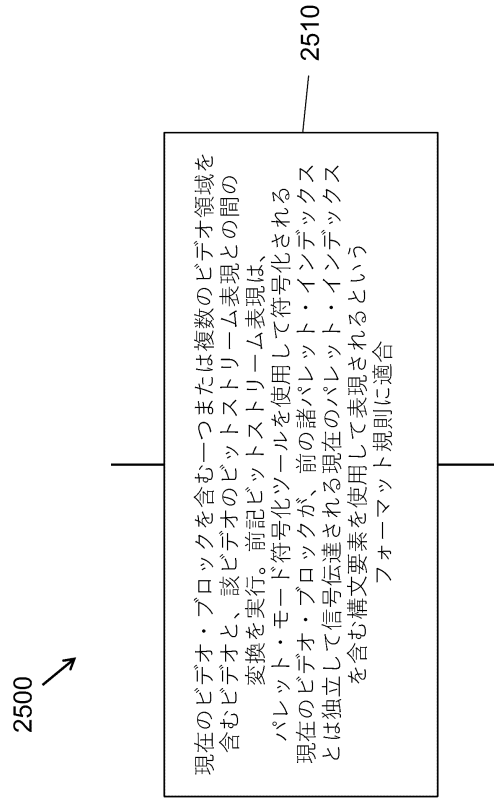


30  
40

【 図 2 5 】

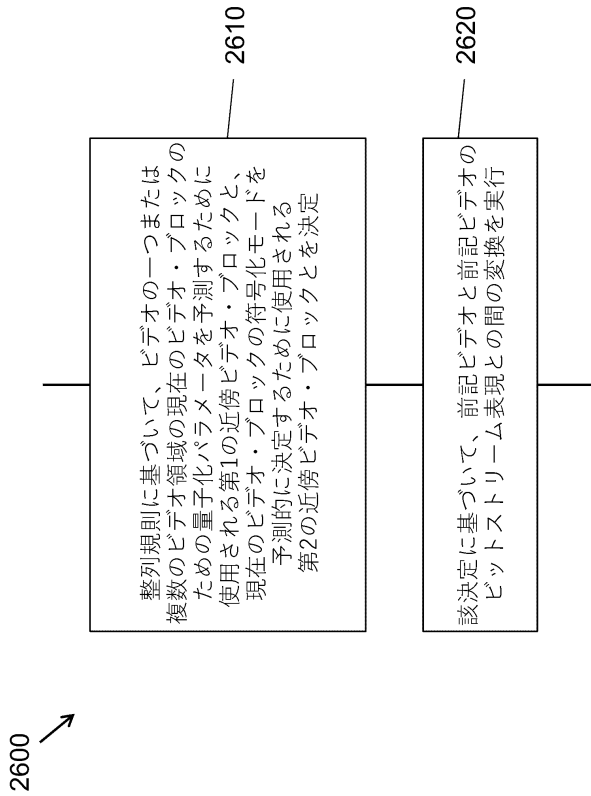


【 図 2 6 】

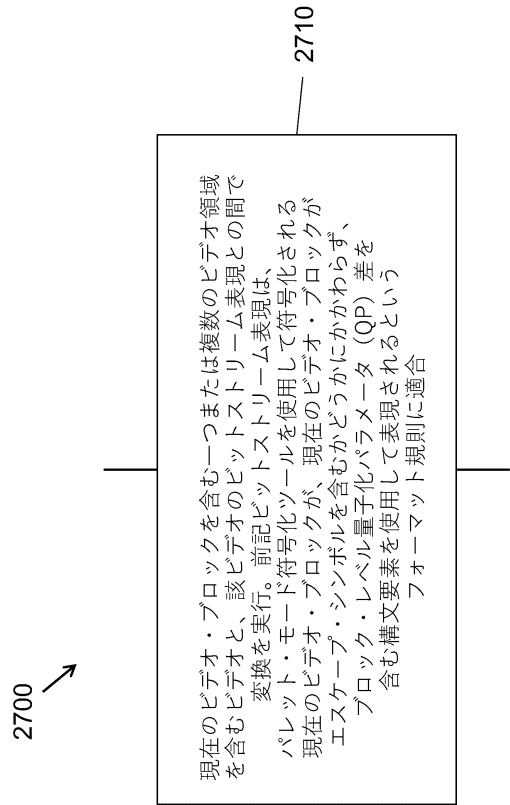


10  
20

【 図 2 7 】

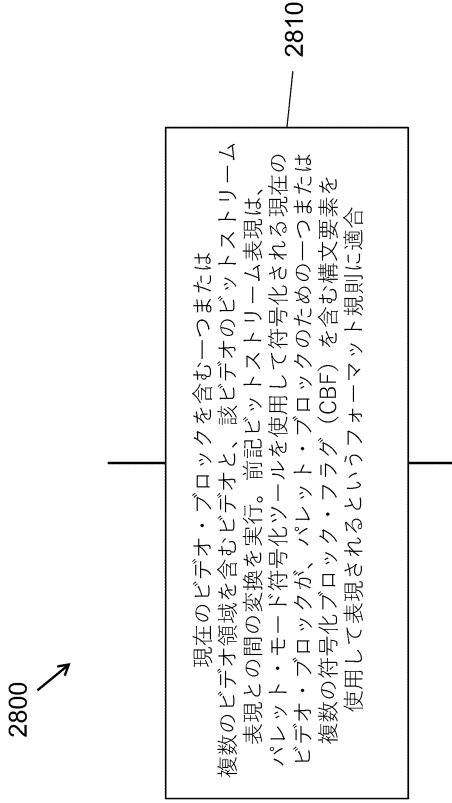


【 図 2 8 】

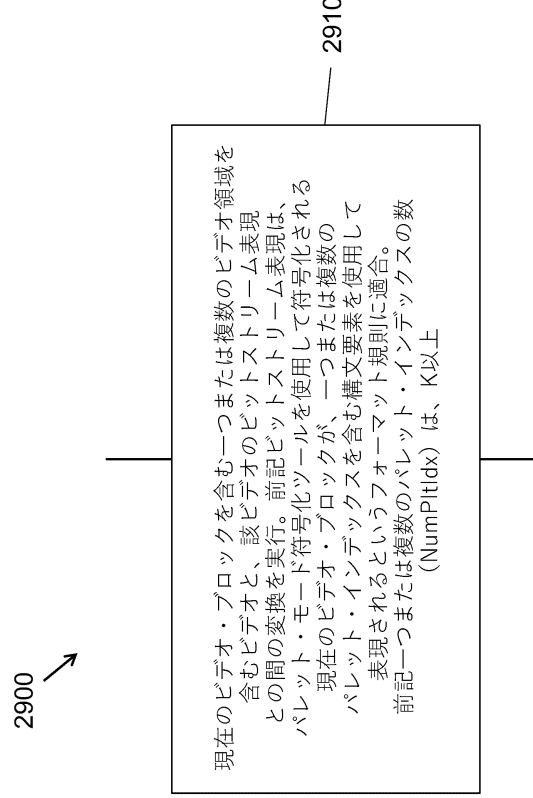


30  
40

【 図 2 9 】

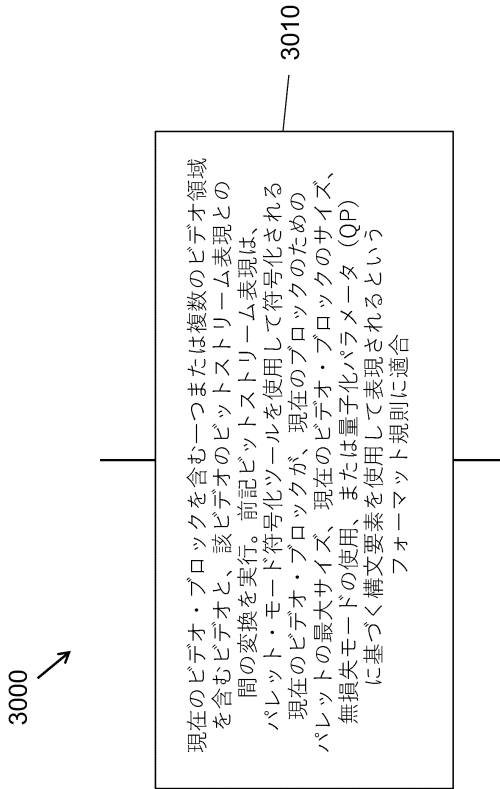


【 図 3 0 】

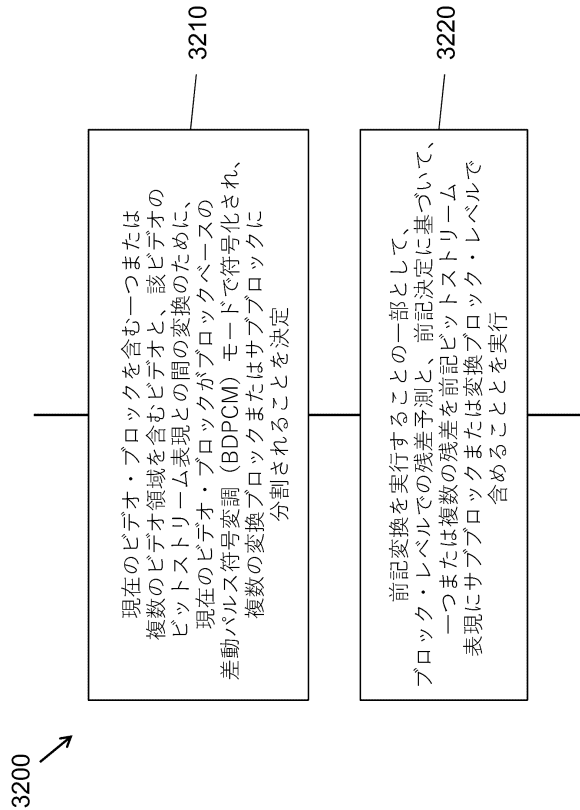


10  
20

【 図 3 1 】



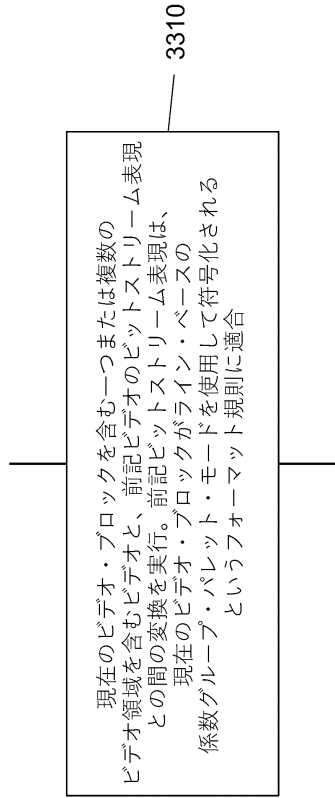
【 図 3 2 】



30  
40  
50

【 図 3 3 】

3300



10

20

30

40

50

【手続補正書】

【提出日】令和4年3月2日(2022.3.2)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

ビデオ・データを処理する方法であって、

10

ビデオの現在のビデオ・ブロックと、該ビデオのビットストリームとの間の変換のために、現在のビデオ・ブロックにある予測モードが適用されることを決定する段階であって、前記予測モードでは、再構成されたサンプルは代表的な色値の集合によって表現され、代表的な色値の前記集合は、前記ビットストリームに含まれる1)パレット予測子、2)エスケープされたサンプル、または3)パレット情報のうちの少なくとも1つを含む、段階と、

少なくとも前記予測モードに基づいて前記変換を実行する段階とを含み、

前記エスケープされたサンプルの量子化された値を指定する第1の構文要素が前記ビットストリームに含まれ、前記第1の構文要素のバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数である、

20

方法。

【請求項2】

K=5である、請求項1に記載の方法。

【請求項3】

前記エスケープされたサンプルは、クリップ関数と、前記エスケープされたサンプルの前記量子化された値とに基づいて再構成される、請求項1または2に記載の方法。

【請求項4】

再構成されたエスケープ・サンプルが、 $\text{Clip}_3(0, (1 - \text{BitDepth}) - 1, \text{tmpVal})$ に基づいて決定され、

$\text{tmpVal}$ は、 $m \cdot (qP/6) + 32$ に6に基づいて決定され、

30

$qP$ は量子化パラメータを指定し、

$m$ は前記エスケープされたサンプルの前記量子化された値に基づいて決定される、請求項3に記載の方法。

【請求項5】

前記 $qP$ は、 $\text{Max}(QpPrimeTsMin, Qp'Y)$ に基づいて決定され、

$QpPrimeTsMin$ は、変換スキップ・モードのための最小の許容される量子化パラメータを表し、

$Qp'Y$ は、ルーマ量子化パラメータを表す、

請求項4に記載の方法。

【請求項6】

40

$QpPrimeTsMin$ は $6 \cdot n + 4$ として定義され、ここで、 $n$ は前記ビットストリームに含まれる第2の構文要素の値である、請求項5に記載の方法。

【請求項7】

前記第2の構文要素は前記ビットストリームのシーケンス・レベルで含まれる、請求項6に記載の方法。

【請求項8】

前記変換は、現在のビデオ・ブロックを前記ビットストリームにエンコードすることを含む、請求項1ないし7のうちいずれか一項に記載の方法。

【請求項9】

前記変換は、現在のビデオ・ブロックを前記ビットストリームからデコードすることを

50

含む、請求項 1 ないし 7 のうちいずれが一項に記載の方法。

【請求項 10】

プロセッサおよび命令を有する非一時的なメモリを有する、ビデオ・データを処理する装置であって、前記命令は、前記プロセッサによって実行されると、前記プロセッサに：  
ビデオの現在のビデオ・ブロックと、該ビデオのビットストリームとの間の変換のために、現在のビデオ・ブロックにある予測モードが適用されることを決定する段階であって、前記予測モードでは、再構成されたサンプルは代表的な色値の集合によって表現され、代表的な色値の前記集合は、前記ビットストリームに含まれる 1) パレット予測子、2) エスケープされたサンプル、または 3) パレット情報のうちの少なくとも 1 つを含む、段階と；

少なくとも前記予測モードに基づいて前記変換を実行する段階とを実行させるものであり、

前記エスケープされたサンプルの量子化された値を指定する第 1 の構文要素が前記ビットストリームに含まれ、前記第 1 の構文要素のバイナリ化は、次数  $K$  の指数ゴロム (EG) 符号を使用し、 $K$  は、3 と等しくない負でない整数である、装置。

【請求項 11】

命令を記憶している非一時的なコンピュータ読み取り可能な記憶媒体であって、前記命令は、プロセッサに：

ビデオの現在のビデオ・ブロックと、該ビデオのビットストリームとの間の変換について、再構成されたサンプルが代表的な色値の集合によって表現され、代表的な色値の前記集合は、前記ビットストリームに含まれる 1) パレット予測子、2) エスケープされたサンプル、または 3) パレット情報のうちの少なくとも 1 つを含む、ことを決定する段階と；

少なくとも前記予測モードに基づいて前記変換を実行する段階とを実行させるものであり、

前記エスケープされたサンプルの量子化された値を指定する第 1 の構文要素が前記ビットストリームに含まれ、前記第 1 の構文要素のバイナリ化は、次数  $K$  の指数ゴロム (EG) 符号を使用し、 $K$  は、3 と等しくない負でない整数である、記憶媒体。

【請求項 12】

ビデオ処理装置によって実行される方法によって生成されたビットストリームを記憶している非一時的なコンピュータ読み取り可能な記録媒体であって、前記方法は：

ビデオの現在のビデオ・ブロックについて、現在のビデオ・ブロックにある予測モードが適用されることを決定する段階であって、前記予測モードでは、再構成されたサンプルは代表的な色値の集合によって表現され、代表的な色値の前記集合は、前記ビットストリームに含まれる 1) パレット予測子、2) エスケープされたサンプル、または 3) パレット情報のうちの少なくとも 1 つを含む、段階と；

少なくとも前記予測モードに基づいて前記ビットストリームを生成する段階とを含み、

前記エスケープされたサンプルの量子化された値を指定する第 1 の構文要素が前記ビットストリームに含まれ、前記第 1 の構文要素のバイナリ化は、次数  $K$  の指数ゴロム (EG) 符号を使用し、 $K$  は、3 と等しくない負でない整数である、記録媒体。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正の内容】

【0001】

10

20

30

40

50

## 関連出願への相互参照

この出願は、2020年8月15日に出願された国際特許出願第PCT/US2020/046581号に基づき、同出願は、2019年8月15日に出願された国際特許出願第PCT/CN2019/100850号、2019年9月29日に出願された同PCT/CN2019/106700号、2019年9月24日に出願された同PCT/CN2019/107494号、2019年9月27日に出願された同PCT/CN2019/108736号、2019年10月1日に出願された同PCT/CN2019/109793号、2019年10月29日に出願された同PCT/CN2019/113931号および2020年1月9日に出願された同PCT/CN2020/071221の優先権および利益を適時に主張してなされる。前述した特許出願はすべて、ここに、参照により本願に組み込まれる。

10

20

30

40

50

## 【手続補正書】

【提出日】令和4年6月13日(2022.6.13)

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【発明の詳細な説明】

【技術分野】

【0001】

関連出願への相互参照

この出願は、2020年8月15日に出願された国際特許出願第PCT/US2020/046581号に基づき、同出願は、2019年8月15日に出願された国際特許出願第PCT/CN2019/100850号、2019年9月29日に出願された同PCT/CN2019/106700号、2019年9月24日に出願された同PCT/CN2019/107494号、2019年9月27日に出願された同PCT/CN2019/108736号、2019年10月1日に出願された同PCT/CN2019/109793号、2019年10月29日に出願された同PCT/CN2019/113931号および2020年1月9日に出願された同PCT/CN2020/071221の優先権および利益を適時に主張してなされる。前述した特許出願はすべて、ここに、参照により本願に組み込まれる。

【0002】

技術分野

本稿は、ビデオおよび画像のコーディング・復号技術に関する。

【背景技術】

【0003】

デジタル・ビデオは、インターネットおよび他のデジタル通信ネットワークにおける最大の帯域幅使用を占める。ビデオを受信および表示できる接続されたユーザー装置の数が増加するにつれて、デジタル・ビデオの利用のための帯域幅需要は増加し続けることが予想される。

【発明の概要】

【課題を解決するための手段】

【0004】

開示される技法は、パレット・モードコーディングおよび復号においてパレット・エスケープ・シンボルのエントロピーコーディングを実行するために、ビデオまたは画像のデコードまたはエンコードの実施形態によって使用されうる。

【0005】

ある例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

【0006】

別の例示的側面では、ビデオ処理の方法が開示される。本方法は、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モードコ

10

20

30

40

50



コーディングツールを用いてコーディングされ、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モードコーディングツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

【0007】

さらに別の例では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長コーディングを使用し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

10

【0008】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

20

【0009】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

30

【0010】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずにコーディングされ、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

40

【0011】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされ、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モードコーディングツール

50

は、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

【0012】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの導出されるパレット・インデックスは最大値をもち、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

10

【0013】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

20

【0014】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいてコーディングされる構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

30

【0015】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、予測的にコーディングされるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

40

【0016】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングスコーディングされる構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

【0017】

50

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0018】

さらに別の例示的側面では、ビデオ処理の方法が開示される。この方法は、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ・ブロックのコーディングモードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップと、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップとを含む。

10

【0019】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ(QP)差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

20

【0020】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数のコーディングブロック・フラグ(CBF)を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

30

【0021】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数(NumPIdx)は、K以上であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、Kは正の整数である。

40

【0022】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失

50

モードの使用、または量子化パラメータ (QP) に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0023】

さらに別の例示的では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調 (BDPCM) モードでコーディングされ、複数の変換ブロックまたはサブブロックに分割されることを決定するステップと、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることを実行するステップとを含む。

10

【0024】

さらに別の例示的側面では、ビデオ処理の方法が開示される。本方法は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ (CG) パレット・モードを使用してコーディングされるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各コーディング単位 (CU) の複数のセグメントを表す。

20

【0025】

さらに別の例示的側面では、上述の方法は、プロセッサを備えるビデオ・エンコーダ装置によって実装されてもよい。

【0026】

さらに別の例示的側面では、上述の方法は、プロセッサを備えるビデオ・デコーダ装置によって実装されてもよい。

【0027】

さらに別の例示的側面では、これらの方法は、プロセッサ実行可能命令の形で具現され、コンピュータ読み取り可能なプログラム媒体に記憶されてもよい。

【0028】

これらおよび他の側面は、本稿においてさらに説明される。

30

【図面の簡単な説明】

【0029】

【図1】パレット・モードでコーディングされるブロックの例を示す図である。

【0030】

【図2】パレット項目を信号伝達するためのパレット予測子の使用例を示す。

【0031】

【図3】水平および垂直横断走査の例を示す。

【0032】

【図4】パレット・インデックスの例示的なコーディングを示す。

40

【0033】

【図5】AおよびBは、最小クロマ・イントラ予測単位 (SCIPU) の例を示す。

【0034】

【図6】ビデオ処理におけるループ内フィルタリングの一例のブロック図を示す。

【0035】

【図7】ローカル・デュアル・ツリーの場合における反復されるパレット項目の例を示す。

【0036】

【図8】コンテキスト導出のプロセスにおける左および上のブロックの例を示す。

【0037】

50

【図 9】本稿に記載される技法を実装するために使用されるハードウェア・プラットフォームの例のブロック図である。

【0038】

【図 10】開示された技法が実装されうる例示的なビデオ処理システムのブロック図である。

【0039】

【図 11】本開示のいくつかの実施形態によるビデオコーディングシステムを示すブロック図である。

【0040】

【図 12】本開示のいくつかの実施形態によるエンコーダを示すブロック図である。 10

【0041】

【図 13】本開示のいくつかの実施形態によるデコーダを示すブロック図である。

【0042】

【図 14】ビデオ処理の例示的方法のフローチャートを示す。

【図 15】ビデオ処理の例示的方法のフローチャートを示す。

【図 16】ビデオ処理の例示的方法のフローチャートを示す。

【図 17】ビデオ処理の例示的方法のフローチャートを示す。

【図 18】ビデオ処理の例示的方法のフローチャートを示す。

【図 19】ビデオ処理の例示的方法のフローチャートを示す。

【図 20】ビデオ処理の例示的方法のフローチャートを示す。 20

【図 21】ビデオ処理の例示的方法のフローチャートを示す。

【図 22】ビデオ処理の例示的方法のフローチャートを示す。

【図 23】ビデオ処理の例示的方法のフローチャートを示す。

【図 24】ビデオ処理の例示的方法のフローチャートを示す。

【図 25】ビデオ処理の例示的方法のフローチャートを示す。

【図 26】ビデオ処理の例示的方法のフローチャートを示す。

【図 27】ビデオ処理の例示的方法のフローチャートを示す。

【図 28】ビデオ処理の例示的方法のフローチャートを示す。

【図 29】ビデオ処理の例示的方法のフローチャートを示す。

【図 30】ビデオ処理の例示的方法のフローチャートを示す。 30

【図 31】ビデオ処理の例示的方法のフローチャートを示す。

【図 32】ビデオ処理の例示的方法のフローチャートを示す。

【図 33】ビデオ処理の例示的方法のフローチャートを示す。

【発明を実施するための形態】

【0043】

本稿は、圧縮解除または復号されたデジタル・ビデオまたは画像の品質を改善するために、画像またはビデオ・ビットストリームのデコーダによって使用されることのできるさまざまな技法を提供する。簡潔のため、本明細書では、用語「ビデオ」は、ピクチャーのシーケンス（伝統的にビデオと呼ばれる）および個々の画像の両方を含むために使用される。さらに、ビデオ・エンコーダも、さらなるエンコードのために使用されるデコードされたフレームを再構成するために、エンコードのプロセス中にこれらの技法を実装しうる。

【0044】

セクション見出しは、理解の容易のために本稿で使用されており、実施形態および技法を対応するセクションに限定するものではない。よって、あるセクションからの実施形態は、他のセクションからの実施形態と組み合わせることができる。

【0045】

#### 1. 概要

本稿は、ビデオコーディング技術に関する。具体的には、パレットコーディングにおけるインデックスおよびエスケープ・シンボルコーディングに関する。これは、HEVCのよ 40

10

20

30

40

50

うな既存のビデオコーディング規格、またはこれから最終化される規格（多用途ビデオコーディング）に適用されてもよい。これは、将来のビデオコーディング規格またはビデオ・コーデックにも適用可能でありうる。

【 0 0 4 6 】

## 2. 背景

ビデオコーディング規格は、主に、周知のITU-TおよびISO/IEC規格の開発を通じて発展してきた。ITU-TはH.261とH.263を生み出し、ISO/IECはMPEG-1とMPEG-4 Visualを生み出し、両組織は合同して、H.262/MPEG-2 VideoとH.264/MPEG-4 Advanced Video Coding (AVC)とH.265/HEVC規格とを生み出した。H.262以来、ビデオコーディング規格は、時間的予測と変換コーディングが利用されるハイブリッドビデオコーディング構造に基づいている。HEVCを越えた将来のビデオコーディング技術を探求するため、2015年にVCEGとMPEGが合同で合同ビデオ探査チーム (JVET) を設立した。それ以来、JVETによって多くの新しい方法が採用され、合同探査モデル (Joint Exploration Model, JEM) と名付けられた参照ソフトウェアに入れられた。2018年4月には、VCEG (Q6/16) とISO/IEC JTC1 SC29/WG11 (MPEG) の間の合同ビデオエキスパートチーム (JVET) が発足し、HEVCに比べ50%のビットレート削減を目指すVVC規格に取り組んでいる。

10

【 0 0 4 7 】

VVCドラフトの最新バージョン、すなわち、Versatile Video Coding (Draft 6) は下記の場所に見出せる：

20

[http://phenix.it-sudparis.eu/jvet/doc\\_end\\_user/documents/15\\_Gothenburg/wg11/JVET-O2001-v14.zip](http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v14.zip)

【 0 0 4 8 】

VVCの最新の参照ソフトウェア (VTM) は、下記の場所に見出せる：

[https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/tags/VTM-5.0](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-5.0)

【 0 0 4 9 】

## 2.1 HEVCスクリーン・コンテンツコーディング拡張 (HEVC-SCC) におけるパレット・モード

### 2.1.1 パレット・モードの概念

【 0 0 5 0 】

30

パレット・モードの背後にある基本的な発想は、CU内のピクセルが代表的な色値の小さな集合によって表現されるということである。この集合はパレットと呼ばれる。エスケープ・シンボルおよびそれに続く（可能性としては量子化された）成分値を信号伝達することによって、パレットの外側にあるサンプルを示すことも可能である。この種のピクセルはエスケープ・ピクセルと呼ばれる。パレット・モードは図1に示されている。図1に示されるように、3つの色成分（ルーマおよび2つのクロマ成分）を有する各ピクセルについて、パレットへのインデックスが設立され、ブロックは、パレットにおける設立された値に基づいて再構成されることができる。

【 0 0 5 1 】

### 2.1.2 パレット項目のコーディング

40

パレット項目のコーディングのために、パレット予測子が維持される。パレットの最大サイズとパレット予測子は、SPSにおいて信号伝達される。HEVC-SCCでは、palette\_predictor\_initializer\_present\_flagがPPSにおいて導入される。このフラグが1である場合、パレット予測子を初期化するための項目が、ビットストリームにおいて信号伝達される。パレット予測子は、各CTU行、各スライスおよび各タイルの先頭において初期化される。palette\_predictor\_initializer\_present\_flagの値に依存して、パレット予測子は0にリセットされるか、PPSにおいて信号伝達されたパレット予測子初期化子項目 (palette predictor initializer entries) を使用して初期化される。HEVC-SCCでは、サイズ0のパレット予測子初期化子により、PPSレベルでパレット予測子初期化の明示的な無効化を許容することができるようになった。

50

## 【0052】

パレット予測子における各項目について、再利用フラグは、それが現在のパレットの一部であるかどうかを示すために信号伝達される。これは図2に示される。再利用フラグは、ゼロのランレングスコーディングを使用して送信される。その後、新しいパレット項目の数が、次数0の指数ゴロム (Exponential Golomb, EG) 符号、すなわちEG-0を使用して信号伝達される。最後に、新しいパレット項目のための成分値が信号伝達される。

## 【0053】

2.1.3 パレット・インデックスのコーディング

パレット・インデックスは、図3に示されるように水平および垂直横断走査を用いてコーディングされる。走査順序は、palette\_transpose\_flagを使用してビットストリームにおいて明示的に信号伝達される。サブセクションの残りについては、走査は水平であると想定される。

## 【0054】

パレット・インデックスは、'COPY\_LEFT' [左をコピー] と 'COPY\_ABOVE' [上をコピー] という2つのパレット・サンプル・モードを使用してコーディングされる。'COPY\_LEFT' モードでは、パレット・インデックスはデコードされたインデックスに割り当てられる。'COPY\_ABOVE' モードでは、上の行のサンプルのパレット・インデックスがコピーされる。'COPY\_LEFT' モードと 'COPY\_ABOVE' モードの両方について、やはり同じモードでコーディングされる後続のサンプルの数を指定するラン値が信号伝達される。

## 【0055】

パレット・モードでは、エスケープ・シンボルについてのインデックスの値は、パレット項目の数である。また、エスケープ・シンボルが 'COPY\_LEFT' または 'COPY\_ABOVE' モードにおけるランの一部である場合、エスケープ成分値は、各エスケープ・シンボルについて信号伝達される。パレット・インデックスのコーディングが図4に示される。

## 【0056】

この構文順序 (syntax order) は、次のように達成される。まず、CUについてのインデックス値の数が信号伝達される。これに続いて、打ち切りされたバイナリコーディングを使用して、CU全体についての実際のインデックス値が信号伝達される。インデックスの数とインデックス値の両方がバイパス・モードでコーディングされる。これは、インデックスに関連したバイパス・ピンをグループ化する。次いで、パレット・サンプル・モード (必要な場合) およびランが、インターリーブされた仕方で信号伝達される。最後に、CU全体についての諸エスケープ・シンボルのための成分エスケープ値と一緒にグループ化され、バイパス・モードでコーディングされる。エスケープ・シンボルのバイナリ化は、3次のEGコーディング、すなわちEG-3である。

## 【0057】

追加的な構文要素 last\_run\_type\_flag が、インデックス値を信号伝達した後に信号伝達される。この構文要素は、インデックスの数とともに、ブロックにおける最後のランに対応するラン値を信号伝達する必要をなくす。

## 【0058】

HEVC-SCCでは、パレット・モードは4:2:2、4:2:0、およびモノクロ・クロマ・フォーマットについても有効にされる。パレット項目とパレット・インデックスの信号伝達は、すべてのクロマ・フォーマットについてほぼ同一である。非モノクロ・フォーマットの場合、各パレット項目は3つの成分で構成される。モノクロ・フォーマットの場合、各パレット項目は単一の成分で構成される。サブサンプリングされたクロマ方向については、クロマ・サンプルは2で割れるルーマ・サンプル・インデックスと関連付けられる。CUのためのパレット・インデックスを再構成した後、サンプルが、それに関連する単一の成分をもつだけである場合、パレット項目の最初の成分のみが使用される。信号伝達における唯一の違いは、エスケープ成分値についてである。各エスケープ・シンボルについて、信号伝達されるエスケープ成分値の数は、そのシンボルに関連する成分の数に依存して

10

20

30

40

50

異なる可能性がある。

【0059】

さらに、パレット・インデックスコーディングにはインデックス調整プロセスがある。パレット・インデックスを信号伝達するとき、左近傍インデックスまたは上近傍インデックスは、現在インデックスとは異なるはずである。よって、1つの可能性を除去することにより、現在のパレット・インデックスの範囲を1だけ減らすことができる。その後、インデックスは、打ち切りされたバイナリ (truncated binary、TB) バイナリ化で信号伝達される。

【0060】

この部分に関連するテキストは次のように示される。ここで、CurrPaletteIndexは現在のパレット・インデックスであり、adjustedRefPaletteIndexは予測インデックスである。

10

【0061】

変数PaletteIndexMap[xC][yC]は、パレット・インデックスを指定する。これは、CurrentPaletteEntriesによって表現される配列へのインデックスである。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。PaletteIndexMap[xC][yC]の値は、0からMaxPaletteIndexの範囲である(両端含む)。変数adjustedRefPaletteIndexは次のように導出される：

```
adjustedRefPaletteIndex = MaxPaletteIndex + 1
if ( PaletteScanPos == 0 ) {
    xcPrev = x0 + TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos - 1][0]
    ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos - 1][1]
    if ( CopyAboveIndicesFlag[xcPrev][ycPrev]==0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[xcPrev][ycPrev]{
(7-157)
    }
    else{
        if ( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[xC][yC - 1]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[xC - 1][yC]
    }
}
```

20

30

【0062】

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```
if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++
```

40

【0063】

さらに、パレット・モードにおけるランレングス要素はコンテキストコーディングされる。JVET-O2011-vEにおいて記載される関連するコンテキスト導出プロセスは、以下のように示される。

構文要素palette\_run\_prefixについてのctxIncの導出プロセス

このプロセスへの入力、ピン・インデックスbinIdxならびに構文要素copy\_above\_palette\_indices\_flagおよびpalette\_idx\_idcである。

このプロセスの出力は変数ctxIncである。

変数ctxIncは次のように導出される：

・ copy\_above\_palette\_indices\_flagが0に等しく、binIdxが0に等しい場合、ctxI

50



ncは以下のように導出される：

$$ctxInc = (palette\_idx\_idc = 1) ? 0 : ((palette\_idx\_idc = 3) ? 1 : 2) \quad (9-69)$$

・それ以外の場合、ctxIncは表1によって与えられる：

【表 1】

表1 -ctxIdxMap[copy\_above\_palette\_indices\_flag][binIdx]の指定

binIdx	0	1	2	3	4	> 4
copy_above_palette_indices_flag == 1	5	6	6	7	7	バイパス
copy_above_palette_indices_flag == 0	0, 1, 2	3	3	4	4	バイパス

10

【 0 0 6 4 】

2.2 VVCにおけるパレット・モード

2.2.1 デュアル・ツリーにおけるパレット

VVCでは、デュアル・ツリーコーディング構造がイントラスライスのコーディングに使用されるため、ルーマ成分と2つのクロマ成分は、異なるパレットおよびパレット・インデックスをもつことがある。さらに、2つのクロマ成分は、同じパレットおよびパレット・インデックスを共有する。

【 0 0 6 5 】

2.2.2 別個のモードとしてのパレット

JVET-N0258および現在のVTMでは、コーディング単位についての予測モードはMODE\_INTRA、MODE\_INTER、MODE\_IBC、MODE\_PLTとすることができる。予測モードのバイナリ化はそれに応じて変更される。

20

【 0 0 6 6 】

IBCがオフにされるときは、1タイル上で、最初の1ピンが、現在の予測モードがMODE\_PLTであるか否かを示すために使用される。P/Bタイル上では、最初のピンは現在の予測モードがMODE\_INTRAであるか否かを示すために使用される。そうでない場合、現在の予測モードがMODE\_PLTまたはMODE\_INTERであることを示すために、1つの追加的なピンが使用される。IBCがオンにされるときは、1タイル上で、最初のピンが、現在の予測モードがMODE\_IBCであるか否かを示すために使用される。もしそうでなければ、現在の予測モードがMODE\_PLTかMODE\_INTRAかを示すために第2のピンが使われる。P/Bタイル上では、最初のピンは現在の予測モードがMODE\_INTRAであるか否かを示すために使用される。イントラモードである場合、現在の予測モードがMODE\_PLTまたはMODE\_INTRAであることを示すために第2のピンが使用される。もしそうでなければ、現在の予測モードがMODE\_IBCまたはMODE\_INTERであることを示すために第2のピンが使用される。

30

【 0 0 6 7 】

JVET-O2001-vEにおける関連するテキストを以下に示す。

【 0 0 6 8 】

コーディング単位構文

40

【表 2】

coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {	記述子
chType = treeType == DUAL_TREE_CHROMA? 1 : 0	
if( slice_type != I    sps_ibc_enabled_flag    sps_palette_enabled_flag ) {	
if( treeType != DUAL_TREE_CHROMA && !( ( cbWidth == 4 && cbHeight == 4 )    modeType == MODE_TYPE_INTRA ) && !sps_ibc_enabled_flag )	10
<b>cu_skip_flag[ x0 ][ y0 ]</b>	ae(v)
if( cu_skip_flag[ x0 ][ y0 ] == 0 && slice_type != I && !( cbWidth == 4 && cbHeight == 4 ) && modeType == MODE_TYPE_ALL )	
<b>pred_mode_flag</b>	ae(v)
if( ( ( slice_type == I && cu_skip_flag[ x0 ][ y0 ] == 0 )    ( slice_type != I && ( CuPredMode[ chType ][ x0 ][ y0 ] != MODE_INTRA    ( cbWidth == 4 && cbHeight == 4 && cu_skip_flag[ x0 ][ y0 ] == 0 ) ) ) ) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA )	20
<b>pred_mode_ibc_flag</b>	ae(v)
if( ( ( ( slice_type == I    ( cbWidth == 4 && cbHeight == 4 )    sps_ibc_enabled_flag ) && CuPredMode[ x0 ][ y0 ] == MODE_INTRA )    ( slice_type != I && !( cbWidth == 4 && cbHeight == 4 ) && !sps_ibc_enabled_flag && CuPredMode[ x0 ][ y0 ] != MODE_INTRA ) ) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[ x0 ][ y0 ] == 0 && modeType != MODE_TYPE_INTER )	30
<b>pred_mode_plt_flag</b>	ae(v)
}	
...	
}	40

## 【 0 0 6 9 】

## 2.2.3 パレット・モード構文

【表 3】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
palettePredictionFinished = 0		
NumPredictedPaletteEntries = 0		
for( predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[ startComp ] && !palettePredictionFinished && NumPredictedPaletteEntries[ startComp ] < palette_max_size; predictorEntryIdx++ ) {		10
<b>palette_predictor_run</b>	ae(v)	
if( palette_predictor_run != 1 ) {		
if( palette_predictor_run > 1 )		
predictorEntryIdx += palette_predictor_run - 1		
PalettePredictorEntryReuseFlags[ predictorEntryIdx ] = 1		
NumPredictedPaletteEntries++		
} else		20
palettePredictionFinished = 1		
}		
if( NumPredictedPaletteEntries < palette_max_size )		
<b>num_signalled_palette_entries</b>	ae(v)	
for( cIdx = startComp; cIdx < ( startComp + numComps); cIdx++ )		
for( i = 0; i < num_signalled_palette_entries; i++ )		
<b>new_palette_entries[ cIdx ][ i ]</b>	ae(v)	30
if( CurrentPaletteSize[ startComp ] > 0 )		
<b>palette_escape_val_present_flag</b>	ae(v)	
if( MaxPaletteIndex > 0 ) {		
<b>num_palette_indices_minus1</b>	ae(v)	
adjust = 0		
for( i = 0; i <= num_palette_indices_minus1; i++ ) {		
if( MaxPaletteIndex - adjust > 0 ) {		40

<b>palette_idx_idc</b>	ae(v)	
PaletteIndexIdc[ i ] = palette_idx_idc		
}		
adjust = 1		
}		
<b>copy_above_indices_for_final_run_flag</b>	ae(v)	
<b>palette_transpose_flag</b>	ae(v)	10
}		
if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {		
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {		
<b>cu_qp_delta_abs</b>	ae(v)	
if( cu_qp_delta_abs )		
<b>cu_qp_delta_sign_flag</b>	ae(v)	
}		20
}		
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {		
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {		
<b>cu_chroma_qp_offset_flag</b>	ae(v)	
if( cu_chroma_qp_offset_flag )		
<b>cu_chroma_qp_offset_idx</b>	ae(v)	
}		
}		30
remainingNumIndices = num_palette_indices_minus1 + 1		
PaletteScanPos = 0		
log2CbWidth = Log2( cbWidth )		
log2CbHeight = Log2( cbHeight )		
while( PaletteScanPos < cbWidth*cbHeight ) {		
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]		40

10

20

30

40

50

if( PaletteScanPos > 0 ) {		
xcPrev	=	
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev	=	
y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
PaletteRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1		10
RunToEnd = 1		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
if( MaxPaletteIndex > 0 )		
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC > 0 ) )		
&& CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 )		
if( remainingNumIndices > 0 && PaletteScanPos < cbWidth* cbHeight - 1 ) {		
<b>copy_above_palette_indices_flag</b>	ae(v)	
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag		20
} else {		
if( PaletteScanPos == cbWidth * cbHeight - 1 && remainingNumIndices > 0 )		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
else		
CopyAboveIndicesFlag[ xC ][ yC ] = 1		
}		
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {		30
currNumIndices = num_palette_indices_minus1 + 1 - remainingNumIndices		
PaletteIndexMap[ xC ][ yC ] = PaletteIndexIdx[ currNumIndices ]		
}		
if( MaxPaletteIndex > 0 ) {		
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 )		
remainingNumIndices - = 1		
if( remainingNumIndices > 0    CopyAboveIndicesFlag[ xC ][ yC ] !=		
copy_above_indices_for_final_run_flag ) {		40

PaletteMaxRunMinus1 = cbWidth * cbHeight - PaletteScanPos - 1 - remainingNumIndices - copy_above_indices_for_final_run_flag	
RunToEnd = 0	
if( PaletteMaxRunMinus1 > 0 ) {	
<b>palette_run_prefix</b>	ae(v)
if( ( palette_run_prefix > 1 ) && ( PaletteMaxRunMinus1 != ( 1 << ( palette_run_prefix - 1 ) ) ) )	
<b>palette_run_suffix</b>	ae(v)
}	
}	
}	
runPos = 0	
while ( runPos <= PaletteRunMinus1 ) {	
xR = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
yR = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	
if( CopyAboveIndicesFlag[ xR ][ yR ] == 0 ) {	
CopyAboveIndicesFlag[ xR ][ yR ] = 0	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xC ][ yC ]	
} else {	
CopyAboveIndicesFlag[ xR ][ yR ] = 1	
if ( !palette_transpose_flag )	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xR ][ yR - 1 ]	
else	
PaletteIndexMap[ xR ][ yR ] = PaletteIndexMap[ xR - 1 ][ yR ]	
}	
runPos++	
PaletteScanPos ++	
}	
}	

10

20

30

40

50

if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++)	
for( sPos = 0; sPos < cbWidth* cbHeight; sPos++) {	
xC = x0 + TraverseScanOrder[ log2CbWidth][ log2CbHeight ][ sPos ][ 0 ]	
yC = y0 + TraverseScanOrder[ log2CbWidth][ log2CbHeight ][ sPos ][ 1 ]	
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {	
<b>palette_escape_val</b>	ae(v)
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val	
}	
}	
}	
}	

10

20

【 0 0 7 0 】

#### 2.2.4 パレット・モードのセマンティクス

以下のセマンティクスでは、配列インデックス $x_0$ 、 $y_0$ は、ピクチャーの左上のルーマ・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・サンプルの位置 $(x_0, y_0)$ を指定する。配列インデックス $x_C$ 、 $y_C$ は、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置 $(x_C, y_C)$ を指定する。配列インデックス $startComp$ は、現在のパレット・テーブルの最初の色成分を指定する。0に等しい $startComp$ はY成分を示し、1に等しい $startComp$ はCb成分を示し、2に等しい $startComp$ はCr成分を示す。 $numComps$ は、現在のパレット・テーブルにおける色成分の数を指定する。

【 0 0 7 1 】

30

予測子パレットは、現在のパレット内の項目を予測するために使用される、以前のコーディング単位からのパレット項目で構成される。

変数 $PredictorPaletteSize[startComp]$ は、現在のパレット・テーブルの最初の色成分 $startComp$ についての予測子パレットのサイズを指定する。 $PredictorPaletteSize$ は、節8.4.5.3において規定されるように導出される。

変数 $PalettePredictorEntryReuseFlags[i]$ が1に等しいことは、予測子パレットにおける $i$ 番目の項目が現在のパレットにおいて再利用されることを指定する。 $PalettePredictorEntryReuseFlags[i]$ が0に等しいことは、予測子パレットの $i$ 番目の項目が現在のパレットにおける項目ではないことを指定する。配列 $PalettePredictorEntryReuseFlags[i]$ のすべての要素が0に初期化される。

40

【 0 0 7 2 】

$palette\_predictor\_run$ は、配列 $PalettePredictorEntryReuseFlags$ におけるゼロでない項目の前にあるゼロの数を決定するために使用される。

$palette\_predictor\_run$ の値は、0から $(PredictorPaletteSize - predictorEntryIdx)$ の範囲内（両端含む）であることがビットストリーム適合性の要件である。ここで、 $predictorEntryIdx$ は、配列 $PalettePredictorEntryReuseFlags$ における現在位置に対応する。変数 $NumPredictedPaletteEntries$ は、予測子パレットから再利用される現在のパレットにおける項目数を指定する。 $NumPredictedPaletteEntries$ の値は、0から $palette\_max\_size$ の範囲内（両端含む）である。

【 0 0 7 3 】

50

num\_signalled\_palette\_entriesは、現在のパレット・テーブルの最初の色成分startCompについて明示的に信号伝達される、現在のパレットにおける項目の数を指定する。num\_signalled\_palette\_entriesが存在しない場合、0に等しいと推定される。

変数CurrentPaletteSize[startComp]は、現在のパレット・テーブルの最初の色成分startCompについての現在のパレットのサイズを指定し、次のように導出される：

```
CurrentPaletteSize[startComp] = NumPredictedPaletteEntries + num_signalled_palette_entries (7-155)
```

CurrentPaletteSize[startComp]の値は、0からpalette\_max\_sizeの範囲内（両端含む）である。

【0074】

new\_palette\_entries[cIdx][i]は、色成分cIdxについてのi番目の信号伝達されるパレット項目についての値を指定する。

変数PredictorPaletteEntries[cIdx][i]は、色成分cIdxについての予測子パレットにおけるi番目の要素を指定する。

変数CurrentPaletteEntries[cIdx][i]は、色成分cIdxについての現在のパレットにおけるi番目の要素を指定し、次のように導出される：

【表4】

```
numPredictedPaletteEntries = 0
```

```
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
```

```
    if( PalettePredictorEntryReuseFlags[ i ] ) {
```

```
        for( cIdx =startComp; cIdx < ( startComp + numComps ); cIdx++ )
```

```
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
```

```
PredictorPaletteEntries[ cIdx ][ i ]
```

```
            numPredictedPaletteEntries++
```

```
        }
```

```
for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ ) (7-156)
```

```
    for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
```

```
        CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = new_palette_entries[ cIdx ][ i ]
```

【0075】

palette\_escape\_val\_present\_flagが1に等しいことは、現在のコーディング単位が少なくとも1つのエスケープコーディングサンプルを含むことを指定する。escape\_val\_present\_flagが0に等しいことは、現在のコーディング単位においてエスケープコーディングされたサンプルがないことを指定する。存在しない場合、palette\_escape\_val\_present\_flagの値は1に等しいと推定される。

変数MaxPaletteIndexは、現在のコーディング単位についてのパレット・インデックスの可能な最大値を指定する。MaxPaletteIndexの値は

CurrentPaletteSize[startComp] - 1 + palette\_escape\_val\_present\_flagに等しく設定される。

【0076】

num\_palette\_indices\_minus1に1を加えたものは、現在のブロックについて明示的に信号伝達されるまたは推定されるパレット・インデックスの数である。num\_palette\_indices\_minus1が存在しない場合、0に等しいと推定される。

palette\_idx\_idcは、パレット・テーブル、CurrentPaletteEntriesへのインデック

10

20

30

40

50



スの指示である。palette\_idx\_idcの値は、ブロック内の最初のインデックスについては0からMaxPaletteIndexの範囲内（両端含む）であり、ブロック内の残りのインデックスについては0から（MaxPaletteIndex - 1）の範囲内（両端含む）である。

palette\_idx\_idcが存在しない場合、0に等しいと推定される。

変数PaletteIndexIdc[i]は、明示的に信号伝達されるまたは推定されるi番目のpalette\_idx\_idcを格納する。配列PaletteIndexIdc[i]のすべての要素は0に初期化される。

【0077】

copy\_above\_indices\_for\_final\_run\_flagが1に等しいことは、コーディング単位の最後の諸位置のパレット・インデックスが、水平横断走査が使用される場合は上の行におけるパレット・インデックスから、垂直横断走査が使用される場合は左の列におけるパレット・インデックスから、コピーされることを指定する。 10

copy\_above\_indices\_for\_final\_run\_flagが0に等しいことは、コーディング単位における最後の諸位置のパレット・インデックスがPaletteIndexIdc[num\_palette\_indices\_minus1]からコピーされることを指定する。

copy\_above\_indices\_for\_final\_run\_flagが存在しない場合、0に等しいと推定される。

【0078】

palette\_transpose\_flagが1に等しいことは、現在のコーディング単位内のサンプルについてのインデックスを走査するために垂直横断走査が適用されることを指定する。palette\_transpose\_flagが0に等しいことは、現在のコーディング単位内のサンプルについてのインデックスを走査するために水平横断走査が適用されることを指定する。存在しない場合、palette\_transpose\_flagの値は0に等しいと推定される。 20

配列TraverseScanOrderは、パレットコーディングのための走査順序配列を指定する。palette\_transpose\_flagが0に等しい場合、TraverseScanOrderは水平走査順序HorTravScanOrderを割り当てられ、palette\_transpose\_flagが1に等しい場合、TraverseScanOrderは垂直走査順序VerTravScanOrderを割り当てられる。

【0079】

copy\_above\_palette\_indices\_flagが1に等しいことは、パレット・インデックスが、水平横断走査が使用される場合には、上の行における同じ位置、または垂直横断走査が使用される場合には、左の列における同じ位置にあるパレット・インデックスに等しいことを指定する。copy\_above\_palette\_indices\_flagが0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいてコーディングされるか、または推定されることを指定する。 30

【0080】

変数CopyAboveIndicesFlag[xC][yC]が1に等しいことは、パレット・インデックスが上の行（水平走査）または左の列（垂直走査）におけるパレット・インデックスからコピーされることを指定する。

CopyAboveIndicesFlag[xC][yC]が0に等しいことは、パレット・インデックスがビットストリームにおいて明示的にコーディングされるか、推定されることを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置（xC,yC）を指定する。PaletteIndexMap[xC][yC]の値は、0から（MaxPaletteIndex - 1）の範囲内（両端含む）とする。 40

【0081】

変数PaletteIndexMap[xC][yC]は、CurrentPaletteEntriesによって表される配列へのインデックスであるパレット・インデックスを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置（xC,yC）を指定する。PaletteIndexMap[xC][yC]の値は、0からMaxPaletteIndexの範囲内（両端含む）である。

【0082】

変数adjustedRefPaletteIndexは次のように導出される：

【表 5】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++ (7-158)

```

【0083】

palette\_run\_prefixは、存在する場合、PaletteRunMinus1のバイナリ化におけるプレフィックス部分を指定する。

【0084】

palette\_run\_suffixは、変数PaletteRunMinus1の導出において使用される。存在しない場合、palette\_run\_suffixの値は0に等しいと推定される。

【0085】

RunToEndが0に等しい場合、変数PaletteRunMinus1は次のように導出される：

- ・ PaletteMaxRunMinus1が0に等しい場合、PaletteRunMinus1は0に等しく設定される。

- ・ それ以外の場合（PaletteMaxRunMinus1が0より大きい場合）、次が適用される：

- ・ palette\_run\_prefixが2より小さい場合、次が適用される：

```

PaletteRunMinus1 = palette_run_prefix (7-159)

```

- ・ それ以外の場合（palette\_run\_prefixが2以上）、次が適用される：

```

PrefixOffset = 1 (palette_run_prefix - 1)

```

```

PaletteRunMinus1 = PrefixOffset + palette_run_suffix (7-160)

```

【0086】

変数PaletteRunMinus1は次のように使用される：

- ・ CopyAboveIndicesFlag[xC][yC]が0に等しい場合、PaletteRunMinus1は、同じパレット・インデックスをもつ連続する位置の数から1を引いた数を指定する。

- ・ それ以外の場合、palette\_transpose\_flagが0に等しい場合、PaletteRunMinus1は、上の行における対応する位置で使用されるのと同じパレット・インデックスをもつ、連続する位置の数から1を引いた数を指定する。

- ・ それ以外の場合、PaletteRunMinus1は、左の列における対応する位置で使用される

のと同じパレット・インデックスをもつ、連続する位置の数から1を引いた数を指定する。

RunToEndが0に等しい場合、変数PaletteMaxRunMinus1はPaletteRunMinus1について可能な最大値を表し、PaletteMaxRunMinus1の値が0以上であることはビットストリーム適合性の要件である。

【0087】

palette\_escape\_valは、ある成分についての量子化されたエスケープコーディングされたサンプル値を指定する。

変数PaletteEscapeVal[cldx][xC][yC]は、PaletteIndexMap[xC][yC]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しいサンプルのエスケープ値を指定する。配列インデックスcldxは、色成分を指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。

PaletteEscapeVal[cldx][xC][yC]が、0に等しいcldxについては0から(1 - (BitDepthY + 1) - 1)の範囲内(両端含む)であり、0に等しくないcldxについては0から(1 - ((BitDepthC + 1)) - 1)の範囲内(両端含む)であることがビットストリーム適合性の要件である。

【0088】

#### 1.1.1 ライン・ベースのCGパレット・モード

ライン・ベースのCGパレット・モードがVVCに採用された。この方法では、パレット・モードの各CUは、横断走査モードに基づいて複数のサンプル・セグメント(この試験では)に分割される。各セグメントにおけるパレット・ランコーディングのためのエンコード順序は、次のとおり: 各ピクセルについて、1つのコンテキストコーディングされたピンrun\_copy\_flag = 0が信号伝達されて、ピクセルが前のピクセルと同じモードであるかどうか、すなわち前の走査されたピクセルと現在のピクセルが両方ともラン・タイプCOPY\_ABOVEであるかどうか、または前の走査されたピクセルと現在のピクセルが両方ともラン・タイプINDEXであり、同じインデックス値であるかどうかを示す。そうでない場合、run\_copy\_flag = 1が信号伝達される。ピクセルと前のピクセルが異なるモードである場合、1つのコンテキストコーディングされたピンcopy\_above\_palette\_indices\_flagが信号伝達されて、ピクセルのラン・タイプ、すなわち、INDEXまたはCOPY\_ABOVEを示す。VTM6.0におけるパレット・モードと同じく、サンプルが最初の行(水平横断走査)または最初の列(垂直横断走査)にある場合、デフォルトでINDEXモードが使用されるため、デコーダはラン・タイプをパースする必要はない。また、前にパースされたラン・タイプがCOPY\_ABOVEである場合も、デコーダはラン・タイプをパースする必要はない。1つのセグメントにおけるピクセルのパレット・ランコーディングの後、インデックス値(INDEXモードについて)および量子化されたエスケープ色は、各ラインCG内のスループットを改善するために、コンテキストコーディングされたピンのエンコード/パースとは別に、バイパスコーディングおよびグループ化される。インデックス値は、VTMのようにパレット・ランコーディングの前に処理されるのではなく、今やランコーディングの後にコーディング/パースされるので、エンコーダは、インデックス値の数num\_palette\_indices\_minus1と最後のラン・タイプcopy\_above\_indices\_for\_final\_run\_flagを信号伝達する必要はない。

JVET-P0077におけるライン・ベースのCGパレット・モードのテキストを以下に示す。

【0089】

パレットコーディング構文

10

20

30

40

【表 6】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
palettePredictionFinished = 0		
NumPredictedPaletteEntries = 0		
for( predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[ startComp ] && !palettePredictionFinished && NumPredictedPaletteEntries[ startComp ] < palette_max_size; predictorEntryIdx++ ) {		10
<b>palette_predictor_run</b>	ae(v)	
if( palette_predictor_run != 1 ) {		
if( palette_predictor_run > 1 )		
predictorEntryIdx += palette_predictor_run - 1		
PalettePredictorEntryReuseFlags[ predictorEntryIdx ] = 1		
NumPredictedPaletteEntries++		
} else		20
palettePredictionFinished = 1		
}		
if( NumPredictedPaletteEntries < palette_max_size )		
<b>num_signalled_palette_entries</b>	ae(v)	
for( cIdx = startComp; cIdx < ( startComp + numComps); cIdx++ )		
for( i = 0; i < num_signalled_palette_entries; i++ )		
<b>new_palette_entries</b> [ cIdx ][ i ]	ae(v)	30
if( CurrentPaletteSize[ startComp ] > 0 )		
<b>palette_escape_val_present_flag</b>	ae(v)	
if( MaxPaletteIndex > 0 ) {		
adjust = 0		
<b>palette_transpose_flag</b>	ae(v)	
}		40

if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {	
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {	
<b>cu_qp_delta_abs</b>	ae(v)
if( cu_qp_delta_abs )	
<b>cu_qp_delta_sign_flag</b>	ae(v)
}	
}	
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {	
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {	
<b>cu_chroma_qp_offset_flag</b>	ae(v)
if( cu_chroma_qp_offset_flag )	
<b>cu_chroma_qp_offset_idx</b>	ae(v)
}	
}	
PreviousRunTypePosition = 0	
PreviousRunType = 0	
for( subSetId = 0; subSetId <= (cbWidth* cbHeight - 1) >> 4; subSetId++) {	
minSubPos = subSetId << 4	
if( minSubPos + 16 > cbWidth * cbHeight)	
maxSubPos = cbWidth * cbHeight	
else	
maxSubPos = minSubPos + 16	
RunCopyMap[ 0 ][ 0 ] = 0	
log2CbWidth = Log2( cbWidth )	
log2CbHeight = Log2( cbHeight )	
PaletteScanPos = minSubPos	
while( PaletteScanPos < maxSubPos ) {	
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	

10

20

30

40

50

if( PaletteScanPos > 0 ) {		
xcPrev =		
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev =		
y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
if( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {		10
<b>run_copy_flag</b>	ae(v)	
RunCopyMap[ xC ][ yC ] = run_copy_flag		
}		
CopyAboveIndicesFlag[ xC ][ yC ] = 0		
if( MaxPaletteIndex > 0 && ! RunCopyMap[startComp][xC][yC] ) {		
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC >		
0 ) )		20
&& CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {		
<b>copy_above_palette_indices_flag</b>	ae(v)	
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag		
}		
PreviousRunType = CopyAboveIndicesFlag[ xC ][ yC ]		
PreviousRunTypePosition = curPos		
} else {		
CopyAboveIndicesFlag[ xC ][ yC ] = CopyAboveIndicesFlag[xcPrev][ ycPrev]		30
}		
}		
PaletteScanPos ++		
}		
PaletteScanPos = minSubPos		
while( PaletteScanPos < maxSubPos ) {		
xC =		
x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		40

<pre> yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ] </pre>		
<pre> if( PaletteScanPos &gt; 0 ) { </pre>		
<pre>     xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ] </pre>		
<pre>     ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ] </pre>		10
<pre> } </pre>		
<pre> if( MaxPaletteIndex &gt; 0 ) { </pre>		
<pre>     if( ! RunCopyMap [ xC ][ yC ] &amp;&amp; CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>         if( MaxPaletteIndex - adjust &gt; 0 ) { </pre>		
<pre>             <b>palette_idx_idc</b> </pre>	ae(v)	
<pre>         } </pre>		
<pre>         adjust = 1 </pre>		
<pre>     } </pre>		20
<pre> } </pre>		
<pre> if( ! RunCopyMap [ xC ][ yC ] &amp;&amp; CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>     CurrPaletteIndex = palette_idx_idc </pre>		
<pre>     if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) { </pre>		
<pre>         PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex </pre>		
<pre>     } else { </pre>		
<pre>         if( !palette_transpose_flag ) </pre>		
<pre>             PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ] </pre>		30
<pre>         else </pre>		
<pre>             PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ] </pre>		
<pre>     } </pre>		
<pre> } </pre>		
<pre> if( palette_escape_val_present_flag ) { </pre>		
<pre>     for( cIdx = startComp; cIdx &lt; ( startComp + numComps ); cIdx++ ) </pre>		
<pre>         for( sPos = minSubPos; sPos &lt; maxSubPos; sPos++ ) { </pre>		40

$x_C = x_0 + \text{TraverseScanOrder}[\log_2 \text{CbWidth}][\log_2 \text{CbHeight}][\text{sPos}][0]$	
$y_C = y_0 + \text{TraverseScanOrder}[\log_2 \text{CbWidth}][\log_2 \text{CbHeight}][\text{sPos}][1]$	
$\text{if}(\text{PaletteIndexMap}[\text{cIdx}][x_C][y_C] == \text{MaxPaletteIndex}) \{$	
<b>palette_escape_val</b>	ae(v)
$\text{PaletteEscapeVal}[\text{cIdx}][x_C][y_C] = \text{palette\_escape\_val}$	
$\}$	
$\}$	
$\}$	
$\}$	
$\}$	

10

## 【0090】

## 7.4.9.6 パレットコーディングのセマンティクス

以下のセマンティクスでは、配列インデックス $x_0$ 、 $y_0$ は、ピクチャーの左上のルーマ・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・サンプルの位置( $x_0, y_0$ )を指定する。配列インデックス $x_C$ 、 $y_C$ は、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置( $x_C, y_C$ )を指定する。配列インデックス $\text{startComp}$ は、現在のパレット・テーブルの最初の色成分を指定する。0に等しい $\text{startComp}$ はY成分を示し、1に等しい $\text{startComp}$ はCb成分を示し、2に等しい $\text{startComp}$ はCr成分を示す。 $\text{numComps}$ は、現在のパレット・テーブルにおける色成分の数を指定する。

20

予測子パレットは、現在のパレット内の項目を予測するために使用される、以前のコーディング単位からのパレット項目で構成される。

変数 $\text{PredictorPaletteSize}[\text{startComp}]$ は、現在のパレット・テーブルの最初の色成分 $\text{startComp}$ についての予測子パレットのサイズを指定する。 $\text{PredictorPaletteSize}$ は、節8.4.5.3において規定されるように導出される。

30

変数 $\text{PalettePredictorEntryReuseFlags}[i]$ が1に等しいことは、予測子パレットにおける $i$ 番目の項目が現在のパレットにおいて再利用されることを指定する。 $\text{PalettePredictorEntryReuseFlags}[i]$ が0に等しいことは、予測子パレットの $i$ 番目の項目が現在のパレットにおける項目ではないことを指定する。配列 $\text{PalettePredictorEntryReuseFlags}[i]$ のすべての要素が0に初期化される。

## 【0091】

$\text{palette\_predictor\_run}$ は、配列 $\text{PalettePredictorEntryReuseFlags}$ におけるゼロでない項目の前にあるゼロの数を決定するために使用される。

$\text{palette\_predictor\_run}$ の値は、0から( $\text{PredictorPaletteSize} - \text{predictorEntryIdx}$ )の範囲内(両端含む)であることがビットストリーム適合性の要件である。ここで、 $\text{predictorEntryIdx}$ は、配列 $\text{PalettePredictorEntryReuseFlags}$ における現在位置に対応する。変数 $\text{NumPredictedPaletteEntries}$ は、予測子パレットから再利用される現在のパレットにおける項目数を指定する。 $\text{NumPredictedPaletteEntries}$ の値は、0から $\text{palette\_max\_size}$ の範囲内(両端含む)である。

40

## 【0092】

$\text{num\_signalled\_palette\_entries}$ は、現在のパレット・テーブルの最初の色成分 $\text{startComp}$ について明示的に信号伝達される、現在のパレットにおける項目の数を指定する。 $\text{num\_signalled\_palette\_entries}$ が存在しない場合、0に等しいと推定される。

変数 $\text{CurrentPaletteSize}[\text{startComp}]$ は、現在のパレット・テーブルの最初の色成分

50



startCompについての現在のパレットのサイズを指定し、次のように導出される：

```
CurrentPaletteSize[startComp] = NumPredictedPaletteEntries + num_signalled_palette_entries (7-155)
```

CurrentPaletteSize[startComp]の値は、0からpalette\_max\_sizeの範囲内（両端含む）である。

【0093】

new\_palette\_entries[cIdx][i]は、色成分cIdxについてのi番目の信号伝達されるパレット項目についての値を指定する。

変数PredictorPaletteEntries[cIdx][i]は、色成分cIdxについての予測子パレットにおけるi番目の要素を指定する。

10

変数CurrentPaletteEntries[cIdx][i]は、色成分cIdxについての現在のパレットにおけるi番目の要素を指定し、次のように導出される：

【表7】

```
numPredictedPaletteEntries = 0
```

```
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
```

```
  if( PalettePredictorEntryReuseFlags[ i ] ) {
```

```
    for( cIdx =startComp; cIdx < ( startComp + numComps ); cIdx++ )
```

```
      CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
```

20

```
      PredictorPaletteEntries[ cIdx ][ i ]
```

```
      numPredictedPaletteEntries++
```

```
    }
```

```
for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ ) (7-156)
```

```
  for( i = 0; i < num_signalled_palette_entries[startComp]; i++ )
```

```
    CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = new_palette_entries[ cIdx ][ i ]
```

30

【0094】

palette\_escape\_val\_present\_flagが1に等しいことは、現在のコーディング単位が少なくとも1つのエスケープコーディングサンプルを含むことを指定する。escape\_val\_present\_flagが0に等しいことは、現在のコーディング単位においてエスケープコーディングされたサンプルがないことを指定する。存在しない場合、palette\_escape\_val\_present\_flagの値は1に等しいと推定される。

変数MaxPaletteIndexは、現在のコーディング単位についてのパレット・インデックスの可能な最大値を指定する。MaxPaletteIndexの値は

```
CurrentPaletteSize[startComp] - 1 + palette_escape_val_present_flag
```

に等しく設定される。

40

【0095】

palette\_idx\_idcは、パレット・テーブル、CurrentPaletteEntriesへのインデックスの指示である。palette\_idx\_idcの値は、ブロック内の最初のインデックスについては0からMaxPaletteIndexの範囲内（両端含む）であり、ブロック内の残りのインデックスについては0から（MaxPaletteIndex - 1）の範囲内（両端含む）である。

palette\_idx\_idcが存在しない場合、0に等しいと推定される。

【0096】

palette\_transpose\_flagが1に等しいことは、現在のコーディング単位内のサンプルについてのインデックスを走査するために垂直横断走査が適用されることを指定する。palette\_transpose\_flagが0に等しいことは、現在のコーディング単位内のサンプルに

50

ついでにインデックスを走査するために水平横断走査が適用されることを指定する。存在しない場合、palette\_transpose\_flagの値は0に等しいと推定される。

配列TraverseScanOrderは、パレットコーディングのための走査順序配列を指定する。palette\_transpose\_flagが0に等しい場合、TraverseScanOrderは水平走査順序HorTravScanOrderを割り当てられ、palette\_transpose\_flagが1に等しい場合、TraverseScanOrderは垂直走査順序VerTravScanOrderを割り当てられる。

【0097】

run\_copy\_flagが1に等しいことは、copy\_above\_palette\_indices\_flagが0に等しい場合、パレット・ラン・タイプが前に走査された位置でのラン・タイプと同じであり、パレット・ラン・インデックスが前の位置でのインデックスと同じであることを指定する。それ以外の場合は、run\_copy\_flagは0に等しい。

10

【0098】

copy\_above\_palette\_indices\_flagが1に等しいことは、パレット・インデックスが、水平横断走査が使用される場合には、上の行における同じ位置、または垂直横断走査が使用される場合には、左の列における同じ位置にあるパレット・インデックスに等しいことを指定する。copy\_above\_palette\_indices\_flagが0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいてコーディングされるか、または推定されることを指定する。

【0099】

変数CopyAboveIndicesFlag[xC][yC]が1に等しいことは、パレット・インデックスが上の行（水平走査）または左の列（垂直走査）におけるパレット・インデックスからコピーされることを指定する。

20

CopyAboveIndicesFlag[xC][yC]が0に等しいことは、パレット・インデックスがビットストリームにおいて明示的にコーディングされるか、推定されることを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対する、サンプルの位置(xC,yC)を指定する。

【0100】

変数PaletteIndexMap[xC][yC]は、CurrentPaletteEntriesによって表される配列へのインデックスであるパレット・インデックスを指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。PaletteIndexMap[xC][yC]の値は、0からMaxPaletteIndexの範囲内（両端含む）である。

30

【0101】

変数adjustedRefPaletteIndexは次のように導出される：

40

50

## 【表 8】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

10

20

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++ (7-158)

```

## 【0 1 0 2】

palette\_escape\_valは、ある成分についての量子化されたエスケープコーディングされたサンプル値を指定する。

変数PaletteEscapeVal[cldx][xC][yC]は、PaletteIndexMap[xC][yC]がMaxPaletteIndexに等しく、palette\_escape\_val\_present\_flagが1に等しいサンプルのエスケープ値を指定する。配列インデックスcldxは、色成分を指定する。配列インデックスxC、yCは、ピクチャーの左上のルーマ・サンプルに対するサンプルの位置(xC,yC)を指定する。

30

PaletteEscapeVal[cldx][xC][yC]が、0に等しいcldxについては0から(1 - (BitDepth<sub>y</sub> + 1) - 1)の範囲内(両端含む)であり、0に等しくないcldxについては0から(1 - ((BitDepth<sub>c</sub> + 1) - 1)の範囲内(両端含む)であることがビットストリーム適合性の要件である。

## 【0 1 0 3】

## 2.3 VVCにおけるローカル・デュアル・ツリー

典型的なハードウェア・ビデオ・エンコーダおよびデコーダでは、ピクチャーがより多くの小さなイントラブロックを有する場合、近傍のイントラブロック間のサンプル処理データ依存性のため、処理スループットが低下する。イントラブロックの予測子生成は、近傍ブロックからの上部および左の境界の再構成されたサンプルを必要とする。よって、イントラ予測はブロックごとに順次処理されなければならない。

40

## 【0 1 0 4】

HEVCでは、最小のCUは8×8ルーマ・サンプルである。最小のイントラCUのルーマ成分は、4つの4×4ルーマ・イントラ予測単位(PU)にさらに分割できるが、最小のイントラCUのクロマ成分はそれ以上分割できない。したがって、最悪ケースのハードウェア処理スループットは、4×4クロマ・イントラブロックまたは4×4ルーマ・イントラブロックが処理されるときに発生する。

50

## 【0105】

VTM5.0では、単一コーディングツリーでは、クロマ・パーティションは常にルーマに従い、最小のイントラCUは $4 \times 4$ ルーマ・サンプルであるため、最小のクロマ・イントラCBは $2 \times 2$ である。したがって、VTM5.0では、単一コーディングツリーにおける最小のクロマ・イントラCBは $2 \times 2$ である。VVC復号のための最悪ケースのハードウェア処理スループットは、HEVC復号の場合の $1/4$ だけである。さらに、クロマ・イントラCBの再構成プロセスは、成分横断線形モデル(cross-component linear model、CCLM)、4タップ補間フィルタ、位置依存イントラ予測結合(position-dependent intra prediction combination、PDPC)、および復号インター・イントラ予測(combined inter intra prediction、CIIP)を含むツリーを採用した後、HEVCの場合よりはるかに複雑になる。ハードウェア・デコーダにおいて高い処理スループットを達成することは困難である。本節では、最悪ケースのハードウェア処理スループットを改善する方法が提案される。

10

## 【0106】

この方法の目標は、クロマ・イントラCBのパーティション分割を制約することにより、 $16$ クロマ・サンプルより小さいクロマ・イントラCBを禁止することである。

## 【0107】

単一コーディングツリーでは、SCIPUは、クロマ・ブロック・サイズがTHクロマ・サンプル以上であり、 $4$ THルーマ・サンプルより小さい少なくとも1つの子ルーマ・ブロックをもつコーディングツリー・ノードとして定義される。ここで、THはこの寄稿では $16$ に設定される。各SCIPUでは、すべてのCBがインターであるか、またはすべてのCBが非インターである、すなわちイントラまたはIBCのいずれかであることが要求される。非インターSCIPUの場合、さらに、非インターSCIPUのクロマがそれ以上分割されず、SCIPUのルーマがさらに分割されることが許されることが要求される。このようにして、最小のクロマ・イントラCBサイズは $16$ クロマ・サンプルであり、 $2 \times 2$ 、 $2 \times 4$ 、および $4 \times 2$ クロマCBが除去される。さらに、クロマ・スケーリングは、非インターSCIPUの場合には適用されない。さらに、ルーマ・ブロックがさらに分割され、クロマ・ブロックが分割されない場合、ローカルのデュアル・ツリーコーディング構造が構築される。

20

## 【0108】

2つのSCIPUの例を図5のAおよびBに示されている。図5のAにおいては、 $8 \times 4$ クロマ・サンプルの1つのクロマCBと3つのルーマCB( $4 \times 8$ 、 $8 \times 8$ 、 $4 \times 8$ のルーマCB)が1つのSCIPUを形成するが、これは、 $8 \times 4$ クロマ・サンプルからの三分木(TT)分割は、 $16$ クロマ・サンプルより小さなクロマCBを生じるためである。図5のBでは、 $4 \times 4$ クロマ・サンプルの一方のクロマCB( $8 \times 4$ クロマ・サンプルの左側)と3つのルーマCB( $8 \times 4$ 、 $4 \times 4$ 、 $4 \times 4$ のルーマCB)が1つのSCIPUを形成し、 $4 \times 4$ サンプルの他方のクロマCB( $8 \times 4$ クロマ・サンプルの右側)と2つのルーマCB( $8 \times 4$ 、 $8 \times 4$ のルーマCB)が1つのSCIPUを形成する。これは、 $4 \times 4$ クロマ・サンプルからの二分木(BT)分割は、 $16$ のクロマ・サンプルよりも小さなクロマCBを生じるためである。

30

## 【0109】

提案される方法では、現在のスライスがIスライスであるか、または現在のSCIPUがさらに1回分割した後に $4 \times 4$ ルーマ・パーティションをその中に有する場合、SCIPUのタイプは非インターであると推定される(VVCではインター $4 \times 4$ は許可されないため)；それ以外の場合、SCIPUのタイプ(インターまたは非インター)は、SCIPUにおいてCUをパースする前に、1つの信号伝達されるフラグによって示される。

40

## 【0110】

上記の方法を適用することにより、 $2 \times 2$ のクロマ・ブロックの代わりに $4 \times 4$ 、 $2 \times 8$ 、または $8 \times 2$ のクロマ・ブロックが処理する場合に、最悪ケースのハードウェア処理スループットが発生する。最悪ケースのハードウェア処理スループットはHEVCの場合と同じであり、VTM5.0の場合の4倍である。

## 【0111】

50

## 2.4 変換スキップ (TS)

HEVCの場合と同様に、ブロックの残差は変換スキップ・モードでコーディングされることができる。構文コーディングの冗長性を避けるために、CUレベルMTS\_CU\_flagがゼロに等しくない場合、変換スキップ・フラグは信号伝達されない。変換スキップのためのブロック・サイズ制限は、JEM4におけるMTSについての場合と同じである。このことは、ブロック幅と高さの両方が32以下の場合に、変換スキップがCUに適用できることを示している。現在のCUについてLFNSTまたはMIPがアクティブ化される場合、暗黙的なMTS変換がDCT2に設定されることに注意されたい。また、暗黙的なMTSは、インターコーディングされたブロックについてMTSが有効にされている場合でも、有効にされることができる。

【0 1 1 2】

また、変換スキップ・ブロックについて、最小の許容される量子化パラメータ (QP) は  $6 * (\text{internalBitDepth} - \text{inputBitDepth}) + 4$  と定義される。

【0 1 1 3】

## 2.5 代替的なルーマ半画素補間フィルタ

JVET-N0309では、代替的な半画素補間フィルタが提案されている。

【0 1 1 4】

半画素ルーマ補間フィルタのスイッチングは、動きベクトル精度に依存して行われる。既存の1/4画素、フル画素、および4画素AMVRモードに加えて、新しい半画素精度AMVRモードが導入される。半画素動きベクトル精度の場合にのみ、代替的な半画素ルーマ補間フィルタが選択されることができる。

【0 1 1 5】

半画素動きベクトル精度 (すなわち、半画素AMVRモード) を使用する非アフィン非マージ・インターコーディングされるCUについて、HEVC/VVC半画素ルーマ補間フィルタと一つまたは複数の代替的な半画素補間との間の切り換えが、新しい構文要素hpellfldxの値に基づいて行われる。構文要素hpellfldxは、半画素AMVRモードの場合にのみ信号伝達される。空間的マージ候補を使用するスキップ/マージ・モードの場合、構文要素hpellfldxの値は近傍ブロックから継承される。

【0 1 1 6】

## 2.6 適応色変換 (Adaptive Color Transform, ACT)

図6は、ACTが適用される復号フローチャートを示す。図6に示されるように、色空間変換は、残差ドメインで実行される。具体的には、1つの追加的な復号モジュール、すなわち逆ACTが、YCgCoドメインからの残差をもとのドメインに戻す変換をするための逆変換の後に、導入される。

【0 1 1 7】

VVCでは、最大変換サイズが1 コーディング 単位 (CU) の幅または高さより小さいのでない限り、1つのCUリーフノードも、変換処理の単位として使用される。よって、提案される実装では、ACTフラグは、残差をコーディング するための色空間を選択するために、1つのCUについて信号伝達される。さらに、HEVC ACT設計に従い、インターおよびIBC CUについて、ACTは、CU内に少なくとも1つのゼロでない係数がある場合にのみ有効にされる。イントラCUについては、ACTは、クロマ成分がルーマ成分の同じイントラ予測モード、すなわちDMモードを選択する場合にのみ、有効にされる。色空間変換のために使用されるコア変換は、HEVCのために使用されるものと同じに保たれる。具体的には、以下の順方向および逆方向YCgCo色変換行列が適用される。

【数 1】

$$\begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} / 4 \quad \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} C'_0 \\ C'_1 \\ C'_2 \end{bmatrix}$$

10

20

30

40

50

## 【 0 1 1 8 】

さらに、色変換前後の残差信号のダイナミックレンジ変化を補償するために、( - 5, - 5, - 3 ) のQP調整が変換残差に適用される。

他方、順色変換と逆色変換は、3つの成分すべての残差にアクセスする必要がある。対応して、提案される実装では、ACTは、3つの成分のすべての残差が利用可能ではない以下の2つのシナリオでは無効にされる。

1. 別個木パーティション ( separate-tree partition ) : 別個の木が適用される場合、1つのCTU内のルーマおよびクロマ・サンプルは、異なる構造によって分割される。その結果、ルーマ・ツリー内のCUはルーマ成分のみを含み、クロマ・ツリー内のCUは2つのクロマ成分のみを含むことになる。イントラ・サブパーティション予測 ( Intra sub-partition prediction、ISP ) : ISPサブパーティションはルーマに適用されるだけであり、一方、クロマ信号は分割なしにコーディングされる。現在のISP設計では、最後の諸ISPサブパーティションを除いて、他のサブパーティションはルーマ成分だけを含む。

10

## 【 0 1 1 9 】

3. 本明細書に記載される技術的解決策および実施形態によって解決される技術的問題

1. エスケープ・シンボルの現在のバイナリ化は固定長ではなく、このことは、一様分布をもつ源に好適でありうる。

2. 現在のパレットコーディング設計は、可能性のある冗長性を除去するためにインデックス調整プロセスを実行し、これは、たとえばエスケープ値インデックスが誤って導出された場合に、パース依存性 ( parsing dependency ) を導入する可能性がある。

20

3. 現在インデックスを導出するために使用される参照インデックスは、エンコーダ制約条件を必要とすることがあり、これは、現在の設計では考慮されず、コーデック設計のために望ましくない。

4. ローカル・デュアル・ツリーが有効にされる場合、前のブロックと現在のブロックのパレット項目は、異なる数の色成分をもつ可能性がある。そのような場合の対処方法は不明である。

5. ローカル・デュアル・ツリーとPLTは、単一ツリー領域からデュアル・ツリー領域にコーディングする際に、いくつかのパレット項目が繰り返される可能性があるため、同時に適用できなかった。一例が図7に示される。

30

6. joint\_cbcモードのためのクロマQPテーブルが制約される可能性がある。

7. ある種の条件下では、エスケープ・サンプルが冗長になることがある。

8. ライン・ベースのCGモードは、高スループットで処理できなかった。

## 【 0 1 2 0 】

4. 実施形態および解決策のリスト

以下のリストは、一般的な概念を説明するための例として考慮されるべきである。これらの項目は狭義に解釈すべきではない。さらに、これらの項目は、任意の仕方で組み合わせることができる。

## 【 0 1 2 1 】

以下の例は、VVCにおけるパレット方式および他のすべてのパレット関連方式に適用されうる。

40

## 【 0 1 2 2 】

以下の項目では、Qpは、JVET-P2001-vEのセクション8.4.5.3におけるqPを示しうる。

## 【 0 1 2 3 】

以下の項目では、QpPrimeTsMinは、変換スキップ・モードのための許容される最小量子化パラメータである。

## 【 0 1 2 4 】

Modulo(x,M)は、xが正の整数の場合は ( x%M ) として定義され、そうでない場合は M - ( - x)%Mとして定義される。

50

## 【0125】

以下では、無損失モードでコーディングされたブロックは、1に等しい`tranquant_by_pass_flag`でコーディングされたブロック、または所与の閾値より大きくないQPおよび1に等しい`transform_skip_flag`でコーディングされたブロックを意味する。

## 【0126】

以下の例は、VVCにおけるパレット方式および他のすべてのパレット関連方式に適用される。

## 【0127】

1. 固定長コーディングが、エスケープ・シンボルをコーディングするために適用されてもよい。

a. 一例では、エスケープ・シンボルは、固定長バイナリ化で信号伝達されてもよい。

b. 一例では、エスケープ・シンボルは、Nビットを使用して固定長バイナリ化で信号伝達されてもよい。

c. 一例では、エスケープ・シンボルを信号伝達するための符号長（たとえば、項目1.bで言及されているN）は、内部ビット深さに依存する。

i. あるいはまた、エスケープ・シンボルを信号伝達するための符号長は、入力ビット深さに依存してもよい。

ii. あるいはまた、エスケープ・シンボルを信号伝達するための符号長は、内部ビット深さと入力ビット深さとの間の差に依存してもよい。

iii. 一例では、Nは入力/内部ビット深さに等しく設定される。

d. 一例では、固定長コーディングの長さは、ビデオ処理単位レベル、たとえば、スライス・サブピクチャー、タイル、ピクチャー、ビデオにおいて信号伝達されてもよい。

e. 一例では、エスケープ・シンボルを信号伝達するための符号長（たとえば、項目1.bで言及されているN）は、量子化パラメータ、すなわち $Q_p$ に依存してもよい。

i. 一例では、エスケープ・シンボルを信号伝達するための符号長は、 $f(Q_p)$ で示されるような量子化パラメータの関数であってもよい。

1. 一例では、関数 $f$ は（内部ビット深さ -  $g(Q_p)$ ）と定義されてもよい。

2. 一例では、Nは（内部ビット深さ -  $\max(16, (Q_p - 4)/6)$ ）に設定されてもよい。

3. 一例では、Nは（内部ビット深さ -  $\max(Q_pPrimeTsMin, (Q_p - 4)/6)$ ）に設定されてもよい。ここで、 $qP$ は、デコードされた量子化パラメータであり、 $Q_pPrimeTsMin$ は、変換スキップ・モードのための最小の許容される量子化パラメータである。

4. あるいはまた、さらに、符号長Nは、 $\max(A, \text{内部ビット深さ} - (\max(Q_pPrimeTsMin, Q_p) - 4)/6)$ に設定されてもよく、ここで、Aは、0または1のような負でない整数値である。

ii. 上記サブ項目において言及された $Q_p$ は、スライスQPを指してもよい。

1. あるいはまた、 $Q_p$ は、スライスQPに一定値を加えたものを指してもよい。

f. 上記の諸例において、Nは、0以上であってもよい。

## 【0128】

2. エスケープ・シンボルのための脱量子化 $Q_p$ は、スライス/ピクチャー/PPSレベル $Q_p$ に基づいていてもよい。

a. 一例では、エスケープ・シンボルのための脱量子化 $Q_p$ は、スライス/ピクチャー/PPSレベル $Q_p$ に所与のオフセットを加えたものに基づいていてもよい。

i. オフセットは定数であってもよい。

ii. オフセットは、暗黙的にまたは明示的に、ビットストリームにおいて示されてもよい。

b. 一例では、ブロック・レベルの $Q_p$ 差はビットストリームにおいてスキップされてもよい。

i. 一例では、 $cbf$ は、0と推定されてもよい。

## 【0129】

10

20

30

40

50

3. エスケープ・シンボルについて、脱量子化の前に左シフトが適用されてもよい。

a. 一例では、 $N$ ビットの左シフト ( $N = 0$ ) が、脱量子化の前に適用されてもよい。

i. 一例では、 $N$ は  $\text{Min}(\text{bitDepth} - 1, (\text{QpPrimeTsMin} - 4)/6)$  に等しくてもよい。ここで、 $\text{bitDepth}$ は内部ビット深さであり、ここで、 $\text{bitDepth}$ は内部ビット深さである。

ii. あるいはまた、 $N$ は、 $\text{bitDepth} - \text{inputBD}$ に等しくてもよく、ここで、 $\text{inputBD}$ は、入力ビット深さである。

1. 一例では、 $\text{inputBD}$ がビットストリームにおいて示されてもよい。

iii. あるいはまた、 $N$ は、 $\text{deltaBD}$ と等しくてもよく、ここで、 $\text{deltaBD}$ は、ビットストリームにおいて示される。

10

【0130】

4. エスケープ・シンボルの脱量子化は  $(Qp - \text{QpPrimeTsMin})$  に依存してもよい。

a. 一例では、 $(Qp - \text{QpPrimeTsMin} + 4)$ は、脱量子化 $Qp$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

b. 一例では、 $\text{Min}(Qp - \text{QpPrimeTsMin} + 4, 63 + \text{QpBdOffset})$ は、脱量子化 $Qp$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

【0131】

5. エスケープ・シンボルの脱量子化は  $(Qp - N * 6)$  に依存してもよい。

a. 一例では、 $N$ は、項目3.aにおける左方シフトの数を指してもよい。

b. 一例では、 $\text{Max}(0, Qp - N * 6)$ が脱量子化 $Qp$ として適用されてもよい。

20

【0132】

6. エスケープ・シンボルの脱量子化は、 $\text{deltaBD}$ 、すなわち、内部ビット深さと入力ビット深さの差に依存してもよい。

a. 一例では、 $(Qp - \text{deltaBD} * 6)$ は、脱量子化 $Qp$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

b. 一例では、 $\text{Min}(\text{Max}(0, Qp - \text{deltaBD} * 6), 63 + \text{QpBdOffset})$ は、脱量子化 $Qp$ としてエスケープ・シンボルの脱量子化のために適用されてもよい。

【0133】

7. 一つのビデオ単位 (たとえば、CU) におけるエスケープ・シンボルの使用を無効にすることが提案される。

30

a. あるいはまた、さらに、エスケープ・シンボルの存在の指示を信号伝達することはスキップされる。

b. 一例では、エスケープ・シンボルの使用を有効/無効にするかどうかは、量子化パラメータおよび/またはビット深さに依存してもよい。

i. 一例では、 $(\text{内部ビット深さ} - (\text{Max}(\text{QpPrimeTsMin}, \text{Qp}) - 4)/6)$ が0より大きくない場合、エスケープ・シンボルの使用は無効にされてもよい。

【0134】

8. 3次のEGを除く可変長コーディングが、エスケープ・シンボルをコーディングするために適用されてもよい。

a. 一例では、エスケープ・シンボルのバイナリ化は、入力パラメータ $K$ をもつ打ち切りされたバイナリ (truncated binary, TB) であってもよい。

40

b. 一例では、エスケープ・シンボルのバイナリ化は、 $K$ 次のEGであってもよい。ここで、 $K$ は3に等しくない。

i. 一例では、エスケープ・シンボルのバイナリ化は、0次のEGであってもよい。

1. あるいはまた、一例では、エスケープ・シンボルのバイナリ化は、1次のEGであってもよい。

2. あるいはまた、一例では、エスケープ・シンボルのバイナリ化は、2次のEGであってもよい。

c. 上記の諸例では、 $K$ は整数であってもよく、以下に依存してもよい。

i. SPS/VPS/PPS/ピクチャー・ヘッダ/スライス・ヘッダ/タイル・グループ

50



- ・ヘッダ/LCU行/LCUのグループ/ブリックにおいて信号伝達されるメッセージ。
  - i i . 内部ビット深さ
  - i i i . 入力ビット深さ
  - i v . 内部ビット深さと入力深さの差
  - v . 現在ブロックのブロック寸法
  - v i . 現在ブロックの現在の量子化パラメータ
  - v i i . カラー・フォーマットの指示 (4:2:0、4:4:4、RGBまたはYUVなど)
  - v i i i . コーディング構造 (単一ツリーまたはデュアル・ツリーなど)
  - i x . 色成分 (ルーマ成分および/またはクロマ成分など)

## 【0135】

9 . エスケープ・シンボルをコーディングするための複数のバイナリ化方法が、ビデオ単位 (たとえば、シーケンス/ピクチャー/スライス/タイル/ブリック/サブピクチャー/CTU行/CTU/CTB/CB/CU/ピクチャー内のサブ領域) および/またはエスケープ・シンボルの一つまたは複数の値に対して適用されてもよい。

a . 一例では、複数のバイナリ化方法のうちの一つをどのようにして選択するかが、ビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について信号伝達されてもよい。

b . 一例では、複数のバイナリ化方法のうちの一つをどのようにして選択するかが、ビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について導出されてもよい。

c . 一例では、一つのビデオ単位および/またはエスケープ・シンボルの一つまたは複数の値について、2つ以上のバイナリ化方法が適用されてもよい。

i . 一例では、インデックスまたはフラグが、選択されたバイナリ化方法を知らせるためにエンコード/デコードされてもよい。

## 【0136】

以下の項目では、pは色成分のシンボル値を表してもよく、bdはビット深さ(たとえば、内部ビット深さまたは入力ビット深さ)を表してもよく、ibdは入力ビット深さを表してもよく、Qpは変換スキップ・ブロックまたは変換ブロックについての量子化パラメータを表してもよい。さらに、ルーマ成分とクロマ成分についてのQPは、異なっても、同じであってもよい。ビット深さは、所与の色成分に関連付けられていてもよい。

## 【0137】

10 . 量子化および/または逆量子化プロセスをどのように適用するかは、ブロックがパレット・モードでコーディングされるか否かに依存してもよい。

a . 一例では、エスケープ・シンボルについての量子化および/または逆量子化プロセスは、量子化が適用された通常のイントラ/インターコーディングされたブロックについて使用されるものとは異なってもよい。

## 【0138】

11 . エスケープ・シンボルについての量子化および/または逆量子化プロセスは、ビットシフトを使用してもよい。

a . 一例では、エスケープ・シンボルを量子化するために右ビットシフトが使用されてもよい。

i . 一例では、エスケープ・シンボルは、 $f(p, Q_p)$ として信号伝達されてもよく、ここで、pは入力シンボル値 (たとえば、入力ルーマ/クロマ・サンプル値) であり、 $Q_p$ は、対応する色成分についての導出された量子化パラメータである。

1 . 一例では、関数fは $p \quad g(Q_p)$ として定義されてもよい。

2 . 一例では、関数fは $(p + (1 \quad (g(Q_p) - 1))) \quad g(Q_p)$ として定義されてもよい。

3 . 一例では、関数fは、 $(0, (1 \quad bd) - 1, (p + (1 \quad (g(Q_p) - 1))) \quad g(Q_p))$ として定義されてもよい。

i i . 一例では、エスケープ・シンボルは、 $h(p)$ として信号伝達されてもよい。

10

20

30

40

50

- 1 . 一例では、関数  $h$  は、 $p \quad N$  として定義されてもよい。
  - 2 . 一例では、関数  $h$  は、 $(p + (1 \quad (N - 1))) \quad N$  として定義されてもよい。
  - 3 . 一例では、`cu_transquant_bypass_flag` が 1 に等しい場合、 $N$  は 0 に設定されてもよい。
  - 4 . 一例では、`cu_transquant_bypass_flag` が 1 に等しい場合、 $N$  は  $(bd - ibd)$  に等しくてもよい。ここで、 $bd$  は内部ビット深さであり、 $ibd$  は入力ビット深さである。
  - 5 . 一例では、関数  $h$  は  $clip(0, (1 \quad (bd - N) - 1, p \quad N)$  として定義されてもよく、ここで、 $bd$  は現在の色成分についての内部ビット深さである。
  - 6 . 一例では、関数  $h$  は  $clip(0, (1 \quad (bd - N) - 1, (p + (1 \quad (N - 1))) \quad N)$  として定義されてもよく、ここで、 $bd$  は現在の色成分についての内部ビット深さである。 10
  - 7 . 上記の例において、 $N$  は  $[0, (bd - 1)]$  の範囲であってもよい。
  - b . 一例では、左ビットシフトは、エスケープ・シンボルを逆量子化するために使用されてもよい。
    - i . 一例では、エスケープ・シンボルは、 $f(p, Qp)$  として脱量子化されてもよい。ここで、 $p$  は、復号されたエスケープ・シンボルであり、 $Qp$  は、対応する色成分についての導出された量子化パラメータである。
      - 1 . 一例では、 $f$  は  $p \quad g(Qp)$  として定義されてもよい。
      - 2 . 一例では、 $f$  は  $(p \quad g(Qp)) + (1 \quad (g(Qp) - 1))$  として定義されてもよい。
    - i i . 一例では、エスケープ・シンボルは、 $f(p, Qp)$  として再構成されてもよい。ここで、 $p$  は、復号されたエスケープ・シンボルである。 20
      - 1 . 一例では、 $f$  は  $clip(0, (1 \quad bd) - 1, p \quad g(Qp))$  として定義されてもよい。
      - 2 . 一例では、 $f$  は、 $clip(0, (1 \quad bd) - 1, (p \quad g(Qp)) + (1 \quad (g(Qp) - 1)))$  として定義されてもよい。
    - i i i . 一例では、エスケープ・シンボルは  $h(p)$  として再構成されてもよい。
      - 1 . 一例では、関数  $h$  は  $p \quad N$  として定義されてもよい。
      - 2 . 一例では、関数  $h$  は、 $(p \quad N) + (1 \quad (N - 1))$  として定義されてもよい。
      - 3 . 一例では、`cu_transquant_bypass_flag` が 1 に等しい場合、 $N$  は 0 に設定されてもよい。
  - 4 . 一例では、`cu_transquant_bypass_flag` が 1 に等しい場合、 $N$  は  $(bd - ibd)$  に等しくてもよく、ここで、 $bd$  は内部ビット深さであり、 $ibd$  は入力ビット深さである。 30
  - 5 . 一例では、 $N$  は  $(\max(QpPrimeTsMin, qP) - 4) / 6$  に設定され、ここで、 $qP$  は復号された量子化パラメータであり、 $QpPrimeTsMin$  は変換スキップ・モードのための最小の許容される量子化パラメータである。
    - a ) 上記の例では、ルーマとクロマの両方が変換スキップ・モードをもつ場合、変換スキップ・モードのための異なる最小の許容される量子化パラメータが、異なる色成分について適用されてもよい。
    - 6 . あるいはまた、上記の諸例について、 $N$  は、 $\min(bd - 1, N)$  のように、さらにクリップされてもよい。 40
    - 7 . 上記の例では、 $N$  は  $[0, (bd - 1)]$  の範囲であってもよい。
- 【 0 1 3 9 】
- 1 2 . 脱量子化として左シフトを適用する場合、エスケープ・シンボル  $p$  の再構成オフセットはビット深さ情報に依存してもよい。
    - a . 一例では、それは、内部ビット深さと入力ビット深さの間の差、すなわち、 $\delta BD = \text{内部ビット深さ} - \text{入力ビット深さ}$  に依存してもよい。
    - b .  $K$  が  $\delta BD$  以下である場合、再構成された値は  $p \quad K$  であってもよい。
    - c .  $K$  が  $\delta BD$  より大きい場合、再構成値は  $(p \quad K) + (1 \quad (K - 1))$  であってもよい。
    - d .  $K$  が  $T0$  以下である場合 (たとえば、 $T0 = 2$ )、再構成された値は  $p \quad K$  であって 50

もよい。

e . KがT1より大きい場合 (たとえば、T1 = 2 )、再構成値は(p K) + (1 (K - 1))であってもよい。

f . 一例では、項目dおよびeにおけるT0およびT1は、シーケンス / ピクチャー / スライス / タイル / ブリック / サブピクチャー・レベルなどで、ビットストリームにおいて信号伝達されてもよい。

g . 一例では、再構成値は、(p K) + ((1 (K - 1) deltaBD deltaBD))であってもよい。

h . 一例では、再構成値は、((p (K + 1)) + (1 K)) (deltaBD + 1) del taBD)であってもよい。

i . 一例では、deltaBDは、シーケンス / ピクチャー / スライス / タイル / ブリック / サブピクチャー・レベルなどでビットストリームにおいて信号伝達されてもよい。

j . 一例では、どの再構成値が使用されるか (たとえば、項目bないしe) は、現在のブロックの量子化パラメータに依存してもよい。

k . 一例では、どの再構成値が使用されるか (たとえば、項目bないしe) は、deltaBDの値に依存してもよい。

l . 一例では、Kは、g(Qp)に設定されてもよい。

【 0 1 4 0 】

1 3 . 上記の諸例では、以下が適用されてもよい :

a . 一例では、エスケープ・シンボルは、コンテキストコーディングされてもよい。

b . 一例では、エスケープ・シンボルはバイパスコーディングされてもよい。

c . 一例では、g(Qp)は、(Qp - 4) / 6またはQP / 8として定義されてもよい。

i . あるいはまた、g(Qp)はQp / 6またはQP / 8として定義されてもよい。

i i . あるいはまた、g(Qp)はmax(16, Qp / 6)として定義されてもよい。

i i i . あるいはまた、g(Qp)はmax(16, (Qp - 4) / 6)として定義されてもよい。

i v . あるいはまた、g(Qp)はmax((bd - ibd) \* 6 + 4, (Qp - 4) / 6)として定義されてもよい。

v . あるいはまた、g(Qp)はmax(M, (Qp - 4) / 6)として定義されてもよい。

1 . 一例では、Mはデコーダに信号伝達されてもよい。

v i . あるいはまた、g(Qp)はmax((M, Qp) - 4) / 6として定義されてもよい。

1 . 一例では、MはSPSにおいて示されてもよい。

2 . 一例では、ルーマ成分およびクロマ成分に同じMまたは異なるMが適用される。

3 . 一例では、Mは(bd - ibd) \* 6 + 4に等しくてもよい。

v i i . あるいはまた、g(Qp)はQp / 6またはQP / 8として定義されてもよい。

v i i i . あるいはまた、g(Qp)は(max(16, Qp) / 6)として定義されてもよい。

i x . あるいはまた、g(Qp)は(max(16, Qp) - 4) / 6として定義されてもよい。

d . 一例では、g(Qp)の値は[0, (bd - 1)]の範囲であってもよい。

e . 一例では、最大関数max(a, i)は(i = a? a : i)として定義されてもよい。

i . あるいはまた、一例では、最大関数max(a, i)は(i a? a : i)として定義されてもよい。

f . 一例では、Nは整数 (たとえば、8または10) であってもよく、以下に依存してもよい。

i . SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ

i i . 内部ビット深さ

i i i . 入力ビット深さ

i v . 内部ビット深さと入力深さの差

v . 現在ブロックのブロック寸法

v i . 現在ブロックの現在の量子化パラメータ

10

20

30

40

50

v i i . カラー・フォーマットの指示 ( 4:2:0、4:4:4、RGB、またはYUVなど )

v i i i . コーディング構造 ( 単一ツリーまたはデュアル・ツリーなど )

i x . 色成分 ( ルーマ成分および / またはクロマ成分など )

x . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

g . 一例では、Nはデコーダに信号伝達されてもよい。

【 0 1 4 1 】

1 4 . エスケープ値についてのQpはクリップされてもよい。

a . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus}$  4と等しくてもよい。

b . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus}$  4に関係してもよい。 10

i . 一例では、エスケープ値に適用される最低のQpは、 $\text{min\_qp\_prime\_ts\_minus}$  4 + 4と等しくてもよい。

c . 一例では、各色成分についての最低のQpは、SPS / PPS / VPD / DPS / タイル / スライス・ヘッダにおいて示されてもよい。

d . 一例では、エスケープ値に適用される最低のQpは、 $(\text{bd} - \text{ibd}) * 6 + 4$ であってもよい。ここで、bdは、内部ビット深さであり、ibdは、ある色成分についての入力ビット深さを示す。

e . 一例では、上記の諸例は、ある色成分に適用されてもよい。

【 0 1 4 2 】

20

1 5 . 上記の諸例では、エスケープ値についてのクロマQpは、マッピングの前 / 後にQpを使用してもよい。

【 0 1 4 3 】

1 6 . パレット・モードで現在のパレット・インデックスを導出する際に参照インデックスを使用しないことが提案される。

a . 一例では、パレット・インデックスは、参照インデックス (たとえば、adjusted RefPaletteIndex) の可能性を排除することなく、直接的に信号伝達されてもよい。

i . あるいはまた、一例では、エンコーダは、参照インデックスが常に現在インデックスと異なることを可能にするように制約されてもよい。そのような場合、パレット・インデックスは、参照インデックスの可能性を排除することによって、信号伝達されてもよい。 30

b . 一例では、パレット・インデックスのバイナリ化は、バイナリ化入力パラメータとして最大パレット・インデックスを用いることを伴う打ち切りされたバイナリ (TB) であってもよい。

c . 一例では、パレット・インデックスのバイナリ化は固定長であってもよい。

d . 一例では、パレット・インデックスのバイナリ化は、K次のEGであってもよい。

i . 一例では、Kは整数 (たとえば、1、2または3) であってもよく、以下に依存してもよい

1 . SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ 40

2 . 内部ビット深さ

3 . 入力ビット深さ

4 . 内部ビット深さと入力深さの差

5 . 現在ブロックのブロック寸法

6 . 現在ブロックの現在の量子化パラメータ

7 . カラー・フォーマットの指示 ( 4:2:0、4:4:4、RGB、またはYUVなど )

8 . コーディング構造 ( 単一ツリーまたはデュアル・ツリーなど )

9 . 色成分 ( ルーマ成分および / またはクロマ成分など )

e . 一例では、上記の諸例は、現在のブロックが少なくとも、1つのエスケープ・サンプルを有する場合にのみ適用されうる。 50

## 【 0 1 4 4 】

17. 現在のパレット・インデックスは、以前の諸パレット・インデックスとは独立して信号伝達されてもよい。

a. 一例では、以前の諸パレット・インデックスを使用するかどうか、および/または、どのように使用するかは、現在のブロックにおいてエスケープ・サンプル（単数または複数）があるかどうかに依存してもよい。

## 【 0 1 4 5 】

18. エスケープ・シンボルについてのインデックスから非エスケープ・シンボルについてのインデックスへの導出は許可されなくてもよい。

a. 一例では、エスケープ・シンボルが適用され、パレット・インデックスがエスケープ・シンボルについてのインデックスと等しくない場合、それらのシンボルをエスケープ・シンボルとしてデコードすることは許されなくてもよい。

10

## 【 0 1 4 6 】

19. 非エスケープ・シンボルについてのインデックスからエスケープ・シンボルについてのインデックスへの導出は、許されなくてもよい。

a. 一例では、エスケープ・シンボルが適用され、パレット・インデックスがエスケープ・シンボルについてのインデックスと等しい場合、それらのシンボルを非エスケープ・シンボルとしてデコードすることは許されなくてもよい。

## 【 0 1 4 7 】

20. 導出されたパレット・インデックスは、現在のパレット・テーブル・サイズによってキャップされてもよい。

20

a. 一例では、パレット・インデックスが  $\text{MaxPaletteIndex}$  より大きい場合、 $\text{MaxPaletteIndex}$  と等しくなるように修正されてもよい。

## 【 0 1 4 8 】

21. 導出されたパレット・インデックスは、エスケープ・シンボルについてのインデックスを除き、現在のパレット・テーブル・サイズによってキャップされてもよい。

a. 一例では、エスケープ・シンボルが適用されず、パレット・インデックスが  $\text{MaxPaletteIndex}$  より大きい場合、 $\text{MaxPaletteIndex}$  と等しくなるように修正されてもよい。

b. 一例では、エスケープ・シンボルが適用され、パレット・インデックスが  $(\text{MaxPaletteIndex} - 1)$  より大きい場合、 $(\text{MaxPaletteIndex} - 1)$  と等しくなるように修正されてもよい。

30

## 【 0 1 4 9 】

22. エスケープ・シンボルを示すインデックスは、修正されることが許されなくてもよい。

a. 一例では、インデックスが  $\text{MaxPaletteIndex}$  と等しいことは、現在のブロックにエスケープ・シンボルが存在する場合、常に、エスケープ・シンボルを示してもよい。

b. 一例では、 $\text{MaxPaletteIndex}$  と等しくないインデックスは、エスケープ・シンボルを示すインデックスとしてデコードされることはできない。

## 【 0 1 5 0 】

23. 参照インデックスと現在のインデックスとの間の差を コーディングすることが提案される。

40

a. 一例では、0に等しい差は、コーディングされることが許されなくてもよい。

b. あるいはまた、パレット コーディングされたブロック内の最初のインデックスについて、インデックスは直接 コーディングされてもよい。

## 【 0 1 5 1 】

24. 参照インデックス（Rと記される）と現在のインデックス（Cと記される）との間の差のモジュロを コーディングすることが提案される。

a. 一例では、 $l = \text{Modulo}(C - R, \text{MaxPaletteIndex})$  が コーディングされてもよい。

i. 一例では、インデックスは、 $\text{Modulo}(l + R, \text{MaxPaletteIndex})$  として再構成

50

されてもよい。

i i . 一例では、0に等しいModulo(C - R, MaxPaletteIndex)は、ビットストリームにおいて許されなくてもよい。

i i i . 一例では、cMax = MaxPaletteIndexである打ち切りされたバイナリ符号が、前記値をコーディングするために使用されてもよい。

i v . あるいはまた、パレットコーディングされたブロック内の最初のインデックスについては、インデックスは直接コーディングされてもよい。

b . 一例では、 $l = \text{Modulo}(C - R, \text{MaxPaletteIndex}) - 1$ がコーディングされてもよい。

i . 一例では、インデックスは、Modulo(l + 1 + R, MaxPaletteIndex)として再構成されてもよい。 10

i i . 一例では、0より小さいModulo(C - R, MaxPaletteIndex) - 1は、ビットストリームにおいて許されなくてもよい。

i i i . 一例では、cMax = (MaxPaletteIndex - 1)である打ち切りされたバイナリ符号が、値lをコーディングするために使用されてもよい。

i v . あるいはまた、パレットコーディングされたブロック内の最初のインデックスについて、Modulo(C - R, MaxPaletteIndex)がコーディングされてもよい。

v . あるいはまた、パレットコーディングされたブロック内の最初のインデックスについては、インデックスは直接コーディングされてもよい。

【0152】

25 . パレット・ブロックのデコードの開始時には、参照インデックスRは-1に等しく設定されてもよい。

a . あるいはまた、参照インデックスRは0に等しく設定されてもよい。

【0153】

26 . パレット・モードとローカル・デュアル・ツリーを排他的に有効にすることが提案される。

a . 一例では、パレット・モードが有効にされている場合、ローカル・デュアル・ツリーは許容されなくてもよい。

i . あるいはまた、一例では、ローカル・デュアル・ツリーが有効にされている場合、パレット・モードは許容されなくてもよい。 30

b . 一例では、ローカル・デュアル・ツリーは、4:4:4のような特定のカラー・フォーマットでは有効にされない。

c . 一例では、パレット・モードは、コーディングツリーがMODE\_TYPE\_INTRAの場合には、許されなくてもよい。

d . ローカル・デュアル・ツリーの使用に基づいてパレット予測子をリセットすることが提案される。

i . 一例では、パレット予測子は、単一ツリーがローカル・デュアル・ツリーに切り換えられるときにリセットされてもよい。

i i . 一例では、パレット予測子は、ローカル・デュアル・ツリーが単一ツリーに切り換えられるときにリセットされてもよい。 40

i i i . あるいはまた、さらに、パレット予測子における諸項目の使用(たとえば、palette\_predictor\_run)を信号伝達するかどうかは、ツリー・タイプに依存してもよい。

1 . 一例では、ローカル・デュアル・ツリーと単一ツリーとの間の切り換えに出会うとき、パレット予測子における諸項目の使用(たとえば、palette\_predictor\_run)の信号伝達は省略される。

【0154】

27 . ローカル・デュアル・ツリーが適用されるとき、パレット予測テーブルにおける反復されるパレット項目を除去することが提案される。

a . 一例では、パレット予測テーブルは、ローカル・デュアル・ツリーが適用される場 50

合にリセットされてもよい。

i . あるいはまた、一例では、デコーダは、ローカル・デュアル・ツリーが適用されるときに、予測テーブル内のすべてのパレット項目をチェックし、反復される項目を除去してもよい。

ii . あるいはまた、一例では、デコーダは、ローカル・デュアル・ツリーが適用されるときに、予測テーブル内の部分的なパレット項目をチェックし、反復された項目を除去してもよい。

iii . 一例では、パレット項目をチェックする際に、フル剪定 (full pruning) または部分剪定 (partial pruning) が適用されてもよい。

1 . 一例では、選択された項目の集合がチェックされてもよい (たとえば、該集合は、パレット予測子における全部のパレット項目または部分的なパレット項目を含む)。

a ) 一例では、フル剪定または部分剪定は、選択された項目に対して適用されてもよい。

2 . 一例では、フル剪定は、1つの項目が、追加されうるすべての項目と比較されることを表してもよい。

3 . 一例では、部分剪定は、1つの項目が、追加されうる部分的な項目と比較されることを表してもよい。

iv . 一例では、2つのパレット項目が同じであるかどうかは、それらのルーマ成分値が同じであるかどうかだけにに基づいていてもよい。

1 . あるいはまた、一例では、2つのパレット項目が同じであるかどうかは、それらのクロマ成分値が同じであるかどうかのみに基づいていてもよい。

2 . あるいはまた、一例では、2つのパレット項目が同じであるかどうかは、それらのルーマとクロマ成分値の両方が同じであるかどうかに基づいていてもよい。

v . 一例では、上記の方法は、ローカル・デュアル・ツリーがルーマ成分を処理し始めるときにのみ、ルーマ・ブロックに対して適用されてもよい。

1 . あるいはまた、一例では、上記の方法は、ローカル・デュアル・ツリーがクロマ成分を処理し始めるときにのみ、クロマ・ブロックに適用されてもよい。

vi . あるいはまた、一例では、エンコーダは、項目の3つの成分が異なる場合に、2つのパレット項目を異なると考える制約条件を追加してもよい。

#### 【0155】

28 . 現在のパレット項目が、パレット予測テーブルの項目と異なる数の色成分をもつ場合、パレット予測テーブルは使用されることが許されなくてもよい。

a . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグ (reused flag) が真であるとマークされることがありうるが、現在のパレット項目が予測とは異なる数の色成分を有する場合には、現在のブロックについては使用されなくてもよい。

b . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグは、現在のパレット項目が予測とは異なる数の色成分を有する場合には、偽としてマークされてもよい。

#### 【0156】

29 . 予測テーブルと現在のパレット・テーブルが異なる色成分 (単数または複数) をもつ場合、パレット予測テーブルは使用されることが許されなくてもよい。

a . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグが真であるとマークされることがありうるが、予測テーブルと現在のパレット・テーブルが異なる色成分を有する場合には、現在のブロックについては使用されなくてもよい。

b . 一例では、パレット予測テーブル内のすべての項目についての再利用フラグは、予測テーブルと現在のパレット・テーブルが異なる色成分を有する場合には、偽としてマークされてもよい。

#### 【0157】

30 . エスケープ・シンボルは、たとえば以前に コーディング されたエスケープ・シンボ

ルに基づいて、予測的にコーディングされてもよい。

a . 一例では、ある成分のエスケープ・シンボルは、同じ色成分におけるコーディングされた値によって予測されてもよい。

i . 一例では、エスケープ・シンボルは、同じ成分における前にコーディングされたあるエスケープ・シンボルを予測子として使用してもよく、それらの間の残差が信号伝達されてもよい。

i i . 代替的に、エスケープ・シンボルは、同じ成分における前にコーディングされたK番目のエスケープ・シンボルを予測子として使用してもよく、それらの間の残差が信号伝達されてもよい。

i i i . あるいはまた、エスケープ・シンボルは、同じ成分における複数の（たとえば、K個の）コーディングされたエスケープ・シンボルから予測されてもよい。 10

1 . 一例では、Kは整数（たとえば、1、2または3）であってもよく、以下に依存してもよい

a ) SPS / VPS / PPS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / LCU行 / LCUのグループ / ブリックにおいて信号伝達されるメッセージ

b ) 内部ビット深さ

c ) 入力ビット深さ

d ) 内部ビット深さと入力深さの差

e ) 現在ブロックのブロック寸法

f ) 現在ブロックの現在の量子化パラメータ 20

g ) カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

h ) コーディング構造（単一ツリーまたはデュアル・ツリーなど）

i ) 色成分（ルーマ成分および / またはクロマ成分など）

b . 一例では、ある成分のエスケープ・シンボルは、別の成分のコーディングされた値によって予測されてもよい。

c . 一例では、ピクセルは複数の色成分を有してもよく、ピクセルがエスケープ・シンボルとして扱われる場合、ある成分の値は、他の成分のサンプルの値によって予測されてもよい。

i . 一例では、エスケープ・シンボルのU成分が、そのシンボルのV成分によって予測されてもよい。 30

d . 一例では、上記の諸方法は、ある色成分に対して（たとえば、ルーマ成分またはクロマ成分に対して）、またはコーディングされた情報などに基づくある種の条件のもとで適用されるだけであってもよい。

【 0 1 5 8 】

3 1 . パレットに関連する構文要素の信号伝達は、パレットの最大サイズ、および / またはブロック寸法、および / または無損失モードの使用および / または量子化パラメータ（QP）に依存してもよい。

a . 一例では、無損失符号ブロックの場合および / またはQPが閾値以下である場合および / または変換スキップが適用される場合、ブロックのパレット・サイズはブロック寸法と等しいと推定される。 40

i . あるいはまた、無損失符号ブロックの場合および / またはQPが閾値以下である場合、ブロックのパレット・サイズは $\min(\text{ブロック寸法}, \text{最大パレット・サイズ})$ に等しいと推定される。

b . ブロック内のエスケープ・サンプルの使用を信号伝達するかどうかは、ブロック寸法および / または無損失コーディングモードの使用（たとえば、QPが所与の値（たとえば4）に等しいか否か；および / またはtransform\_skip\_flagが1に等しいか；またはtransquant\_bypass\_flagがtrueに等しいか否か）および / またはQPに依存してもよい。

i . あるいはまた、さらに、エスケープ・サンプルの使用を信号伝達するかどうかは、現在のブロックのブロック寸法と現在のパレット・サイズとの間の関係に依存してもよ 50



い。

1. 一例では、それを信号伝達するかどうかは、ブロック寸法が現在のパレット・サイズに等しいかどうか依存してもよい。

a) あるいはまた、さらに、ブロック寸法が現在のパレット・サイズに等しい場合、それは信号伝達されず、偽であると推定される。

2. あるいはまた、それを信号伝達するかどうかは、ブロック寸法が現在のパレット・サイズより小さくないかどうか依存してもよい。

a) あるいはまた、さらに、ブロック寸法が現在のパレット・サイズよりも小さくない場合、それは信号伝達されず、偽であると推定される。

i i . あるいはまた、さらに、エスケープ・サンプルの使用を信号伝達するかどうかは、ブロック寸法、パレットの最大サイズ、および/または無損失モードの間の関係に依存してもよい。

10

1. 一例では、あるブロックが無損失モードでコーディングされ、かつブロック寸法がパレットの最大サイズより小さい場合、エスケープ・サンプルの使用の信号伝達は省略されてもよく、それは偽であると推定される。

2. 一例では、あるブロックが閾値より大きくないQPでコーディングされ、かつブロック寸法がパレットの最大サイズより小さい場合、エスケープ・サンプルの使用の信号伝達は省略されてもよく、それは偽であると推定される。

i i i . エスケープ・サンプルの使用の指示(たとえば、`palette_escape_val_present_flag`)は、ある種の条件下では推定されてもよい。

20

1. 一例では、現在のブロック・サイズが最大許容されるパレット・サイズ(たとえば、`palette_max_size`)以下である場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

a) あるいはまた、一例では、現在のブロック・サイズが最大許容されるパレット・サイズより大きい場合、エスケープ・サンプルの使用の指示が信号伝達されてもよい。

b) あるいはまた、一例では、現在のブロック・サイズが最大許容されるパレット・サイズより大きい場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

2. 一例では、上記の諸方法は、無損失コーディング条件の下で適用されうる。

30

3. 一例では、上記の諸方法は、無損失コーディングされるCUに適用されうる。

4. 一例では、現在のブロック・サイズが現在のブロックのパレット・サイズ以下である場合、エスケープ・サンプルの使用の指示は偽に推定されてもよい。

5. 一例では、エスケープ・サンプルの使用フラグが推定されるとき、対応する構文要素、たとえば`palette_escape_val_present_flag`はビットストリームにおいてスキップされてもよい。

【0159】

32. パレット・モードにおけるランレングスコーディングのためのコンテキストは、パレット項目をインデックス付けするためのパレット・インデックスに依存してもよい。

a. 一例では、デコーダでのインデックス調整プロセス(セクション2.1.3において述べた)の後のパレット・インデックスが、長さ要素のプレフィックス(たとえば、`palette_run_prefix`)のためのコンテキストを導出するために用いられてもよい。

40

b. あるいはまた、一例では、項目13で定義されたIは、長さ要素のプレフィックス(たとえば、`palette_run_prefix`)のためのコンテキストを導出するために、パレット・インデックスを置き換えてもよい。

【0160】

33. 量子化パラメータ予測子のための導出プロセスにおいて使用される左近傍ブロックおよび/または上近傍ブロックの位置を、モード/MV(たとえばMPM)導出において使用される近傍左ブロックおよび/または上近傍ブロックの位置と整列させることが提案される。

50

a . 量子化パラメータのための導出プロセスで使用される左近傍ブロックおよび / または上近傍ブロックの位置は、マージ / AMVP 候補リスト導出プロセスにおいて使用される位置と整列されてもよい。

b . 一例では、量子化パラメータのための導出プロセスにおいて使用される近傍左ブロックおよび / または上ブロックの位置は、図 8 に示される左 / 上の隣接ブロックであってもよい。

【 0 1 6 1 】

3 4 . ブロック・レベルのQP差は、エスケープ・サンプルが現在のブロックに存在するかどうかに関係なく、送信されてもよい。

a . 一例では、ブロック・レベルのQP差を送信するかどうか、および / またはどのように送信するかは、パレット以外のモードでコーディングされたブロックに従うことができる。

b . 一例では、ブロック・レベルのQP差は、あるパレット・ブロックについて常に送信されないのでもよい。

c . 一例では、ブロック幅が閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

d . 一例では、ブロック高さが閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

e . 一例では、ブロック・サイズが閾値よりも大きい場合、ブロック・レベルのQP差がパレット・ブロックについて送信されてもよい。

f . 一例では、上記の諸例はルーマ・ブロックまたはクロマ・ブロックにのみ適用される。

【 0 1 6 2 】

3 5 . パレット・ブロックについてのコーディングブロック・フラグ ( coded block flag、CBF ) ( たとえば、cbf\_luma、cbf\_cb、cbf\_cr ) のうちの一つまたは複数に設定されてもよい。

a . 一例では、パレット・ブロックについてのCBFは、常に1に等しく設定されてもよい。

b . パレット・ブロックについてのCBFのうちの一つまたは複数は、エスケープ・ピクセルが現在のブロックに存在するかどうか依存してもよい。

i . 一例では、パレット・ブロックがエスケープ・サンプルを有する場合、そのcbfを1に設定されてもよい。

i i . あるいはまた、パレット・ブロックがエスケープ・サンプルを有しない場合、そのcbfは0に設定されてもよい。

c . あるいはまた、近傍のパレットコーディングされたブロックにアクセスするとき、それは、1に等しいCBFを有するイントラコーディングされたブロックとして扱われてもよい。

【 0 1 6 3 】

3 6 . パレット・ブロックに適用されるルーマおよび / またはクロマQPと、そのブロックについて導出されるQP ( たとえば、JVET-02001-vE仕様における $Q_{p\gamma}$ または $Q_{p'\gamma}$  ) との間の差は、諸パレット・ブロックのための固定された値に等しく設定されてもよい。

a . 一例では、ルーマおよび / またはクロマQPオフセットが0に設定されてもよい。

b . 一例では、CbとCrについてのクロマQPオフセットが異なってもよい。

c . 一例では、ルーマQPオフセットとクロマQPオフセットは異なってもよい。

d . 一例では、クロマQPオフセット ( 単数または複数 ) がDPS / VPS / SPS / PPS / スライス / ブリック / タイル・ヘッダにおいて示されてもよい。

【 0 1 6 4 】

3 7 . NumPalletIdxで示される、現在のブロックについて明示的に信号伝達されるまたは推測されるパレット・インデックスの数 ( たとえば、num\_palette\_indices\_minus1

10

20

30

40

50

+ 1) は、K以上に制約されてもよい。

a . 一例では、Kは、現在のパレット・サイズ、エスケープ・フラグ、および/またはパレットコーディングされたブロックの他の情報に基づいて決定されてもよい。Sを現在のブロックの現在のパレット・サイズ、Eをエスケープ存在フラグ(たとえば、palette\_escape\_val\_present\_flag)の値とする。BlkSを現在のブロック・サイズとする。

i . 一例では、KはSに等しく設定されてもよい。

i i . あるいはまた、一例では、Kは、S + Eに等しく設定されてもよい。

i i i . あるいはまた、一例では、Kは(予測されたパレット項目の数 + 信号伝達されたパレット項目の数 + palette\_escape\_val\_present\_flag)(たとえば、NumPredictedPaletteEntries + num\_signalled\_palette\_entries + palette\_escape\_val\_present\_flag)に等しく設定されてもよい。

i v . あるいはまた、一例では、Kは、(パレット・インデックスの最大値(たとえば、MaxPaletteIndex) + 1)に等しく設定されてもよい。

v . あるいはまた、一例では、Kはデコーダに信号伝達されてもよい。

i . 一例では、Kは固定された整数値であってもよい。

i i . 一例では、Kは整数であり、以下に基づいて決定されてもよい

- 1 . 以前にコーディングされたブロック/現在のブロックのデコードされた情報
- 2 . 現在ブロック/近傍(隣接または非隣接)ブロックの量子化パラメータ
- 3 . ビデオ・コンテンツ(たとえば、スクリーン・コンテンツまたは自然コンテンツ)

4 . DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大コーディング単位(Largest coding unit、LCU) / コーディング単位(CU) / LCU行 / LCUのグループ / TU / PUブロック / ビデオコーディング単位において信号伝達されるメッセージ

5 . CU / PU / TU / ブロック / ビデオコーディング単位の位置

6 . 現在ブロックおよび/またはその近傍ブロックのブロック寸法

7 . 現在ブロックおよび/またはその近傍ブロックのブロック形状

8 . カラー・フォーマットの指示(4:2:0、4:4:4、RGB、またはYUVなど)

9 . コーディングツリー構造(デュアル・ツリーまたは単一ツリーなど)

10 . スライス / タイル・グループ・タイプおよび/またはピクチャー・タイプ

11 . 色成分(たとえば、ルーマ成分および/またはクロマ成分に対してのみ適用されうる)

12 . 時間的レイヤーID

13 . 規格のプロファイル / レベル / 層(Tier)

b . 一例では、num\_palette\_indices\_minus1の代わりに(NumPtlIdx - K)が信号伝達 / パースされてもよい。

i . あるいはまた、さらに、それは、(S + E)が1より小さくない場合にのみ、信号伝達されてもよい。

i i . 一例では、(NumPtlIdx - K)の値は、バイナリ化されたピン・ストリングがプレフィックス(たとえば、打ち切りされた単進(truncated unary))および/または第m EG符号をもつサフィックスを有していてもよいというバイナリ化方法で信号伝達されてもよい。

i i i . 一例では、(NumPtlIdx - K)の値は、打ち切りされたバイナリ・バイナリ化法(truncated binary binarization method)で信号伝達されてもよい。

i v . 一例では、(NumPtlIdx - K)の値は、打ち切りされた単進バイナリ化法(truncated unary binarization method)で信号伝達されてもよい。

v . 一例では、(NumPtlIdx - K)の値は、第m EGバイナリ化法(m-th EG binarization method)で信号伝達されてもよい。

v i . 一例では、BlkS - Kの値は、打ち切りされた単進 / 打ち切りされたバイナリ・バイナリ化方法に対する最大値として使用されるなど、上記の諸バイナリ化方法にお

る入力パラメータ（たとえば、cMax）として使用されてもよい。

- c．一例では、適合ビットストリームは、NumPitIdxがK以上であることを満たす。
- d．一例では、適合ビットストリームは、NumPitIdxがK'以下であることを満たす。
- i．一例では、K'は（ブロック幅\*ブロック高さ）に設定される。
- ii．一例では、K'は（ブロック幅\*ブロック高さ - K）に設定される。

#### 【0165】

38．上記の諸方法を適用するかどうか、および/または、どのように適用するかは、下記に基づいていてもよい：

- a．ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）
- b．DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイ  
ル・グループ・ヘッダ / 最大コーディング単位（Largest coding unit、LCU） / コ  
ーディング単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオコ  
ーディング単位において信号伝達されるメッセージ
- c．CU / PU / TU / ブロック / ビデオコーディング単位の位置
- d．現在ブロックおよび/またはその近傍ブロックのブロック寸法
- e．現在ブロックおよび/またはその近傍ブロックのブロック形状
- f．カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）
- g．コーディングツリー構造（デュアル・ツリーまたは単一ツリーなど）
- h．スライス/タイル・グループ・タイプおよび/またはピクチャー・タイプ
- i．色成分（たとえば、ルーマ成分および/またはクロマ成分に対してのみ適用されう  
る）
- j．時間的レイヤーID
- k．規格のプロファイル/レベル/層（Tier）
- l．現在のブロックが1つのエスケープ・サンプルをもつか否か
  - i．一例では、上記の諸方法は、現在のブロックが少なくとも、1つのエスケープ・  
サンプルを有する場合にのみ適用されうる。
- m．現在のブロックが無損失モードでコーディングされるか否か（たとえば、cu\_tra  
nsquant\_bypass\_flag）
  - i．一例では、上記の諸方法は、現在のブロックが無損失モードでコーディングされ  
ない場合にのみ適用されうる。
- n．無損失コーディングが有効にされているか否か（たとえば、transquant\_bypas  
s\_enabled、cu\_transquant\_bypass\_flag）
  - i．一例では、上記の諸方法は、無損失コーディングが無効にされている場合にのみ  
適用されうる。

#### 【0166】

##### ライン・ベースのCGパレット・モード関連

39．これらがエスケープ・サンプルであるかどうかは、各CGについて示されてもよい。

- a．一例では、各CGについて、構文要素、たとえばpalette\_escape\_val\_present  
\_flagが、エスケープ・サンプルが存在するか否かを示すためにビットストリームにおい  
て送られてもよい。
- i．一例では、palette\_escape\_val\_present\_flagは、CGサイズ、現在のプロ  
ック中のデコードされたサンプルの数、および/または現在のブロックのパレット・サイ  
ズに基づいて推定される、または信号伝達されるのでもよい。
- b．一例では、現在のCGについて、エスケープ・サンプルが存在しない場合、インデ  
ックス調整が適用されてもよい。
- c．一例では、現在のCGについて、エスケープ・サンプルが存在する場合、インデッ  
クス調整は適用されるべきではない。
- d．あるいはまた、上記の諸方法は、現在のブロックがエスケープ・サンプルを含む場  
合にのみ適用されうる。

#### 【0167】

10

20

30

40

50

40 . ライン・ベースのCGパレット・モードでは、上のインデックスをコピーの使用の指示（たとえばcopy\_above\_palette\_indices\_flag）は、コンテキストコーディングされなくてもよい。

e . あるいはまた、一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）は、いかなるコンテキストも使用することなくバイパスコーディングされてもよい。

i . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）および現在のセグメントにおけるコピー・フラグ（たとえば、run\_copy\_flag）は、インターリーブされて信号伝達されてもよい。

f . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）は、現在のセグメントにおけるすべてのコピー・フラグ（たとえば、run\_copy\_flag）の後にコーディングされてもよい。

g . 一例では、上のインデックスをコピーの使用の指示（たとえば、copy\_above\_palette\_indices\_flag）および信号伝達されるインデックスは、インターリーブされてコーディングされてもよい。

h . 上記の諸方法は、他のパレット・ベースのコーディングモードにも適用されうる。

【0168】

41 . コピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値は、インターリーブされて信号伝達されてもよい。

i . 一例では、第1のコピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値が順にコーディングされてもよく；続いて第2のコピー・フラグ、ラン・タイプ、上のインデックスをコピーの使用の指示、およびエスケープ値がコーディングされてもよい。

j . あるいはまた、さらに、所与のCGについて、上記の方法が適用されてもよい。

【0169】

42 . ライン・ベースのCGパレット・モードは、Thと記される所与の閾値以下のサイズのブロックについては無効にされてもよい。

k . 一例では、Thは、ライン・ベースのCGパレット・モードにおけるセグメントのサンプル数に等しい。

l . 一例では、Thは固定した数（たとえば16）であり、以下に基づいてもよい

i . ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）

ii . DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大コーディング単位（Largest coding unit、LCU） / コーディング単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオコーディング単位において信号伝達されるメッセージ

iii . CU / PU / TU / ブロック / ビデオコーディング単位の位置

iv . 現在ブロックおよび / またはその近傍ブロックのブロック寸法

v . 現在ブロックおよび / またはその近傍ブロックのブロック形状

vi . カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

vii . コーディングツリー構造（デュアル・ツリーまたは単一ツリーなど）

viii . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

ix . 色成分（たとえば、ルーマ成分および / またはクロマ成分に対してのみ適用されうる）

x . 時間的レイヤーID

xi . 規格のプロファイル / レベル / 層（Tier）

xii . 現在のブロックの量子化パラメータ

xiii . 現在のブロックが1つのエスケープ・サンプルをもつか否か

xiv . 無損失コーディングが有効にされているか否か（たとえば、transquant\_bypass\_enabled、cu\_transquant\_bypass\_flag）

10

20

30

40

50

## 【 0 1 7 0 】

BDPCM関連

4 3 . 1つのブロックがBDPCMでコーディングされ、それが複数の変換ブロックまたはサブブロックに分割される場合、残差予測はブロック・レベルで行われてもよく、残差の信号伝達はサブブロック / 変換ブロック・レベルで行われる。

a . あるいはまた、さらに、あるサブブロックの再構成は、別のサブブロックの再構成プロセスにおいては許可されない。

b . あるいはまた、残差予測および残差の信号伝達は、サブブロック / 変換ブロック・レベルで行われる。

i . このようにして、あるサブブロックの再構成は、別のサブブロックの再構成プロセスにおいて利用されうる。 10

## 【 0 1 7 1 】

クロマQPテーブル関連

4 4 . 所与のインデックスについて、joint\_cb\_crモードについてのクロマQPテーブルの値は、CbについてのクロマQPテーブルの値とCrについてのクロマQPテーブルの値の両方によって制約されうる。

c . 一例では、joint\_cb\_crモードについてのクロマQPテーブルの値の値は、CbについてのクロマQPテーブルの値とCrについてのクロマQPテーブルの値との間（両端含む）に制約されうる。

## 【 0 1 7 2 】

ブロッキング解除関連

4 5 . ブロッキング解除におけるMVの比較は、代替的な半画素補間フィルタが使用されるかどうか（たとえば、JVET-O2001-vE仕様におけるhpellIdxによって示される）に依存してもよい。

d . 一例では、異なる補間フィルタを使用するブロックは、異なるMVを有するものとして扱われてもよい。

e . 一例では、代替的な半画素補間フィルタが関与する場合、ブロッキング解除比較のためにMV差に一定のオフセットが加算されてもよい。

## 【 0 1 7 3 】

一般的なクレーム

4 6 . 上記の諸方法を適用するかどうか、および / または、どのように適用するかは下記に基づいていてもよい：

a . ビデオ・コンテンツ（たとえば、スクリーン・コンテンツまたは自然コンテンツ）

b . DPS / SPS / VPS / PPS / APS / ピクチャー・ヘッダ / スライス・ヘッダ / タイル・グループ・ヘッダ / 最大コーディング単位（Largest coding unit、LCU） / コーディング単位（CU） / LCU行 / LCUのグループ / TU / PUブロック / ビデオコーディング単位において信号伝達されるメッセージ

c . CU / PU / TU / ブロック / ビデオコーディング単位的位置

d . 現在ブロックおよび / またはその近傍ブロックのブロック寸法

e . 現在ブロックおよび / またはその近傍ブロックのブロック形状 40

f . 現在ブロックの量子化パラメータ

g . カラー・フォーマットの指示（4:2:0、4:4:4、RGB、またはYUVなど）

h . コーディングツリー構造（デュアル・ツリーまたは単一ツリーなど）

i . スライス / タイル・グループ・タイプおよび / またはピクチャー・タイプ

j . 色成分（たとえば、ルーマ成分および / またはクロマ成分に対してのみ適用される）

k . 時間的レイヤーID

l . 規格のプロファイル / レベル / 層（Tier）

m . 現在のブロックが1つのエスケープ・サンプルをもつか否か

i . 一例では、上記の諸方法は、現在のブロックが少なくとも、1つのエスケープ・ 50

サンプルを有する場合にのみ適用されうる。

n . 現在のブロックが無損失モードでコーディングされるか否か (たとえば、`cu_transquant_bypass_flag`)

i i . 一例では、上記の諸方法は、現在のブロックが無損失モードでコーディングされない場合にのみ適用されうる。

o . 無損失コーディングが有効にされているか否か (たとえば、`transquant_bypass_enabled`、`cu_transquant_bypass_flag`)

【0174】

## 5. 実施形態

実施形態は、JVET-O2001-vEに基づく。新しく追加されたテキストは、二重括弧で囲まれる。たとえば、`{{a}}`は、“a”が追加されたことを示す。削除されるテキストは、二重ブラケットで囲まれる。たとえば、`[[b]]`は、“b”が削除されることを示す。

【0175】

### 5.1 実施形態#1

#### パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (`xCb, yCb`)
- ・パレット・テーブルにおける最初の色成分を指定する変数 `startComp`
- ・現在のブロックの色成分を指定する変数 `cIdx`、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数 `nCbW` と `nCbH`

このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列 `recSamples[x][y]` であり、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$

【0176】

`cIdx`の値に依存して、変数 `nSubWidth` および `nSubHeight` は次のように導出される：

- ・`cIdx`が0に等しい場合、`nSubWidth`は1に設定され、`nSubHeight`は1に設定される。
- ・それ以外の場合、`nSubWidth`は`SubWidthC`に設定され、`nSubHeight`は`SubHeightC`に設定される。

【0177】

位置 (`xCb, yCb`) における再構成されたサンプル配列 `recSamples` の (`nCbW × nCbH`) ブロックは、`recSamples[x][y]` ( $x = 0..nCbW - 1$  および  $y = 0..nCbH - 1$ ) によって表され、0から`nCbW - 1`の範囲 (両端含む) における各 `x` および 0から`nCbH - 1`の範囲 (両端含む) における各 `y` についての `recSamples[x][y]` の値は、以下のように導出される：

- ・変数 `xL` および `yL` は次のように導出される：

$xL = \text{palette\_transpose\_flag} ? x * nSubHeight : x * nSubWidth$  (8-268)

$yL = \text{palette\_transpose\_flag} ? y * nSubWidth : y * nSubHeight$  (8-269)

- ・変数 `blsEscapeSample` は、次のように導出される：

・ `PaletteIndexMap[xCb + xL][yCb + yL]` が `MaxPaletteIndex` に等しく、`palette_escape_val_present_flag` が 1 に等しい場合、`blsEscapeSample` は 1 に等しく設定される。

- ・それ以外の場合、`blsEscapeSample` は 0 に等しく設定される。

- ・`blsEscapeSample` が 0 に等しい場合、以下が適用される：

$\text{recSamples}[x][y] = \text{CurrentPaletteEntries}[cIdx][\text{PaletteIndexMap}[xCb + xL][yCb + yL]]$  (8-270)

- ・それ以外の場合、`cu_transquant_bypass_flag` が 1 に等しい場合、以下が適用される：

$\text{recSamples}[x][y] = \text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL]$  (8-271)

- ・それ以外の場合 (`blsEscapeSample` が 1 に等しく、`cu_transquant_bypass_flag` が 0 に等しい)、次の順序付けられたステップが適用される：

10

20

30

40

50

## 【 0 1 7 8 】

1 . 量子化パラメータ $qP$ は次のように導出される :

- ・  $cIdx$ が0に等しい場合、  
 $qP = \text{Max}(0, Qp'Y)$  (8-272)
- ・ それ以外で、 $cIdx$ が1に等しい場合、  
 $qP = \text{Max}(0, Qp'Cb)$  (8-273)
- ・ それ以外の場合 ( $cIdx$ が2に等しい) 、  
 $qP = \text{Max}(0, Qp'Cr)$  (8-274)

2 . 変数 $bitDepth$ は次のように導出される :

$$bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C \quad (8-275)$$

10

3 . [[リスト $levelScale$ ]は $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$ として指定される。ここで、 $k = 0..5$ ]

4 . 下記が適用される :

$$[[tmpVal = ( PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] * levelScale[qP \% 6] ) \quad (qP / 6) + 32 ) \quad 6 \quad (8-276)]]$$

{{ $T$ は、成分 $cIdx$ について ( $internal\_bit\_depth - input\_bit\_depth$ ) [内部ビット深さ - 入力ビット深さ] に等しく設定される。

$$Nbits = \text{max}(T, (qP - 4) / 6)$$

- ・  $Nbits$ が $T$ に等しい場合、

$$recSamples[x][y] = PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] \quad Nbits \quad 20$$

- ・ それ以外の場合、

$$recSamples[x][y] = ( PaletteEscapeVal[cIdx][xCb + xL][yCb + yL] \quad Nbits ) + ( 1 \quad (Nbits - 1) )$$

$$[[recSamples[x][y] = Clip3( 0, (1 \quad bitDepth) - 1, tmpVal ) \quad (8-277)]]$$

## 【 0 1 7 9 】

以下の条件 :

- ・  $cIdx$ が0に等しく、 $numComps$ が1に等しい ;
- ・  $cIdx$ が2に等しい

のいずれかが真である場合、

変数 $PredictorPaletteSize[startComp]$ と配列 $PredictorPaletteEntries$ は次のように導出または修正される :

30

40

50



## 【表 9】

```

for( i = 0; i < CurrentPaletteSize[ startComp ]; i++ )
  for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ )
    newPredictorPaletteEntries[ cIdx ][ i ] = CurrentPaletteEntries[ cIdx ][ i ]
newPredictorPaletteSize = CurrentPaletteSize[ startComp ]
for( i = 0; i < PredictorPaletteSize && newPredictorPaletteSize <
PaletteMaxPredictorSize; i++ )
  if( !PalettePredictorEntryReuseFlags[ i ] ) {
    for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++ )
      newPredictorPaletteEntries[ cIdx ][ newPredictorPaletteSize ] =
        PredictorPaletteEntries[ cIdx ][ i ]
    newPredictorPaletteSize++
  }
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )
  PredictorPaletteEntries[ cIdx ][ i ] = newPredictorPaletteEntries[ cIdx ][ i ]
PredictorPaletteSize[ startComp ] = newPredictorPaletteSize

```

PredictorPaletteSize[startComp]の値が0からPaletteMaxPredictorSizeまでの範囲（両端含む）であることはビットストリーム適合性の要件である。

## 【 0 1 8 0 】

## 5.2 実施形態#2

この実施形態は、パレット・インデックス導出を記載する。  
パレットコーディングのセマンティクス  
 [[変数adjustedRefPaletteIndex]は次のように導出される：

40

50

## 【表 1 0】

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev =
x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev =
y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```

if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++

```

## 【 0 1 8 1】

palette\_idx\_idcについてのバイナリ化プロセス

このプロセスへの入力、構文要素palette\_idx\_idcについてのバイナリ化の要求と変数MaxPaletteIndexである。

このプロセスの出力は、前記構文要素のバイナリ化である。

変数cMaxは次のように導出される：

- ・ [[このプロセスが現在のブロックについて初めて呼び出される場合、]]cMaxはMaxPaletteIndexに等しく設定される。
- ・ [[それ以外の場合（このプロセスが現在のブロックについて初めて呼び出されたのではない）、cMaxはMaxPaletteIndexから1を引いたものに等しく設定される。]]

palette\_idx\_idcについてのバイナリ化は、cMaxで節9.3.3.4で規定されているTBバイナリ化プロセスを呼び出すことによって導出される。

## 【 0 1 8 2】

5.3 実施形態#3

【表 1 1】

テーブル 9-77 – 構文要素および関連するバイナリ化

構文要素	入力パラメータ	
	プロセス	
palette_escape_val	[[EG3]] {{FL}}	{{bitDepth - (Max(QpPrimeTsMin, Qp'Y) - 4)/6}}

10

## 【 0 1 8 3 】

## 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力には以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (xCb, yCb)
- ・パレット・テーブルにおける最初の色成分を指定する変数 startComp
- ・現在のブロックの色成分を指定する変数 cIdx、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数 nCbW と nCbH

20

このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列 recSamples[x][y] であり、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$

## 【 0 1 8 4 】

cIdx の値に依存して、変数 nSubWidth および nSubHeight は次のように導出される：

....

- ・それ以外の場合 (blsEscapeSample が 1 に等しく、cu\_transquant\_bypass\_flag が 0 に等しい)、次の順序付けられたステップが適用される：

## 【 0 1 8 5 】

5 . 量子化パラメータ qP は次のように導出される：

- ・ cIdx が 0 に等しい場合、
- $qP = \text{Max}(0, Qp'Y)$  (8-272)
- ・ それ以外で、cIdx が 1 に等しい場合、
- $qP = \text{Max}(0, Qp'Cb)$  (8-273)
- ・ それ以外の場合 (cIdx が 2 に等しい)、
- $qP = \text{Max}(0, Qp'Cr)$  (8-274)

30

6 . 変数 bitDepth は次のように導出される：

$$\text{bitDepth} = (\text{cIdx} == 0) ? \text{BitDepth}_Y : \text{BitDepth}_C \quad (8-275)$$

3 . [[リスト levelScale[] は levelScale[k] = {40, 45, 51, 57, 64, 72} として指定される。ここで、 $k = 0..5$ ]]

4 . 下記が適用される：

40

$$[[\text{tmpVal} = (\text{PaletteEscapeVal}[\text{cIdx}][\text{xCb} + \text{xL}][\text{yCb} + \text{yL}] * \text{levelScale}[\text{qP} \% 6]) \quad (\text{qP} / 6) + 32) \quad 6 \quad (8-276)]]$$

$$\{\{\text{shift} = (\text{max}(\text{QpPrimeTsMin}, \text{qP}) - 4) / 6$$

$$\text{tmpVal} = (\text{PaletteEscapeVal}[\text{cIdx}][\text{xCb} + \text{xL}][\text{yCb} + \text{yL}] \quad \text{shift}) \}\}$$

$$\text{recSamples}[\text{x}][\text{y}] = \text{Clip3}(0, (1 \quad \text{bitDepth}) - 1, \text{tmpVal}) \quad (8-277)$$

## 【 0 1 8 6 】

## 5.4 実施形態 #4

copy\_above\_palette\_indices\_flag が 1 に等しいことは、パレット・インデックスが、水平横断走査が使用される場合は上の行における同じ位置、または垂直横断走査が使用される場合は左の列における同じ位置にあるパレット・インデックスに等しいことを指

50

定する。copy\_above\_palette\_indices\_flagが0に等しいことは、サンプルのパレット・インデックスの指示がビットストリームにおいてコーディングされるか、または推定されることを指定する。

… …

変数adjustedRefPaletteIndexは次のように導出される：

【表 1 2】

```
adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 {{&& !palette_escape_val_present_flag}}) {
    xcPrev = x0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 0 ]
    ycPrev = y0 + TraverseScanOrder[ log2CbWidth ][ log2bHeight ][ PaletteScanPos - 1 ][ 1 ]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {
        adjustedRefPaletteIndex = PaletteIndexMap[ xcPrev ][ ycPrev ] { (7-157)
    }
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}
```

CopyAboveIndicesFlag[xC][yC]が0に等しい場合、変数CurrPaletteIndexは次のように導出される：

```
if ( CurrPaletteIndex == adjustedRefPaletteIndex )
    CurrPaletteIndex++ (7-158)
【 0 1 8 7 】
```

## 5.5 実施形態#5

【表 1 3】

テーブル 9-77 – 構文要素およびその関連するバイナリ化

構文要素	入力パラメータ	
	プロセス	
palette_escape_val	[[EG3]] {{FL}}	$\{\{\max(1, \text{bitDepth} - (\text{Max}(\text{QpPrimeTsMin}, \text{Qp}^Y) - 4)/6)\}\}$

### 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のブロックの左上のルーマ・サンプルを指定する位置 (xCb, yCb)
- ・パレット・テーブルにおける最初の色成分を指定する変数startComp
- ・現在のブロックの色成分を指定する変数cIdx、
- ・現在のブロックの幅と高さをそれぞれ指定する2つの変数nCbWとnCbH

このプロセスの出力は、ブロックについての再構成されたサンプル値を指定する配列  $recSamples[x][y]$  であり、 $x = 0..nCbw - 1$ 、 $y = 0..nCbh - 1$

【 0 1 8 8 】

$cIdx$ の値に依存して、変数 $nSubWidth$ および $nSubHeight$ は次のように導出される：

....

・それ以外の場合 ( $blsEscapeSample$ が1に等しく、 $cu\_transquant\_bypass\_flag$ が0に等しい)、次の順序付けられたステップが適用される：

【 0 1 8 9 】

9 . 量子化パラメータ $qP$ は次のように導出される：

・  $cIdx$ が0に等しい場合、

$$qP = \text{Max}(0, Qp'Y) \quad (8-272)$$

・ それ以外で、 $cIdx$ が1に等しい場合、

$$qP = \text{Max}(0, Qp'Cb) \quad (8-273)$$

・ それ以外の場合 ( $cIdx$ が2に等しい)、

$$qP = \text{Max}(0, Qp'Cr) \quad (8-274)$$

10 . 変数 $bitDepth$ は次のように導出される：

$$bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C \quad (8-275)$$

11 . [[リスト $levelScale[]$ は $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$ として指定される。ここで、 $k = 0..5$ ]]

12 . 下記が適用される：

[[ $tmpVal = (\text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL] * levelScale[qP \% 6]) / ((qP / 6) + 32) \quad (8-276)$ ]]

[[ $shift = \min(bitDepth - 1, (\max(QpPrimeTsMin, qP) - 4) / 6)$ ]]

$tmpVal = (\text{PaletteEscapeVal}[cIdx][xCb + xL][yCb + yL] \ll shift)$ ]]

$recSamples[x][y] = \text{Clip3}(0, (1 \ll bitDepth) - 1, tmpVal) \quad (8-277)$

【 0 1 9 0 】

## 5.6 実施形態#6

この実施形態は、変換スキップのための変換シフトをスキップするための設計を示しており、JVET-O2001-vEに基づいている。

【 0 1 9 1 】

### 8.7.2 スケーリングおよび変換プロセス

このプロセスへの入力には以下の通りである：

・現在のピクチャーの左上のルーマ・サンプルに対して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位置 ( $xTbY, yTbY$ )

・現在のブロックの色成分を指定する変数 $cIdx$

・変換ブロック幅を指定する変数 $nTbW$

・変換ブロック高さを指定する変数 $nTbH$ 。

このプロセスの出力は、 $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプルの  $(nTbW) \times (nTbH)$  配列  $resSamples[x][y]$  である。

変数 $bitDepth$ 、 $bdShift$ および $tsShift$ は、次のように導出される：

$$bitDepth = (cIdx == 0) ? BitDepth_Y : BitDepth_C \quad (8-942)$$

$$bdShift = \text{Max}(20 - bitDepth, 0) \quad (8-943)$$

[[ $tsShift = 5 + ((\text{Log2}(nTbW) + \text{Log2}(nTbH)) / 2) \quad (8-944)$ ]]

変数 $codedCIdx$ は次のように導出される：

・ $cIdx$ が0に等しい、または $TuCResMode[xTbY][yTbY]$ が0に等しい場合、 $codedCIdx$ は $cIdx$ に等しく設定される。

・それ以外で、 $TuCResMode[xTbY][yTbY]$ が1または2に等しい場合、 $codedCIdx$ は1に等しく設定される。

・それ以外の場合、 $codedCIdx$ は2に等しく設定される。

変数 $cSign$ は  $(1 - 2 * slice\_joint\_cbcr\_sign\_flag)$  に等しく設定される。

10

20

30

40

50

残差サンプルの $(nTbW) \times (nTbH)$ 配列resSampleは以下のように導出される：

【 0 1 9 2 】

1. 節8.7.3に規定される変換係数についてのスケーリング・プロセスは、変換ブロック位置 $(xTbY, yTbY)$ 、変換ブロック幅 $nTbW$ 、変換ブロック高さ $nTbH$ 、codedCIdxに等しく設定された色成分変数cIdx、および現在の色成分ビット深さbitDepthを入力として呼び出され、出力はスケーリングされた変換係数 $d$ の $(nTbW) \times (nTbH)$ 配列である。

2. 残差サンプル $r$ の $(nTbW) \times (nTbH)$ 配列は以下のように導出される：

・ [[transform\_skip\_flag[xTbY][yTbY]が1に等しく、cIdxが0に等しい場合、 $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプル配列値 $r[x][y]$ は次のように導出される：

10

$r[x][y] = d[x][y] \quad \text{tsShift} \quad (8-945)$

・ [[それ以外の場合 (transform\_skip\_flag[xTbY][yTbY]が0に等しい、またはおよびcIdxが0に等しくない場合)]節8.7.4.1で規定されたスケーリングされた変換係数のための変換プロセスが、変換ブロック位置 $(xTbY, yTbY)$ 、変換ブロック幅 $nTbW$ および変換ブロック高さ $nTbH$ 、色成分変数cIdxならびにスケーリングされた変換係数 $d$ の $(nTbW) \times (nTbH)$ 配列を入力として呼び出され、出力は、残差サンプル $r$ の $(nTbW) \times (nTbH)$ 配列である。

3.  $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、中間残差サンプル $res[x][y]$ は、次のように導出される：

20

・ {{transform\_skip\_flag[xTbY][yTbY]が1に等しく、かつcIdxが0に等しい場合、次が適用される：

$res[x][y] = d[x][y]$

・ {{それ以外の場合 (transform\_skip\_flag[xTbY][yTbY]が0に等しい、またはおよびcIdxが0に等しくない)、次が適用される：}}

$res[x][y] = (r[x][y] + (1 \quad (bdShift - 1))) \quad bdShift \quad (8-946)$

4.  $x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$ として、残差サンプル $resSamples[x][y]$ は次のように導出される：

・ cIdxがcodedCIdxに等しい場合、次が適用される：

$resSamples[x][y] = res[x][y] \quad (8-947)$

30

・ それ以外で、TuCResMode[xTbY][yTbY]が2に等しい場合、次が適用される：

$resSamples[x][y] = cSign * res[x][y] \quad (8-948)$

・ それ以外の場合、次が適用される：

$resSamples[x][y] = (cSign * res[x][y]) \quad 1$

【 0 1 9 3 】

### 8.7.3 変換係数のためのスケーリング・プロセス

....

変数rectNonTsFlagは次のように導出される：

$rect[[NonTs]]Flag = ((\text{Log2}(nTbW) + \text{Log2}(nTbH)) \& 1) == 1 \quad [ \& \& ]$

(8-955)

40

$[[transform\_skip\_flag[xTbY][yTbY]=]] = 0$

変数bdShift、rectNormおよびbdOffsetは次のように導出される：

・ {{transform\_skip\_flag[xTbY][yTbY]が1に等しく、かつcIdxが0に等しい場合、次が適用される：

$bdShift = 10$

・ {{それ以外の場合、次が適用される：}}

$bdShift = bitDepth + (\text{rect}[[NonTs]]Flag \quad ? \quad 1 \quad : \quad 0) +$

$(\text{Log2}(nTbW) + \text{Log2}(nTbH)) / 2) - 5 + dep\_quant\_enabled\_flag \quad (8-956)$

$dOffset = (1 \quad bdShift) \quad 1 \quad (8-957)$

リストlevelScale[][]は、levelScale[j][k] = {{40,45,51,57,64,72},{57,64,72,

50

80, 90, 102}として指定される。ここで、 $j = 0..1$ 、 $k = 0..5$ 。

$(nTbW) \times (nTbH)$  配列  $dz$  は、 $(nTbW) \times (nTbH)$  配列  $TransCoeffLevel[xTbY][yTbY][cIdx]$  に等しく設定される。

$x = 0..nTbW - 1$ 、 $y = 0..nTbH - 1$  として、スケーリングされた変換係数  $d[x][y]$  の導出については、以下が適用される：

- ・ 中間スケーリング因子  $m[x][y]$  は以下のように導出される：

- ・ 次の条件のうちの一つまたは複数が真である場合、 $m[x][y]$  は 16 に等しく設定される：

- ・  $sps\_scaling\_list\_enabled\_flag$  が 0 に等しい。

- ・  $transform\_skip\_flag[xTbY][yTbY]$  が 1 に等しい。

10

- ・ それ以外の場合、次が適用される：

$$m[x][y] = ScalingFactor[Log2(nTbW)][Log2(nTbH)][matrixId][x][y]$$

ここで、 $matrixId$  はテーブル 7 において指定される (8-958)

- ・ スケーリング因子  $ls[x][y]$  は次のように導出される：

- ・  $dep\_quant\_enabled\_flag$  が 1 に等しい場合、次が適用される：

$$ls[x][y] = (m[x][y] * levelScale[rect[[NonTs]]Flag][(qP + 1) \% 6]) \quad ((qP + 1) / 6) \quad (8-960)$$

- ・ それ以外の場合 ( $dep\_quant\_enabled\_flag$  が 0 に等しい)、次が適用される：

$$ls[x][y] = (m[x][y] * levelScale[rect[[NonTs]]Flag][qP \% 6]) \quad (qP / 6) \quad (8-961)$$

20

- ・  $BdpcmFlag[xTbY][yYbY]$  が 1 に等しい場合、 $dz[x][y]$  は次のように修正される：

- ・  $BdpcmDir[xTbY][yYbY]$  が 0 に等しく、 $x$  が 0 より大きい場合、次が適用される：

$$dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x - 1][y] + dz[x][y]) \quad (8-961)$$

- ・ それ以外で、 $BdpcmDir[xTbY][yYbY]$  が 1 に等しく、 $y$  が 0 より大きい場合、次が適用される：

$$dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x][y - 1] + dz[x][y]) \quad (8-962)$$

- ・ 値  $dnc[x][y]$  は次のように導出される：

$$dnc[x][y] = (dz[x][y] * ls[x][y] + bdOffset) \quad bdShift \quad (8-963)$$

- ・ 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る：

$$d[x][y] = Clip3(CoeffMin, CoeffMax, dnc[x][y]) \quad (8-964)$$

30

【 0 1 9 4 】

### 5.7 実施形態 #7

この実施形態は、パレット・インデックスの数を信号伝達するための設計を示す。

【 0 1 9 5 】

#### 7.3.8.6 パレットコーディング構文

40

50

【表 1 4】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子	
...		
if( MaxPaletteIndex > 0 ) {		
<b>num_palette_indices</b> {_{diff}}[_minus1]	ae(v)	
adjust = 0		
for( i = 0; i <= [[num_palette_indices_minus1]][NumPaletteIndices]; i++ ) {		10
if( MaxPaletteIndex - adjust > 0 ) {		
<b>palette_idx_idc</b>	ae(v)	
PaletteIndexIdc[ i ] = palette_idx_idc		
}		
adjust = 1		
}		
<b>copy_above_indices_for_final_run_flag</b>	ae(v)	
<b>palette_transpose_flag</b>	ae(v)	20
}		
if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag ) {		
if( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {		
<b>cu_qp_delta_abs</b>	ae(v)	
if( cu_qp_delta_abs )		
<b>cu_qp_delta_sign_flag</b>	ae(v)	
}		30
}		
if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag ) {		
if( cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {		
<b>cu_chroma_qp_offset_flag</b>	ae(v)	
if( cu_chroma_qp_offset_flag )		
<b>cu_chroma_qp_offset_idx</b>	ae(v)	
}		40



}	
remainingNumIndices = [[num_palette_indices_minus1 + 1]][NumPaletteIndices])	
...	
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
currNumIndices = [[num_palette_indices_minus1 + 1]][NumPaletteIndices] - remainingNumIndices	10
PaletteIndexMap[ xC ][ yC ] = PaletteIndexIdc[ currNumIndices ]	
}	
...	
}	

num\_palette\_indices{{\_diff}}[\_minus1]に[[1]] ( {{MaxPaletteIndex + 1}} )を加えたものは、現在のブロックについて明示的に信号伝達されるまたは推定されるパレット・インデックスの数である。

{{NumPaletteIndicesは ( num\_palette\_indices\_diff + MaxPaletteIndex + 1 ) に設定される。}}

num\_palette\_indices{{\_diff}}[\_minus1]が存在しない場合は、0に等しいと推定される。

{{num\_palette\_indices\_diffの値は、0からcbWidth\*cbHeight - ( MaxPaletteIndex + 1 ) ( 両端含む ) の範囲とする。}}

copy\_above\_indices\_for\_final\_run\_flagが1に等しいことは、コーディング単位における最後の諸位置のパレット・インデックスが、水平横断走査が使用される場合は上の行におけるパレット・インデックスから、垂直横断走査が使用される場合は左の列におけるパレット・インデックスからコピーされることを指定する。

copy\_above\_indices\_for\_final\_run\_flagが0に等しいことは、コーディング単位における最後の諸位置のパレット・インデックスがPaletteIndexIdc[ [[num\_palette\_indices\_minus1]][NumPaletteIndices - 1] ]からコピーされることを指定する。

【 0 1 9 6 】

### 9.5.3.13 num\_palette\_indices{{\_diff}}[\_minus1]についてのバイナリ化プロセス

このプロセスへの入力は、構文要素num\_palette\_indices{{\_diff}}[\_minus1]についてのバイナリ化の要求およびMaxPaletteIndexである。

このプロセスの出力は、前記構文要素のバイナリ化である。

変数cRiceParamは次のように導出される：

$$cRiceParam = 3 + ( ( MaxPaletteIndex + 1 ) - 3 ) \quad (9-26)$$

変数cMaxは、cRiceParamから次のように導出される：

$$cMax = 4 - cRiceParam \quad (9-27)$$

構文要素num\_palette\_indices{{\_diff}}[\_minus1]のバイナリ化は、プレフィックスのピン・ストリングと ( 存在する場合は ) サフィックスのピン・ストリングの連結である。

プレフィックス・ピン・ストリングの導出のためには、次が適用される：

・ num\_palette\_indices{{\_diff}}[\_minus1]のプレフィックス値prefixValは次

10

20

30

40

50

のように導出される：

$$\text{prefixVal} = \text{Min}(\text{cMax}, \text{num\_palette\_indices}\{\{\_diff\}\}[\[_minus1\]]) \quad (9-28)$$

・プレフィックス・ビン・ストリングは、prefixValについて節9.3.3.3で規定されるTRバイナリ化プロセスを、変数cMaxとcRiceParamを入力として呼び出すことによって指定される。

プレフィックス・ビン・ストリングが長さ4ですべてのビットが1に等しいビット・ストリングに等しい場合、サフィックス・ビン・ストリングが存在し、それは次のように導出される：

・num\_palette\_indices{\{\\_diff\}\}[\[\_minus1\]}のサフィックス値suffixValは次のように導出される：

$$\text{suffixVal} = \text{num\_palette\_indices}\{\{\_diff\}\}[\[_minus1\]} - \text{cMax} \quad (9-29)$$

・サフィックス・ビン・ストリングは、指数ゴロム次数kをcRiceParam + 1に等しく設定して、suffixValのバイナリ化について節9.3.3.5で規定されたk次のEGkバイナリ化プロセスを呼び出すことによって指定される。

【表15】

テーブル9-77 – 構文要素および関連するバイナリ化

構文構造	構文要素	バイナリ化	
		プロセス	入力パラメータ
palette_coding()	num_palette_indices{\{\_diff\}\}[\[_minus1\]}	9.5.3.13	MaxPaletteIndex

テーブル9-82 – ctxInc のコンテキスト符号化されたビンを用いた構文要素への割り当て

構文要素	binIdx					
	0	1	2	3	4	>= 5
num_palette_indices{\{\_diff\}\}[\[_minus1\]}	バイパス	バイパス	バイパス	バイパス	バイパス	バイパス

【0197】

### 5.8 実施形態#8

この実施形態は、ライン・ベースのCGパレット・モードにおけるインターリーブされた信号伝達の設計を示す。本実施形態は、JVET-P2001-v4で提供されるドラフトに基づく。

【表 1 6】

palette_coding( x0, y0, cbWidth, cbHeight, startComp, numComps ) {	記述子
palettePredictionFinished = 0	
...	
PreviousRunPosition = 0	
PreviousRunType = 0	
for( subSetId = 0; subSetId <= ( cbWidth * cbHeight - 1 ) / 16; subSetId++ ) {	10
minSubPos = subSetId * 16	
if( minSubPos + 16 > cbWidth * cbHeight)	
maxSubPos = cbWidth * cbHeight	
else	
maxSubPos = minSubPos + 16	
RunCopyMap[ x0 ][ y0 ] = 0	
PaletteScanPos = minSubPos	20
log2CbWidth = Log2( cbWidth )	
log2CbHeight = Log2( cbHeight )	
while( PaletteScanPos < maxSubPos ) {	
x <sub>C</sub> = x <sub>0</sub> + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
y <sub>C</sub> = y <sub>0</sub> + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	
if( PaletteScanPos > 0 ) {	
x <sub>C</sub> Prev = x <sub>0</sub> +	30
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]	
y <sub>C</sub> Prev = y <sub>0</sub> +	
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]	
}	
if( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {	
<b>run_copy_flag</b>	ac(v)
	40

RunCopyMap[ xC ][ yC ] = run_copy_flag	
}	
CopyAboveIndicesFlag[ xC ][ yC ] = 0	
if( MaxPaletteIndex > 0 && !RunCopyMap[ xC ][ yC ] ) {	
if( ( ( !palette_transpose_flag && yC > 0 )    ( palette_transpose_flag && xC > 0 ) ) && CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 ) {	10
<b>copy_above_palette_indices_flag</b>	ae(v)
CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag	
}	
PreviousRunType = CopyAboveIndicesFlag[ xC ][ yC ]	
PreviousRunPosition = curPos	
} else {	
CopyAboveIndicesFlag[ xC ][ yC ] = CopyAboveIndicesFlag[ xcPrev ][ ycPrev ]	20
}	
{{ if( MaxPaletteIndex > 0 ) {	
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
if( MaxPaletteIndex - adjust > 0 ) {	
palette_idx_idc	{{ae(v)}}}
}	
adjust = 1	
}	30
}	
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
CurrPaletteIndex = palette_idx_idc	
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex	
} else {	
if( !palette_transpose_flag )	40

10

20

30

40

50

PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ]		
else		
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ]		
}		
}		
if( palette_escape_val_present_flag ) {		
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++){		10
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]		
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {		
<b>palette_escape_val</b>		ae(v)
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val		
}		
}		
}}		20
PaletteScanPos ++		
}		
[[PaletteScanPos = minSubPos		
while( PaletteScanPos < maxSubPos ) {		
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]		
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]		
if( PaletteScanPos > 0 ) {		30
xcPrev	=x0 +	
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 0 ]		
ycPrev	=	y0 +
TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos - 1 ][ 1 ]		
}		
if( MaxPaletteIndex > 0 ) {		40

if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
if( MaxPaletteIndex - adjust > 0 ) {	
<b>palette_idx_idc</b>	[[ac(v)]]
}	
adjust = 1	
}	
}	10
if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
CurrPaletteIndex = palette_idx_idc	
if( CopyAboveIndicesFlag[ xC ][ yC ] == 0 ) {	
PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex	
} else {	
if( !palette_transpose_flag )	
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ]	20
else	
PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ]	
}	
}	
if( palette_escape_val_present_flag ) {	
for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++ )	
for( sPos = minSubPos; sPos < maxSubPos; sPos++ ) {	
xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]	30
yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]	
if( PaletteIndexMap[ cIdx ][ xC ][ yC ] == MaxPaletteIndex ) {	
<b>palette_escape_val</b>	[[ac(v)]]
PaletteEscapeVal[ cIdx ][ xC ][ yC ] = palette_escape_val	
}	
}	
}	
}	40

}]]	
}	

【 0 1 9 8 】

## 5.9 実施形態#9

変更はJVET-P2001-vEに基づく。

【 0 1 9 9 】

### 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・現在のピクチャーの左上のルーマ・サンプルに対して現在のコーディングブロックの左上のサンプルを指定する位置 ( $x_{CbComp}, y_{CbComp}$ )
- ・単一ツリーが使用されるかデュアル・ツリーが使用されるかを指定する変数  $treeType$ 。  
デュアル・ツリーが使用される場合、これは、現在のツリーがルーマ成分に対応するかクロマ成分に対応するかを指定する。
- ・現在のブロックの色成分を指定する変数  $cIdx$ 、
- ・現在のコーディングブロックの幅および高さをそれぞれ指定する2つの変数  $nCbW$  および  $nCbH$

10

このプロセスの出力は、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$ として、ブロックについての再構成されたサンプル値を指定する配列  $recSamples[x][y]$  である。

#### 【0200】

$treeType$ の値に依存して、変数  $startComp$  および  $numComps$  は次のように導出される：

- ・  $treeType$  が SINGLE\_TREE に等しい場合：
  - $startComp = 0$  (444)
  - $numComps = 3$  (445)
- ・ それ以外の場合、 $treeType$  は DUAL\_TREE\_LUMA に等しい：
  - $startComp = 0$  (446)
  - $numComps = 1$  (447)
- ・ それ以外の場合、 $treeType$  は DUAL\_TREE\_CHROMA に等しい：
  - $startComp = 1$  (448)
  - $numComps = 2$  (449)

20

#### 【0201】

$cIdx$ の値に依存して、変数  $nSubWidth$  および  $nSubHeight$  は次のように導出される：

・  $cIdx$  が 0 より大きく、 $startComp$  が 0 に等しい場合、 $nSubWidth$  は  $SubWidthC$  に設定され、 $nSubHeight$  は  $SubHeightC$  に設定される。

・ それ以外の場合は、 $nSubWidth$  は 1 に設定され、 $nSubHeight$  は 1 に設定される。

位置 ( $x_{CbComp}, y_{CbComp}$ ) における再構成されたサンプル配列  $recSamples$  の ( $nCbW \times nCbH$ ) ブロックは、 $recSamples[x][y]$  (ここで、 $x = 0..nCbW - 1$  および  $y = 0..nCbH - 1$ ) によって表され、0 から  $nCbW - 1$  の範囲 (両端含む) における各  $x$  および 0 から  $nCbH - 1$  の範囲 (両端含む) における各  $y$  についての  $recSamples[x][y]$  の値は、次のように導出される：

30

・ 変数  $xL$ 、 $yL$ 、 $xCbL$ 、 $yCbL$  は次のように導出される：

$$xL = x * nSubWidth \quad (450)$$

$$yL = y * nSubHeight \quad (451)$$

$$xCbL = x_{CbComp} * nSubWidth \quad (452)$$

$$yCbL = y_{CbComp} * nSubHeight \quad (453)$$

・ 変数  $bIsEscapeSample$  は、次のように導出される：

40

・  $PaletteIndexMap[xCbL + xL][yCbL + yL]$  が  $MaxPaletteIndex$  に等しく、 $palette\_escape\_val\_present\_flag$  が 1 に等しい場合、 $bIsEscapeSample$  は 1 に等しく設定される。

#### 【0202】

・ それ以外の場合、 $bIsEscapeSample$  は 0 に設定される。

・  $bIsEscapeSample$  が 0 に等しい場合、以下が適用される：

$recSamples[x][y] = CurrentPaletteEntries[cIdx][PaletteIndexMap[xCbL + xL][yCbL + yL]]$  (454)

・ それ以外の場合 ( $bIsEscapeSample$  が 1 に等しい)、以下の順序付けられたステップが適用される：

50

1. 量子化パラメータ $qP$ は次のように導出される：

- ・  $cldx$ が0に等しい場合、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Y)$  (455)

- ・ それ以外で、 $cldx$ が1に等しい場合、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Cb)$  (456)

- ・ それ以外の場合 ( $cldx$ は2に等しい)、  
 $qP = \text{Max}(QpPrimeTsMin, Qp'Cr)$  (457)

2. リスト $levelScale[]$ は、 $k = 0..5$ として、 $levelScale[k] = \{40, 45, 51, 57, 64, 72\}$ として指定される。

3. 次が適用される：

```

{{shift = Min(bitDepth - 1, (QpPrimeTsMin - 4) / 6)}}
[[tmpVal = (PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] *
  levelScale[qP%6]) (qP/6) + 32) 6 (458)]]
{{tmpVal = ((PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] shift) *
  levelScale[(qP - QpPrimeTsMin + 4) % 6]) ((qP - QpPrimeTsMin +
  4) / 6) + 32) 6 (458)}}
recSamples[x][y] = Clip3(0, (1 - BitDepth) - 1, tmpVal) (459)

```

【0203】

#### 5.10 実施形態#10

変更はJVET-P2001-vEに基づく。

【0204】

#### 8.4.5.3 パレット・モードについてのデコード・プロセス

このプロセスへの入力は以下の通り：

- ・ 現在のピクチャーの左上のルーマ・サンプルに対して現在のコーディングブロックの左上のサンプルを指定する位置 ( $xCbComp, yCbComp$ )

- ・ 単一ツリーが使用されるかデュアル・ツリーが使用されるかを指定する変数 $treeType$ 。デュアル・ツリーが使用される場合、これは、現在のツリーがルーマ成分に対応するかクロマ成分に対応するかを指定する。

- ・ 現在のブロックの色成分を指定する変数 $cldx$ 、

- ・ 現在のコーディングブロックの幅および高さをそれぞれ指定する2つの変数 $nCbW$ および $nCbH$

このプロセスの出力は、 $x = 0..nCbW - 1$ 、 $y = 0..nCbH - 1$ として、ブロックについての再構成されたサンプル値を指定する配列 $recSamples[x][y]$ である。

【0205】

$treeType$ の値に依存して、変数 $startComp$ および $numComps$ は次のように導出される：

- ・  $treeType$ が $SINGLE\_TREE$ に等しい場合：

$startComp = 0$  (444)

$numComps = 3$  (445)

- ・ それ以外の場合、 $treeType$ は $DUAL\_TREE\_LUMA$ に等しい：

$startComp = 0$  (446)

$numComps = 1$  (447)

- ・ それ以外の場合、 $treeType$ は $DUAL\_TREE\_CHROMA$ に等しい：

$startComp = 1$  (448)

$numComps = 2$  (449)

【0206】

$cldx$ の値に依存して、変数 $nSubWidth$ および $nSubHeight$ は次のように導出される：

- ・  $cldx$ が0より大きく、 $startComp$ が0に等しい場合、 $nSubWidth$ は $SubWidthC$ に設定され、 $nSubHeight$ は $SubHeightC$ に設定される。

- ・ それ以外の場合、 $nSubWidth$ は1に設定され、 $nSubHeight$ は1に設定される。

10

20

30

40

50



位置 (xCbComp, yCbComp) における再構成されたサンプル配列 recSamples の (nCbw × nCbH) ブロックは、recSamples[x][y] (ここで、x = 0..nCtbW - 1 および y = 0..nCbH - 1) によって表され、0 から nCtbW - 1 の範囲 (両端含む) における各 x および 0 から nCbH - 1 の範囲 (両端含む) における各 y についての recSamples[x][y] の値は、次のように導出される：

- ・変数 xL、yL、xCbL、yCbL は次のように導出される：

$$xL = x * nSubWidth \quad (450)$$

$$yL = y * nSubHeight \quad (451)$$

$$xCbL = xCbComp * nSubWidth \quad (452)$$

$$yCbL = yCbComp * nSubHeight \quad (453)$$

10

- ・変数 blsEscapeSample は、次のように導出される：

・PaletteIndexMap[xCbL + xL][yCbL + yL] が MaxPaletteIndex に等しく、palette\_escape\_val\_present\_flag が 1 に等しい場合、blsEscapeSample は 1 に等しく設定される。

【0207】

- ・それ以外の場合、blsEscapeSample は 0 に設定される。

- ・blsEscapeSample が 0 に等しい場合、以下が適用される：

recSamples[x][y] = CurrentPaletteEntries[cldx][PaletteIndexMap[xCbL + xL][yCbL + yL]] (454)

・それ以外の場合 (blsEscapeSample が 1 に等しい)、以下の順序付けられたステップが適用される： 20

【0208】

4. 量子化パラメータ qP は次のように導出される：

- ・cldx が 0 に等しい場合、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Y) \quad (455)$$

- ・それ以外で、cldx が 1 に等しい場合、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Cb) \quad (456)$$

- ・それ以外の場合 (cldx は 2 に等しい)、

$$qP = \text{Max}(QpPrimeTsMin, Qp'Cr) \quad (457)$$

5. リスト levelScale[] は、k = 0..5 として、levelScale[k] = {40, 45, 51, 57, 64, 72} 30 } として指定される。

6. 次が適用される：

{{shift = Min(bitDepth - 1, (QpPrimeTsMin - 4) / 6)}}

[[tmpVal = (PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] \* levelScale[qP%6]) (qP/6) + 32) 6 (458)]]

{{qP' = Max(0, qP - 6 \* shift)}}

tmpVal = ((PaletteEscapeVal[cldx][xCbL + xL][yCbL + yL] \* shift) \* levelScale[qP'%6]) (qP'/6) + 32) 6 (458)}}

recSamples[x][y] = Clip3(0, (1 - BitDepth) - 1, tmpVal) (459)

【0209】

40

図9は、ビデオ処理装置900のブロック図である。装置900は、本明細書に記載される方法のうちの一つまたは複数を実装するために使用されてもよい。装置900は、スマートフォン、タブレット、コンピュータ、モノのインターネット (IoT) 受信機などにおいて具現されてもよい。装置900は、一つまたは複数のプロセッサ902、一つまたは複数のメモリ904、およびビデオ処理ハードウェア906を含んでいてもよい。プロセッサ902は、本稿に記載される一つまたは複数の方法を実装するように構成されてもよい。メモリ (単数または複数) 904は、本明細書に記載される方法および技法を実装するために使用されるデータおよびコードを記憶するために使用されてもよい。ビデオ処理ハードウェア906は、ハードウェア回路において、本稿に記載されるいくつかの技法を実装するために使用されてもよい。いくつかの実施形態では、ハードウェア906は、プロセッサ9 50

02、たとえばグラフィックスコプロセッサに対して少なくとも部分的に内部であってもよい。

【0210】

開示された技術のいくつかの実施形態は、ビデオ処理ツールまたはモードを有効にする判断または決定を行うことを含む。一例では、ビデオ処理ツールまたはモードが有効にされている場合、エンコーダは、ビデオのブロックの処理においてそのツールまたはモードを使用または実装するが、必ずしもそのツールまたはモードの使用に基づく、結果として生じるビットストリームを修正しなくてもよい。すなわち、ビデオのブロックからビデオのビットストリーム表現への変換は、前記判断または決定に基づいて有効にされているとき、そのビデオ処理ツールまたはモードを使用する。別の例では、ビデオ処理ツールまたはモードが有効にされている場合、デコーダは、ビットストリームがそのビデオ処理ツールまたはモードに基づいて修正されているという知識を用いて、そのビットストリームを処理する。すなわち、ビデオのビットストリーム表現からビデオのブロックへの変換は、前記判断または決定に基づいて有効にされたビデオ処理ツールまたはモードを使用して実行される。

10

【0211】

開示された技術のいくつかの実施形態は、ビデオ処理ツールまたはモードを無効にする判断または決定を行うことを含む。一例では、ビデオ処理ツールまたはモードが無効にされている場合、エンコーダは、ビデオのブロックをビデオのビットストリーム表現に変換する際に、そのツールまたはモードを使用しない。別の例では、ビデオ処理ツールまたはモードが無効にされている場合、デコーダは、ビットストリームが、判断または決定に基づいて有効にされたビデオ処理ツールまたはモードを使用して修正されていないという知識を用いて、そのビットストリームを処理する。

20

【0212】

図10は、本明細書に開示されるさまざまな技法が実装されうる例示的なビデオ処理システム1000を示すブロック図である。さまざまな実装は、システム1000の構成要素の一部または全部を含んでいてもよい。システム1000は、ビデオ・コンテンツを受信するための入力1002を含んでいてもよい。ビデオ・コンテンツは、生または非圧縮フォーマット、たとえば、8または10ビットの多成分ピクセル値の形で受領されてもよく、または圧縮またはエンコードされたフォーマットで受領されてもよい。入力1002は、ネットワークインターフェース、周辺バスインターフェース、または記憶インターフェースを表すことができる。ネットワークインターフェースの例は、イーサネット、受動光ネットワーク(PON)などの有線インターフェース、およびWi-Fiまたはセルラーインターフェースなどの無線インターフェースを含む。

30

【0213】

システム1000は、本稿に記載されるさまざまなコーディングまたはエンコード方法を実装しうるコーディングコンポーネント1004を含んでいてもよい。コーディングコンポーネント1004は、入力1002からコーディングコンポーネント1004の出力へ、ビデオの平均ビットレートを低減して、ビデオのコーディング表現を生成してもよい。よって、コーディング技法は、時に、ビデオ圧縮またはビデオトランスコード技法と呼ばれる。コーディングコンポーネント1004の出力は、コンポーネント1006によって表されるように接続された通信を介して、記憶されるか、または送信されてもよい。入力1002で受信されたビデオの記憶されたまたは通信されたビットストリーム(またはコーディングされた)表現は、ディスプレイインターフェース1010に送られるピクセル値または表示可能なビデオを生成するために、コンポーネント1008によって使用されうる。ビットストリーム表現からユーザーが見ることができるビデオを生成するプロセスは、時に、ビデオ圧縮解除と呼ばれる。さらに、ある種のビデオ処理動作は、「コーディング」操作またはツールと称されるが、コーディングツールまたは操作は、エンコーダにおいて使用され、コーディングの結果を反転させる対応する復号ツールまたは操作がデコーダにおいて実行されることが理解されるであろう。

40

50

## 【0214】

周辺バスインターフェースまたはディスプレイインターフェースの例は、ユニバーサルシリアルバス（USB）または高精細度マルチメディアインターフェース（MDMI）またはディスプレイポートなどを含む。記憶インターフェースの例は、SATA（シリアルアドバンスドテクノロジーアタッチメント）、PCI、IDEインターフェースなどを含む。本稿に記載される技法は、携帯電話、ラップトップ、スマートフォン、またはデジタルデータ処理および/またはビデオ表示を実行することができる他の装置のようなさまざまな電子装置において具現されうる。

## 【0215】

図11は、本開示の技表を利用することができる例示的なビデオコーディングシステム100を示すブロック図である。

10

## 【0216】

図11に示されるように、ビデオコーディングシステム100は、源装置110および宛先装置120を含んでいてもよい。源装置110は、ビデオ・エンコード装置と称されてもよく、エンコードされたビデオ・データを生成する。宛先装置120は、ビデオ・デコード装置と称されてもよく、源装置110によって生成されたエンコードされたビデオ・データをデコードしてもよい。

## 【0217】

源装置110は、ビデオ源112、ビデオ・エンコーダ114、および入出力（I/O）インターフェース116を含むことができる。

20

## 【0218】

ビデオ源112は、ビデオ捕捉装置のような源、ビデオ・コンテンツ・プロバイダーからビデオ・データを受信するためのインターフェース、および/またはビデオ・データを生成するためのコンピュータ・グラフィックス・システム、またはそのような源の組み合わせを含んでいてもよい。ビデオ・データは、一つまたは複数のピクチャーを含んでいてもよい。ビデオ・エンコーダ114は、ビデオ源112からのビデオ・データをエンコードしてビットストリームを生成する。ビットストリームは、ビデオ・データのコーディング表現を形成するビットのシーケンスを含んでいてもよい。ビットストリームは、コーディングされたピクチャーおよび関連するデータを含んでいてもよい。コーディングピクチャーは、ピクチャーのコーディング表現である。関連するデータは、シーケンスパラメータセット、ピクチャーパラメータセット、および他の構文構造を含んでいてもよい。I/Oインターフェース116は、変調器/復調器（モデム）および/または送信器を含んでいてもよい。エンコードされたビデオ・データは、ネットワーク130aを通じてI/Oインターフェース116を介して宛先装置120に直接送信されてもよい。エンコードされたビデオ・データはまた、宛先装置120によるアクセスのために記憶媒体/サーバー130b上に記憶されてもよい。

30

## 【0219】

宛先装置120は、I/Oインターフェース126、ビデオ・デコーダ124、および表示装置122を含んでいてもよい。

## 【0220】

I/Oインターフェース126は、受信機および/またはモデムを含んでいてもよい。I/Oインターフェース126は、源装置110または記憶媒体/サーバー130bから、エンコードされたビデオ・データを取得してもよい。ビデオ・デコーダ124は、エンコードされたビデオ・データをデコードしてもよい。表示装置122は、デコードされたビデオ・データをユーザーに対して表示してもよい。表示装置122は、宛先装置120と一体化されてもよく、または外部表示装置とインターフェースするように構成された宛先装置120の外部にあってもよい。

40

## 【0221】

ビデオ・エンコーダ114およびビデオ・デコーダ124は、高効率ビデオコーディング（HEVC）標準、多用途ビデオコーディング（VVC）標準、およびその他の現行のおよ

50

び/またはさらなる標準のようなビデオ圧縮標準に従って動作してもよい。

【0222】

図12は、図11に示されるシステム100内のビデオ・エンコーダ114であってもよいビデオ・エンコーダ200の一例を示すブロック図である。

【0223】

ビデオ・エンコーダ200は、本開示の技法のいずれかまたはすべてを実行するように構成されてもよい。図12の例では、ビデオ・エンコーダ200は、複数の機能コンポーネントを含む。本開示に記載される技法は、ビデオ・エンコーダ200のさまざまなコンポーネント間で共有されてもよい。いくつかの例では、プロセッサが、本開示に記載される技法のいずれかまたはすべてを実行するように構成されてもよい。

10

【0224】

ビデオ・エンコーダ200の機能コンポーネントは、分割部201と、モード選択部203、動き推定部204、動き補償部205、およびイントラ予測部206を含みうる予測部202と、残差生成部207と、変換部208と、量子化部209と、逆量子化部210と、逆変換部211と、再構成部212と、バッファ213と、エン트로ピーエンコーディング部214とを含んでいてもよい。

【0225】

他の例では、ビデオ・エンコーダ200は、より多くの、より少ない、または異なる機能コンポーネントを含んでいてもよい。一例では、予測部202は、イントラブロックコピー(IBC)部を含んでいてもよい。IBC部は、少なくとも1つの参照ピクチャーが現在のビデオ・ブロックが位置するピクチャーであるIBCモードで予測を実行することができる。

20

【0226】

さらに、動き推定部204や動き補償部205のようないくつかのコンポーネントは、高度に統合されてもよいが、説明のために図12の例では別個に表現されている。

【0227】

分割部201は、ピクチャーを一つまたは複数のビデオ・ブロックにパーティション分割することができる。ビデオ・エンコーダ200およびビデオ・デコーダ300は、さまざまなビデオ・ブロック・サイズをサポートすることができる。

【0228】

モード選択部203は、たとえば誤差結果に基づいて、コーディングモードのうちの1つ、イントラまたはインターを選択し、結果として得られるイントラまたはインターコーディングされたブロックを、残差ブロック・データを生成する残差生成部207に、また、参照ピクチャーとして使用するためにエンコードされたブロックを再構成する再構成部212に提供することができる。いくつかの例では、モード選択部203は、予測がインター予測信号およびイントラ予測信号に基づく、イントラおよびインター予測の組み合わせ(CIIP)モードを選択してもよい。モード選択部203は、また、インター予測の場合、ブロックについての動きベクトルの分解能(たとえば、サブピクセルまたは整数ピクセル精度)を選択してもよい。

30

【0229】

現在のビデオ・ブロックに対してインター予測を実行するために、動き推定部204は、バッファ213からの一つまたは複数の参照フレームを現在のビデオ・ブロックと比較することによって、現在のビデオ・ブロックについての動き情報を生成してもよい。動き補償部205は、現在のビデオ・ブロックに関連するピクチャー以外のバッファ213からのピクチャーのデコードされたサンプルと前記動き情報とに基づいて、現在のビデオ・ブロックについて予測されるビデオ・ブロックを決定することができる。

40

【0230】

動き推定部204と動き補償部205は、たとえば、現在のビデオ・ブロックがIスライス内にあるか、Pスライス内にあるか、Bスライス内にあるかによって、現在のビデオ・ブロックについて異なる動作を実行してもよい。

50

## 【0231】

いくつかの例では、動き推定部204は、現在のビデオ・ブロックについて一方向の予測を実行してもよく、動き推定部204は、現在のビデオ・ブロックについての参照ビデオ・ブロックを求めて、リスト0またはリスト1の参照ピクチャーを探索してもよい。次いで、動き推定部204は、参照ビデオ・ブロックを含むリスト0またはリスト1内の参照ピクチャーを示す参照インデックスと、現在のビデオ・ブロックと参照ビデオ・ブロックとの間の空間的変位を示す動きベクトルとを生成することができる。動き推定部204は、参照インデックス、予測方向指示子、および動きベクトルを現在のビデオ・ブロックの動き情報として出力することができる。動き補償部205は、現在のビデオ・ブロックの動き情報によって示される参照ビデオ・ブロックに基づいて、現在のブロックの予測されたビデオ・ブロックを生成することができる。 10

## 【0232】

他の例では、動き推定部204は、現在のビデオ・ブロックについて双方向予測を実行してもよく、動き推定部204は、現在のビデオ・ブロックについての参照ビデオ・ブロックを求めてリスト0内の参照ピクチャーを探索してもよく、現在のビデオ・ブロックについての別の参照ビデオ・ブロックを求めてリスト1内の参照ピクチャーをも探索してもよい。次いで、動き推定部204は、参照ビデオ・ブロックを含むリスト0およびリスト1における参照ピクチャーを示す参照インデックスと、参照ビデオ・ブロックと現在のビデオ・ブロックとの間の空間的変位を示す動きベクトルとを生成してもよい。動き推定部204は、現在のビデオ・ブロックの参照インデックスおよび動きベクトルを、現在のビデオ・ブロックの動き情報として、出力することができる。動き補償部205は、現在のビデオ・ブロックの動き情報によって示される参照ビデオ・ブロックに基づいて、現在のビデオ・ブロックの予測ビデオ・ブロックを生成することができる。 20

## 【0233】

いくつかの例では、動き推定部204は、デコーダのデコード処理のための動き情報のフルセットを出力してもよい。

## 【0234】

いくつかの例では、動き推定部204は、現在のビデオについての動き情報のフルセットは出力しなくてもよい。むしろ、動き推定部204は、別のビデオ・ブロックの動き情報を参照して現在のビデオ・ブロックの動き情報を信号伝達してもよい。たとえば、動き推定部204は、現在のビデオ・ブロックの動き情報が近傍のビデオ・ブロックの動き情報と十分に類似していると判断することができる。 30

## 【0235】

一例では、動き推定部204は、現在のビデオ・ブロックに関連する構文構造において、現在のビデオ・ブロックが別のビデオ・ブロックと同じ動き情報を有することをビデオ・デコーダ300に対して示す値を示してもよい。

## 【0236】

別の例では、動き推定部204は、現在のビデオ・ブロックに関連する構文構造において、別のビデオ・ブロックおよび動きベクトル差(MVD)を同定してもよい。動きベクトル差は、現在のビデオ・ブロックの動きベクトルと示されたビデオ・ブロックの動きベクトルとの間の差を示す。ビデオ・デコーダ300は、示されたビデオ・ブロックの動きベクトルと動きベクトル差とを使用して、現在のビデオ・ブロックの動きベクトルを決定することができる。 40

## 【0237】

上述のように、ビデオ・エンコーダ200は、動きベクトルを予測的に信号伝達してもよい。ビデオ・エンコーダ200によって実装されうる予測的な信号伝達技法の2つの例は、先進動きベクトル予測(AMVP)およびマージ・モード信号伝達を含む。

## 【0238】

イントラ予測部206は、現在のビデオ・ブロックに対してイントラ予測を実行することができる。イントラ予測部206が現在のビデオ・ブロックに対してイントラ予測を実 50

行するとき、イントラ予測部206は、同じピクチャー内の他のビデオ・ブロックのデコードされたサンプルに基づいて、現在のビデオ・ブロックのための予測データを生成することができる。現在のビデオ・ブロックのための予測データは、予測されるビデオ・ブロックおよびさまざまな構文要素を含んでいてもよい。

【0239】

残差生成部207は、現在のビデオ・ブロックから現在のビデオ・ブロックの予測ビデオ・ブロックを差し引くことによって（たとえば、マイナス記号によって示される）、現在のビデオ・ブロックのための残差データを生成することができる。現在のビデオ・ブロックの残差データは、現在のビデオ・ブロック内のサンプルの異なるサンプル成分に対応する残差ビデオ・ブロックを含んでいてもよい。

10

【0240】

他の例では、たとえばスキップ・モードにおいて、現在のビデオ・ブロックのための現在のビデオ・ブロックのための残差データが存在しないことがあり、残差生成部207は減算演算を実行しないことがある。

【0241】

変換処理部208は、現在のビデオ・ブロックに関連する残差ビデオ・ブロックに一つまたは複数の変換を適用することによって、現在のビデオ・ブロックのための一つまたは複数の変換係数ビデオ・ブロックを生成することができる。

【0242】

変換処理部208が現在のビデオ・ブロックに関連する変換係数ビデオ・ブロックを生成した後、量子化部209は、現在のビデオ・ブロックに関連する一つまたは複数の量子化パラメータ（QP）値に基づいて、現在のビデオ・ブロックに関連する変換係数ビデオ・ブロックを量子化することができる。

20

【0243】

逆量子化部210および逆変換部211は、変換係数ビデオ・ブロックから残差ビデオ・ブロックを再構成するために、変換係数ビデオ・ブロックに対して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。再構成部212は、再構成された残差ビデオ・ブロックを、予測部202によって生成された一つまたは複数の予測ビデオ・ブロックからの対応するサンプルに加算して、バッファ213に記憶するために現在のブロックに関連する再構成されたビデオ・ブロックを生成する。

30

【0244】

再構成部212がビデオ・ブロックを再構成した後、ビデオ・ブロック内のビデオ・ブロックロッキング・アーチファクトを低減するためにループ・フィルタリング動作が実行されてもよい。

【0245】

エントロピーエンコーディング部214は、ビデオ・エンコーダ200の他の機能コンポーネントからデータを受領することができる。エントロピーエンコーディング部214がデータを受領すると、エントロピーエンコーディング部214は、一つまたは複数のエントロピーエンコーディング動作を実行してエントロピーエンコーディングデータを生成し、エントロピーエンコーディングデータを含むビットストリームを出力することができる。

40

【0246】

図13は、図11に示されるシステム100内のビデオ・デコーダ114であってもよいビデオ・デコーダ300の例を示すブロック図である。

【0247】

ビデオ・デコーダ300は、本開示の技法のいずれかまたはすべてを実行するように構成されうる。図13の例では、ビデオ・デコーダ300は、複数の機能的コンポーネントを含む。本開示に記載される技法は、ビデオ・デコーダ300のさまざまなコンポーネント間で共有されてもよい。いくつかの例では、プロセッサが、本開示に記載される技法のいずれかまたはすべてを実行するように構成されてもよい。

50

## 【0248】

図13の例では、ビデオ・デコーダ300は、エントロピー復号部301、動き補償部302、イントラ予測部303、逆量子化部304、逆変換部305、再構成部306、バッファ307を含む。ビデオ・デコーダ300は、いくつかの例では、ビデオ・エンコーダ200(図12)に関して説明したエンコード・パス(pass)とおおむね逆のデコード・パスを実行することができる。

## 【0249】

エントロピー復号部301は、エンコードされたビットストリームを取り出すことができる。エンコードされたビットストリームはエントロピーコーディングされたビデオ・データ(たとえば、ビデオ・データのエンコードされたブロック)を含むことができる。エントロピー復号部301はエントロピーコーディングされたビデオ・データを復号することができる、エントロピー復号されたビデオ・データから、動き補償部302は、動きベクトル、動きベクトル精度、参照ピクチャー・リスト・インデックス、および他の動き情報を含む動き情報を決定することができる。動き補償部302は、たとえば、AMVPおよびマージ・モードを実行することによって、そのような情報を決定することができる。

10

## 【0250】

動き補償部302は、可能性としては補間フィルタに基づく補間を実行して、動き補償されたブロックを生成することができる。サブピクセル精度で使用される補間フィルタのための識別子が、構文要素に含まれてもよい。

## 【0251】

動き補償部302は、ビデオ・ブロックのエンコード中にビデオ・エンコーダ200によって使用される補間フィルタを使用して、参照ブロックの整数未満ピクセルについての補間された値を計算することができる。動き補償部302は、受領された構文情報に従ってビデオ・エンコーダ200によって使用される補間フィルタを決定し、該補間フィルタを使用して予測ブロックを生成することができる。

20

## 【0252】

動き補償部302は、構文情報の一部を使用して、エンコードされたビデオ・シーケンスのフレームおよび/またはスライスをエンコードするために使用されるブロックのサイズ、エンコードされたビデオ・シーケンスのピクチャーの各マクロブロックがどのようにパーティション分割されるかを記述する分割情報、各パーティションがどのようにエンコードされるかを示すモード、各インターコーディングされたブロックについての一つまたは複数の参照フレーム(および参照フレーム・リスト)、およびエンコードされたビデオ・シーケンスをデコードするための他の情報を決定することができる。

30

## 【0253】

イントラ予測部303は、たとえばビットストリームにおいて受領されたイントラ予測モードを使用して、空間的に隣接するブロックから予測ブロックを形成してもよい。逆量子化部303は、ビットストリーム内で提供され、エントロピー復号部301によって復号された量子化ビデオ・ブロック係数を逆量子化する、すなわち、脱量子化する。逆変換部303は、逆変換を適用する。

## 【0254】

再構成部306は、残差ブロックを、動き補償部202またはイントラ予測部303によって生成された対応する予測ブロックと加算して、デコードされたブロックを形成することができる。所望であれば、ブロック性アーチファクトを除去するために、デコードされたブロックをフィルタリングするようブロック解除フィルタも適用されてもよい。次いで、デコードされたビデオ・ブロックはバッファ307に格納され、バッファ307は、その後の動き補償/イントラ予測のための参照ブロックを提供し、表示装置上に提示するためのデコードされたビデオをも生成する。

40

## 【0255】

いくつかの実施形態では、以下の方法は、上記に列挙した例および実施形態のリストに基づく。一例では、これらの方法は、図9~図13に示される実装を使用して実施できる

50

が、これに限定されない。

【0256】

図14は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1400は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1410)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

10

【0257】

図15は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1500は、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1510)を含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされ、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モードコーディングツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される

20

図16は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1600は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1610)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長コーディングを使用し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

30

【0258】

図17は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1700は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1710)を含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

40

【0259】

図18は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1800は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行すること(1810)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代

50



表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

【0260】

図19は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法1900は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(1910)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずにコーディングされ、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

10

【0261】

図20は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2000は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2010)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされ、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

20

【0262】

図21は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2100は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2110)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの導出されるパレット・インデックスは最大値をもち、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

30

【0263】

図22は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2200は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2210)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

40

【0264】

図23は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2300は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2310)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいてコーディングされる構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して

50

現在のビデオ・ブロックを表現する。

【0265】

図24は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2400は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2410)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、予測的にコーディングされるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

10

【0266】

図25は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2500は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行すること(2510)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングスコーディングされる構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する。

20

【0267】

図26は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2600は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2610)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0268】

30

図27は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2700は、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ・ブロックのコーディングモードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップ(2710)と、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップ(2720)とを含む。

【0269】

図28は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法2800は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行すること(2810)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ(QP)差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される。

40

【0270】

図29は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるよ

50

うに、方法2900は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(2910)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数のコーディングブロック・フラグ(CBF)を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0271】

図30は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3000は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(3010)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数(NumPltIdx)は、K以上であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、Kは正の整数である。

【0272】

図31は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3100は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること(3110)を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ(QP)に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する。

【0273】

図32は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3200は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調(BDPCM)モードでコーディングされ、複数の変換ブロックまたはサブブロックに分割されることを決定するステップ(3210)と、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることとを実行するステップ(3220)とを含む。

【0274】

図33は、ビデオ処理のための例示的方法のフローチャートである。そこに示されるように、方法3300は、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行すること(3310)を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ(CG)パレット・モードを使用してコーディングされるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各コーディング単位(CU)の複数のセグメントを表す。

【0275】

以下の解決策は、いくつかの実施形態の好ましい特徴として、前のセクションに列挙されている項目(たとえば、項目1)に記載されている追加の技法とともに実装されてもよい。

【0276】

10

20

30

40

50

1. ビデオ処理の方法であって：現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのためのエスケープ・シンボルのバイナリ化は、次数Kの指数ゴロム(EG)符号を使用し、Kは、3と等しくない負でない整数であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

【0277】

2.  $K = 0$ である、解決策1の方法。

【0278】

3.  $K = 1$ である、解決策1の方法。

【0279】

4.  $K = 2$ である、解決策1の方法。

【0280】

ビデオ処理の方法であって、一つまたは複数のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、前記一つまたは複数のビデオ・ブロックのうちの現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされ、現在のビデオ・ブロックについてのエスケープ・シンボルのバイナリ化が固定長バイナリ化を用いるというフォーマット規則に適合し、前記パレット・モードコーディングツールは代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

【0281】

6. 前記固定長バイナリ化は、Nビットを使用し、Nは1より大きい整数である、解決策5の方法。

【0282】

7. Nが、内部ビット深さに基づく、解決策6の方法。

【0283】

8. Nの値が、スライス・サブピクチャー、タイル、ピクチャー、またはビデオにおいて信号伝達される、解決策6の方法。

【0284】

9. Nが量子化パラメータに基づいている、解決策6の方法。

【0285】

10. Nが、 $f(Q_p)$ として示される量子化パラメータ( $Q_p$ )の関数( $f()$ )に基づく、解決策9の方法。

【0286】

11. Nが $(ibd - \max(16, (Q_p - 4)/6))$ に設定され、ここで、ibdが内部ビット深さである、解決策9の方法。

【0287】

12. Nが $(ibd - \max(Q_pPrimeTsMin, (Q_p - 4)/6))$ に設定され、ここで、ibdは内部ビット深さであり、 $Q_pPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータである、解決策9の方法。

【0288】

13. Nが $\max(A, (ibd - \max(16, (Q_pPrimeTsMin - 4)/6)))$ に設定され、ここで、ibdは内部ビット深さであり、 $Q_pPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータであり、Aは負でない整数である、解決策9の方法。

【0289】

10

20

30

40

50

14.  $A = 0$  または  $A = 1$  である、解決策 13 の方法。

【0290】

15. 前記量子化パラメータが、前記ビデオのスライスの量子化パラメータと定数値との和であり、前記定数値が整数値である、解決策 9 ないし 14 のうちいずれか一項の方法。

【0291】

16. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるというフォーマット規則に適合し、前記現在のビデオ・ブロックのエスケープ・シンボルのバイナリ化は、可変長コーディングを使用し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

10

【0292】

17. 前記可変長コーディングが次数 3 の指数ゴロム符号を除外する、解決策 16 の方法。

【0293】

18. 前記可変長コーディングが、入力パラメータ  $K$  をもつ打ち切りされたバイナリ (TB) 符号であり、 $K$  は整数である、解決策 16 の方法。

20

【0294】

19.  $K$  は、(a) シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大コーディング単位 (LCU) 行、LCU のグループ、またはブリック、(b) 内部ビット深さ、(c) 入力ビット深さ、(d) 内部ビット深さと入力ビット深さとの間の相違、(e) 現在ビデオ・ブロックの寸法、(f) 現在ビデオ・ブロックの現在の量子化パラメータ、(g) 前記ビデオのカラー・フォーマットの指示、(h) コーディングツリー構造、または (i) 前記ビデオの色成分に基づいている、解決策 18 の方法。

30

【0295】

20. 前記エスケープ・シンボルの複数の値が、複数のバイナリ化方法を使用して信号伝達される、解決策 5 の方法。

【0296】

21. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記変換は、現在のビデオ・ブロックに対する量子化または逆量子化プロセスの適用を含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされるかどうかに基づいて、前記量子化または逆量子化プロセスの適用を構成するフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

40

【0297】

ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックのエスケープ・シンボルがバイナリシフト演算を使用して量子化および/または脱量子化されるように表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シン

50

ボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

【0298】

23．前記量子化が右ビットシフトに対応する、解決策22の方法。

【0299】

24．前記エスケープ・シンボルが $f(p, Q_p)$ としてコーディングされ、 $f()$ は関数であり、 $p$ は入力シンボル値であり、 $Q_p$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、解決策22の方法。

【0300】

25． $f$ が $p$   $g(Q_p)$ として定義される、解決策24の方法。

10

【0301】

26． $f$ が、 $(p + (1 - (g(Q_p) - 1)))$   $g(Q_p)$ として定義される、解決策24の方法。

【0302】

27． $f$ は、 $\text{clip}(0, (1 - bd) - 1, (p + (1 - (g(Q_p) - 1)))$   $g(Q_p)$ )として定義され、 $\text{clip}(x, \text{min}, \text{max})$ はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $\text{max}$ は整数である、解決策24の方法。

【0303】

28．前記エスケープ・シンボルは $h(p)$ としてコーディングされ、 $h()$ は関数であり、 $p$ は入力値シンボルである、解決策22の方法。

20

【0304】

29． $h$ が $p$   $N$ として定義され、 $N$ は負でない整数である、解決策28の方法。

【0305】

30． $h$ は、 $(p + (1 - (N - 1)))$   $N$ として定義され、 $N$ は負でない整数である、解決策28の方法。

【0306】

31． $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = 0$ である、解決策29または30の方法。

【0307】

32． $\text{cu\_transquant\_bypass\_flag} = 1$ のとき $N = (bd - \text{ibd})$ であり、 $bd$ は内部ビット深さであり、 $\text{ibd}$ は入力ビット深さである、解決策29または30の方法。

30

【0308】

33． $h$ は、 $\text{clip}(0, (1 - (bd - N) - 1, p - N)$ として定義され、ここで、 $bd$ は、現在のビデオ・ブロックの現在の色成分の内部ビット深さであり、 $N$ は負でない整数であり、 $\text{clip}(x, \text{min}, \text{max})$ は、クリッピング関数であり、 $x$ 、 $\text{min}$ 、および $\text{max}$ は整数である、解決策28の方法。

【0309】

34． $h$ は、 $\text{clip}(0, (1 - (bd - N) - 1, (p + (1 - (N - 1)))$   $N)$ として定義され、ここで、 $bd$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $N$ は、負でない整数であり、 $\text{clip}(x, \text{min}, \text{max})$ はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $\text{max}$ は整数である、解決策28の方法。

40

【0310】

35． $N$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策29ないし34のうちいずれか一項の方法。

【0311】

36．前記脱量子化は、左ビットシフトに対応する、解決策22の方法。

【0312】

37．前記エスケープ・シンボルは、 $f(p, Q_p)$ として脱量子化され、ここで、 $f()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルであり、 $Q_p$ は、現在のビデオ・ブ

50

ロックを表す対応する色成分についての導出された量子化パラメータである、解決策 3 6 の方法。

【 0 3 1 3 】

3 8 .  $f$ が  $p$   $g(Qp)$ として定義される、解決策 3 7 の方法。

【 0 3 1 4 】

3 9 . 前記エスケープ・シンボルは  $f(p, Qp)$ として再構成され、ここで、 $f()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルであり、 $Qp$ は、現在のビデオ・ブロックを表す対応する色成分についての導出された量子化パラメータである、解決策 3 6 の方法。

【 0 3 1 5 】

4 0 .  $f$ は、 $\text{clip}(0, (1 - bd) - 1, p \quad g(Qp))$ として定義され、 $bd$ は、現在のビデオ・ブロックの現在の色成分についての内部ビット深さであり、 $\text{clip}(x, \text{min}, \text{max})$ はクリッピング関数であり、 $x$ 、 $\text{min}$ 、 $\text{max}$ は整数である、解決策 3 9 の方法。

【 0 3 1 6 】

4 1 . クリッピング関数  $\text{clip}(x, \text{min}, \text{max})$ は

【数 2】

$$\text{clip}(x, \text{min}, \text{max}) = \begin{cases} \text{min} & x < \text{min} \\ x & \text{min} \leq x \leq \text{max} \\ \text{max} & x > \text{max} \end{cases}$$

10

20

として定義される、解決策 2 7、3 3、3 4 または 4 0 の方法。

【 0 3 1 7 】

4 2 . 前記エスケープ・シンボルは  $h(p)$ として再構成され、 $h()$ は関数であり、 $p$ はデコードされたエスケープ・シンボルである、解決策 3 6 の方法。

【 0 3 1 8 】

4 3 .  $h$ は  $p \quad N$ として定義され、 $N$ は負でない整数である、解決策 4 2 の方法。

【 0 3 1 9 】

4 4 .  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき  $N = 0$ である、解決策 4 2 または 4 3 の方法。

30

【 0 3 2 0 】

4 5 .  $\text{cu\_transquant\_bypass\_flag} = 1$ のとき  $N = (bd - ibd)$ であり、 $bd$ は内部ビット深さであり、 $ibd$ は入力ビット深さである、解決策 4 2 または 4 3 の方法。

【 0 3 2 1 】

4 6 .  $N = (\text{max}(QpPrimeTsMin, qP) - 4) / 6$ であって、 $qP$ はデコードされた量子化パラメータであり、 $QpPrimeTsMin$ は変換スキップ・モードのための最小の許容される量子化パラメータである、解決策 4 2 または 4 3 の方法。

【 0 3 2 2 】

4 7 .  $N$ がさらに  $\text{min}(bd - 1, N)$ としてクリップされ、 $bd$ が現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策 4 3 ないし 4 6 のうちいずれか一項の方法。

40

【 0 3 2 3 】

4 8 .  $N$ が  $[0, (bd - 1)]$ の範囲内であり、 $bd$ は現在のビデオ・ブロックの現在の色成分の内部ビット深さである、解決策 4 3 ないし 4 7 のうちいずれか一項の方法。

【 0 3 2 4 】

4 9 . 前記エスケープ・シンボルの再構成オフセットは、ビット深さ情報に基づく、解決策 3 6 の方法。

【 0 3 2 5 】

5 0 . 前記ビット深さ情報は、内部ビット深さと入力ビット深さとの間の差 ( $BD$ と記される)を含む、解決策 4 9 の方法。

50

## 【0326】

51.  $K_{BD}$ である場合、再構成されたオフセットは $p_{BD}$ に等しく、ここで、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ は、整数である、解決策50の方法。

## 【0327】

52. 前記再構成されたオフセットは、 $K_{T0}$ の場合、 $p_{BD}$ に等しく、 $p$ は、デコードされたエスケープ・シンボルであり、 $K$ および $T0$ は整数である、解決策49の方法。

## 【0328】

53.  $T0 = 2$ である、解決策50の方法。

## 【0329】

54. 再構成されたオフセットは、 $(p_{BD} - K) + ((1 - (K - 1))_{BD} - 1)_{BD}$ に等しく、 $p$ はデコードされたエスケープ・シンボルであり、 $K$ は整数である、解決策50の方法。

10

## 【0330】

55.  $BD$ が、シーケンス・レベル、ピクチャー・レベル、スライス・レベル、タイル・レベル、ブリック・レベル、またはサブピクチャー・レベルでビットストリーム表現において信号伝達される、解決策50の方法。

## 【0331】

56. 前記エスケープ・シンボルがコンテキストコーディングされる、解決策22ないし55のうちいずれか一項の方法。

20

## 【0332】

57. 前記エスケープ・シンボルがバイパスコーディングされる、解決策22ないし55のうちいずれか一項の方法。

## 【0333】

58.  $g(Qp)$ が $(Qp - 4)/6$ として定義される、解決策25ないし27、38または40のうちいずれか一項の方法。

## 【0334】

59.  $g(Qp)$ が $(\max(M, Qp) - 4)/6$ として定義され、 $M$ は整数である、解決策25ないし27、39または40のうちいずれか一項の方法。

## 【0335】

60.  $M$ が、シーケンスパラメータセット (SPS) において信号伝達される、解決策59の方法。

30

## 【0336】

61.  $g(Qp)$ が $[0, (bd - 1)]$ の範囲内であり、 $bd$ が、現在のビデオ・ブロックの現在の色成分についての内部ビット深さである、解決策58ないし60のうちいずれか一項の方法。

## 【0337】

62. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの一つまたは複数のパレット・インデックスは、参照インデックスを用いずにコーディングされ、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

40

## 【0338】

63. 前記一つまたは複数のパレット・インデックスのバイナリ化が、バイナリ化入力パラメータとして最大パレット・インデックスを有する打ち切りされたバイナリ (TB) 符号である、解決策62の方法。

## 【0339】

64. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビ

50



デオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを使用してコーディングされ、エスケープ・シンボルのインデックスと非エスケープ・シンボルのインデックスとの間の導出を制約するというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

【0340】

65．前記非エスケープ・シンボルのインデックスから前記エスケープ・シンボルのインデックスを導出することが許されない、解決策64の方法。 10

【0341】

66．前記エスケープ・シンボルのインデックスから前記非エスケープ・シンボルのインデックスを導出することが許されない、解決策64の方法。

【0342】

67．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがパレット・モードコーディングツールを用いてコーディングされるというフォーマット規則に適合し、前記パレット・モードコーディングツールの導出されるパレット・インデックスは最大値をもち、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。 20

【0343】

68．前記最大値は、現在のパレット・テーブル・サイズである、解決策67の方法。

【0344】

69．前記最大値は、一つまたは複数のエスケープ・シンボルについてのインデックスを除外する現在のパレット・テーブル・サイズであり、前記一つまたは複数のエスケープ・シンボルのうちのあるエスケープ・シンボルが、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、解決策67の方法。

【0345】

70．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックがエスケープ・シンボルを含む構文要素を用いて表現されるというフォーマット規則に適合し、前記エスケープ・シンボルを示すインデックスの値は、前記一つまたは複数のビデオ領域のそれぞれについて変更されず、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を用いずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。 40

【0346】

71．前記インデックスがMaxPaletteIndexに等しい、解決策70の方法。

【0347】

72．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在インデックスおよび参照インデックスに基づいてコーディングされる構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。 50

## 【0348】

73. 前記現在インデックスと前記参照インデックスとの間の差がコーディングされる、解決策72の方法。

## 【0349】

74. 前記差のコーディングされた表現は、ゼロ値の差を除外する、解決策73の方法。

## 【0350】

75. 前記現在インデックスと前記参照インデックスとの間の差のモジュロがコーディングされる、解決策72の方法。

## 【0351】

76. 前記モジュロが  $l = \text{modulo}(C - R, \text{MaxPaletteIndex})$  として表され、Cは前記現在インデックスであり、Rは前記参照インデックスであり、MaxPaletteIndexは所定の負でない整数である、解決策75の方法。

## 【0352】

77. 前記パレット・モードコーディングツールのパレット・ブロックの始めにおいて、前記参照インデックスが-1に設定される、解決策72の方法。

## 【0353】

78. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、予測的にコーディングされるエスケープ・シンボルを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

## 【0354】

79. 前記エスケープ・シンボルは、以前にコーディングされたエスケープ・シンボルに基づいて予測的にコーディングされる、解決策78の方法。

## 【0355】

80. 前記ビデオのある色成分における前記エスケープ・シンボルは、同じ色成分における値に基づいて予測的にコーディングされる、解決策78の方法。

## 【0356】

81. 前記ビデオの第1の色成分における前記エスケープ・シンボルは、前記第1の色成分とは異なる前記ビデオの第2の色成分における値に基づいて予測的にコーディングされる、解決策78の方法。

## 【0357】

82. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを用いてコーディングされる現在のビデオ・ブロックが、パレット項目をインデックス付けするためのパレット・インデックスに基づくコンテキストでランレングスコーディングされる構文要素を用いて表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを用いて現在のビデオ・ブロックを表現する、方法。

## 【0358】

83. 長さ要素のプレフィックスについてのコンテキストは、デコーダにおけるインデックス調整プロセスの後のパレット・インデックスに基づく、解決策82の方法。

## 【0359】

84. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行すること

10

20

30

40

50

を含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、前の諸パレット・インデックスとは独立して信号伝達される現在のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

【0360】

85．前記前の諸パレット・インデックスを使用することは、現在のビデオ・ブロックが一つまたは複数のエスケープ・シンボルを含むかどうかに基づき、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのためにエスケープ・シンボルが使用される、解決策84の方法。

10

【0361】

86．ビデオ処理の方法であって、整列規則に基づいて、ビデオの一つまたは複数のビデオ領域の現在のビデオ・ブロックのための量子化パラメータを予測するために使用される第1の近傍ビデオ・ブロックと、現在のビデオ・ブロックのコーディングモードを予測的に決定するために使用される第2の近傍ビデオ・ブロックとを決定するステップと、該決定に基づいて、前記ビデオと前記ビデオのビットストリーム表現との間の変換を実行するステップとを含む、方法。

【0362】

87．前記第1の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、解決策86の方法。

20

【0363】

88．前記第2の近傍ビデオ・ブロックが、左上の近傍ビデオ・ブロックまたは上の近傍ビデオ・ブロックである、解決策86または87の方法。

【0364】

89．前記コーディングモードが、現在のビデオ・ブロックについての最確モード (most probable mode、MPM) を含む、解決策86ないし88のうちいずれか一項の方法。

【0365】

90．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間で変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかにかかわらず、ブロック・レベル量子化パラメータ (QP) 差を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、方法。

30

【0366】

91．前記QP差が、閾値よりも大きい幅を有するパレット・ブロックについてコーディングされる、解決策90の方法。

40

【0367】

92．前記QP差が、閾値よりも高い高さを有するパレット・ブロックについてコーディングされる、解決策90の方法。

【0368】

93．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、パレット・ブロックのための一つまたは複数のコーディングブロック・フラグ (CBF) を含む構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な

50

色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

【0369】

94．前記CBFのそれぞれが1に等しく設定される、解決策93の方法。

【0370】

95．前記一つまたは複数のCBFの値は、前記現在のビデオ・ブロックがエスケープ・シンボルを含むかどうかに基づいており、前記エスケープ・シンボルは、前記代表的な色値を使用せずにコーディングされる前記現在のビデオ・ブロックのサンプルのために使用される、解決策93の方法。

【0371】

96．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、一つまたは複数のパレット・インデックスを含む構文要素を使用して表現されるというフォーマット規則に適合し、前記一つまたは複数のパレット・インデックスの数 (NumPltIdx) は、K以上であり、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現し、Kは正の整数である、方法。

10

【0372】

97．Kは、現在のパレット・サイズ (S)、エスケープ・フラグ (E)、または現在のビデオ・ブロックのサイズ (BlkS) に基づく、解決策96の方法。

20

【0373】

98． $K = S + E$ である、解決策97の方法。

【0374】

99．Kが、パレット・インデックスの最大値 (MaxPaletteIndex) に1を加えたものに等しい、解決策96の方法。

【0375】

100．前記構文要素の1つがNumPltIdx - Kを含む、解決策96の方法。

【0376】

101．(NumPltIdx - K)の値のバイナリ化が、打ち切りされたバイナリ符号である、解決策100の方法。

30

【0377】

102．(NumPltIdx - K)の値のバイナリ化が打ち切りされた単進符号である、解決策100の方法。

【0378】

103．(BlkS - K)がバイナリ化入力パラメータであり、BlkSが現在のビデオ・ブロックのサイズである、解決策101または102の方法。

【0379】

104．ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、パレット・モードコーディングツールを使用してコーディングされる現在のビデオ・ブロックが、現在のブロックのためのパレットの最大サイズ、現在のビデオ・ブロックのサイズ、無損失モードの使用、または量子化パラメータ (QP) に基づく構文要素を使用して表現されるというフォーマット規則に適合し、前記パレット・モードコーディングツールは、代表的な色値のパレットを使用して現在のビデオ・ブロックを表現する、方法。

40

【0380】

105．無損失モードが適用されたこと、QPが閾値より大きいこと、または変換スキップが適用されたことを判別したとき、現在のブロックのためのパレットのサイズは、現在のビデオ・ブロックのサイズに等しいと推定される、解決策104の方法。

【0381】

50

106. 前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット (DPS)、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、適応パラメータセット (APS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大コーディング単位 (LCU)、コーディング単位 (CU)、LCU行、LCUのグループ、変換単位 (TU)、予測単位 (PU) ブロック、またはビデオコーディング単位において信号伝達されるメッセージ、前記ビデオのカラー・フォーマットの指示、コーディングツリー構造、時間的IDレイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、解決策 1 ないし 105 のうちいずれか一項の方法。

【0382】

10

107. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、該ビデオのビットストリーム表現との間の変換のために、現在のビデオ・ブロックがブロックベースの差動パルス符号変調 (BDPCM) モードでコーディングされ、複数の変換ブロックまたはサブブロックに分割されることを決定するステップと、前記変換を実行することの一部として、ブロック・レベルでの残差予測と、前記決定に基づいて、一つまたは複数の残差を前記ビットストリーム表現にサブブロックまたは変換ブロック・レベルで含めることとを実行するステップとを含む、方法。

【0383】

108. ビデオ処理の方法であって、現在のビデオ・ブロックを含む一つまたは複数のビデオ領域を含むビデオと、前記ビデオのビットストリーム表現との間の変換を実行することを含み、前記ビットストリーム表現は、現在のビデオ・ブロックがライン・ベースの係数グループ (CG) パレット・モードを使用してコーディングされるというフォーマット規則に適合し、前記ライン・ベースのCGパレット・モードは、代表的な色値のパレットを使用して、現在のビデオ・ブロックの各コーディング単位 (CU)の複数のセグメントを表す、方法。

20

【0384】

109. 前記ビットストリーム表現は、各係数グループについてエスケープ・サンプルが存在するかどうかの指示を含み、エスケープ・サンプルは、前記代表的な色値を使用せずにコーディングされる現在のビデオ・ブロックのサンプルのために使用される、解決策 108 の方法。

30

【0385】

110. 前記ビットストリーム表現は、コンテキストコーディングされない上のインデックスをコピーの使用の指示を含む、解決策 108 の方法。

【0386】

111. 前記指示がバイパスコーディングされる、解決策 110 の方法。

【0387】

112. 一つまたは複数のコピー・フラグ、一つまたは複数のラン・タイプ、上のインデックスをコピーすること使用の一つまたは複数の指示、およびエスケープ値が、前記ビットストリーム表現において、インターリーブされた仕方で信号伝達される、解決策 108 の方法。

40

【0388】

113. 前記ライン・ベースのCGパレット・モードは、前記現在のビデオ・ブロックのサイズが閾値 (Th) 以下であると判定されると、無効にされる、解決策 108 の方法。

【0389】

114. 前記変換を実行することが、前記ビデオのビデオ・コンテンツ、デコーダパラメータセット (DPS)、シーケンスパラメータセット (SPS)、ビデオパラメータセット (VPS)、ピクチャーパラメータセット (PPS)、適応パラメータセット (APS)、ピクチャー・ヘッダ、スライス・ヘッダ、タイル・グループ・ヘッダ、最大コーディング単位 (LCU)、コーディング単位 (CU)、LCU行、LCUのグループ、変換単位 (TU)、予測単位 (PU) ブロック、またはビデオコーディング単位において信号伝達されるメッ

50

ページ、前記ビデオのカラー・フォーマットの指示、コーディングツリー構造、時間的I/Dレイヤー、または標準のプロファイル、レベル、もしくは階層のうちの一つまたは複数にさらに基づいている、解決策107ないし113のうちいずれか一項の方法。

【0390】

115．前記変換を実行することが、前記一つまたは複数のビデオ領域から前記ビットストリーム表現を生成することを含む、解決策1ないし114のうちいずれか一項の方法。

【0391】

116．前記変換を実行することが、前記ビットストリーム表現から前記一つまたは複数のビデオ領域を生成することを含む、解決策1ないし114のうちいずれか一項の方法。

【0392】

117．プロセッサと、命令を有する非一時メモリとを備えるビデオ・システム内の装置であって、前記命令は、前記プロセッサによる実行の際に、前記プロセッサに、解決策1ないし116のうちいずれか一項の方法を実施させる、装置。

【0393】

118．非一時的なコンピュータ読み取り可能媒体に記憶されたコンピュータ・プログラム・プロダクトであって、解決策1ないし116のうちいずれか一項の方法を実行するためのプログラム・コードを含む、コンピュータ・プログラム・プロダクト。

【0394】

本明細書に記載された、開示されたおよびその他の解決策、例、実施形態、モジュール、および機能的な動作は、デジタル電子回路において、または本明細書に開示された構造およびそれらの構造的等価物を含むコンピュータ・ソフトウェア、ファームウェア、またはハードウェアにおいて、またはそれらの一つまたは複数の組み合わせにおいて実装されることができる。開示されたおよびその他の実施形態は、一つまたは複数のコンピュータ・プログラム・プロダクト、すなわち、データ処理装置による実行のため、またはデータ処理装置の動作を制御するために、コンピュータ読み取り可能媒体上にエンコードされたコンピュータ・プログラム命令の一つまたは複数のモジュールとして実装されることができる。コンピュータ読み取り可能媒体は、機械読み取り可能な記憶デバイス、機械読み取り可能な記憶基板、メモリ装置、機械読み取り可能な伝搬信号を実現する物質の組成、またはそれらの一つまたは複数の組み合わせでありうる。用語「データ処理装置」は、たとえば、プログラム可能なプロセッサ、コンピュータ、または複数のプロセッサまたはコンピュータを含む、データを処理するためのすべての装置、デバイス、および機械を包含する。装置は、ハードウェアに加えて、問題のコンピュータ・プログラムのための実行環境を生成するコード、たとえば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはそれらの一つまたは複数の組み合わせを構成するコードを含むことができる。伝搬信号は、人工的に生成された信号、たとえば、好適な受信器装置に送信するための情報をエンコードするために生成される機械生成された電気信号、光信号、または電磁信号である。

【0395】

コンピュータ・プログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、またはコードとしても知られる）は、コンパイルされるまたはインタープリットされる言語を含む、任意の形のプログラミング言語で書くことができ、それは、スタンドアロン・プログラムとして、またはコンピューティング環境での使用に好適なモジュール、コンポーネント、サブルーチン、または他のユニットとして展開することを含め、任意の形で展開できる。コンピュータ・プログラムは、必ずしもファイルシステム内のファイルに対応するものではない。プログラムは、他のプログラムまたはデータを保持するファイルの一部に（たとえば、マークアップ言語文書に格納される一つまたは複数のスクリプト）、問題のプログラム専用の単一のファイルに、または複数の協調させられるファイル（たとえば、一つまたは複数のモジュール、サブプログラム、またはコード

10

20

30

40

50

部分を格納するファイル)に格納されることができる。コンピュータ・プログラムは、1つのコンピュータ上で、または1つのサイトに位置するか、または複数のサイトに分散され、通信ネットワークによって相互接続される複数のコンピュータ上で実行されるように展開されることができる。

【0396】

本稿に記載されるプロセスおよび論理フローは、一つまたは複数のコンピュータ・プログラムを実行する一つまたは複数のプログラム可能なプロセッサによって実行されて、入力データに対して作用し、出力を生成することによって機能を実行することができる。プロセスおよび論理フローは、特殊目的の論理回路、たとえばFPGA(フィールドプログラマブルゲートアレイ)またはASIC(特定用途向け集積回路)によって実行されることも

10

【0397】

コンピュータ・プログラムの実行に好適なプロセッサは、たとえば、汎用および専用マイクロプロセッサの両方、および任意の種類デジタル・コンピュータの任意の一つまたは複数のプロセッサを含む。一般に、プロセッサは、読み出し専用メモリまたはランダムアクセスメモリまたはその両方から命令およびデータを受領する。コンピュータの必須の要素は、命令を実行するためのプロセッサと、命令およびデータを記憶するための一つまたは複数のメモリデバイスである。一般に、コンピュータはまた、データを記憶するための一つまたは複数の大容量記憶装置、たとえば、磁気ディスク、磁気光ディスク、または光ディスクをも含む、またはそれらからデータを受領したり、またはそれにデータを転送したり、またはその両方をしたりするよう、動作上結合される。しかしながら、コンピュータは、そのような装置を有する必要はない。コンピュータ・プログラム命令およびデータを記憶するのに好適なコンピュータ読み取り可能媒体は、あらゆる形の不揮発性メモリ、媒体およびメモリデバイスを含み、たとえば、半導体メモリデバイス、たとえばEPROM、EEPROM、およびフラッシュメモリデバイス；磁気ディスク、たとえば、内蔵ハードディスクまたはリムーバブルディスク；光磁気ディスク；ならびにCD-ROMおよびDVD-ROMディスクを含む。プロセッサおよびメモリは、特殊目的論理回路によって補足されるか、またはそれに組み込まれることができる。

20

【0398】

この特許文献は多くの個別的事項を含んでいるが、これらは、いずれかの主題の範囲または特許請求されうるものの範囲に対する限定として解釈されるべきではなく、むしろ特定の技法の特定の実施形態に特有でありうる特徴の記述と解釈されるべきである。この特許文献において別個の実施形態の文脈に記載されているある種の特徴は、単一の実施形態において組み合わせられて実施されることもできる。逆に、単一の実施形態の文脈において記載されるさまざまな特徴が、複数の実施形態において別個に、または任意の好適なサブコンビネーションにおいて実施されることもできる。さらに、特徴は、ある種の組み合わせにおいて作用するものとして上述され、さらには当初はそのようなものとして請求項に記載されることがあるが、請求項に記載された組み合わせからの一つまたは複数の特徴が、場合によっては、組み合わせから削除されてもよく、請求項に記載された組み合わせは、サブコンビネーションまたはサブコンビネーションの変形に向けられてもよい。

30

40

【0399】

同様に、図面には特定の順序で動作が示されているが、これは、所望の結果を達成するために、そのような動作が示されている特定の順序で、あるいは逐次順に実行されること、あるいはすべての示されている動作が実行されることを要求すると解釈されるべきではない。さらに、この特許文献に記載されている実施形態におけるさまざまなシステムコンポーネントの分離は、すべての実施形態においてそのような分離を必要とするものとして理解されるべきではない。

【0400】

少数の実装および例のみが記述されており、この特許文献に記載され、説明されている内容に基づいて、他の実装、向上および変形がなされてもよい。

50

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/46581

A. CLASSIFICATION OF SUBJECT MATTER  
 IPC - H04N 19/124, H04N 19/157 (2020.01)  
 CPC - H04N 19/13, H04N 19/174, H04N 19/593, H04N 19/70, H04N 19/157

According to International Patent Classification (IPC) or to both national classification and IPC

10

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
 See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
 See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
 See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/0288415 A1 (Microsoft Technology Licensing, LLC), 04 October 2018 (04.10.2018), entire document, especially Abstract; para [0005], [0060]-[0063], [004]-[0108], [0158], [0165]-[0166]	1-4
A	US 2016/0100177 A1 (QUALCOMM Incorporated), 07 April 2016 (07.04.2016), entire document, especially Abstract;	1-4
A	US 2017/0127077 A1 (HFI INNOVATION INC), 04 May 2017 (04.05.2017), entire document, especially Abstract;	1-4

20

30

Further documents are listed in the continuation of Box C.  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

"D" document cited by the applicant in the international application

"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 13 December 2020	Date of mailing of the international search report <b>12 JAN 2021</b>
---	--

40

Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer <b>Lee Young</b> Telephone No. PCT Helpdesk: 571-272-4300
---	--

50



INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US 20/46581

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

- 1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
- 2.  Claims Nos.:  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
  
- 3.  Claims Nos.: 35, 47-48, 56-57, 61, 89, 106 and 115-118  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

10

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

--- ( See Continuation in Supplemental Box ) ---

- 1.  As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
  
- 2.  As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
  
- 3.  As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
- 4.  No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:  
Claims 1-4

20

30

Remark on Protest

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

40

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/46581

Continuation of:

Box III. Observations where unity of invention is lacking

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be searched, the appropriate additional search fees must be paid.

Group I - Claims 1-4 are directed to a method of video processing using exponential-Golomb (EG) code.

Group II - Claims 5-15 are directed to a method of video processing that uses fixed length binarization.

Group III - Claims 16-20 are directed to a method of video processing that uses a variable length coding.

Group IV - Claims 21 are directed to a method of video processing using a binary shift operation.

Group V - Claims 22-34, 36-46, 49-55, 58-60, 64-86 are directed to a method of video processing without using the representative color values.

Group VI - Claims 62-63, 67-85, 90-105 are directed to a method of video processing without using a reference index.

Group VII - Claims 86-88 are directed to a method of video processing based on neighboring video block used for predicting a quantization parameter.

Group VIII - Claims 107, 114/107 are directed to a method of video processing based on block-based differential pulse code modulation.

Group IX - Claims 108-113, 114/(108-113) are directed to a method of video processing based on a line-based coefficient group.

The inventions listed as Groups I-IX do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features:

The invention of Group I included the features of a binarization of an escape symbol for the current video block uses an exponential-Golomb (EG) code of order K, wherein K is a non-negative integer that is unequal to three, not required by any other group.

The invention of Group II included the features of a binarization of an escape symbol for the current video block uses a fixed length binarization, not required by any other group.

The invention of Group III included the features of a binarization of an escape symbol of the current video block uses a variable length coding, not required by any other group.

The invention of Group IV included the features the bitstream representation conforms to a format rule that configures the application of the quantization or the inverse quantization process based on whether the current video block is coded using a palette mode coding tool, and an escape symbol of the current video block is quantized and/or dequantized using a binary shift operation, not required by any other group.

The invention of Group V included the features of constrains a derivation between an index of an escape symbol and an index of a non-escape symbol, wherein the escape symbol is used for a sample of the current video block coded without using the representative color values, not required by any other group.

The invention of Group VI included the features of one or more palette indexes of the palette mode coding tool are coded without using a reference index; a derived palette index of the palette mode coding tool has a maximum value; palette mode coding tool is represented using syntax elements comprising an escape symbol, wherein a value of an index indicating the escape symbol is unchanged for each of the one or more video regions; represented using syntax elements that are coded based on the current index and a reference index; represented using syntax elements comprising an escape symbol that is predictively coded; represented using syntax elements that are run-length coded with a context based on a palette index for indexing palette entries; represented using syntax elements comprising a current palette index that is signaled independently of previous palette indices; represented using syntax elements comprising a block-level quantization parameter (QP) difference regardless of whether the current video block comprises an escape symbol; represented using syntax elements comprising one or more coded block flags (CBFs) for a palette block; represented using syntax elements comprising one or more palette indices, wherein a number of the one or more palette indices (Numplad0 is greater than or equal to K, K is a positive integer; represented using syntax elements based on a maximum size of a palette for the current block, a size of the current video block, a usage of a lossless mode, or a quantization parameter, not required by any other group.

--continued in next sheet --

10

20

30

40

50

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/46581

-- continued from previous sheet --

The invention of Group VII included the features of determining, based on an alignment rule, a first neighboring video block used for predicting a quantization parameter for a current video block of one or more video regions of a video and a second neighboring video block used for predictively determining a coding mode of the current video block; and performing, based on the determining, a conversion between the video and a bitstream representation of the video, not required by any other group.

10

The invention of Group VIII included the features of determining, that the current video block is coded with a block-based differential pulse code modulation (BDPCM) mode and split into multiple transform blocks or sub-blocks; performing, as part of performing the conversion, a residual prediction at a block level and an inclusion of one or more residuals in the bitstream representation at the sub-block or transform block level based on the determining, not required by any other group.

The invention of Group IX included the features of the bitstream representation conforms to a format rule that the current video block that is coded using a line-based coefficient group (CG) palette mode, wherein the line-based CG palette mode represents multiple segments of each coding unit (CU) of the current video block using a palette of representative color values, not required by any other group.

Common Technical Features

Groups I-IX share the features of conversion between the video and a bitstream representation of the video.

Groups I-VI share the features of performing a conversion between a video comprising one or more video regions comprising a current video block and a bitstream representation of the video, wherein the bitstream representation conforms to a format rule that the current video block is coded using a palette mode coding tool; the palette mode coding tool represents the current video block using a palette of representative color values.

20

Groups I-V share the features of the escape symbol is used for a sample of the current video block coded without using the representative color values.

However, the shared technical features do not represent a contribution over prior art as being anticipated by US 2017/0127077 A1 (HFI INNOVATION INC), 04 May 2017 (04.05.2017).

HFI INNOVATION INC teaches a conversion between the video and a bitstream representation of the video (para [0047]- encoder may include such constraint during encoding process.); performing a conversion between a video comprising one or more video regions comprising a current video block and a bitstream representation of the video, wherein the bitstream representation conforms to a format rule that the current video block is coded using a palette mode coding tool (para [0024]-[0025], [0078]- video coding system incorporating palette coding; current palette size of a current palette table of the current CU, or both predicted palette size and new palette size associated with the current palette table; method for palette coding of image and video data to avoid issues associated with zero-size palette); the palette mode coding tool represents the current video block using a palette of representative color values (para [0076]-[0077]- color index); the escape symbol is used for a sample of the current video block coded without using the representative color values (para [0051]-[0054], [0077] - all the pixels may still be encoded as escape pixels, where a major color index equal to zero represents the escape pixel; where the escape indication flag is asserted if there is any sample in the current CU coded as the escape pixel and the escape indication flag is not asserted if there is no sample in the current CU coded as the escape pixel. The size information related to palette size of a current palette table of the current CU is conditionally signaled based on the escape indication flag in step 130).

30

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups. Therefore, Groups I-IX lack unity under PCT Rule 13.

40

## フロントページの続き

(32)優先日 令和1年9月24日(2019.9.24)

(33)優先権主張国・地域又は機関  
中国(CN)

(31)優先権主張番号 PCT/CN2019/108736

(32)優先日 令和1年9月27日(2019.9.27)

(33)優先権主張国・地域又は機関  
中国(CN)

(31)優先権主張番号 PCT/CN2019/109793

(32)優先日 令和1年10月1日(2019.10.1)

(33)優先権主張国・地域又は機関  
中国(CN)

(31)優先権主張番号 PCT/CN2019/113931

(32)優先日 令和1年10月29日(2019.10.29)

(33)優先権主張国・地域又は機関  
中国(CN)

(31)優先権主張番号 PCT/CN2020/071221

(32)優先日 令和2年1月9日(2020.1.9)

(33)優先権主張国・地域又は機関  
中国(CN)

(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,IT,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,ST,SV,SY,TH,TJ,TM,TN,TR,T,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

弁理士 伊東 忠重

(74)代理人 100070150

弁理士 伊東 忠彦

(74)代理人 100135079

弁理士 宮崎 修

(72)発明者 ズー, ウエイジア

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン  
ボールヴァード 12655, シックスス フロアー, スイート ナンバー・137

(72)発明者 シュイ, ジェジョン

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン  
ボールヴァード 12655, シックスス フロアー, スイート ナンバー・137

(72)発明者 ザン, リー

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン  
ボールヴァード 12655, シックスス フロアー, スイート ナンバー・137

(72)発明者 チュアン, シャオ チアン

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン  
ボールヴァード 12655, シックスス フロアー, スイート ナンバー・137

F ターム(参考) 5C159 LC09 MA01 MA04 MA05 MA21 MC11 ME01 ME13 ME15 RC12

TA57 TB04 TB06 TB08 TC28 UA02 UA05 UA16