

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号
特開2023-123772
(P2023-123772A)

(43)公開日 令和5年9月5日(2023.9.5)

(51)国際特許分類		F I	
H 0 4 N	19/117 (2014.01)	H 0 4 N	19/117
H 0 4 N	19/186 (2014.01)	H 0 4 N	19/186
H 0 4 N	19/13 (2014.01)	H 0 4 N	19/13
H 0 4 N	19/70 (2014.01)	H 0 4 N	19/70
審査請求 有 請求項の数 1 O L (全44頁)			
(21)出願番号	特願2023-108523(P2023-108523)	(71)出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	令和5年6月30日(2023.6.30)	(74)代理人	110003281 弁理士法人大塚国際特許事務所
(62)分割の表示	特願2021-569987(P2021-569987))の分割	(72)発明者	タケ, ジョナサン フランス国 レンヌーアタラント, セデックス セッソンーセヴィニエ 35517, リュドゥラ トゥッシュランペール キヤノン リサーチ センター フランス エス. エー. エス 内
原出願日	令和2年8月18日(2020.8.18)	(72)発明者	ジスケ, クリストフ フランス国 レンヌーアタラント, セデックス セッソンーセヴィニエ 35517, リュドゥラ トゥッシュランペール
(31)優先権主張番号	1911952.8	最終頁に続く	
(32)優先日	令和1年8月20日(2019.8.20)		
(33)優先権主張国・地域又は機関	英国(GB)		

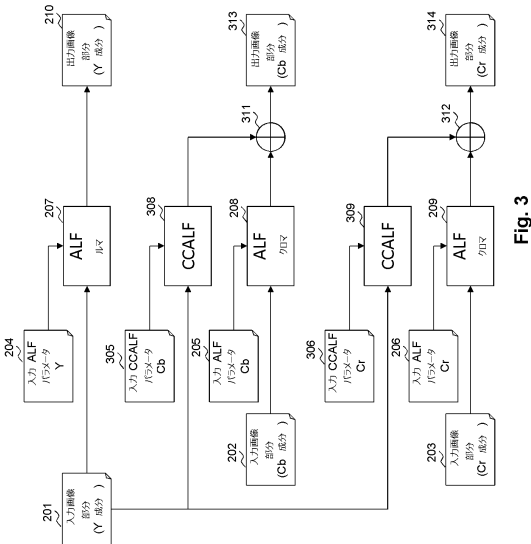
(54)【発明の名称】 クロマのためのクロスコンポーネント適応（アダプティブ）ループフィルタ

(57)【要約】 (修正有)

【課題】クロスコンポーネントフィルタを簡略化することにより、クロスコンポーネント適応ループフィルタ（CCALF）処理の一部の符号化性能を向上させる方法を提供する。

【解決手段】方法は、基準サンプルの周辺の複数のサンプルに対応する第1成分の複数のサンプル値を取得すること、ビットストリームから複数のフィルタ係数を取得すること、複数のフィルタ係数と、第1成分の複数のサンプル値とを用いて、第1成分とは異なる第2成分のために、CCALFの出力を生成すること、を含む。複数のフィルタ係数は、ビットストリームにおいてプレフィックスなしで表され、複数のフィルタ係数の値は、規定のビット数を使用して定義される2のべき乗の値で表される範囲に制限され、複数のフィルタ係数を取得することは、規定のビット数を用いて、複数のフィルタ係数におけるフィルタ係数の絶対値を決定することを含む。

【選択図】 図3



【特許請求の範囲】

【請求項 1】

クロスコンポーネント適応ループフィルタを実行する方法であって、
基準サンプルの周辺の複数のサンプルに対応する第 1 成分の複数のサンプル値を取得することと、

ビットストリームから複数のフィルタ係数を取得することと、

前記複数のフィルタ係数と、前記第 1 成分の複数のサンプル値とを用いて、前記第 1 成分とは異なる第 2 成分のために、クロスコンポーネント適応ループフィルタの出力を生成することと、

を含み、

前記複数のフィルタ係数は、前記ビットストリームにおいてプレフィックスなしで表され、

前記複数のフィルタ係数の値は、規定のビット数を使用して定義される 2 のべき乗の値で表される範囲に制限され、

前記複数のフィルタ係数を取得することは、前記規定のビット数を用いて、前記複数のフィルタ係数におけるフィルタ係数の絶対値を決定することを含む

方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオ成分（コンポーネント）の複数のブロックの符号化または復号化に関するものである。本発明の実施形態は、そのような成分の複数のサンプルをフィルタするためにフィルタを制御するときに、特に（しかし排他的ではない）用途を見出す。特に、適応（アダプティブ）ループフィルタを制御することであるが、これに限定されるものではない。

【背景技術】

【0002】

ビデオ符号化には、画像符号化（画像はビデオの 1 フレームに相当）を含む。ビデオ符号化では、変換係数の量子化や動き補償（補間フィルタを用いて行われることが多い）などの符号化ツールにより、歪みのバイアス／影響（系統的と思われる歪み、あるいは少なくとも所与のコンテキストではランダムでない歪み）が生じることがある。これらのバイアスやアーティファクトを補正し、符号化効率を向上させる（または少なくとも良好なレベルを維持する）ために、ポストフィルタまたはインループフィルタと呼ばれるいくつかの特定の符号化ツールが使用される。デブロッキングフィルタ（DBF）、サンプル適応（アダプティブ）オフセット（SAO）フィルタ、適応ループフィルタ（ALF）などは、そのような符号化ツールの一例である。インループフィルタは、現在のフレームに提供される画質の向上が、現在のフレームに基づいて符号化される後続のフレームの符号化効率の向上を可能にするように、符号化ループ内に適用される。例えば、DCT 係数の量子化はビデオ圧縮には効率的であるが、圧縮されたサンプルブロックの境界でブロッキングアーチファクト（バイアス）が発生することがある。デブロッキングフィルタは、このよ

【0003】

本発明は、特に適応ループフィルタ（ALF）に関するものであり、これは、復号化されたフレーム／画像における望ましくない圧縮アーティファクトを低減するためのインループフィルタとしても適用される。ALF は、ビデオ符号化エキスパーツグループ／動画画像エキスパーツグループ（VCEG／MPEG）標準化グループによって研究されており

10

20

30

40

50

、例えばVVCテストモデルソフトウェアの第5バージョン（VTM-5.0またはVVCドラフトバージョン5）において、多用途ビデオ符号化（VVC）規格に使用することが検討されている。クロスコンポーネントALF（CCALF）は、JVET-00636（スウェーデン国のヨーテボリにて2019年7月3日-12日に開催されたジョイントビデオエキスパートチーム（JVET）の第15回ミーティング）で提案され、ALFの補助フィルタであり、ルマチャネルのサンプル値を用いて、クロマチャネルのサンプル値に対して行われるアーティファクトの追加のエラー訂正を見積もるものである。CCALFは、各クロマチャネルに適用されてもよい。

【発明の概要】

【0004】

本発明は、クロスコンポーネントフィルタを簡略化することにより、CCALF処理の一部の符号化性能を向上させることに関するものである。

【0005】

本発明の実施形態は、複数のフィルタ係数または複数のクロスコンポーネント入力サンプルのいずれかまたは両方の（内部ビット深度と比較して）低減されたビット深度表現を使用することに関するものである。この制限は、ビットシフト演算及び／又はフィルタ係数／入力サンプルの特定のビットを無視することによって実行されてもよい。

【0006】

本発明の一態様では、画像部分をフィルタする方法が提供され、該方法は、参照サンプルを取り囲む複数のサンプルに対応する第1成分の複数のサンプル値と複数のフィルタ係数とを受信することと、前記複数のフィルタ係数と前記第1成分の複数のサンプル値とをクロスコンポーネントフィルタに入力しクロスコンポーネントフィルタ出力を生成することと、を含み、前記クロスコンポーネントフィルタは、前記複数のフィルタ係数および／または前記第1成分の複数のサンプル値を表すための制限されたビット数を使用して前記フィルタ出力を生成する。

【0007】

このような方法では、フィルタ計算が簡略化され、CCALF符号化の改善の大部分が驚くほど維持される。

【0008】

オプションで、前記制限されたビット数は、内部ビット深度によって定義されるビット数よりも低い。

【0009】

効率的な実施のために、前記複数のフィルタ係数および／または前記第1成分の複数のサンプル値を表すための前記制限されたビット数は、前記フィルタ出力が16ビット以下で表されるようにする。

【0010】

オプションで、本方法は、前記クロスコンポーネントフィルタ出力を異なる成分に対応するフィルタの出力と結合することをさらに含む。オプションで、前記第1成分はルマであり、前記異なる成分はクロマ成分である。

【0011】

効率的な実施のために、本方法は、フィルタ乗算演算の前にビット数を削減させるようにビットシフト動作を実行することを含む。これにより、フィルタ計算を簡略化することができる。

【0012】

複数のサンプル値の制限

本発明の一態様では、サンプル値を表すために使用されるビット数が削減される。

【0013】

オプションで、第3の変形例によれば、前記制限することは、特定の複数のビットのみを考慮することによってサンプル値の精度を削減させることを含む。

【0014】

10

20

30

40

50

オプションで、所定の数の最下位ビットが無視される；例えば、最下位ビット、最下位 2 ビット、最下位 3 ビット又は最下位 4 ビットである。

【0015】

オプションで、所定の数の最上位ビットが保持される。例えば、最上位 8、7、6、5 または 4 ビットである。

【0016】

さらに簡略化するために、この方法は、除去された最上位のビットを使用してサンプル値を丸めることを含んでいてもよい。

【0017】

オプションで、第 4 の変形例によれば、前記複数のサンプル値を前記フィルタに入力することは、前記第 1 成分のサンプル値と参照サンプル値との値の差を表す数値を入力することを含む。

10

【0018】

オプションで、本方法は、固定クリッピングパラメータ C によって定義されるクリッピング関数に基づいて、前記差をクリッピングすることをさらに含む。

【0019】

前記クリッピング関数 K は、以下のように表されてもよい：

$$K(d, C) = \max(-C, \min(C, d))$$

ここで、C はクリッピングパラメータであり、d は差を表す数値である。したがって、クリッピング関数の入力、差を表す数値とクリッピングパラメータであり、出力は -C の最大値または C と d の間の小さい方の値である。

20

【0020】

有利な一実施形態では、C は $2^{(IBitDepth - n)} - 1$ であり、IBitDepth は複数のサンプル値を表すために用いられるビット数であり、n は $n < IBitDepth$ を満足する整数である。オプションで、n = 2 または 3 である。

【0021】

複数の係数値の制限

本発明の一態様では、複数のフィルタ係数値を表現するために使用されるビット数が削減される。

【0022】

オプションで、第 2 の変形例によれば、前記制限することは、複数のフィルタ係数の可能な値の範囲を制限することを含む。これにより、より少ないビット数で係数を表現することができ、その後の計算がより簡単になる。

30

【0023】

オプションで、第 1 の変形例によれば、前記制限することは、入力係数値の固定小数点小数精度を削減させたものを使用することを含む。

【0024】

特定の利点を有する実施形態は、前記削減された固定小数点小数精度は 8 ビット、7 ビット、または 6 ビットのうちの 1 つである場合を含む。

【0025】

1 つのオプションの実施形態において、前記範囲を制限することは、閾値より高いかなるビットも使用しないことを含み、例えば、前記閾値は固定小数点小数精度の第 7、第 6、第 5、第 4 または第 3 のビットである。

40

【0026】

本発明の 1 つの有利な実施態様では、各乗算はビット単位のシフトおよび加算の演算に置き換えられ、これらはハードウェアおよびソフトウェアの両方で実施がより簡単である。このように、前記フィルタ演算は複数のサンプル値と 1 つのフィルタ係数の乗算を含み、前記乗算は該乗算の演算をビット単位のシフトおよび加算の演算で置き換えるようにビット単位のシフトおよび／または他のフィルタ係数との組み合わせを介して実施される。

【0027】

50

フィルタ形状

本発明の別の態様では、フィルタの形状の修正が提案される。そのような修正には、CCALFによって提供される利得を維持しながら、考慮すべきサンプルの数を減らすこと（したがって、結果の計算を単純化すること）が含まれる。

【0028】

オプションで、第1成分の各サンプル値は、参照サンプルに対して定義された位置の1つのサンプル値であり、サンプルの前記位置はフィルタ形状によって定義される。

【0029】

実施を容易にするために、前記フィルタ形状は、16個以下のサンプルを含むようにされる。

【0030】

一実施形態では、前記フィルタ形状は、前記第1成分のサンプルが前記参照サンプルに対して位置 $(x+u, y+v)$ を有するパターンを含み、 u および v は $[-2, 2]$ 整数区間に属する。

【0031】

別の実施形態では、前記フィルタ形状は、すべてのサンプルが前記参照サンプルに対して位置 $(x+u, y+v)$ を有するパターンを含み、 u は $[-2, 2]$ に属し、 v は $[-1, 2]$ に属する。変形例では、前記フィルタは「菱形(diamond-shaped)」であり、例えば、 $|u|=2$ のとき、 v は $[0, 1]$ に属する。

【0032】

一実施形態では、前記フィルタ形状は、すべてのサンプルが前記参照サンプルに対して位置 $(x+u, y+v)$ を有するパターンを含み、 u は $[-1, 1]$ に属し、 v は $[-1, 2]$ に属する。変形例では、前記フィルタは「菱形」であり、例えば、 $|u|=1$ のとき、 v は $[0, 1]$ に属する。

【0033】

別の実施形態では、前記フィルタ形状は、すべてのサンプルが前記参照サンプルに対して位置 $(x+u, y+v)$ を有するパターンを含み、 u, v は $[0, 1]$ に属する。

【0034】

シグナリング

本発明の実施形態は、ビットストリーム内の複数のCCALFフィルタ係数をシグナリングすることに関するものでもある。本発明の目的は、CCALFに関連するビットレートコストを最小化するように、シグナリングを改善することである。

【0035】

オプションで、本方法は、前記フィルタ係数および／または複数のサンプル値に対する削減されたビット数をビットストリーム内でシグナリングすることをさらに含む。これにより、異なる複数の画像に対して一貫した処理を行うことができ、必要に応じて処理に柔軟性を持たせることができる。

【0036】

本発明の一態様によれば、本方法は、所与の成分に対してクロスコンポーネント適応ループフィルタ(CCALF)が指示されているかどうかを決定することと、前記ビットストリームから前記複数のフィルタ係数を復号化することと、をさらに含む。

【0037】

本発明の別の態様によれば、CCALFを用いて符号化された画像部分を復号化する方法が提供され、該方法は、所与の成分に対してCCALFが指示されているかどうかを決定することと、ビットストリームから複数のCCALF係数を復号化することと、前記複数のCCALF係数を用いて前記画像部分を復号化することと、を含み、前記複数のCCALF係数は、前記ビットストリームにプレフィックス無しで符号化されている。

【0038】

この態様は、従来技術に対して簡略化を表し、特に、本明細書に記載のように複数のフィルタ係数を表すために用いられるビット数を削減することと組み合わせた場合、CCA

10

20

30

40

50

L F の性能を大きく損なうことなくより低いビットレートをもたらす。

【0039】

オプションで、所与の成分に対して C C A L F が指示されているかどうかを前記決定することは、フラグを復号化することを含む。

【0040】

本発明の別の態様によれば、クロスコンポーネント適応ループフィルタ (C C A L F) を用いて画像部分を符号化する方法が提供され、該方法は、所与の成分に対して C C A L F が指示されているかどうかを決定することと、複数の C C A L F 係数をビットストリームに符号化することと、前記複数の C C A L F 係数を用いて前記画像部分を符号化することと、を含み、前記複数の C C A L F 係数は、固定表現を用いてスライスヘッダに符号化される。

10

【0041】

オプションで、所与の成分に対して C C A L F が指示されているかどうかを前記決定することは、フラグをスライスヘッダに符号化することを含む。

【0042】

第1の変形例では、実施を容易にするために、前記複数の C C A L F 係数はスライスヘッダに直接的に符号化されてもよい。

【0043】

前記複数の C C A L F 係数は、符号付き単項符号化を使用してビットストリームに符号化されてもよい。このような符号化方式は、符号化および復号化が簡単であり、そのために他のパラメータ (例えば、プレフィックス) に依存することはない。

20

【0044】

オプションで、第3の変形例によれば、前記符号付き単項符号化は、前記係数の符号を示す1つのビットと、前記係数の振幅を示す一連のビットと、を含む。

【0045】

オプションで、第2の変形例によれば、前記符号付き単項符号化は、前記係数がゼロであるか否かを示す前記ビットストリーム内の第1ビットと、前記係数がゼロでない場合、前記係数の符号を示す1つのビットと、前記係数の振幅を示し1に等しいビットシーケンスと、を含む。

【0046】

オプションで、前記第1ビットが1に等しいことは前記係数がゼロであることを示す。

30

【0047】

オプションで、1に等しい前記符号ビットは、前記係数が負であることを示す。

【0048】

オプションで、パッシングを容易にするために、前記符号付き単項符号化は、前記係数のシグナリングの終了を示すシーケンスの終了にゼロに等しい1つのビットをさらに含む。

【0049】

本発明のさらに別の態様は、請求項48によって定義されるような、画像をフィルタするための装置に関する。

40

【0050】

本発明のさらに別の態様は、請求項49および50によってそれぞれ定義されるような、符号化器および復号化器に関する。

【0051】

本発明のさらに別の態様は、請求項51によって定義されるようなプログラムに関する。プログラムは、それ自体で提供されてもよいし、キャリア媒体上で提供されてもよいし、キャリア媒体によってまたはキャリア媒体内で提供されてもよい。キャリア媒体は、非一時的なもの、例えば記憶媒体、特にコンピュータ可読記憶媒体であってもよい。キャリア媒体はまた、例えば信号または他の伝送媒体のような一時的なものであってもよい。信号は、インターネットを含む任意の適切なネットワークを介して伝送されてもよい。

50

【0052】

本発明のさらなる特徴は、他の独立請求項および従属請求項によって特徴付けられる。

【0053】

本発明の一態様における任意の特徴は、任意の適切な組み合わせで、本発明の他の態様に適用され得る。特に、方法の態様は、装置の態様に適用されてもよく、その逆もまた然りである。

【0054】

さらに、ハードウェアに実施された特徴は、ソフトウェアに実施されてもよく、その逆もまた然りである。本明細書におけるソフトウェア及びハードウェアの特徴へのいかなる言及も、それに応じて解釈されるべきである。

【0055】

本明細書に記載されているような装置の特徴は、方法の特徴として提供されることもあり、その逆もまた然りである。本明細書で使用されるように、手段プラス機能 (means plus function) 特徴は、適切にプログラムされたプロセッサ及び関連するメモリのよ

10

うなそれらの対応する構造の観点から代替的に表現され得る。

【0056】

また、本発明の任意の態様において説明され定義された様々な特徴の特定の組み合わせは、独立して実施および／または供給および／または使用することができることを理解されたい。

【図面の簡単な説明】

20

【0057】

次に、本発明の実施形態を、例示としてのみ、以下の図面を参照しながら説明する。

【0058】

【図1】図1は、VTM-5.0の典型的な復号化ループにおいてALFが行われる場所を示す。

【図2】図2は、VTM-5.0のALFフィルタの概要を説明するフローチャートである。

【図3】図3は、VTM-5.0のALFに加え、CCALFフィルタの概要を示すフローチャートである。

【図4-a】図4-aは、本発明の実施形態によるCCALFフィルタのフィルタ形状および係数配置を提供する。

30

【図4-b】図4-bは、CCALFフィルタ係数の典型的なビット単位のメモリ表現を示す。

【図4-c】図4-cは、10ビットの内部ビット深度を有するCCALFフィルタ中に使用されるサンプル値の典型的なビット単位のメモリ表現を示す。

【図4-d】図4-dは、CCALFフィルタを実行中の、CCALFフィルタ係数とサンプル値との乗算の出力の典型的なビット単位のメモリ表現を示す。

【図4-e】図4-eは、CCALFフィルタを実行中の、複数のCCALFフィルタ係数と複数のサンプル値との乗算の複数の結果を合計するために実行される加算演算の出力の典型的なビット単位のメモリ表現を示す。

40

【図5-a】図5-aは、本発明の第1の変形例による、係数値の精度を削減することによってCCALFフィルタ係数のビット数を削減するビット単位のメモリ表現を示す。

【図5-b】図5-bは、本発明の第2の変形例による、可能な係数値の範囲を制限することによってCCALFフィルタ係数のビット数を削減するビット単位のメモリ表現を示す。

【図5-c】図5-cは、本発明の第1及び第2の変形例を組み合わせた実施例による、係数値の精度を削減しかつ可能な係数値の範囲を制限することによってCCALFフィルタ係数のビット数を削減するビット単位のメモリ表現を示す。

【図5-d】図5-dは、本発明の第3の変形例による、最下位ビットを無視することによってフィルタ処理の内部で使用するサンプル値のビット数を削減するビット単位のメ

50

メモリ表現を示す。

【図 6 - a】図 6 - a は、本発明の第 4 の変形例による、フィルタ処理の内部で使用される複数のサンプル値の差のビット単位のメモリ表現を示す。

【図 6 - b】図 6 - b は、本発明の第 4 の変形例による、フィルタ処理の内部でビット数を削減するためにクリッピング操作を使用する場合の複数のサンプル値の差のビット単位のメモリ表現を示す。

【図 6 - c】図 6 - c は、本発明の第 3 及び第 4 の変形例を組み合わせた実施例による、フィルタ処理の内部でビット数を削減するためにクリッピング操作を使用しかつ最下位ビットを無視する場合の複数のサンプル値の差のビット単位のメモリ表現を示す。

【図 7 - a】図 7 - a は、本発明の変形例による C C A L F フィルタを実行中の、C C A L F フィルタ係数とサンプル値との乗算の出力の典型的なビット単位のメモリ表現を示す。

【図 7 - b】図 7 - b は、本発明の変形例による C C A L F フィルタを実行中の、複数の C C A L F フィルタ係数と複数のサンプル値との乗算の複数の結果を合計するために実行される加算演算の出力の典型的なビット単位のメモリ表現を示す。

【図 7 - c】図 7 - c は、本発明の第 5 の変形例による、C C A L F フィルタ処理で使用するビット数及び演算を減らすために 16 個のサンプルを使用する C C A L F フィルタの一例のフィルタ形状及び係数配置を提供する。

【図 7 - d】図 7 - d は、本発明の第 5 の変形例による、C C A L F フィルタ処理で使用するビット数及び演算を減らすために 16 個のサンプルを使用する C C A L F フィルタの一例のフィルタ形状及び係数配置を提供する。

【図 7 - e】図 7 - e は、本発明の第 5 の変形例による、C C A L F フィルタ処理で使用するビット数及び演算を減らすために 12 個のサンプルを使用する C C A L F フィルタの一例のフィルタ形状及び係数配置を提供する。

【図 7 - f】図 7 - f は、本発明の第 5 の変形例による、C C A L F フィルタ処理で使用するビット数及び演算を減らすために 8 個のサンプルを使用する C C A L F フィルタの一例のフィルタ形状及び係数配置を提供する。

【図 7 - g】図 7 - g は、本発明の第 5 の変形例による、C C A L F フィルタ処理で使用するビット数及び演算を減らすために 4 つのサンプルを使用する C C A L F フィルタの一例のフィルタ形状及び係数配置を提供する。

【図 8】図 8 は、本発明の実施形態による符号化方法の複数のステップを示すフローチャートである。

【図 9】図 9 は、本発明の実施形態に係る復号化方法の複数のステップを示すフローチャートである。

【図 10】図 10 は、本発明の 1 つ以上の実施形態が実施され得るデータ通信システムを模式的に示すブロック図である。

【図 11】図 11 は、本発明の 1 つ以上の実施形態が実施され得る処理装置の構成要素を示すブロック図である。

【図 12】図 12 は、本発明の 1 つ以上の実施形態が実施され得るネットワークカメラシステムを示す図である。

【図 13】図 13 は、本発明の 1 つ以上の実施形態が実施され得るスマートフォンを示す図である。

【発明を実施するための形態】

【0059】

以下に説明する本発明の実施形態は、画像の符号化および復号化の改善に関するものである。

【0060】

本明細書において「シグナリング」は、フィルタを制御するための 1 つ以上のパラメータに関する情報（例えばモード／スキームの、使用、不使用、有効、または無効、または他のフィルタ制御関連情報）をビットストリームに挿入する（中に提供する／含める／符

10

20

30

40

50

号化する)、またはそこから抽出する／取得する(復号化する)ことを指す場合がある。

【0061】

本明細書では、画像部分の例として「スライス」という用語を使用する(このような画像部分の他の例は、タイルまたは(タイル(複数可)のグループ／セットである)タイルグループである)。本発明の実施形態は、スライスの代わりに画像部分(例えばタイル又はタイルグループ)、及び画像部分／タイル／タイルグループのヘッダ(スライスヘッダの代わり)、画像部分／タイル／タイルグループのタイプ(スライスタイプの代わり)、及び画像部分／タイル／タイルグループの統計値(スライス統計値の代わり)などの適切に変更したパラメータ／値／構文に基づいて実施することができるということも理解され得る。また、スライスヘッダ又はシーケンスパラメータセット(SPS)の代わりに、ALF又はCCALFパラメータ(又は(CC)ALFフィルタを用いるための情報)をシグナリングするために、適応(アダプティブ)パラメータセット(APS)又はタイル(グループ)ヘッダが使用されてもよいことが理解される。APSが(CC)ALFパラメータ(または(CC)ALFフィルタを使用するための情報)をシグナリングするために使用される場合、スライスヘッダまたはタイルグループヘッダは、例えば、適応(アダプティブ)セット識別子(aps_id)を示すことによって、(CC)ALFパラメータ(または(CC)ALFフィルタを使用するための情報)を得るためにどのAPSを使用しなければならないかを示すのに使用されてもよい。また、スライス、タイルグループ、タイル、符号化ツリーユニット(CTU)／最大符号化ユニット(LCU)、符号化ツリーブロック(CTB)、符号化ユニット(CU)、予測ユニット(PU)、変換ユニット(TU)、又は画素／サンプルのブロックのいずれかが画像部分と呼ばれることがあることが理解される。

【0062】

また、次のことも理解される：フィルタまたはツールが「アクティブ」と記述されるとき、フィルタ／ツールは「有効」または「使用可能」または「使用される」であり、「非アクティブ」と記述されるとき、フィルタ／ツールは「無効」または「使用不可」または「使用されない」であり；「クラス」は1以上の要素のグループ、グループ化、カテゴリ、または分類を意味する。さらに、フラグが「アクティブ」と記述されている場合、そのフラグは関連するフィルタ／ツールが「アクティブ」であることを示すことを意味することも理解される。

【0063】

適応ループフィルタ(ALF)

図1は、VTM-5.0の典型的な復号化ループにおいて、ALFが行われる場所を示している。101では、画像部分(例えばスライス)が符号化ツリーユニット(CTU：VVUの最大の符号化ユニットで、通常128x128サンプル／画素サイズ)単位で復号化される。CTUは、特定の予測スキーム／モードを使用して符号化される矩形ブロック、または符号化ユニット(CU)に分割され、しばしば、残差ブロックのロシー符号化も行われる。ブロックベースの符号化の使用により、符号化されたブロック間の境界でブロッキングアーチファクトが見えることがある。102では、復号化された画像部分は、DBFによって処理され、これらのアーティファクトを低減／除去する。一般に、ブロック予測のための残差(ブロック)を符号化するために、残差値は(残差エネルギーを少数の係数に圧縮するために)DCTのような変換を用いて変換され、変換された係数は符号化コストを低減するために量子化される。この量子化は、再構成されたブロック(すなわち、フレームバッファ106に格納された参照フレーム内のブロック)において、しばしばいくつかのリングングアーチファクトを導入する。103において、DBFの出力画像部分は、次に、低い計算コストでこれらのアーティファクトのいくつかを低減するのに有用であるSAOフィルタによって処理される。104において、SAOフィルタの出力画像部分は、次に、ALFによって処理される。ALFは、例えば「リングング」のようなアーティファクトをさらに低減することができる。ALFは、より高次のエラーモデリング能力を有するが、より高い計算コストである。次に、ALFの出力画像部分は、出力

(例えば、ディスプレイまたはディスプレイと通信するための通信インタフェース) 105に送られる。また、(時間予測ツールが使用される場合) 時間予測に使用できるように、フレームバッファ106に(そこに格納される参照フレームの一部として) 入れることができる。これが、DBF、SAOフィルタ、ALFが「インループ」フィルタと呼ばれる理由である。符号化器は、復号化時にそれらがバイパスされるように(すなわち、フィルタが実行されず、無効にされたツールに対応するステップの出力はその入力と同じである)、インループフィルタの一部を無効化することができる。また、場合によっては、処理される画像部分はスライスに限定されず、1つまたは複数のスライスを含むフルフレームであってもよく、スライス境界(複数存在する場合)にわたってフィルタを適用して、それらの境界上のアーティファクトを低減する可能性がある。複数成分の画像(例えばYCrCbフォーマットの画像)の場合、DBF、SAOフィルタ、ALF処理は各成分に個別に、場合によっては異なる方法で(例えば他の成分と異なるフィルタリングパラメータを使用して)適用される。

10

【0064】

図2は、VTM-5.0におけるALFフィルタの概要を示している。各チャンネル(ルマチャンネル: Y、クロマチャンネル: Cb、Cr)について、入力画像部分201、202、203は、フィルタパラメータ204、205、206とともに、適応ループフィルタ(ALF)処理207、208、209に供給される。ALFプロセスは、各チャンネル210、211、212のための処理された出力画像部分を生成する。208、209におけるALFクロマ処理は、両方のクロマチャンネルに対して同じであるが、205、206において提供される特定のパラメータにより、各チャンネルに対して異なるように動作するように構成され得る。207で行われるALFルマ処理は、208で行われるALFクロマ処理よりも複雑であることが多い。これには2つの理由がある。第1の理由は、ルマチャンネルが、しばしばより滑らかであるクロマチャンネルよりも多くの構造(輪郭、テクスチャなど)を含むので、ルマチャンネルのコンテンツは、しばしばクロマチャンネルよりも処理/符号化するのがより複雑であることである。第2の理由は、人間の目はクロマチャンネルのエラーよりもルマチャンネルのエラーに敏感であることである。したがって、より高度で複雑な(すなわち、より多くの電力を消費するおよび/またはより遅い)処理は、より良い符号化利得およびより良い視覚品質を提供するために、ルマチャンネルに対して好ましいものである。

20

30

【0065】

204、205、及び206で提供される入力パラメータ情報は、適応パラメータセット(APS)及び/又はスライスヘッダで伝送される。また、204、205、及び206は、いくつかの共通情報を共有することができる。

【0066】

ALF処理207、208、209は、それぞれのタスク結果が他のもののタスク結果に依存しないため、各チャンネルで並列に実行することが可能である。このような並列処理の機会は、処理にかかる総時間を減少させることができ、ハードウェアまたはソフトウェアのいずれにおいても効率的に実施することが可能である。

【0067】

204、205、206で提供される入力パラメータ情報は、内部ビット深度情報を含むこともできる。内部ビット深度は、サンプル値を表現するために使用できる最大ビット数である(すなわち、サンプル値の精度を表す)。VTM-5.0では、ルマチャンネルに対して、クロマチャンネルに対する内部ビット深度とは異なる内部ビット深度を使用することが許可されている(ビット深度情報は、ルマとクロマの両方のチャンネルタイプのシーケンスパラメータセット(SPS)においてシグナリングされる)。以下の説明では、「IBDL」はルマの内部ビット深度、「IBDC」はクロマの内部ビット深度を意味する。内部ビット深度に「IBD」を使用する場合、IBDLはIBDCと同等と見なすことを意味する。記載がない場合、または指定がない場合は、内部ビット深度が議論されている処理に影響しないか、または内部ビット深度がオーソドックスに扱われるため省略されて

40

50

いることを示す。

【0068】

図3は、図2のALFフィルタを拡張し、クロスコンポーネントALF（CCALF）処理を使用するようにしたものである。

【0069】

各クロマチャネル202、203について、ルマチャネル201の入力画像部分は、フィルタパラメータ305、306とともに、クロスコンポーネント適応ループフィルタ処理（CCALF）308、309に提供される。CCALFフィルタ処理の出力は、311、312において、ALFクロマフィルタ処理208、209の出力に加えられ、各クロマチャネルについて、処理済出力画像部分313、314が生成される。308、309におけるCCALF処理は、両方のクロマチャネルに対して同じであるが、305及び306において提供される特定のパラメータにより、各チャネルに対して異なって動作するように構成され得る。例えば、各クロマチャネルは、特定のフィルタ係数を用いて処理されてもよい。

10

【0070】

いくつかの変形例では、クロマチャネルは、例えばYUV4:2:0色表現を使用するビデオのために、ルマチャネルと比較してサブサンプリングされる。そのような変形例では、CCALFフィルタ処理308、309は、各ルマサンプル位置には適用されず、クロマサンプル位置をアップサンプリングするときに得られる整数のルマサンプル位置のみに適用される。したがって、YUV4:2:0表現では、フィルタはルマサンプル位置の1/4にのみ適用される。

20

【0071】

ALFフィルタ処理207、208、209と、CCALFフィルタ処理308、309は独立しているので、並行して行うことが可能である。図2と比較すると、図3では、各チャネルの2つのクロマフィルタの結果を結合するために311と312で実行される加算演算によって、レイテンシが導入されている。しかしながら、CCALF及びクロマALFは一般にルマALFよりも複雑ではないので、VTM-5.0では、追加のレイテンシなしに3チャネルの全体処理を実現することが可能であるはずである（すなわち、レート決定ステップはルマALFであると考えられ、そのため他の2チャネルにおける追加のレイテンシは全体処理のレイテンシに影響を及ぼさない）。

30

【0072】

305及び306で提供される入力パラメータ情報は、適応パラメータセット（APS）及び／又はスライスヘッダで伝送される。305、および306はまた、いくつかの共通の情報を共有することができる。

【0073】

ある変形例では、305、306は線形畳み込みフィルタのフィルタ係数を含む（例えば図4参照）。1つの変形例では、305及び306は、例えば、非線形ALFフィルタのためにVTM-5.0で使用されるものと同様の非線形フィルタのためのフィルタ係数及びクリッピングパラメータを含む。線形及び非線形フィルタの両方が符号化性能の利得を提供することが見出されているが、線形フィルタはより複雑ではなく、そのようなものとして特定の実施において好まれるかもしれない。

40

【0074】

いくつかの変形例では、305および306で提供される入力パラメータ情報は、チャネル間のデータ（例えばサンプル値）を「正規化」する（すなわち、同等の精度値を使用する）ためのいくつかの左または右ビットシフト演算パラメータ（または乗除演算パラメータ）を推論するためのIBDLおよびIBDCを含んでもよい。

【0075】

いくつかの変形例では、ALFフィルタが実行されていない場合でも、CCALFを適用できることが理解される。また、ルマチャネルまたは他の種類のチャネルは、クロマチャネル以外の種類のチャネルを補正するために使用され得ることが理解される。例えば、

50

赤チャネルは、RGB色表現を使用するビデオにおいて、緑チャネル及び／又は青チャネルを補正するために使用され得る。

【0076】

また、いくつかの変形例では、別のチャネルのエラーを低減するために2つ以上のチャネルが使用されることが理解される。例えば、クロマチャネル202及び203の両方の入力画像部分は、両方ともCCALFフィルタでフィルタされ、各画像部分のフィルタの結果は、ルマチャネルの残りのエラーを修正するためにALFルマフィルタ207の出力に加えられる可能性がある。このような変形例は、クロマチャネルがルマチャネルと比較してサブサンプリングされる場合に特に適用され得る。例えば、YUV4:2:0色表現において、4つの異なるフィルタがクロマサンプルに使用され得、1つのクロマサンプル位置にダウンサンプルされ／関連付けられる4つのルマサンプル位置の各々に対して1つのフィルタが使用され得る。言い換えれば、2x2ブロックのルマサンプルに対して、2x2ブロックの位置(0,0)のルマサンプルを補正するために対応するクロマ位置に1番目のフィルタが適用され、それぞれ位置(0,1)、(1,0)、(1,1)のルマサンプルを補正するために同じクロマサンプル位置に対して2番目、3番目、4番目のフィルタが適用される。変形例では、4つのルマサンプル位置に対して1つのCCALFフィルタのみが使用されるが、フィルタされたクロマサンプル値は、ルマサンプルと同じグリッド上に整列したクロマサンプルを得るために、クロマサンプルをアップサンプリングすることによって得られる。

【0077】

図4-aは、線形畳み込みフィルタでCCALFを行う際に使用できるフィルタ形状401の一例である。このフィルタ形状は、YUV4:2:0色表現を使用するビデオのクロマチャネルのエラーを修正するためにルマチャネルをフィルタするのに適しているが、他の種類の色表現にも使用することができる。各ボックスは、参照サンプル位置（背景が灰色ドットのボックス）の周囲に配置された複数のサンプルに対応している。各ボックス内の数値は、ボックスの位置のサンプルに適用するフィルタ係数のインデックスに対応する。

【0078】

位置 (x_c, y_c) のCb成分クロマサンプルは、その値に以下の量を追加することでフィルタ／補正される。

【0079】

【数1】

$$\begin{aligned}
 O_{(x_c, y_c)}^{Cb} = & f_0^{Cb} \cdot I_{(x, y-2)} \\
 & + f_1^{Cb} \cdot I_{(x-1, y-1)} + f_2^{Cb} \cdot I_{(x, y-1)} + f_3^{Cb} \cdot I_{(x+1, y-1)} \\
 & + f_4^{Cb} \cdot I_{(x-2, y)} + f_5^{Cb} \cdot I_{(x-1, y)} + f_6^{Cb} \cdot I_{(x, y)} + f_7^{Cb} \cdot I_{(x+1, y)} + f_8^{Cb} \cdot I_{(x+2, y)} \\
 & + f_9^{Cb} \cdot I_{(x-2, y+1)} + f_{10}^{Cb} \cdot I_{(x-1, y+1)} + f_{11}^{Cb} \cdot I_{(x, y+1)} + f_{12}^{Cb} \cdot I_{(x+1, y+1)} + \\
 & f_{13}^{Cb} \cdot I_{(x+2, y+1)} \\
 & + f_{14}^{Cb} \cdot I_{(x-1, y+2)} + f_{15}^{Cb} \cdot I_{(x, y+2)} + f_{16}^{Cb} \cdot I_{(x+1, y+2)} \\
 & + f_{17}^{Cb} \cdot I_{(x, y+3)}
 \end{aligned}
 \tag{1}$$

【0080】

ここで、 f^{Cb}_i は、Cbクロマ成分のフィルタのインデックスiのフィルタ係数である。

【0081】

Cr成分についても同様のフィルタが行われるが、 f^{Cr}_i フィルタ係数が使用される。

【0082】

同様のフィルタを行うために、以下の式表記を用い、表記を簡略化する。

10

20

30

40

50

【 0 0 8 3 】

【 数 2 】

$$O_{(x_c, y_c)}^{Cb} = \sum_i f_{(u_i, v_i)}^{Cb} \cdot I_{(x+u_i, y+v_i)} \quad (2)$$

【 0 0 8 4 】

ここで、 $f_{(u_i, v_i)}^{Cb}$ は、位置 (x, y) を持つ参照サンプル（401の灰色ドット背景のボックス）のオフセット (u_i, v_i) でフィルタ形状内のサンプル $I_{(x+u_i, y+v_i)}$ で適用するフィルタ係数を示す。

【 0 0 8 5 】

式（2）の実施は、通常、固定小数点演算を用いた実数演算の近似のための整数演算を用いて行われる。そして、式（2）は次のように書き換えることができる。

【 0 0 8 6 】

【 数 3 】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot I_{(x+u_i, y+v_i)} \right) + (1 \ll (B-1)) \right) \gg B \quad (3)$$

【 0 0 8 7 】

ここで、 $\ll$ は整数左ビットシフト演算、 $\gg$ は整数右ビットシフト演算、 B はフィルタ係数 $f_{(u_i, v_i)}^{Cb}$ （図4-b参照）の固定小数点表現の小数精度（通常10ビット）、 $I_{(x+u_i, y+v_i)}$ はコーデックのIBDと同じ整数精度（通常10ビット、図4-c参照）を有する整数サンプル値である。

【 0 0 8 8 】

YUV4:2:0色表現では、ルマサンプル位置 (x, y) は、アップサンプリングされたクロマ位置 $(x, y) = (2 \cdot x_c, 2 \cdot y_c)$ に最も近い整数値に対応する。

【 0 0 8 9 】

いくつかの変形例では、IBDLがIBDCより大きいとき、式（3）は次のように書き換えられ得る。

【 0 0 9 0 】

【 数 4 】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg (IBDL - IBDC)) \right) + (1 \ll (B-1)) \right) \gg B \quad (4)$$

【 0 0 9 1 】

または、

【 0 0 9 2 】

【 数 5 】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i \left(f_{(u_i, v_i)}^{Cb} \cdot I_{(x+u_i, y+v_i)} \right) \gg (IBDL - IBDC) \right) + (1 \ll (B-1)) \right) \gg B \quad (5)$$

【 0 0 9 3 】

または、

【 0 0 9 4 】

【 数 6 】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i \left(f_{(u_i, v_i)}^{Cb} \cdot I_{(x+u_i, y+v_i)} \right) \gg (IBDL - IBDC) \right) + (1 \ll (B-1)) \right) \gg B \quad (6)$$

【 0 0 9 5 】

であり、IBDLがIBDCより小さいとき、式(3)は次のように書き換えられ得る。

【0096】

【数7】

$$O_{(x_c, y_c)}^{Cb} = \left(\sum_i f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} \ll (IBDC - IBDL) + (1 \ll (B-1)) \right) \gg B \quad (7)$$

【0097】

または、B+IBDLがIBDCより厳密に大きいとき、等価的に：

【0098】

【数8】

$$O_{(x_c, y_c)}^{Cb} = \left(\sum_i f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} + (1 \ll (B1-1)) \right) \gg B1 \quad (8)$$

【0099】

であり、B1=B+IBDL-IBDCである。

【0100】

式(4)、(5)、(6)、(7)のビット単位のシフト演算、または式(8)のビット単位のシフトパラメータの変更により、フィルタの結果を正規化してクロマのサンプル値と同じ精度を維持することができる。そのために、式(4)ではルマサンプル値を正規化し、式(5)ではルマサンプル値と係数値との乗算結果を正規化し、または式(6)、(7)、(8)では乗算出力の和の結果（すなわちフィルタの結果）を正規化する。

【0101】

また、式(6)は、式(8)のように書き換えても同じ結果になる。

【0102】

式(7)では、IBDLがIBDCより小さい場合、どこで左シフトしても計算結果が同じになるため、意図的にビット単位のシフトを括弧で囲まないようにしている。これは、ハードウェアまたはソフトウェアの設計上の選択の問題である。また、(IBDL-IBDC)による右シフトを行う前に $1 \ll (IBDC - IBDL - 1)$ を加算して、式(4)のシフト値を丸めることも可能である。

【0103】

ハードウェア／ソフトウェア設計の観点からは、IBDLがIBDCよりも大きい場合、式(8)の定式化に従った実施が、より少ない演算で済むため、好ましいと考えられる。IBDLがIBDCよりも常に大きい特定のケースでは、乗算の入力のビット数を制限するために、式(4)が好ましいと考えられる。式(4)を用いるシステムの実施で得られるビット数は、そのシステムにおいてIBDCを取り得る最大値とIBDLを取り得る最大値によって決まる（処理されなければならないのは最悪のケースであるため）。IBDCがIBDLより大きい場合、特にIBDCを取り得る最大値がIBDLを取り得る最大値より常に大きい場合、乗算の入力ビット深度を増やさないために式(8)を使用することが望ましい場合がある。IBDCの最大値とIBDLの最大値が等しい場合、式(8)は最良の妥協点である。

【0104】

B+IBDLが厳密にはIBDCより大きくない場合、式(8)は次のように書き換えることができる。

【0105】

【数9】

$$O_{(x_c, y_c)}^{Cb} = \left(\sum_i f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} \right) \ll B2 \quad (9)$$

【0106】

ここで、 $B2 = IBDC - IBDL - B$ である。

【0107】

図4-bは、式(3)におけるCCALFフィルタ係数の典型的なビット単位のメモリ表現である。CCALFフィルタ係数のビット表現402は、典型的には、実数の固定小数点値近似を格納するために、16ビットの符号付き整数表現(tow表現の補数)を使用する。10ビット精度は小数部に使用され、他のビットは整数部と符号を保存するために使用される。

【0108】

図4-cは、VTM-5.0の内部ビット深度10ビットの式(3)のサンプル値を16ビットの整数メモリ空間403(またはレジスタ)に格納した場合の典型的なビット単位のメモリ表現である。クロスしたボックスは使用されないビットを表している。

10

【0109】

図4-dは、CCALFフィルタ係数表現が図4-bで使用されたものに対応するとき、10ビット内部ビット深度のサンプル値とCCALFフィルタ係数との乗算の結果(例えば所与のiに対して $f^{Cb}(u_i, v_i) \cdot I(x+u_i, y+v_i)$)を格納するための32ビット符号付き整数の典型的なビット単位のメモリ表現を示している。見てわかるように、24ビット精度(403の10ビット精度と402の14ビット精度)と符号のための1ビットが必要である(すなわち、25ビットの符号付き整数表現)。

【0110】

図4-eは、CCALFフィルタ係数表現が図4-bで使用されたものに対応し、図4-aのフィルタパターンを使用する場合の、10ビット内部ビット深度サンプル値のCCALFフィルタ係数による乗算の結果の累積(和)(例えば $\sum_i f^{Cb}(u_i, v_i) \cdot I(x+u_i, y+v_i)$)の中間結果を格納するための32ビット符号付き整数の典型的なビット単位のメモリ表現を示している。これは、図4-bのように、18個の25ビット符号付き整数の和を格納するために必要なビット単位のメモリに相当する。見てわかるように、29ビット精度(403の10ビット精度+402の14ビット精度+18個の加算の最悪の場合の結果を保存するための5ビット)と符号のための1ビットが必要である(すなわち、30ビットの符号付き整数表現)。

20

【0111】

本発明の実施形態によれば、CCALFフィルタのための式(3)のフィルタ演算は、特に、論理ゲート数がコストを有し、乗算を行うためのゲート数が入力の最大ビット数に依存するハードウェア専用設計において、図4-bの典型的な係数表現を用いる場合に比べ、非常に簡素化されている。設計の大幅な簡略化を伴う場合、符号化効率に影響を与えながらも、大幅な簡略化を達成できることが意外に知られている。一般に、以下に開示される簡略化は、フィルタ係数($f^{Cb}(u_i, v_i)$)またはクロスコンポーネント入力サンプル($I(x+u_i, y+v_i)$)のいずれかまたは両方の(内部ビット深度と比較して)低減されたビット深度表現を使用することに関連する。低減されたビット深度表現は、第1成分サンプルの内部ビット深度と比較して制限され、または典型的なフィルタ係数のビット深度と比較して制限される(両方の一般的な例は10ビットである)。

30

【0112】

ビット深度の低減の特定の組み合わせは、(例えば、16ビットアーキテクチャで実行できる計算につながる場合に)特に有利であることを理解されたい。いずれにせよ、制限されたビット数を使用する計算は高速であり、(単なるALFと比較して)CCALFの追加コスト(時間または計算力の増加のいずれか)は低いことを意味する。

40

【0113】

式(3)の乗算入力のビット数を減らす第1の変形例は、固定小数点小数精度を下げることである。これは、図4-bの最下位ビットのいくつかを削除することで可能である。例えば、10ビットの代わりに7ビットを使用することは、CCALFの符号化利得をあまり減らさないように、(例えば10ビットのサンプル内部ビット深度を使用する場合)良い妥協点を示すことが分かっている。いくつかの変形例では、10ビットの代わりに6

50

ビットを使用すると、CCALFの符号化利得が少し減少するが、それでも興味深いハードウェア設計の妥協点となっている。

【0114】

図5-aは、固定小数点小数精度を10ビットではなく7ビットに制限した場合の、式(3)のCCALFフィルタ係数のビット単位のメモリ表現の一例を示している。そして、402のビット単位の表現は、501において13ビット(精度12ビット、符号1ビット)に削減される。

【0115】

式(3)の乗算入力のビット数を減らすための第2の変形例は、フィルタ係数の取り得る値の範囲を制限することである。これは、図4-bに示す最上位ビットのいくつかを削除することによって達成することができる。驚くべきことに、係数の取り得る値の範囲を大幅に制限しても、CCALFは依然として効率的に動作することが分かっている。この効果はCCALFに特有のものであり、例えばALFでは再現されないため、特に驚くべきことである。このような単純化の符号化効率に対する驚くほど低い影響は、本願発明者らが、クロスコンポーネントALFは、ルマサンプルの勾配が強いところ(例えば、エッジの近く)で最も有用であり、その場所でのクロマエラーは一般に強く、ルマサンプルの勾配と相関していることを発見したからであると説明される。さらに、そのような場所でのCCALF出力は一般的に小さい/制限された値である(一方、他の場所ではほとんどゼロである)。つまり、数学的に最適なフィルタ係数(実数領域)は一般に振幅が小さく、少ないビット数の整数表現(固定小数点小数)で十分に近似できる。そのため、符号化性能の向上の大部分は、複数の係数のサブセットから達成することができる。本発明の実施形態は、この性質を利用して、符号化性能に大きな影響を与えることなく、計算を簡略化することができる。これは、以下に示す第3及び第4の変形例が機能する理由も説明する: サンプル/係数の精度を下げると、CCALF出力の小さな値の推定に大きな影響を与えない一種の「丸め誤差ノイズ」を生成する。

【0116】

まとめると、以下のことが判明した。

- i) 係数を $-0.03125 \sim 0.03125$ の範囲に制限する(例えば、小数点以下6番目以上のビットを使用しない)ことは、許容できる符号化インパクトである。
- ii) 係数を $-0.0625 \sim 0.0625$ の範囲に制限する(例えば、小数点以下5番目以上のビットを使用しない)ことは、符号化に大きな影響を与えない。
- iii) 係数を $-0.125 \sim 0.125$ の範囲に制限する(例えば、小数点以下4番目以上のビットを使用しない)ことは、符号化にほとんど影響を与えない。
- iv) 係数を $-0.25 \sim 0.25$ の範囲に制限する(例えば、小数点以下3番目以上のビットを使用しない)ことによる符号化への影響はごくわずかです。
- v) 係数を $-0.015625 \sim 0.015625$ の範囲に制限する(例えば、小数点以下7番目より高いビットを使用しない)ことは、特定の実施において許容され得る符号化インパクトを有する。

【0117】

範囲の限界値は、使用可能な係数値に含めても含めなくてもよい。これら3つの範囲を使用した例を以下に詳述するが、代替の範囲を使用することも可能である。

【0118】

以下の説明では、結果を提供する。これらの結果は、JVET-N01010文書に定義された「SDRビデオのJVET共通試験条件およびソフトウェア参照構成」を用いて得られたものであるが、試験条件をビデオシーケンスの17個の最初のフレームに適用するように制限している。このように、VTM-5.0ソフトウェア(またはVTM-6.0ソフトウェアを提供するための開発過程での中間開発バージョン)において、本発明の変形例が実施されている。ルマチャネルと両クロマチャネルについて、“全イントラ”(AI)、“ランダムアクセス”(RA)、“低遅延B”(LB)、“低遅延P”(LP)構成の平均BDレート符号化利得結果を参照(例えば、VTM-5.0)と比較して求めている。

これらの平均チャネル符号化利得は、各クロマチャネルの符号化利得にルマ符号化利得の1.4倍を加算し、合計を1.6で割って平均YUV符号化利得を得ることによって結合される。これは、クロマチャネルがルマチャネルよりも符号化しやすいため、ルマチャネルとクロマチャネルの結果を組み合わせる従来の方法である。AI、RA、LB、LP構成で得られた平均YUV符号化利得は一緒に平均化され、これらの最終平均符号化利得値は符号化利得比較を提供するために一緒に比較される。符号化利得の比較は、（明確に異なることが指定されていない場合）JVET-00636 寄書で規定された、CCALFを実施した参照ソフトウェア（すなわち、VTM-5.0またはそれ以降の中間開発版）であるが、それを簡単にするための発明は含まれない参照に対するパーセントとして表現される。比較された変形例は、その参照を簡略化して実施されている（すなわち、同じVTM-5.0またはそれ以降の中間開発バージョンを使用し、本発明で簡略化されたCCALFが実施されている）。パーセンテージは、変形例の符号化利得（これは、同じ品質に対するレート低減のパーセンテージで表される平均的なBDレート利得である）と、参照の符号化利得（これも平均的なBDレート利得）との間の比率を計算することによって得られる。なお、VTM-5.0における参照のCCALFを使用すると、VTM-5.0と比較して約-1.2%の平均YUV符号化利得が得られる。

10

【0119】

以下に示す符号化利得の比較結果は参考であり、別の符号化器の実施において本発明を実施する場合、または別のコーデック仕様において本発明を実施する場合、報告された符号化利得の割合は異なる可能性がある。

20

【0120】

以下の説明において、範囲表記は以下の意味で使用される。

-] x , y [は、x と y の間の値の範囲を意味し、x も y も含まない。
- [x , y [は、x と y の間の値の範囲を意味し、x を含み y を含まない。
-] x , y] は、x と y の間の値の範囲を意味し、y を含み x を含まない。
- [x , y] は、x と y の間の値の範囲を意味し、x と y を含む。

【0121】

[0.03125 ; 0.03125 [の範囲の係数を使う（つまり、固定小数点小数精度の6番目より上のビットを使わない）と、CCALFの大部分の符号化利得を維持し、乗算に使うビット数を大幅に削減することができる。この変形例の一例を図5-bに示す。

30

【0122】

図5-bは、10ビットの固定小数点小数精度を使用し、係数の可能な値の範囲を範囲[-0.03125 ; 0.03125 [に制限する場合の、式(3)のCCALFフィルタ係数のビット単位のメモリ表現の一例を示している。そして、402のビット表現は、502で6ビット（精度5ビット、符号1ビット）に削減される。

【0123】

第2の変形例（図5-b）は、図5-cに示されるように、第1の変形例（図5-a）に組み合わせることができる。

【0124】

図5-cは、固定小数点小数精度を10ビットではなく7ビットに制限し、係数の可能な値の範囲を範囲[-0.03125 ; 0.03125 [に制限した場合の、式(3)のCCALFフィルタ係数のビット単位のメモリ表現の一例を示している。そして、402のビット表現は、503で3ビット（精度2ビット、符号1ビット）に削減される。

40

【0125】

さらに、係数を]-0.03125 ; 0.03125 [の範囲に制限する（すなわち、フィルタ係数は値の集合{-0.0234375, -0.015625, -0.0078125, 0, 0.0078125, 0.015625, 0.0234375}の値のみを取ることができる）ことにより、もう少し単純化することができ、各乗算を2つの加算／減算（符号処理による）およびビット単位のシフト（図5-cに示すケース）で置き換えるこ

50

とが可能である。例えば、式（３）の乗算は、例えば次のように置き換えることができる。

【０１２６】

【数１０】

$$0_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i \left(\begin{cases} I_{(x+u_i, y+v_i)} \text{ if } f_{(u_i, v_i)}^{Cb} \in \{0.015625, 0.0234375\}, \\ -I_{(x+u_i, y+v_i)} \text{ if } f_{(u_i, v_i)}^{Cb} \in \{-0.015625, -0.0234375\}, \\ 0 \text{ else.} \end{cases} + \begin{cases} I_{(x+u_i, y+v_i)} \ll 1 \text{ if } f_{(u_i, v_i)}^{Cb} \in \{0.0078125, 0.0234375\}, \\ -I_{(x+u_i, y+v_i)} \ll 1 \text{ if } f_{(u_i, v_i)}^{Cb} \in \{-0.0078125, -0.0234375\}, \\ 0 \text{ else.} \end{cases} \right) \right) \gg B \quad (10)$$

10

【０１２７】

ここで、 $\ll$ は整数左ビットシフト演算、 $\gg$ は整数右ビットシフト演算、 B はフィルタ係数（ $f_{(u_i, v_i)}^{Cb}$ ）（図５－ｃ参照）の固定小数点表現の小数精度（７ビット）、 $I_{(x+u_i, y+v_i)}$ はコーデックの内部ビット深度（通常１０ビット、図４－ｃ参照）と同じ整数精度を有する整数サンプル値である。

【０１２８】

乗算は、１つの加算／減算（符号処理に依存）とビット単位のシフトだけで置き換えることもできる。しかし、この場合、 $CCALF$ の符号化利得にさらなる損失が発生する。例えば、固定小数点の精度を６ビットに制限し、係数を $[-0.03125; 0.03125]$ の範囲に制限する場合（すなわち、フィルタ係数は値の集合 $\{-0.015625, 0, 0.015625\}$ の値のみを取ることができる）実行可能である。また、例えば、固定小数点の精度を７ビットに制限し、係数を $[-0.015625; 0.015625]$ の範囲に制限する場合（すなわち、フィルタ係数は値の集合 $\{-0.0078125, 0, 0.0078125\}$ の値のみを取ることができる）も実行可能である。第１の変形例を用いると、１０ビット小数精度を用い、 $[-1; 1]$ の範囲のフィルタ係数を用いた場合と比較して、約７８％の $CCALF$ 利得を維持することができる。第２の変形例を用いると、 $CCALF$ の利得を約７６％に抑えることができる。

20

30

【０１２９】

他のいくつかの変形例では、係数の値の範囲は、 $[-0.0625; 0.0625]$ 、 $[-0.0625; 0.0625]$ 、 $[0.125; 0.125]$ 、 $[0.125; 0.125]$ 、 $[-0.25; 0.25]$ 、または、 $[-0.25; 0.25]$ である。すなわち、１、２、３個の追加の最上位ビットは係数の（小数）精度について保持されている。例えば、図５－ｃの変形例として、フィルタ係数のために２ビットの代わりに３または４ビットが保持される。このような変形例では、追加ビットは図５－ｃで保持されるビットよりも１または２高いビットである。この４つの変形例の２番目と４番目は、それぞれ３および４の加算／減算（符号処理による）とビット単位のシフトによって乗算を置換することができる。

40

【０１３０】

式（３）の乗算入力のビット数を減らす３つ目の変形例は、サンプル値の精度を下げることである。これは、サンプル値の最下位ビットを考慮しないことで実現できる（これは、 CPU 実施を使用した右へのビット単位のシフトを使用することと同等である）。例えば、式（３）は次のように書き換えることができる。

【０１３１】

【数１１】

$$0_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg DB) \right) + (1 \ll (B - DB - 1)) \right) \gg B - DB \quad (11)$$

50

【0132】

ここで、DBはデッドビット（無視される／考慮されないビット）の数に相当する。本変形例の乗算に用いられるサンプル値 $I(x+u_i, y+v_i) \gg DB$ のビット表現の一例を図5-dに示す。

【0133】

図5-dは、VTM-5.0の内部ビット深度が10ビットの場合に、式(3)の乗算演算のハードウェア実施を簡単にするためにサンプル値の精度を10ビットから6ビットに制限した場合の式(3)で使用するサンプル値のビット単位のメモリ表現の一例を示したものである。クロスしたボックスは使用されないビットを表す。403のビット表現が504で6ビットに削減される。

【0134】

変形例としては、例えば、除去された最上位ビットの値を用いてサンプル値を（例えば、除去された最上位ビットが1である場合、ビット数が削減された表現に1を加えることによって、または等価的に、不要なビットを除去／考慮しない前に除去された最上位ビットの1に対応する値を加えることによって）「丸める」ことが挙げられる。サンプル精度を下げるのがCCALFの符号化効率に大きな影響を与える場合、このような変形例は興味深いものであるが、最下位ビットを無視するだけの場合と比較して、追加の演算やメモリの使用を意味することにもなる。

【0135】

CCALFによって改善されたVTM-5.0では、フィルタを行う際にサンプル値の上位6ビットのみを保持しても、符号化効率に大きな影響を与えないことが分かっている。例えば、内部ビット深度が10ビットの場合、サンプル値の下位4ビットは無視でき、CCALFの効率に与える影響は小さくなる。サンプル値の最上位6ビットより少ないビットを保持することは可能であるが、CCALFの符号化効率は低下する。例えば、フィルタ処理を行う際にサンプル値の最上位4ビットを保持する（つまり、10ビットサンプルの最下位6ビットを削除する）と、驚くほど興味深い利得が得られる（平均して、10ビットサンプルの全サンプル精度を使用した場合と比較して、約88%のCCALF利得が維持される）。一方、最上位5ビットを維持すると、完全なサンプル精度を使用した場合と比較して、約95%のCCALF利得が維持される。最上位6ビットの場合、約98%の利得が保持される。最上位ビットを6つ以上保持しても、結果はわずかに増加するだけである。例えば、7または8個の最上位ビットを保持する（すなわち、10ビットの内部ビット深度の下位3または2ビットを削除する）ことで、完全なサンプル精度のCCALF利得の99%程度の結果が得られる。

【0136】

第4の変形例として、式(2)で表されるCCALFフィルタは、VTM-5.0の非線形ALF(NLALF)のフィルタ定式化に近くなるように修正されている。

【0137】

【数12】

$$O_{(x,y)}^{cb} = f_{(0,0)}^{cb} \cdot I_{(x,y)} + \sum_{i>0} f_{(u_i,v_i)}^{cb} \cdot (I_{(x+u_i,y+v_i)} - I_{(x,y)}) \quad (12)$$

【0138】

ここで、 $f^{cb}(u_i, v_i)$ は依然としてフィルタ係数であるが、参照サンプルのオフセット (u_i, v_i) におけるフィルタ形状のサンプル $I(x+u_i, y+v_i)$ と参照サンプル $I(x, y)$ の値の差分を表す数値を乗じて表現している。（401の灰色ドット背景のボックス）は、位置が (x, y) である。差分値を表す図形は、参照サンプルに隣接するサンプルに対する差分に相当するグラデーションであってもよい。グラデーションを用いると、参照サンプルに隣接するサンプルと隣接しないサンプルの両方に対する差分の正規化が行われることを表すことになる。

【0139】

式(12)は、式(2)が式(3)で整数演算で実施されたのと同様に、整数演算で実施することができる。

【0140】

【数13】

$$O_{(x_c, y_c)}^{Cb} = \left(f_{(0,0)}^{Cb} \cdot I_{(x,y)} + \left(\sum_{i>0} f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} - I_{(x,y)}) \right) \right) \gg B \quad (13)$$

+(1 << (B-1))

【0141】

ここで、<<は整数左ビットシフト演算、>>は整数右ビットシフト演算、Bはフィルタ係数 $f^{Cb}(u_i, v_i)$ の固定小数点表現の小数精度（通常10ビット）であり（図4-b参照）、 $I(x+u_i, y+v_i)$ と $I(x, y)$ はコーデックの内部ビット深度（通常10ビット、図4-c参照）と同じ整数精度での整数サンプル値である。この定式化を使用する場合、 $I(x+u_i, y+v_i)$ のビット表現と比較し、 $(I(x+u_i, y+v_i) - I(x, y))$ のビット表現は、符号を示す1ビットが追加されることになる。これを図6-aに示す。いくつかの代替例では、 $f^{Cb}(0, 0)$ は常に0に等しく、ビットストリームには符号化されない。他の代替例では $f^{Cb}(0, 0) \cdot I(x, y)$ は必ずしも0にならず、この乗算は参照サンプルの差分を使用しない他の乗算と同様に処理される。

【0142】

図6-aは、VTM-5.0の10ビット内部ビット深度の式(13)におけるサンプル値の差分 $(I(x+u_i, y+v_i) - I(x, y))$ が16ビット整数メモリ空間601（またはレジスタ）に格納された場合のビット単位のメモリ表示を示す図である。クロスしたボックスは使用されないビットを表す。

【0143】

ハードウェア設計の観点からは、参照サンプルとの差は、ルマサンプルのNLALFフィルタのために既に存在／計算されているので、コストはかからない（CCALFフィルタのフィルタ形状は、NLALFのルマ用フィルタの形状に含まれる）。

【0144】

式(3)のフィルタを使用した場合のCCALFの符号化利得は、式(13)のフィルタを使用した場合と同等である。

【0145】

この表現の利点は、乗算の入力のビット数を減らすために、より少ないビット数で表現できる範囲にとどまるように差分値をクリップすることができることである。

【0146】

【数14】

$$O_{(x_c, y_c)}^{Cb} = \left(f_{(0,0)}^{Cb} \cdot I_{(x,y)} + \left(\sum_{i>0} f_{(u_i, v_i)}^{Cb} \cdot K(I_{(x+u_i, y+v_i)} - I_{(x,y)}, C) \right) \right) \gg B \quad (14)$$

+(1 << (B-1))

【0147】

ここで、Kは例えばクリッピング関数である。

【0148】

【数15】

$$K(d, C) = \max(-C, \min(C, d)) \quad (15)$$

【0149】

またはそれに準じて、 $K(d, C) = \max(C, \min(-C, d))$ であり、ここでCはクリッピングパラメータ、dは差分を表す数値である。この式は、クリッピングパラメータCが各サンプル位置で同じであり、その値は固定で、乗算の入力である $K(I(x+u_i, y+v_i) - I(x, y), C)$ を格納するビット数を制限するために選ばれるの

に対し、非線形 A L F ではクリッピングパラメータ C が各サンプル位置で変化しその値がフィルタ係数と共に (A P S で) 送信される点で、V T M - 5 . 0 の非線形 A L F と特に異なることに注意する必要がある。

【0150】

C の特に有利な固定値は、 $2^{I B i t D e p t h} - 2 - 1$ または $2^{I B i t D e p t h} - 3 - 1$ であることが判明しており、 $I B i t D e p t h$ はコーデックの内部ビット深度 (すなわち、サンプル値を表すためのビット数) である。例えば、10 ビットの内部ビット深度では、C の良い値は 255 または 127 である。この例のクリッピング演算では、乗算の入力において 2 ~ 3 ビットを節約することができる。

【0151】

C の値を小さくすることも可能であるが、C C A L F の符号化効率を低下させる傾向があり、逆に値を大きくするとクリッピング処理による単純化の効果が減少する。そのため、クリッピングパラメータ C の一般化は $2^{I B i t D e p t h} - n - 1$ となり、 n は $I B i t D e p t h$ よりも小さい整数となる。 n が小さいと、符号化効率の低下は少ないが、簡略化に伴う利点も少なく、 n が大きいとその逆となる。

【0152】

図 6 - b は、内部ビット深度が 10 ビット、 $C = 255$ の場合の、式 (14) のクリップされた差分 $K(I(x + u_i, y + v_i) - I(x, y), C)$ を 16 ビット整数メモリ空間 602 (またはレジスタ) に記憶する場合のビット単位のメモリ表現例である。

【0153】

第 4 の変形例は、第 3 の変形例と組み合わせて、クリップされた差分値 $K(I(x + u_i, y + v_i) - I(x, y), C)$ の低精度ビットを無視できるようにすることが可能である。特に有利な組み合わせは、第 2 変形例と同じ数の下位ビット除去を維持することであり、C C A L F 符号化利得に大きな影響を与えないようにすることである。この例を図 6 - c で説明する。

【0154】

図 6 - c は、内部ビット深度 10 ビット、 $C = 255$ の場合の、下位 4 ビットを無視した $K(I(x + u_i, y + v_i) - I(x, y), C)$ を記憶するビット単位のメモリ表現の一例を示す図である。すると、乗算の入力では、 $K(I(x + u_i, y + v_i) - I(x, y), C)$ を表すのに 10 ビットではなく、残りの精度 4 ビットと符号用の 1 ビット 603 だけでよいことになる。これは、計算されるビット数の大幅な節約を意味し、より単純な実施に帰結する。

【0155】

いくつかの実施形態では、4 つの変形例の組み合わせが実行される。例えば、乗算演算の出力結果が 10 精度ビット + 1 符号ビットで表現できるように、変形例 1、2、3、および任意に 4 を組み合わせ、乗算結果の合計が最大 15 ビット + 1 符号ビットで表現できる (したがって、16 ビットレジスタまたはメモリ空間に表現可能 / 記憶可能) ようにすることが可能である。このような実施形態は、特にハードウェア実施に適している。図 7 - a、図 7 - b は、クロマサンプル値に結果を加算するための最終丸めおよびビットシフト前の、フィルタ演算の中間表現 (乗算出力) および最終表現 (合計結果) の一例を示している。

【0156】

図 7 - a は、サンプル値の上位 8 ビットのみを保持し (例えば図 5 - d のように)、係数の小数部分を 7 ビット精度で保持し、 $[-0.03125; 0.03125]$ [または] $[-0.03125; 0.03125]$ [の係数範囲 (例えば図 5 - c のように 2 ビット + 符号ビット) で $f^{C b}(u_i, v_i) \cdot I(x + u_i, y + v_i)$ を格納する場合のビット単位のメモリ表現例であり、これにより 10 ビット + 1 符号ビットで乗算結果を格納できることになる。

【0157】

ルマビット深度 $I B D L$ とクロマビット深度 $I B D C$ が異なる場合の式 (4) と、ルマ

10

20

30

40

50

サンプル値に対して保持する最高精度ビット数 KB を $DB = IBDL - KB$ とした式 (11) (この例では $KB = 8$) を用いると、式 (3) は次のように書き換えることができる。

【0158】

【数16】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg B3) \right) + (1 \ll (B4 - 1)) \right) \gg B4 \quad (16)$$

【0159】

かつ、 $B3 = DB + (IBDL - IBDL - KB) = 2$ 、 $IBDL - IBDL - KB$ 、 $B4 = BDP C - B3$ で、 $BDP C$ は係数の小数部のビット数 (例えば、この例では係数の小数部の精度を7ビットとして $BDP C = 7$)、 $f_{(u_i, v_i)}^{Cb}$ の係数の整数表現 (すなわち、この例では、 $[-0.03125; 0.03125]$ [または] $-0.03125; 0.03125$ [の範囲の固定小数値の整数値、この例では、整数値は $[-4; 3]$ または $[-3; 3]$ の範囲の値をとる) である。この式は、 $KB \leq 2 \cdot IBDL - IBDL \leq BDP C + KB - 1$ の場合に有効である。 $KB \leq 2 \cdot IBDL - IBDL$ かつ $2 \cdot IBDL - IBDL = BDP C + KB$ (すなわち $B4 = 0$) の場合、式 (16) は次のように置き換えることができる。

【0160】

【数17】

$$O_{(x_c, y_c)}^{Cb} = \sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg B3) \quad (17)$$

【0161】

$KB \leq 2 \cdot IBDL - IBDL$ かつ $2 \cdot IBDL - IBDL > BDP C + KB$ のとき、式 (16) は、以下のように置き換えることができる。

【0162】

【数18】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg B3) \right) \right) \ll (-B4) \quad (18)$$

【0163】

であり、 $2 \cdot IBDL - IBDL < KB$ のとき、式 (16) は、次のように置き換えることができる。

【0164】

【数19】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \ll (-B3)) \right) + (1 \ll (B4 - 1)) \right) \gg B4 \quad (19)$$

【0165】

所与の例では、 $KB = 8$ で、10ビットのルマとクロマのビット深度 (すなわち、 $IBDL = IBDL = 10$) に対して、式 (16) は次のように書き換えることができる。

【0166】

【数20】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg 2) \right) + 16 \right) \gg 5 \quad (20)$$

【0167】

この結果、各係数は2ビットと符号ビットで表され、各サンプルは削減された8ビットと、オプションとして (実施／設計の選択に応じて) 符号ビット (0に等しい) で表され

10

20

30

40

50

、その結果、乗算結果は8ビットと符号で表すことができるのである。

【0168】

図7-aは、サンプル値の上位6ビットのみを保持し（例えば図5-dより上位2ビット）、係数の小数部の精度を7ビットとし、 $[-0.125; 0.125]$ [または] $[-0.125; 0.125]$ [の係数の範囲（すなわち、4ビット+符号ビットでこれらを格納するために例えば、図5-cより2ビット多い）で $f^{Cb}(u_i, v_i) \cdot I(x+u_i, y+v_i)$ を格納する場合のビット単位のメモリ表現例を提供するのに有効である。乗算結果は10ビット+1符号ビットに格納することができる。

【0169】

この例では、 $KB=6$ である。ルマとクロマのビット深度が10ビット（すなわち $IBDL=IBDC=10$ ）の場合、式(16)は次のように書くことができる。

【0170】

【数21】

$$O_{(x_c, y_c)}^{Cb} = \left(\left(\sum_i f_{(u_i, v_i)}^{Cb} \cdot (I_{(x+u_i, y+v_i)} \gg 4) \right) + 4 \right) \gg 3 \quad (21)$$

【0171】

図7-bは、図7-aで説明したように $f^{Cb}(u_i, v_i) \cdot I(x+u_i, y+v_i)$ で表現され計算される場合、和演算 $\sum_i f^{Cb}(u_i, v_i) \cdot I(x+u_i, y+v_i)$ を計算し保存するためのビット単位のメモリ表現の例を提供する。これは、フィルタが図4-aのように、18の乗算結果のうち17の異なる加算を行う場合であることを理解する必要がある。17回の加算のうち最悪の場合（固定小数点値を格納する10ビットの整数値を考慮した場合は、整数値 $-2^{10} = -1024$ の18倍（すなわち15ビットと符号ビットで格納できる -18432 ）となり、15ビットと符号ビットで加算結果を表現できることを意味する。

【0172】

さらなる（第5の）変形例では、図4-aのフィルタパターンを変更し、18サンプルの代わりに16サンプル以下がフィルタ処理に使用されるようにする。このため、和演算において1ビットを節約することができる。また、乗算演算がなくなるので、フィルタの計算が簡単になる利点もある。例えば、図4-bでは、29精度ビットと1符号ビットを使用する代わりに、28精度ビットと1符号ビットだけが使用され、図7-bでは、15精度ビットを使用する代わりに、14精度ビットだけが使用されることになる。その第5の変形例を用いたいくつかの変形例では、加算演算で節約されたビットは、乗算結果の精度を高めるために費やされる。例えば、図7-aでは、10ビットの代わりに11ビット精度が使われる。ある変形例では、10精度ビットのうち小数部を（7ビットから8ビットへ）1ビット増やす。これは、係数値の小数部に1つ低い精度のビットを追加することで実現することができる。例えば図5-cでは、一番右の使用済みのビットが再び使用されることになる。あるいは、第3の変形例が使用される場合、サンプル値に対してさらに1つの精度ビットを使用することによって達成することができる。例えば、図5-dでは、一番右のビットが再び使用されるであろう。第2の代替として、10精度ビットのうち整数部が（3ビットから4ビットへ）1ビット増加する。これは、係数値に1つ高い精度のビットを追加する（つまり、可能な係数値の範囲を広げる）ことで達成される。例えば図5-cでは、一番左の使用済みのビットが再び使用されることになる。

【0173】

これらの第5の変形例は、固定サイズを有するレジスタを有する既存のハードウェアを使用し、結果がレジスタに適合するときはいつでも乗算のコストが同じである場合（例えば、所定のサイズのレジスタまたはメモリ位置を意味する動作を処理するCPUに対する命令として動作するソフトウェア実施の場合）、特に興味深いものである。また、ハードウェア設計において、固定サイズの演算を実施するために既存の所定の回路テンプレートを使用し、一部のビットが使用されていなくても問題（例えば、開発速度／コストの問題

10

20

30

40

50

）ない場合にも有用である。この場合、例えば16ビット入力と32ビット出力で実施した場合、10ビットと11ビットの乗算のコスト（実行レイテンシと実施に使用する論理ゲート数）は変わらないが、CCALFの符号化利得は、1ビット多く使用するので、第2バージョンの方がわずかに優れている。

【0174】

代替として、8サンプル（またはそれ以下）をフィルタに使用し、合計演算で2ビット（またはそれ以上）を節約することができる。例えば、8サンプルを使用する図7-f（2ビット節約）、または4サンプルだけを使用する図7-g（3ビット節約）などを参照されたい。その代替の変形例では、合計で節約された2ビット（またはそれ以上）は、例えばフィルタ処理中に、より精度を持ち16ビットレジスタの精度を完全に使用するために乗算の入力間で再分配される。

10

【0175】

図7-cは、フィルタのために16サンプルのみを使用するフィルタパターン706の一例を提供する。そのフィルタパターンを用いると、フィルタの動作は以下になる。

【0176】

$$\begin{aligned} 0_{(x_c, y_c)}^{Cb} = & f_0^{Cb} \cdot I_{(x-1, y-1)} + f_1^{Cb} \cdot I_{(x, y-1)} + f_2^{Cb} \cdot I_{(x+1, y-1)} \\ & + f_3^{Cb} \cdot I_{(x-2, y)} + f_4^{Cb} \cdot I_{(x-1, y)} + f_5^{Cb} \cdot I_{(x, y)} + f_6^{Cb} \cdot I_{(x+1, y)} + f_7^{Cb} \cdot I_{(x+2, y)} \\ & + f_8^{Cb} \cdot I_{(x-2, y+1)} + f_9^{Cb} \cdot I_{(x-1, y+1)} + f_{10}^{Cb} \cdot I_{(x, y+1)} + f_{11}^{Cb} \cdot I_{(x+1, y+1)} + f_{12}^{Cb} \cdot I_{(x+2, y+1)} \\ & + f_{13}^{Cb} \cdot I_{(x-1, y+2)} + f_{14}^{Cb} \cdot I_{(x, y+2)} + f_{15}^{Cb} \cdot I_{(x+1, y+2)} \end{aligned}$$

20

【0177】

このサンプルの選択は、「菱形（ダイヤモンド）」形状として記述することができ、各サンプルは、参照サンプルに対して位置（ $x+u$, $y+v$ ）を有し、 $u \in [-2, 2]$ および $v \in [-1, 2]$ であり、 $|u| = 2$ のとき $v \in [0, 1]$ である。

【0178】

図7-dは、フィルタに16個のサンプルを使用したフィルタパターン707の他の例を示している。このパターンは、YUV4:2:0ビデオ表現を使用する場合、図7-cのものよりも適しており、クロマサンプルのサブサンプル位置は、707のサンプル5、サンプル6、サンプル9、サンプル10の位置の中間に位置している。一変形例によれば、2つ以上のフィルタパターンが利用可能であり、YUV4:2:0におけるクロマサブサンプリングに従って1つのフィルタパターンが選択される。1つの変形例では、フィルタパターンインデックスは、例えばパラメータセット（例えばSPS又はPPS）において、ビットストリームにおいて提供／シグナリングされる。そのフィルタパターンで、フィルタ動作は以下になる。

30

【0179】

$$\begin{aligned} 0_{(x_c, y_c)}^{Cb} = & f_0^{Cb} \cdot I_{(x-1, y-1)} + f_1^{Cb} \cdot I_{(x, y-1)} + f_2^{Cb} \cdot I_{(x+1, y-1)} + f_3^{Cb} \cdot I_{(x-2, y-1)} \\ & + f_4^{Cb} \cdot I_{(x-1, y)} + f_5^{Cb} \cdot I_{(x, y)} + f_6^{Cb} \cdot I_{(x+1, y)} + f_7^{Cb} \cdot I_{(x+2, y)} \\ & + f_8^{Cb} \cdot I_{(x-1, y+1)} + f_9^{Cb} \cdot I_{(x, y+1)} + f_{10}^{Cb} \cdot I_{(x+1, y+1)} + f_{11}^{Cb} \cdot I_{(x+2, y+1)} \\ & + f_{12}^{Cb} \cdot I_{(x-1, y+2)} + f_{13}^{Cb} \cdot I_{(x, y+2)} + f_{14}^{Cb} \cdot I_{(x+1, y+2)} + f_{15}^{Cb} \cdot I_{(x+2, y+2)} \end{aligned}$$

40

【0180】

このサンプルの選択は、各サンプルが、参照サンプルに対して位置（ $x+u$, $y+v$ ）を有し、 $u \in [-1, 2]$ 、 $v \in [-1, 2]$ であると表現することができる。

50

【0181】

図7-eは、フィルタに12サンプルを使用したフィルタパターン708の例である。このパターンでは、図7-cのパターンよりも少ないサンプル数でアクセスでき、ビットストリーム内の係数を少なく符号化できるが、合計演算に使用されるビット数をさらに減らすことはできない。そのフィルタパターンで、フィルタの動作は、以下のようになる。

【0182】

$$\begin{aligned} 0_{(x_c, y_c)}^{Cb} = & f_0^{Cb} \cdot I_{(x-1, y-1)} + f_1^{Cb} \cdot I_{(x, y-1)} + f_2^{Cb} \cdot I_{(x+1, y-1)} \\ & + f_3^{Cb} \cdot I_{(x-1, y)} + f_4^{Cb} \cdot I_{(x, y)} + f_5^{Cb} \cdot I_{(x+1, y)} \\ & + f_6^{Cb} \cdot I_{(x-1, y+1)} + f_7^{Cb} \cdot I_{(x, y+1)} + f_8^{Cb} \cdot I_{(x+1, y+1)} \\ & + f_9^{Cb} \cdot I_{(x-1, y+2)} + f_{10}^{Cb} \cdot I_{(x, y+2)} + f_{11}^{Cb} \cdot I_{(x+1, y+2)} \end{aligned}$$

10

【0183】

このサンプルの選択は、各サンプルが、参照サンプルに対して位置（ $x + u$ 、 $y + v$ ）を有し、 $u \in [-1, 1]$ 、 $v \in [-1, 2]$ であると表現することができる。

【0184】

図7-fは、フィルタのために8つのサンプルだけを使用するフィルタパターン709の例を提供する。このパターンでは、図7-cのパターンと比較して、さらに加算演算に使用するビット数を1ビット削減することができる。また、図4-aのパターンと比較して、2ビット削減することが可能である。このフィルタパターンでは、フィルタの動作は以下のようになる。

20

【0185】

$$\begin{aligned} 0_{(x_c, y_c)}^{Cb} = & f_0^{Cb} \cdot I_{(x, y-1)} \\ & + f_1^{Cb} \cdot I_{(x-1, y)} + f_2^{Cb} \cdot I_{(x, y)} + f_3^{Cb} \cdot I_{(x+1, y)} \\ & + f_4^{Cb} \cdot I_{(x-1, y+1)} + f_5^{Cb} \cdot I_{(x, y+1)} + f_6^{Cb} \cdot I_{(x+1, y+1)} \\ & + f_7^{Cb} \cdot I_{(x, y+2)} \end{aligned}$$

30

【0186】

このサンプルの選択は「菱形」と表現することができ、各サンプルは参照サンプルに対して位置（ $x + u$ 、 $y + v$ ）を有し、 $u \in [-1, 1]$ 、 $v \in [-1, 2]$ であり、 $|u| = 1$ のとき $v \in [0, 1]$ である。

【0187】

図7-gは、フィルタのために4つのサンプルだけを使用するフィルタパターン710の例を提供する。このパターンでは、図7-fのパターンと比較して、和演算に使用されるビット数をさらに1つ減らすことができる。また、図4-aのパターンと比較して、3ビットの削減が可能である。このフィルタパターンで、フィルタの動作は、以下のようになる。

40

【0188】

$$0_{(x_c, y_c)}^{Cb} = f_0^{Cb} \cdot I_{(x, y)} + f_1^{Cb} \cdot I_{(x+1, y)} + f_2^{Cb} \cdot I_{(x, y+1)} + f_3^{Cb} \cdot I_{(x+1, y+1)}$$

【0189】

このサンプルの選択は、各サンプルが参照サンプルに対して位置（ $x + u$ 、 $y + v$ ）を有し、 u と v が共に $[0, 1]$ に属すると表現することができる。

【0190】

50

他のいくつかの変形例では、加算のコストは実質的にゼロである。乗算入力、乗算出力／結果が16ビット（すなわち、15精度ビットと1符号ビット）の符号付き数値になるように制限される。次に、サンプル値に10ビットを使用する場合、係数値は、5ビット+1符号ビットで表現できるように、所定の精度で、所定の範囲に制限される。これらの変形例では、乗算演算は16ビットのレジスタで行われるものもあります。

【0191】

一実施形態では、係数の小数精度は、例えば、パラメータセット（例えば、APS、SPSまたはPPS）、またはスライスヘッダにおいて、第1の精度情報として、ビットストリーム内でシグナリングされる。

【0192】

一実施形態では、係数の範囲（または可能な範囲のセット内のインデックス）は、例えば、パラメータセット（例えば、APS、SPSまたはPPS）内またはスライスヘッダ内で、第2の精度情報としてビットストリーム内でシグナリングされる。

【0193】

一実施形態では、サンプル値の保持ビット数（すなわちKB）または代替的にデッドビット数（すなわちDB）は、第3の精度情報として、例えばパラメータセット（例えばAPS、SPSまたはPPS）内またはスライスヘッダ内で、ビットストリーム内でシグナリングされる。

【0194】

一実施形態では、第1、第2、および／または第3の精度情報のいずれか1つの組み合わせが、例えばパラメータセット（例えばAPS、SPSまたはPPS）内、またはスライスヘッダ内で、ビットストリーム内でシグナリングされる。

【0195】

一変形例では、ビットストリーム内でシグナリングされる第1、第2および／または第3の精度情報のいずれかのかかる組み合わせの許容値は、例えば16などの固定最大ビット数を使用して乗算演算が計算され得る最悪ケースを保証するものとする。変形例として、最悪ケースは、加算動作中に使用される最大ビット数も考慮する。

【0196】

一実施形態では、第1、第2及び／又は第3の精度情報のいずれかの組み合わせの可能な値のセットにおけるインデックスがシグナリングされる。

【0197】

1つの変形例では、そのようなセットは、乗算演算が固定された最大ビット数、例えば16を使用して計算され得る最悪ケースを保証するために定義される。変形例では、最悪ケースは、加算演算中に使用されるビットの最大数も考慮する。

【0198】

第1、第2及び／又は第3の精度情報のいずれかの組み合わせが、第1及び第2の精度情報（すなわち、小数精度及び保持範囲）を含む実施形態に対する変形例では、浮動小数点表現の仮数サイズ（ビット数）及び固定指数をシグナリングすることである。

【0199】

ある変形例では、フィルタ係数に固定小数点表現を使用する代わりに、特定の浮動小数点表現を使用し、限られたビット数の演算で計算されることが保証されている。言い換えれば、各係数インデックスに対して、第1、第2、第3の精度情報のいずれかの組み合わせが通知される。

【0200】

説明された例、変形例及び／又は実施形態において、使用されるビット深度及びサンプル値の精度に応じて、係数の表現の最下位ビットを同じ量だけシフトさせながら（同じ精度を保つために）、係数の範囲を上下にシフトし得る（すなわち2のべき乗で掛ける）ことが理解され、同等の数学結果を提供することができる。つまり、係数値（またはサンプル値）の「範囲」は、サンプル値（または係数値の「範囲」）のスケーリングの逆数でスケーリングされる。

10

20

30

40

50

【0201】

【数22】

$$f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} = a \cdot f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} \cdot \frac{1}{a} = \frac{1}{b} \cdot f'_{(u_i, v_i)}{}^{Cb} \cdot I_{(x+u_i, y+v_i)} \cdot b \quad (22)$$

【0202】

これはビット単位のシフトでも同じであり、Bによる左ビット単位のシフトは 2^B による乗算と等しく、右シフトは 2^B による整数の除算とほぼ等しい（違いは、負の数の場合、左シフトでは丸めがマイナス無限大に向かうのに対し、整数の除算では0に向かうことである）。

10

【0203】

また、サンプル値やフィルタ係数のビット表現の簡略化は、上記のビット演算によって「フィルタ中」に行われたが、フィルタ前にこれらの値を個別に前処理することも可能であることが理解されるべきである。

【0204】

シグナリング

図3の入力CCALFパラメータ305および306は、異なる粒度レベルでCCALFを有効／無効にするためのフラグを含んでもよい。フラグのシグナリングの一例を以下に示す。

【0205】

特定の実施形態では、1つのフラグが、例えばシーケンスパラメータセット（SPS）又はピクチャパラメータセット（PPS）というパラメータセットにおいてシグナリングされる。このフラグは、シーケンスレベルまたはピクチャーレベルでCCALFツールを作動させるか否かを可能にする。CCALFツールが有効な場合、スライスヘッダ内のクロマ成分ごとに1つのフラグがシグナリングされる。これにより、スライス内のクロマ成分毎にCCALFの有効／無効を設定することができる。スライスにおいてCCALFが有効な場合（すなわち、少なくとも1つのクロマ成分においてCCALFが有効な場合）、CCALFが有効なクロマ成分毎に、`ccalf__ctb__flag__cb[x][y]`（Cb成分用）又は`ccalf__ctb__flag__cr[x][y]`（Cr成分用）フラグは、x，yが画像における水平、垂直CTU座標／インデックスであるスライスの各CTUのCTUデータにおいて符号化される。このフラグにより、クロマCTB内のCCALFが有効か無効かの信号が出力される。クロマCTBでCCALFが有効な場合、クロマサンプルはCTBのルマサンプルを使って（例えば、前述したように）CCALFで補正／フィルタされる。

20

30

【0206】

フラグはCABAC（コンテキスト適応二値算術符号化）を使用して通知され、クロマ成分ごとに3つのCABACコンテキストを使用する。インデックス`componentIdx`を持つ与えられたクロマ成分に対して、コンテキストは、同じスライスに属する上（`[x][y-1]`）と左（`[x-1][y]`）CTUでアクティブである`ccalf__ctb__flag__cb`（Cb成分用）または`ccalf__ctb__flag__cr`（Cr成分用）の数（それらが存在し／利用できる場合）を数え、コンテキストインデックスとしてその値を用いることによって選択される。

40

【0207】

VTM-5.0に対応する多用途ビデオ符号化（ドラフト5）と同じ表記と規則で、`ccalf__ctb__flag__cb[x][y]`と`ccalf__ctb__flag__cr[x][y]`のシグナリングの例を表1に示す。

【0208】

50

【表 1】

coding_tree_unit() {	Descriptor
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(slice_sao_luma_flag slice_sao_chroma_flag)	
sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)	
if(slice_alf_enabled_flag){	
...	
}	
if(slice_ccalf_enable_flag_cb)	
ccalf_ctb_flag_cb[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(slice_ccalf_enable_flag_cr)	
ccalf_ctb_flag_cr[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(slice_type == I && qbtbt_dual_tree_intra_flag)	
dual_tree_implicit_qt_split(xCtb, yCtb, CtbSizeY, 0)	
else	
coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 1, 0, 0, 0, 0, 0, SINGLE_TREE, MODE_TYPE_ALL)	
}	

10

coding_tree_unit () 構文

【0209】

実施形態によれば、スライス専用フィルタ係数を適合させて送信できるようにするために、従来スライスパラメータで行われていたように、CCALFフィルタの係数はスライスヘッダでシグナリングされる。あるクロマ成分に対してCCALFが有効であることが示された場合、対応するフィルタの各係数はビットストリーム内／ビットストリームから順次符号化／復号化される。

20

【0210】

CCALFの係数は、通常、ゴロム (Golomb) 可変長符号を用いて符号化される。しかし、上述した係数を表すビットの削減を実施する場合、符号化を簡略化することができる。具体的には、係数値を格納するビット数が削減されるため、符号化効率を落とさずに、より解析が簡単な符号化方式を用いることができる。

【0211】

特に、ゴロム符号化では、係数を正しく復号化するために「調整可能パラメータ」 (M) をシグナリングする必要があり、これはビットストリームのオーバーヘッドを増加させる。本発明は、他のパラメータを参照することなく、スライスヘッダ内の係数を直接シグナリングすることに関する。言い換えれば、係数は、固定表現を用いて（すなわち、「調整可能パラメータ」を参照することなく）符号化される。これは、符号ワードがデリミタ（推論または明示的に符号化されたもの）のない単一の要素であるという点で、「プレフィックスフリー」または「カンマフリー」であると表現することができる。

30

【0212】

第1の代替では、上述の第1および第2の変形例を組み合わせる場合に特に適しているが、係数値の使用／有効ビットは、圧縮（すなわち、値が何であれ、符号を表すための固定ビット数）なしでビットストリームにそのまま格納される。例えば、図5-cのように係数が表現される場合、係数はビットストリーム中の3ビット（例えば符号ビットと2つの小数精度ビット）に格納することができる。

40

【0213】

第2の代替として、低振幅の係数のコストを削減するために、一種の符号付き単項符号化が使用される。ビットストリームの第1のビットは係数がゼロかどうかを示し、次に係数がゼロでない場合、ビットは係数の符号を示し、次に1に等しいビットのシーケンスは係数の振幅を示す。1に等しいビットの数は、「係数の整数表現」の絶対値から1を引いた値に等しい。係数の整数表現とは、503の整数値のビットで表現される整数値のことである。「マイナス1」を用いるのは、ゼロに等しい情報が既に符号化されているためである。そして、絶対値が最大可能／許可された値より低い場合、ゼロに等しいビットは、その数値のシグナリングの終了を示す。これは、絶対値から1を引いた値の単項符号化に

50

相当する。この表現に対する復号化処理は、表 2 に示すアルゴリズムでまとめることができる。

【 0 2 1 4 】

【 表 2 】

decode_ccalf_coefficient() {
if (read_one_bit())
return 0
sign = read_one_bit()
max_val = CCALF_MAX_COEFF_VALUE
if (sign)
max_val = - CCALF_MIN_COEFF_VALUE
abs_val = 1
while (abs_val < max_val && read_one_bit())
abs_val += 1
if (sign)
return - abs_val
return abs_val
}

10

第 2 の代替の係数復号化アルゴリズム

【 0 2 1 5 】

可能な係数値の非対称範囲が使用されるいくつかの変形例では、表 2 において「CCALF__MAX__COEFF__VALUE」が「-CCALF__MIN__COEFF__VALUE」に等しくない場合、数の最大可能／認可値は負の数と正の数で異なることができる。例えば、図 5 - c の表現を用いた場合、可能な係数値の範囲が「-0.03125 ; 0.03125」[(すなわち「-0.03125 ; 0.0234375」)]であれば、最大可能／許可値は、負の数では（-0.03125 は整数の -4 で表されるので）4、正の数では（0.0234375 は整数の 3 で表されるので）3 である。この例では、第 1 のビットが 1 であれば数値は 0 であり、符号ビットが 1 であれば数値は負であることを意味する。そして、ビットストリームにおける係数の表現は次のようになる：-0.03125 に対して「01111」（すなわち、-4 = CCALF__MIN__COEFF__VALUE）、-0.0234375 に対して「01110」（すなわち -3）、-0.015625 に対して「0110」（すなわち -2）、-0.0078125 に対して「010」（すなわち -1）、0.0 に対して「1」、0.0078125 に対して「000」（すなわち 1）、0.015625 に対して「0010」（すなわち 2）、0.0234375 に対して「0011」（すなわち、3 = CCALF__MAX__COEFF__VALUE）である。いくつかの変形例では、第 1 のビットが 1 に等しい場合、その数値が 0 でないことを意味することがある。いくつかの変形例では、符号ビットが 1 に等しいと、数値が正であることを意味することがある。いくつかの変形例において、数の単項符号化は、（数が最大値より低い場合）1 に等しいビットシーケンスが後続する 0 に等しいビットシーケンスであり得る。

20

30

【 0 2 1 6 】

第 3 の代替として、ビットストリームは、まず、係数の絶対値に対する整数表現の単項符号化を行う。非ゼロの場合は、符号ビットがそれに続く。表 3 のアルゴリズムは、この表現に対する復号化処理の一例を示している。この第 3 の代替では、係数の絶対値を復号化する前に符号を知ることにはできない。そして、単項符号化に用いられる最大値は、正の値と負の値で同じである（表 3 の CCALF__MAX__COEFF__ABSOLUTE__VALUE）。例として、図 5 - c の表現は、可能な係数値の範囲を「-0.0234375 ; 0.0234375」に等しく設定して使用する（すなわち CCALF__MAX__COEFF__ABSOLUTE__VALUE = 3）。そして、ビットストリームにおける係数の表現は次のようになる：-0.0234375 に対して「1111」（すなわち -3）、-0.015625 に対して「1101」（すなわち -2）、-0.0078125 に対して「101」（すなわち -1）、0.0 に対して「0」、0.0078125 に

40

50

対して「100」（すなわち1）、0.015625に対して「1100」（すなわち2）、0.0234375に対して「1110」（すなわち3）である。

【0217】

【表3】

decode_ccalf_coefficient() {
abs_val = 0
while (abs_val < CCALF_MAX_COEFF_ABSOLUTE_VALUE && read_one_bit())
abs_val += 1
if (abs_val > 0 && read_one_bit())
return -abs_val
return abs_val
}

10

第3の代替の係数復号化アルゴリズム

【0218】

いくつかの変形例では、CCALFフィルタの係数はAPS（または別のパラメータセット）でシグナリングされる。slice_ccalf_enable_flag_cbがスライスヘッダの1に等しいとき、スライスヘッダは、Cbクロマ成分のためにCCALFフィルタの係数が検索されなければならないAPS（または他のパラメータセット）のインデックス／識別子も提供する。スライスヘッダのslice_ccalf_enable_flag_crが1のとき、スライスヘッダは、Crクロマ成分についてCCALFフィルタのフィルタ係数を取得しなければならないAPS（または他のパラメータセット）のインデックスも提供する。図4-aに示されるようなフィルタパターンを使用する場合、いくつかの係数はいくつかのサンプル位置と共有されるので、ビットストリームに存在するフィルタ係数の数は、フィルタ形状に使用されるサンプルの数よりも小さくなる。

20

【0219】

本発明の実施の形態について

前述の1つまたは複数の実施形態は、前述の1つまたは複数の実施形態の方法のステップを実行する符号化器または復号化器の形態で実施され得る。以下の実施形態は、そのような実施を例示する。

30

【0220】

例えば、前述の実施形態のいずれかによる適応ループフィルタは、図8の符号化器によって実行されるポストフィルタ9415または図9の復号化器によって実行されるポストフィルタ9567において使用されてもよい。

【0221】

図8は、本発明の一実施形態による符号化器のブロック図である。符号化器は、接続されたモジュールによって表され、各モジュールは、例えば、装置の中央処理装置（CPU）によって実行されるプログラミング命令の形態で、本発明の1つ以上の実施形態による画像シーケンスの画像を符号化する少なくとも1つの実施形態を実施する方法の少なくとも1つの対応するステップを実施するように適合されている。

40

【0222】

デジタル画像i0からin9401のオリジナルシーケンスは、符号化器9400によって入力として受信される。各デジタル画像は、時には複数の画素とも呼ばれるサンプルの集合によって表される（以下、これらを複数の画素と呼ぶ）。符号化処理の実施後、符号化器9400によってビットストリーム9410が出力される。ビットストリーム9410は、複数の符号化ユニットまたはスライスなどの画像部分のデータを含み、各スライスは、スライスの符号化に使用される符号化パラメータの符号化値を送信するためのスライスヘッダと、符号化されたビデオデータからなるスライスボディとを含む。

【0223】

入力されたデジタル画像i0からin9401は、モジュール9402によって複数の

50

画素の複数のブロックに分割される。ブロックは画像部分に対応し、可変サイズ（例えば、 4×4 、 8×8 、 16×16 、 32×32 、 64×64 、 128×128 画素およびいくつかの矩形ブロックサイズも考慮できる）であってもよい。符号化モードは、各入力ブロックごとに選択される。空間予測符号化（イントラ予測）と時間予測符号化（インター符号化，MERGE，SKIP）の2つの符号化方式がある。可能な符号化モードがテストされる。

【0224】

モジュール9403は、符号化される所定のブロックが、符号化されるブロックの近傍の画素から計算される予測器によって予測される、イントラ予測処理を実行する。選択されたイントラ予測器の表示と、与えられたブロックとその予測器との間の差は、イントラ符号化が選択された場合、残差を提供するために符号化される。

10

【0225】

時間予測は、動き推定モジュール9404と動き補償モジュール9405によって実施される。まず参照画像のセット9416の中から参照画像が選択され、参照画像のうち、符号化されるべき所定のブロックに最も近い（画素値の類似性の点で最も近い）領域である、参照領域または画像部分とも呼ばれる部分が、動き推定モジュール9404によって選択される。そして、動き補償モジュール9405は、選択された領域を用いて、符号化されるべきブロックを予測する。選択された参照領域と与えられたブロックとの間の差は、残留ブロック／データとも呼ばれ、動き補償モジュール9405によって計算される。選択された参照領域は、動き情報（例えば、動きベクトル）を用いて示される。

20

【0226】

このように、両方の場合（空間予測および時間予測）において、残差は、SKIPモードでないときに元のブロックから予測器を減算することによって計算される。

【0227】

モジュール9403によって実施されるイントラ予測では、予測方向が符号化される。モジュール9404、9405、9416、9418、9417によって実施されるインター予測では、少なくとも1つの動きベクトルまたはその動きベクトルを識別するための情報（データ）が時間予測用に符号化される。

【0228】

インター予測が選択された場合、動きベクトルおよび残差ブロックに関連する情報が符号化される。さらにビットレートを下げるために、動きが均質であると仮定して、動きベクトル予測器との差分によって動きベクトルを符号化する。動き情報予測器候補のセットからの動きベクトル予測器は、動きベクトル予測および符号化モジュール9417によって、動きベクトルフィールド9418から得られる。

30

【0229】

符号化器9400は、レート歪み基準などの符号化コスト基準を適用して符号化モードを選択するための選択モジュール9406をさらに備えている。さらに冗長性を減らすために、変換モジュール9407によって残差ブロックに変換（DCTなど）を施し、得られた変換データを量子化モジュール9408で量子化し、エントロピー符号化モジュール9409でエントロピー符号化を行う。最後に、SKIPモードでなく、選択された符号化モードが残差ブロックの符号化を必要とする場合、符号化されている現在のブロックの残差ブロックがビットストリーム9410に挿入される。

40

【0230】

符号化器9400はまた、後続の画像の動き推定のための参照画像（例えば、参照画像／ピクチャ9416内のもの）を生成するために、符号化された画像の復号化を実行する。これにより、ビットストリームを受信する符号化器および復号化器は、同じ参照フレームを有することができる（例えば、再構成された画像または再構成された画像部分が使用される）。逆量子化（「量子化解除」）モジュール9411は、量子化されたデータの逆量子化（「量子化解除」）を行い、その後に変換モジュール9412によって逆変換が実行される。イントラ予測モジュール9413は、予測情報を使用して、所定のブロック

50

に対してどの予測器を使用するかを決定し、動き補償モジュール 9 4 1 4 は、モジュール 9 4 1 2 によって得られた残差を、一連の参照画像 9 4 1 6 から得られた参照領域に実際に追加する。その後、モジュール 9 4 1 5 によってポストフィルタが適用され、複数の画素の再構成されたフレーム（画像または画像部分）をフィルタして、参照画像 9 4 1 6 のセットのための別の参照画像を得る。

【0 2 3 1】

図 9 は、本発明の実施形態による符号化器からデータを受信するために使用され得る復号化器 9 5 6 0 のブロック図である。復号化器は、接続されたモジュールによって表され、各モジュールは、例えば、デバイスの CPU によって実行されるプログラミング命令の形態で、復号化器 9 5 6 0 によって実施される方法の対応するステップを実施するように適合される。

10

【0 2 3 2】

復号化器 9 5 6 0 は、符号化されたユニット（例えば、画像部分、ブロックまたは符号化ユニットに対応するデータ）からなるビットストリーム 9 5 6 1 を受け取り、各々は、符号化パラメータに関する情報を含むヘッダと、符号化ビデオデータを含むボディとを含む。図 8 に関して説明すると、符号化されたビデオデータはエントロピー符号化されており、動き情報（例えば動きベクトル予測器のインデックス）は、所定の画像部分（例えばブロックまたは CU）に対して、所定のビット数で符号化される。受信した符号化されたビデオデータは、モジュール 9 5 6 2 によってエントロピー復号化される。その後、モジュール 9 5 6 3 によって残留データが量子化解除され、その後、モジュール 9 5 6 4 によ

20

【0 2 3 3】

また、符号化モードを示すモードデータをエントロピー復号化し、そのモードに基づいて、符号化された画像データのブロック（ユニット／セット／グループ）に対してイントラ型復号化またはインター型復号化が行われる。イントラモードの場合、ビットストリームで指定されたイントラ予測モードに基づいて、イントラ予測モジュール 9 5 6 5 によってイントラ予測器が決定される（例えば、イントラ予測モードは、ビットストリームに提供されるデータを用いて決定可能である）。モードがインターモードである場合、符号化器が使用する参照領域を見つける（特定する）ように、ビットストリームから動き予測情報が抽出／取得される。動き予測情報は、例えば、参照フレームインデックスと動きベクトル残差とから構成される。動きベクトル予測情報は、動きベクトル復号化モジュール 9 5 7 0 によって、動きベクトル残差に加算され、動きベクトルを得るために使用される。

30

【0 2 3 4】

動きベクトル復号化モジュール 9 5 7 0 は、動き予測によって符号化された各画像部分（例えば、現在のブロックまたは CU）に対して動きベクトル復号化を適用する。現在のブロックに対する動きベクトル予測器のインデックスが得られると、画像部分（例えば現在のブロックまたは CU）に関連する動きベクトルの実際の値が復号化され、モジュール 9 5 6 6 によって動き補償を適用するために使用され得る。復号化された動きベクトルによって示される参照画像部分は、モジュール 9 5 6 6 が動き補償を実行できるように、参照画像 9 5 6 8 のセットから抽出／取得される。動きベクトルフィールドデータ 9 5 7 1 は、後続の復号化された動きベクトルの予測に使用されるために、復号化された動きベクトルで更新される。

40

【0 2 3 5】

最後に、復号化されたブロックが得られる。適切な場合、ポストフィルタモジュール 9 5 6 7 によってポストフィルタが適用される。復号化されたビデオ信号 9 5 6 9 が最終的に得られ、復号化器 9 5 6 0 によって提供される。

【0 2 3 6】

図 10 は、本発明の 1 つ以上の実施形態が実施され得るデータ通信システムを示す図である。データ通信システムは、データ通信ネットワーク 9 2 0 0 を介して、データストリーム 9 2 0 4 のデータパケットを受信装置（この場合はクライアント端末 9 2 0 2）に送

50

信するように動作可能な送信装置（この場合はサーバ 9201）を含む。データ通信ネットワーク 9200 は、ワイドエリアネットワーク（WAN）であっても、ローカルエリアネットワーク（LAN）であってもよい。このようなネットワークは、例えば、無線ネットワーク（Wi-Fi / 802.11a または b または g）、イーサネットネットワーク、インターネットネットワーク、または複数の異なるネットワークからなる混合ネットワークであってもよい。本発明の特定の実施形態では、データ通信システムは、サーバ 9201 が複数のクライアントに同じデータコンテンツを送信するデジタルテレビ放送システムであってもよい。

【0237】

サーバ 9201 が提供するデータストリーム 9204 は、ビデオやオーディオデータを表すマルチメディアデータで構成されてもよい。オーディオおよびビデオデータストリームは、本発明のいくつかの実施形態において、サーバ 9201 によって、マイクロフォンおよびカメラをそれぞれ使用して取り込まれることができる。いくつかの実施形態では、データストリームは、サーバ 9201 に格納されてもよいし、サーバ 9201 が他のデータプロバイダから受信してもよいし、サーバ 9201 で生成してもよい。サーバ 9201 は、特に、符号化器への入力として提示されたデータのよりコンパクトな表現である送信用の圧縮ビットストリームを提供するために、ビデオ及びオーディオストリームを符号化するための符号化器を備える。送信データの量に対する送信データの質のより良い比率を得るために、ビデオデータの圧縮は、例えば、高効率ビデオ符号化（HEVC）形式または H.264 / アドバンスドビデオ符号化（AVC）形式または多用途ビデオ符号化（VV C）形式に従ってもよい。クライアント 9202 は、送信されたビットストリームを受信し、再構成されたビットストリームを復号化して、ディスプレイ装置でビデオ画像を再生し、ラウドスピーカーでオーディオデータを再生する。

【0238】

本実施形態ではストリーミングシナリオを考慮したが、本発明のいくつかの実施形態では、符号化器と復号化器の間のデータ通信は、例えば光ディスクのようなメディア記憶装置を使用して実行されてもよいことが理解されよう。本発明の 1 つ以上の実施形態では、ビデオ画像は、最終画像にフィルタされた複数の画素を提供するために画像の再構成された複数の画素に適用するための補償オフセットを代表するデータと共に伝送されてもよい。

【0239】

図 11 は、本発明の少なくとも 1 つの実施形態を実施するように構成された処理装置 9300 を模式的に示す図である。処理装置 9300 は、マイクロコンピュータ、ワークステーション、ユーザ端末、または軽量携帯機器などの装置であってもよい。デバイス / 装置 9300 は、

- マイクロプロセッサのような中央処理装置 9311（CPU と表記される）、
- 装置 9300 の演算及び / 又は本発明の実施のためのコンピュータプログラム / 命令を記憶するための ROM で示される読み取り専用メモリ 9307、
- 本発明の実施形態の方法の実行可能コード、並びに本発明の実施形態によるデジタル画像のシーケンスを符号化する方法及び / 又はビットストリームを復号化する方法を実施するために必要な変数及びパラメータを記録するために適合されたレジスタを記憶するための RAM と呼ばれるランダムアクセスメモリ 9312、及び、
- 処理されるデジタルデータが送信または受信される通信ネットワーク 9303 に接続された通信インタフェース 9302、

に接続された通信バス 9313 を含む。

【0240】

オプションで、装置 9300 は、以下の構成要素も含むことができる：

- 本発明の 1 つ以上の実施形態の方法を実施するためのコンピュータプログラム、及び本発明の 1 つ以上の実施形態の実施中に使用又は生成されるデータを記憶するための、ハードディスクなどのデータ記憶手段 9304、

ーディスク 9306（例えば記憶媒体）用のディスクドライブ 9305 であって、ディスクドライブ 9305 は、ディスク 9306 からデータを読み取るか、またはディスク 9306 にデータを書き込むように適合されている、ディスクドライブ、または、キーボード 9310、タッチスクリーン、またはその他のポインティング／入力手段によって、データを表示し、および／またはユーザーとのグラフィカルインタフェースとして機能するためのスクリーン 9309。

【0241】

装置 9300 は、例えばデジタルカメラ 9320 又はマイクロフォン 9308 などの様々な周辺機器に接続することができ、各々は、装置 9300 にマルチメディアデータを供給するように入力／出力カード（図示せず）に接続されている。

【0242】

通信バス 9313 は、装置 9300 に含まれる、又はそれに接続される様々な要素間の通信及び相互運用性を提供する。バスの表現は限定的ではなく、特に中央処理装置 9311 は、装置 9300 の任意の要素に直接又は装置 9300 の別の要素によって命令を伝達するように動作可能である。

【0243】

ディスク 9306 は、例えば、書き換え可能か否かを問わないコンパクトディスク（CD-ROM）、ZIP ディスクまたはメモ리카ードなどの任意の情報媒体で置き換えることができ、一般的には、マイクロコンピュータまたはプロセッサによって読み取り可能な情報記憶手段で、装置に組み込まれているか否かを問わず、おそらく取り外し可能で、その実行により本発明によるデジタル画像シーケンスを符号化する方法および／またはビットストリームを復号化する方法が実施できる 1 または複数のプログラムの格納に適合させることが可能なものである。

【0244】

実行可能コードは、読み取り専用メモリ 9307、ハードディスク 9304 上、または前述したような例えばディスク 9306 のような取り外し可能なデジタル媒体上のいずれかに記憶され得る。変形例によれば、プログラムの実行可能コードは、実行される前に装置 9300 の記憶手段の 1 つ（例えばハードディスク 9304）に記憶されるように、インタフェース 9302 を介して、通信ネットワーク 9303 によって受信され得る。

【0245】

中央処理装置 9311 は、本発明によるプログラムまたはプログラムのソフトウェアコードの命令または部分、前述の記憶手段の 1 つに格納される命令の実行を制御し指示するように適合されている。電源投入時に、例えばハードディスク 9304、ディスク 9306、または読み取り専用メモリ 9307 の不揮発性メモリに格納されているプログラムまたはプログラムは、ランダムアクセスメモリ 9312 に転送され、その後、プログラムまたはプログラムの実行可能コード、および本発明を実施するために必要な変数およびパラメータを格納するためのレジスタが格納される。

【0246】

本実施形態では、本装置は、ソフトウェアを用いて本発明を実現するプログラム可能装置である。しかしながら、代替的に、本発明は、ハードウェアで（例えば、特定用途向け集積回路（ASIC）の形態で）実施され得る。

【0247】

また、本発明の他の実施形態によれば、前述の実施形態による復号化器は、コンピュータ、携帯電話（セルラーフォン）、タブレット、またはコンテンツをユーザに提供／表示することができる他のタイプの装置（例えば、表示装置）などのユーザ端末に設けられることが理解される。さらに別の実施形態によれば、前述の実施形態による符号化器は、カメラ、ビデオカメラまたはネットワークカメラ（例えば、閉回路テレビまたはビデオ監視カメラ）をも含む撮像装置に設けられ、符号化器が符号化するためのコンテンツを撮像して提供する。2 つのそのような実施形態が、図 12 および図 13 を参照して以下に提供される。

10

20

30

40

50

【0248】

図12は、ネットワークカメラ9452とクライアント装置9454とを含むネットワークカメラシステム9450を説明するための図である。

【0249】

ネットワークカメラ9452は、撮像部9456、符号化部9458、通信部9460、及び制御部9462を含む。ネットワークカメラ9452とクライアント装置9454は、ネットワーク9200を介して相互に通信可能に接続されている。撮像部9456は、レンズとイメージセンサ（例えば、CCD（電荷結合素子）やCMOS（相補型金属酸化膜半導体））を含み、対象物の画像を撮像し、その画像に基づく画像データを生成する。この画像は、静止画像であってもよいし、ビデオであってもよい。撮像ユニットはまた、（光学的またはデジタル的に）それぞれズームまたはパンするように適合されているズーム手段および／またはパン手段を含んでいてもよい。符号化ユニット9458は、前述の実施形態の1つまたは複数で説明した符号化方法を用いて画像データを符号化する。符号化部9458は、前述の実施形態で説明した符号化方法のうち少なくとも1つを用いる。他の例として、符号化部9458は、前述の実施形態で説明した符号化方法を組み合わせて用いることができる。

10

【0250】

ネットワークカメラ9452の通信部9460は、符号化部9458により符号化された符号化画像データを、クライアント装置9454に送信する。さらに、通信部9460は、クライアント装置9454からコマンドを受信してもよい。コマンドは、符号化部9458による符号化のためのパラメータを設定するコマンドで構成される。制御部9462は、通信部9460が受信したコマンドやユーザ入力に従って、ネットワークカメラ9452内の他のユニットを制御する。

20

【0251】

クライアント装置9454は、通信部9464、復号化部9466、および制御部9468を含む。クライアント装置9454の通信部9464は、ネットワークカメラ9452にコマンドを送信してもよい。さらに、クライアント装置9454の通信部9464は、ネットワークカメラ9452から、符号化された画像データを受信する。復号化部9466は、前述の1つ以上の実施形態で説明した復号化方法を用いて、符号化された画像データを復号化する。別の例として、復号化ユニット9466は、前述の実施形態で説明した復号化方法の組合せを用いることができる。クライアント装置9454の制御部9468は、ユーザの操作または通信部9464が受信したコマンドに従って、クライアント装置9454内の他のユニットを制御する。また、クライアント装置9454の制御部9468は、復号化部9466により復号化された画像を表示するように、表示装置9470を制御してもよい。

30

【0252】

また、クライアント装置9454の制御部9468は、ネットワークカメラ9452のパラメータ、例えば、符号化部9458による符号化のためのパラメータの値を指定するためのGUI（グラフィカルユーザインタフェース）を表示するように、表示装置9470を制御してもよい。また、クライアント装置9454の制御部9468は、表示装置9470が表示するGUIに対するユーザの操作入力に従って、クライアント装置9454内の他のユニットを制御してもよい。また、クライアント装置9454の制御部9468は、表示装置9470が表示するGUIに対するユーザ操作入力に応じて、ネットワークカメラ9452に対するパラメータの値を指定するコマンドをネットワークカメラ9452に送信するように、クライアント装置9454の通信部9464を制御してもよい。

40

【0253】

図13は、スマートフォン9500を説明するための図である。スマートフォン9500は、通信部9502、復号化／符号化部9504、制御部9506、及び表示部9508から構成される。

【0254】

50

通信部 9502 は、ネットワーク 9200 を介して、符号化された画像データを受信する。復号化／符号化部 9504 は、通信部 9502 が受信した符号化された画像データを復号化する。復号化／符号化部 9504 は、前述の 1 つ以上の実施形態で説明した復号化方法を用いて、符号化画像データを復号化する。また、復号化／符号化部 9504 は、前述の実施形態で説明した符号化方法または復号化方法のうち少なくとも 1 つを用いることも可能である。別の例として、復号化／符号化部 9504 は、前述の実施形態で説明した復号化方法または符号化方法を組み合わせて使用することができる。

【0255】

制御部 9506 は、ユーザの操作または通信部 9502 が受信したコマンドに従って、スマートフォン 9500 内の他のユニットを制御する。例えば、制御部 9506 は、復号化／符号化部 9504 によって復号化された画像を表示するように、表示部 9508 を制御する。

【0256】

スマートフォンは、画像または動画を記録するための画像記録デバイス 9510（例えばデジタルカメラおよび関連回路）をさらに備えてもよい。このような記録された画像又は動画は、制御部 9506 の指示の下、復号化／符号化部 9504 によって符号化されてもよい。スマートフォンは、モバイルデバイスの向きを感知するように適合されたセンサ 9512 をさらに含んでもよい。そのようなセンサは、加速度計、ジャイロスコプ、コンパス、グローバルポジショニング（GPS）ユニット、または類似の位置センサを含み得る。そのようなセンサ 9512 は、スマートフォンが向きを変えたかどうかを判断することができ、そのような情報は、ビデオストリームを符号化するときに使用され得る。

【0257】

本発明を実施形態を参照して説明したが、本発明は開示された実施形態に限定されないことが理解されよう。添付の特許請求の範囲に定義された本発明の範囲から逸脱することなく、様々な変更および修正がなされ得ることは、当業者には理解されよう。本明細書（添付の請求項、要約書及び図面を含む）に開示された全ての特徴、及び／又はそのように開示された任意の方法又はプロセスの全てのステップは、かかる特徴及び／又はステップの少なくとも一部が相互に排他的である組み合わせを除き、任意の組み合わせで組み合わせることが可能である。本明細書（添付の請求項、要約および図面を含む）に開示された各特徴は、明示的に別段の記載がない限り、同一、同等または類似の目的を果たす代替の特徴によって置き換えることができる。したがって、明示的に別段の記載がない限り、開示された各特徴は、同等または類似の特徴の一般的な一連の一例でしかない。

【0258】

また、上述の比較、決定、評価、選択、実行（execution）、実行（performing）、または検討の任意の結果（例えば符号化またはフィルタ処理中に行われる選択）は、ビットストリーム中のデータにおいて示されるか、またはビットストリーム中のデータから決定／参照可能であり、例えばフラグまたは結果を示す情報であり、示されるかまたは決定／参照された結果が、例えば復号化処理中に比較、決定、評価、選択、実行、実行、または検討を実際に行う代わりに処理において使用され得ることが理解されるであろう。

【0259】

特許請求の範囲において、用語“含む（comprising）”は他の要素やステップを排除するものではなく、不定冠詞“a”や“an”は、複数を排除するものではない。異なる特徴が相互に異なる従属請求項に記載されているという単なる事実は、これらの特徴の組合せが有利に使用され得ないことを示すものではない。特許請求の範囲に現れる参照数字は、説明のためのものであり、特許請求の範囲に限定的な影響を及ぼさないものとする。

【0260】

先の実施形態において、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組み合わせで実施することができる。ソフトウェアで実施される場合、機能は、1 つ以上の命令又はコードとして、コンピュータ可読媒体に格納されるか又はそれを介して伝送され、ハードウェアベースの処理ユニットによって実行されても

10

20

30

40

50

よい。

【0261】

コンピュータ可読媒体は、データ記憶媒体などの有形の媒体に相当するコンピュータ可読記憶媒体、または、例えば通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含んでもよい。このように、コンピュータ可読媒体は、一般に、(1) 非一時的である有形のコンピュータ可読記憶媒体、または(2) 信号または搬送波などの通信媒体に対応することができる。データ記憶媒体は、本開示に記載の技術を実施するための命令、コード及び／又はデータ構造を取得するために、1以上のコンピュータ又は1以上のプロセッサによってアクセスされ得る任意の利用可能な媒体であってよい。コンピュータプログラム製品は、コンピュータ読取可能な媒体を含んでもよい。

10

【0262】

限定ではなく例として、このようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMまたは他の光ディスクストレージ、磁気ディスクストレージ、または他の磁気記憶装置、フラッシュメモリ、または命令またはデータ構造の形態で所望のプログラムコードを格納するために使用でき、コンピュータによってアクセスできる任意の他の媒体で構成することができる。また、どのような接続も適切にコンピュータ可読媒体と呼ばれる。例えば、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者線(DSL)、または赤外線、ラジオ、マイクロ波などの無線技術を使用してウェブサイト、サーバ、または他のリモートソースから命令を送信する場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、ラジオ、マイクロ波などの無線技術も媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的な媒体を含まず、代わりに、非一時的な有形記憶媒体に向けられることを理解されたい。本明細書で使用されるディスクおよびディスクは、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多用途ディスク(DVD)、フロッピーディスクおよびブルーレイディスクを含み、ディスクは通常磁氣的にデータを再生し、ディスクはレーザーを用いて光学的にデータを再生する。また、上記の組み合わせもコンピュータ読み取り可能な媒体の範囲に含まれるものとする。

20

【0263】

命令は、1つ以上のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルロジックアレイ(FPGA)、または他の同等の集積またはディスクリット論理回路などの1つ以上のプロセッサによって実行され得る。したがって、本明細書で使用される「プロセッサ」という用語は、前述の構造のいずれか、または本明細書に記載の技術の実施に適した他の構造を指す場合がある。さらに、いくつかの態様において、本明細書に記載される機能は、符号化および復号化のために構成された専用のハードウェアおよび／またはソフトウェアモジュールで提供されてもよく、または複合コーデックに組み込まれてもよい。また、技法は、1つまたは複数の回路または論理要素内で完全に実施され得る。

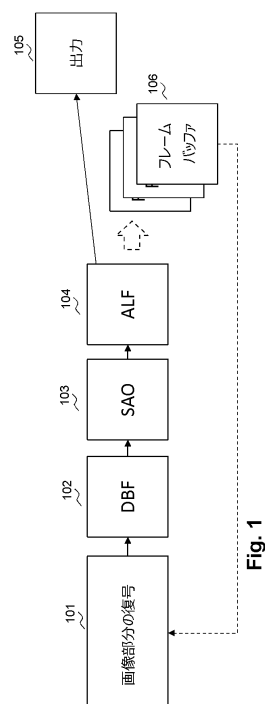
30

40

50

【図面】

【図 1】



【図 2】

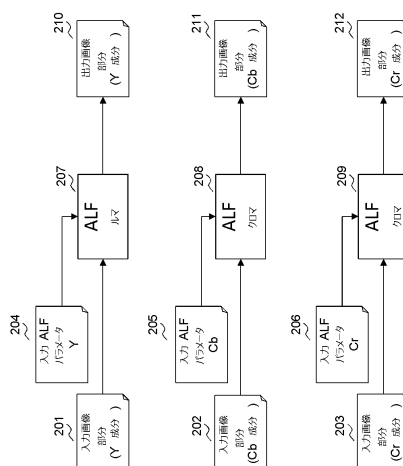
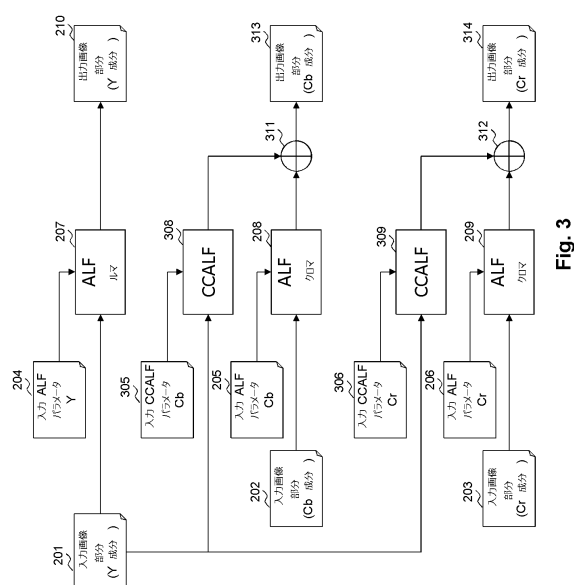
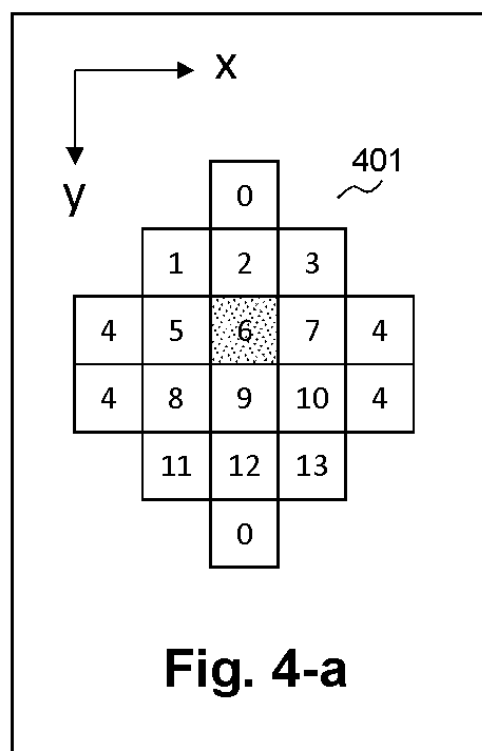


Fig. 2

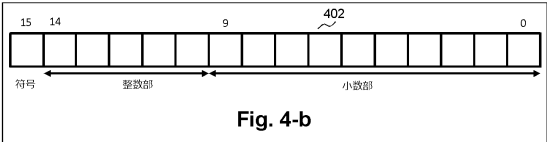
【圖 3】



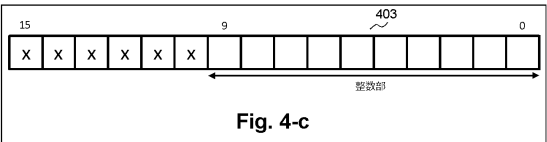
【図 4 - a】



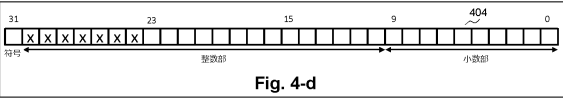
【図 4 - b】



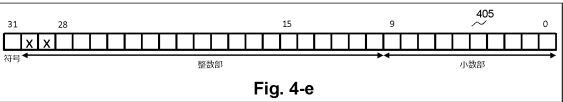
【図 4 - c】



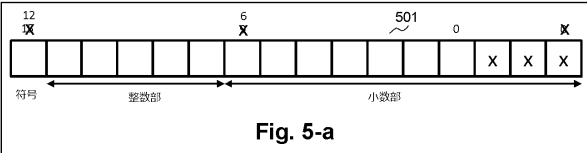
【図 4 - d】



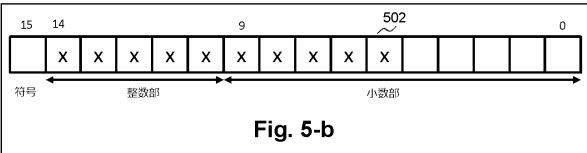
【図 4 - e】



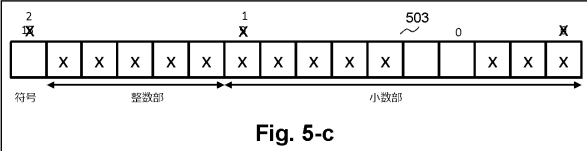
【図 5 - a】



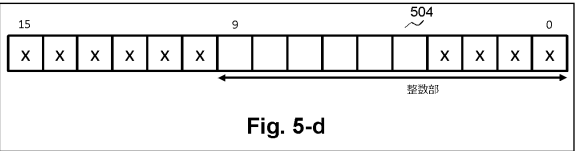
【図 5 - b】



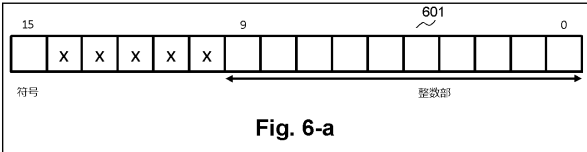
【図 5 - c】



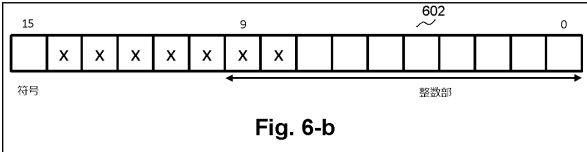
【図 5 - d】



【図 6 - a】



【図 6 - b】



10

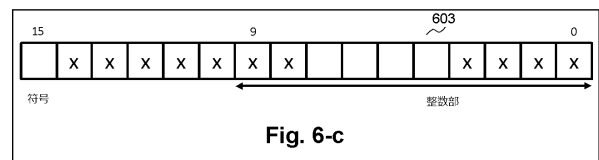
20

30

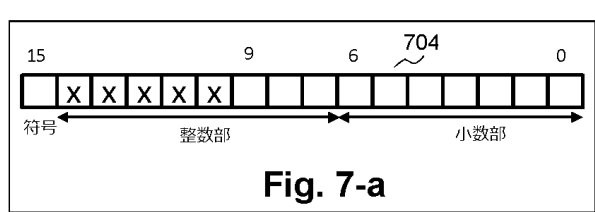
40

50

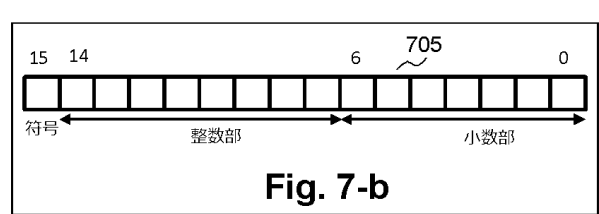
【図 6 - c】



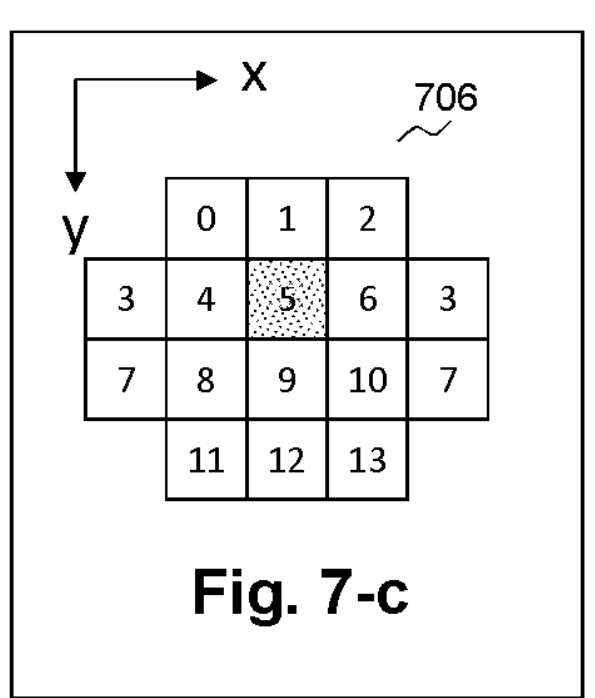
【図 7 - a】



【図 7 - b】



【図 7 - c】



10

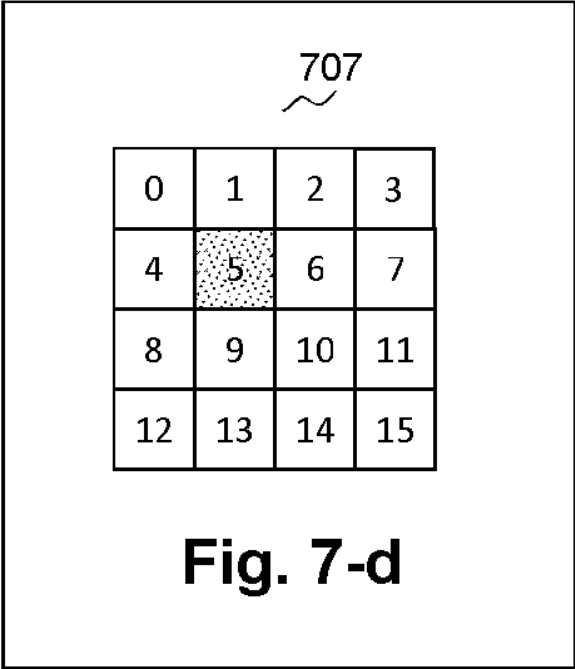
20

30

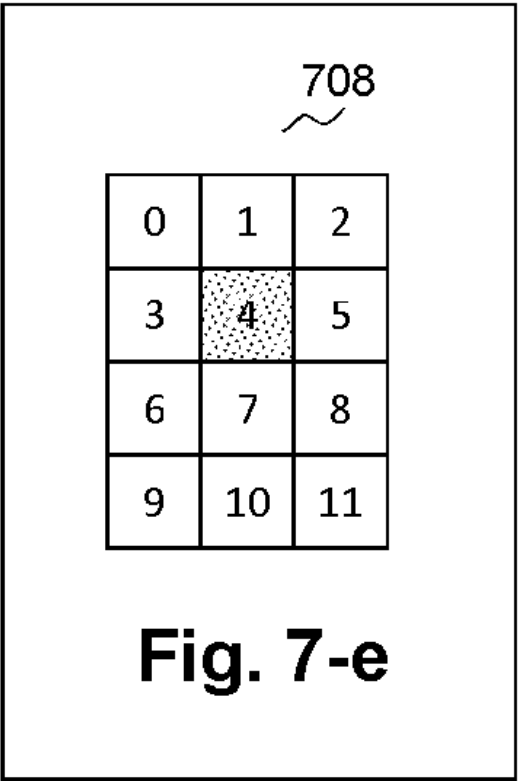
40

50

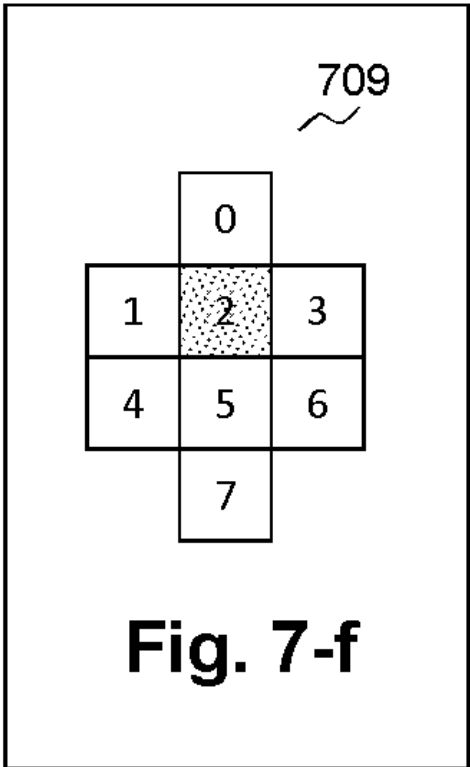
【図 7 - d】



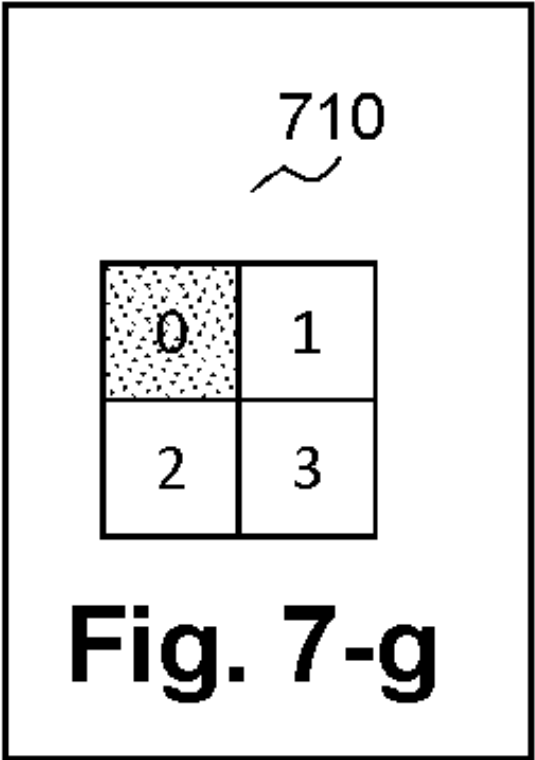
【図 7 - e】



【図 7 - f】



【図 7 - g】



10

20

30

40

50

【圖 8】

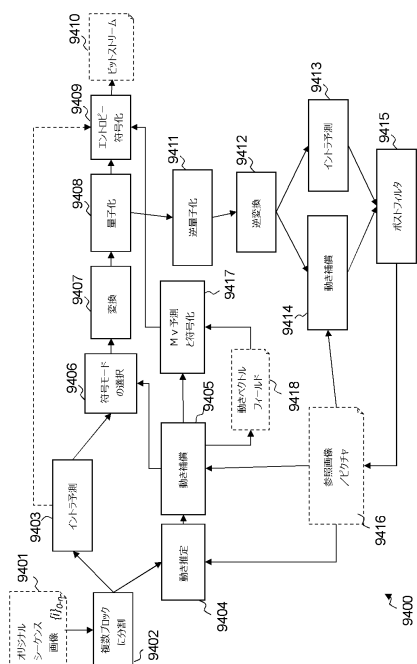


Fig. 8

【図 9】

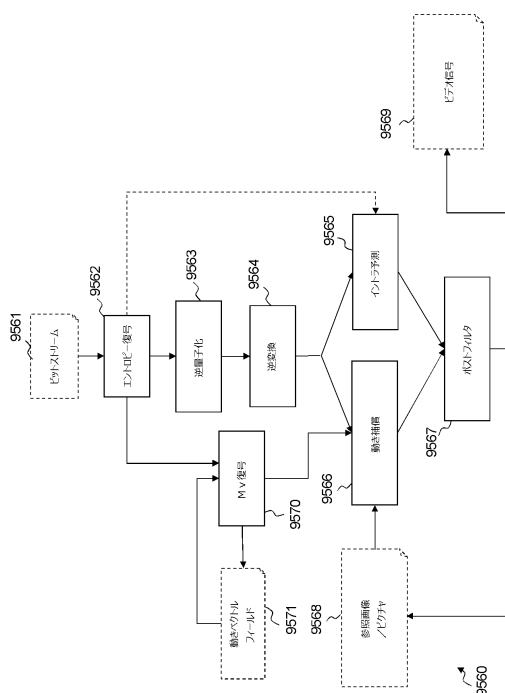


Fig. 9

【図 10】

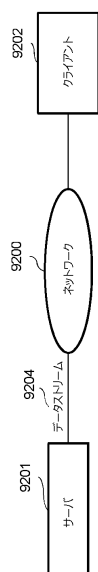


Fig. 10

【図 11】

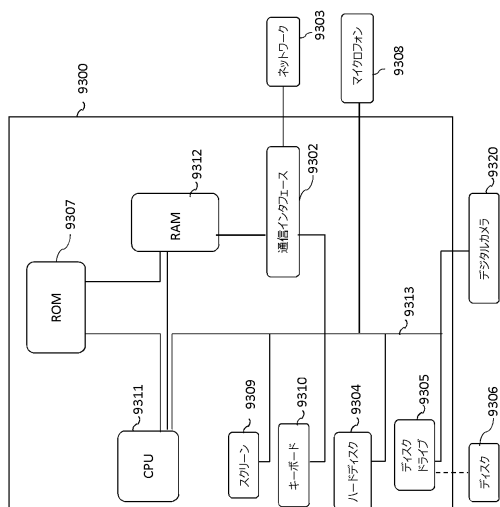


Fig. 11

【図 1 2】

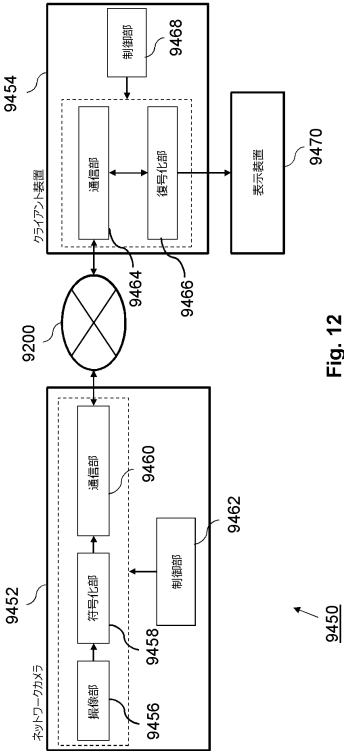


Fig. 12

【図 1 3】

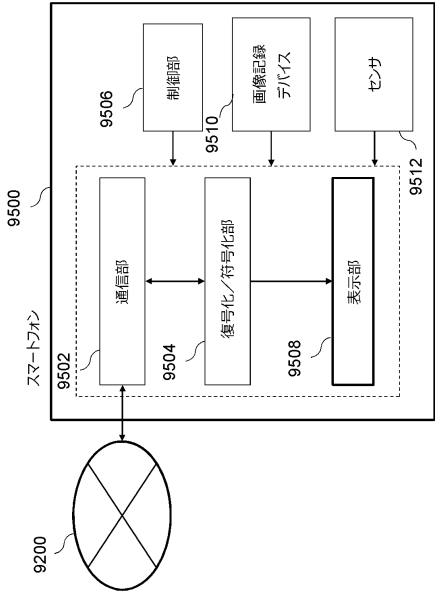


Fig. 13

10

20

30

40

50

フロントページの続き

- ル キヤノン リサーチ センター フランス エス. エー. エス 内
(72)発明者 ラロシュ, ギローム
フランス国 レンヌーアタラント, セデックス セッソンーセヴィニエ 3 5 5 1 7, リュ ドゥ
ラ トゥッシュランベール キヤノン リサーチ センター フランス エス. エー. エス 内
(72)発明者 オンノ, パトリス
フランス国 レンヌーアタラント, セデックス セッソンーセヴィニエ 3 5 5 1 7, リュ ドゥ
ラ トゥッシュランベール キヤノン リサーチ センター フランス エス. エー. エス 内