

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7701271号
(P7701271)

(45)発行日 令和7年7月1日(2025.7.1)

(24)登録日 令和7年6月23日(2025.6.23)

(51)国際特許分類

F I

H 0 4 N 19/70 (2014.01)

H 0 4 N 19/82 (2014.01)

H 0 4 N 19/70

H 0 4 N 19/82

請求項の数 9 (全72頁)

(21)出願番号	特願2021-559943(P2021-559943)	(73)特許権者	520476341
(86)(22)出願日	令和2年4月16日(2020.4.16)		北京字節跳動網絡技術有限公司
(65)公表番号	特表2022-527012(P2022-527012 A)		Beijing Bytedance Network Technology Co., Ltd.
(43)公表日	令和4年5月27日(2022.5.27)		中華人民共和國100041北京市石景山区実興大街30号院3号楼2層B-0035房間
(86)国際出願番号	PCT/CN2020/085075		Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District Beijing 100041 China
(87)国際公開番号	WO2020/211810		
(87)国際公開日	令和2年10月22日(2020.10.22)		
審査請求日	令和3年10月8日(2021.10.8)	(73)特許権者	520477474
審判番号	不服2023-16502(P2023-16502/J 1)		バイトダンス インコーポレイテッド
審判請求日	令和5年9月29日(2023.9.29)		最終頁に続く
(31)優先権主張番号	PCT/CN2019/082855		
(32)優先日	平成31年4月16日(2019.4.16)		
(33)優先権主張国・地域又は機関			

(54)【発明の名称】 映像符号化のためのON適応ループフィルタリング

(57)【特許請求の範囲】
【請求項1】

映像の第1の映像領域と前記映像のビットストリームとの間での変換のため、前記第1の映像領域に対応する第1の構文要素が前記ビットストリームに存在することを判定することと、前記第1の構文要素は、前記第1の映像領域が参照する適応パラメータセットのインデックスを示し、前記第1の映像領域は映像ピクチャまたは映像スライスであり、前記判定することに基づいて前記変換を行うことと、を含み、前記適応パラメータセットが無効である場合、前記第1の構文要素は、前記ビットストリームに存在せず、

前記第1の映像領域のサンプルに対して、異なる方向の複数のサンプルの差分に基づいてフィルタインデックスが導出され、前記適応パラメータセットの輝度フィルタセットと、前記輝度フィルタセットに含まれない別の輝度フィルタとから、前記フィルタインデックスは、前記フィルタインデックスに基づいて特定の輝度フィルタを選択するために使用されることが許容され、

前記映像領域は複数の映像コーディングツリーブロックを含み、前記映像コーディングツリーブロックは複数のM×M個の映像ブロックに分割され、Mは2又は4に等しく、同じフィルタインデックスが、M×M個の映像ブロックの各サンプルに対して適用され、M×M個の映像ブロックごとに、異なる方向の前記複数のサンプルの差分は、1：2のサブサンプリングレートに基づいて導出される、映像データ処理方法。

【請求項 2】

前記適応パラメータセットは、適応ループフィルタリングのための適応ループフィルタリング適応パラメータセットを含む、請求項 1 に記載の方法。

【請求項 3】

前記フィルタインデックスの最大値が N で示され、前記輝度フィルタセットにおける輝度フィルタの数が $N + 1$ 以下である、請求項 1 に記載の方法。

【請求項 4】

前記フィルタインデックスは、固定フィルタセットに含まれる別の特定の輝度フィルタを導出するために使用されることが許容される、請求項 1 に記載の方法。

【請求項 5】

前記変換は、前記第 1 の映像領域を前記ビットストリームに符号化することを含む、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記変換は、前記第 1 の映像領域を前記ビットストリームから復号化することを含む、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

【請求項 7】

処理装置と、命令を備える非一時的メモリとを備える映像データを処理するための装置であって、前記命令が処理装置によって実行されると、前記処理装置に、

映像の第 1 の映像領域と前記映像のビットストリームとの間での変換のため、前記第 1 の映像領域に対応する第 1 の構文要素が前記ビットストリームに存在することを判定することと、前記第 1 の構文要素が、前記第 1 の映像領域が参照する適応パラメータセットのインデックスを示し、前記第 1 の映像領域は映像ピクチャまたは映像スライスであり、

前記判定することに基づいて前記変換を行うことと、を行わせ、
前記適応パラメータセットが無効である場合、前記第 1 の構文要素は、前記ビットストリームに存在せず、

前記第 1 の映像領域のサンプルに対して、異なる方向の複数のサンプルの差分に基づいてフィルタインデックスが導出され、前記適応パラメータセットの輝度フィルタセットと、前記輝度フィルタセットに含まれない他の輝度フィルタとから、前記フィルタインデックスは、前記フィルタインデックスに基づいて特定の輝度フィルタを選択するために使用されることが許容され、

前記映像領域は複数の映像コーディングツリーブロックを含み、前記映像コーディングツリーブロックは複数の $M * M$ 個の映像ブロックに分割され、 M は 2 又は 4 に等しく、

同じフィルタインデックスが、 $M * M$ 個の映像ブロックの各サンプルに対して適用され、
 $M * M$ 個の映像ブロックごとに、異なる方向の前記複数のサンプルの差分は、1 : 2 のサブサンプリングレートに基づいて導出される、

映像データを処理するための装置。

【請求項 8】

命令を記憶する非一時的なコンピュータ可読記憶媒体であって、この命令は、処理装置に、

映像の第 1 の映像領域と前記映像のビットストリームとの間での変換のため、前記第 1 の映像領域に対応する第 1 の構文要素が前記ビットストリームに存在することを判定することと、前記第 1 の構文要素が、前記第 1 の映像領域が参照する適応パラメータセットのインデックスを示し、前記第 1 の映像領域は映像ピクチャまたは映像スライスであり、

前記判定することに基づいて前記変換を行うことと、を行わせ、
前記適応パラメータセットが無効である場合、前記第 1 の構文要素は、前記ビットストリームに存在せず

前記第 1 の映像領域のサンプルに対して、異なる方向の複数のサンプルの差分に基づいてフィルタインデックスが導出され、前記適応パラメータセットの輝度フィルタセットと、前記輝度フィルタセットに含まれない他の輝度フィルタとから、前記フィルタインデックスは、前記フィルタインデックスに基づいて特定の輝度フィルタを選択され、

10

20

30

40

50

前記映像領域は複数の映像コーディングツリーブロックを含み、前記映像コーディングツリーブロックは複数の $M \times M$ 個の映像ブロックに分割され、 M は2又は4に等しく、同じフィルタインデックスが、 $M \times M$ 個の映像ブロックの各サンプルに対して適用され、 $M \times M$ 個の映像ブロックごとに、異なる方向の前記複数のサンプルの差分は、1:2のサブサンプリングレートに基づいて導出される、非一時的なコンピュータ可読記憶媒体。

【請求項9】

映像のビットストリームを記憶する方法であって、前記方法は、前記映像の第1の映像領域に対して、前記第1の映像領域に対応する第1の構文要素が前記ビットストリームに存在することを判定することと、前記第1の構文要素は、前記第1の映像領域が参照する適応パラメータセットのインデックスを示し、前記第1の映像領域は映像ピクチャまたは映像スライスであり、前記判定することに基づいて前記ビットストリームを生成することと、前記ビットストリームを非一時的なコンピュータ可読記録媒体に記憶することと、を含み、前記適応パラメータセットが無効である場合、前記第1の構文要素は、前記ビットストリームに存在せず、

前記第1の映像領域のサンプルに対して、異なる方向の複数のサンプルの差分に基づいてフィルタインデックスが導出され、前記適応パラメータセットの輝度フィルタセットと、前記輝度フィルタセットに含まれない他の輝度フィルタとから、前記フィルタインデックスは、前記フィルタインデックスに基づいて特定の輝度フィルタを選択するために使用されることが許容され、

前記映像領域は複数の映像コーディングツリーブロックを含み、前記映像コーディングツリーブロックは複数の $M \times M$ 個の映像ブロックに分割され、 M は2又は4に等しく、同じフィルタインデックスが、 $M \times M$ 個の映像ブロックの各サンプルに対して適用され、 $M \times M$ 個の映像ブロックごとに、異なる方向の前記複数のサンプルの差分は、1:2のサブサンプリングレートに基づいて導出される、方法。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本願は、2019年4月16日出願の国際特許出願第PCT/CN2019/082855号の優先権と利益を主張する、2020年4月16日出願の国際特許出願第PCT/CN2020/085075号に基づく。米国法に基づくすべての目的のために、上記出願の開示全体は、本明細書の開示の一部として参照により援用される。

【0002】

この特許文献は、映像符号化技術、デバイスおよびシステムに関する。

【背景技術】

【0003】

映像圧縮の進歩にもかかわらず、デジタル映像は、依然として、インターネット及び他のデジタル通信ネットワークにおいて最大の帯域幅の使用量を占めている。映像を受信及び表示することが可能である接続されたユーザ機器の数が増加するにつれ、デジタル映像の使用に対する帯域幅需要は増大し続けることが予測される。

【発明の概要】

【0004】

デジタル映像符号化に関し、具体的には、映像符号化のための適応ループフィルタリングに関するデバイス、システム、および方法について記載する。記載された方法は、既存の映像符号化規格（例えば、高効率映像符号化（HEVC））および将来の映像符号化規格（例えば、汎用映像符号化（VVC））、又はコーデックの両方に適用され得る。

【0005】

映像符号化規格は、主に周知のITU-TおよびISO/IEC規格の開発によって発展してきた。ITU-TはH.261とH.263を作り、ISO/IECはMPEG-1とMPEG-4 Visualを作り、両団体はH.262/MPEG-2 VideoとH.264/MPEG-4 AVC (Advanced Video Coding)とH.265/HEVC規格を共同で作った。H.262以来、映像符号化規格は、時間予測と変換符号化が利用されるハイブリッド映像符号化構造に基づく。HEVCを超えた将来の映像符号化技術を探索するため、2015年には、VCEGとMPEGが共同でJVET (Joint Video Exploration Team)を設立した。それ以来、多くの新しい方法がJVETによって採用され、JEM (Joint Exploration Mode)と呼ばれる参照ソフトウェアに組み込まれてきた。2018年4月には、VCEG (Q6/16)とISO/IEC JTC1 SC29/WG11 (MPEG)の間にJoint Video Expert Team (JVET)が発足し、HEVCと比較して50%のビットレート削減を目標にVVV規格の策定に取り組んでいる。

10

【0006】

1つの代表的な態様において、開示される技術は、映像処理の方法を提供するために使用してもよい。この方法は、映像の現在の映像ブロックに対して、フィルタ係数を使用し、少なくとも1つの中間結果を伴う2つ以上の演算を含むフィルタリング処理を行うことと、前記少なくとも1つの中間結果にクリッピング演算を適用することと、前記少なくとも1つの中間結果に基づいて、前記現在の映像ブロックと前記映像のビットストリーム表現との間での変換を行うこととを含み、前記少なくとも1つの中間結果は、前記フィルタ係数の重み付けの合計と、前記現在の映像ブロックの現在のサンプルと前記現在のサンプルの近傍のサンプルとの間の差とに基づく。

20

【0007】

別の代表的な態様では、開示される技術は、映像処理の方法を提供するために使用してもよい。この方法は、映像の現在の映像ブロックを前記映像のビットストリーム表現に符号化することであって、前記現在の映像ブロックは、適応ループフィルタ (ALF) で符号化される、符号化することと、時間的適応フィルタの1つ以上のセットの可用性または使用に基づいて、前記ビットストリーム表現における前記時間的適応フィルタの前記1つ以上のセット内の時間的適応フィルタのセットの指示を選択的に含むことと、を含む。

30

【0008】

さらに別の代表的な態様では、開示される技術は、映像処理の方法を提供するために使用してもよい。この方法は、映像のビットストリーム表現における時間的適応フィルタのセットの指示に基づいて、適応ループフィルタ (ALF) で符号化される映像の現在の映像ブロックに適用可能な時間的適応フィルタの前記セットを備える1つ以上のセットの前記時間的適応フィルタの可用性または使用を判定することと、前記判定することに基づいて、時間的適応フィルタの前記セットを選択的に適用することによって、前記ビットストリーム表現から復号化された現在の映像ブロックを生成することと、を含む。

【0009】

さらに別の代表的な態様では、開示される技術は、映像処理の方法を提供するために使用してもよい。この方法は、適応ループフィルタで符号化された現在の映像ブロックに対して、利用可能な時間的ALF係数セットに基づいて、時間的適応ループフィルタリング (ALF) 係数セットの数を判定することであって、前記利用可能な時間的ALF係数セットは、前記判定する前に符号化または復号化されており、前記ALF係数セットの数は、前記現在の映像ブロックを構成するタイルグループ、タイル、スライス、ピクチャ、符号化ツリーブロック (CTB)、または映像ユニットに使用される、判定することと、時間的ALF係数セットの前記数に基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現の間での変換を行うことと、を含む。

40

【0010】

さらに別の代表的な態様では、開示される技術は、映像処理の方法を提供するために使

50

用してもよい。この方法は、映像の現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域のヘッダにおける適応ループフィルタリング (ALF) の指示が、前記ビットストリーム表現に関連付けられた適応パラメータセット (APS) ネットワーク抽象化層 (NAL) ユニットにおける ALF の指示と等しいことを判定することと、前記変換を行うことと、を含む。

【0011】

さらに別の代表的な態様では、開示される技術は、映像処理の方法を提供するために使用してもよい。この方法は、現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域で使用される適応ループフィルタのタイプに基づいて、非線形適応ループフィルタリング (ALF) 動作を選択的に有効化することと、前記選択的に有効化した後に、前記変換を行うことと、を含む。

10

【0012】

さらに別の代表的な態様において、上記方法は、処理装置が実行可能なコードの形式で実施され、コンピュータ可読プログラム媒体に記憶される。

【0013】

さらに別の代表的な態様において、上述した方法を行うように構成された、または動作可能なデバイスが開示される。この装置は、この方法を実装するようにプログラムされた処理装置を含んでもよい。

【0014】

さらに別の代表的な態様において、映像デコード装置は、本明細書で説明されるような方法を実装してもよい。

20

【0015】

開示される技術の上記および他の態様および特徴は、図面、説明および特許請求の範囲でより詳細に説明される。

【図面の簡単な説明】

【0016】

【図1】映像符号化のためのエンコーダブロック図の例を示す。

【図2A】形状変換に基づく適応ループフィルタ (GALF) フィルタ形状の例を示す。

【図2B】形状変換に基づく適応ループフィルタ (GALF) フィルタ形状の例を示す。

【図2C】形状変換に基づく適応ループフィルタ (GALF) フィルタ形状の例を示す。

30

【図3】GALF エンコーダ決定のためのフローグラフの例を示す。

【図4A】適応ループフィルタ (ALF) 分類のための例示的なサブサンプリングされたラブラシアン計算を示す。

【図4B】適応ループフィルタ (ALF) 分類のための例示的なサブサンプリングされたラブラシアン計算を示す。

【図4C】適応ループフィルタ (ALF) 分類のための例示的なサブサンプリングされたラブラシアン計算を示す。

【図4D】適応ループフィルタ (ALF) 分類のための例示的なサブサンプリングされたラブラシアン計算を示す。

【図5】輝度フィルタの形状の例を示す。

40

【図6】ワイド映像グラフィックアレイ (WVGA) シーケンスの領域分割の例を示す。

【図7】再整形を伴う復号化の流れの例示的なフロー図を示す。

【図8】双方向オブティカルフロー (BIO) アルゴリズムで使用するオブティカルフローの軌跡の例を示す。

【図9A】ブロック拡張なしの双方向オブティカルフロー (BIO) アルゴリズムを使用した例示的なスナップショットを示す。

【図9B】ブロック拡張なしの双方向オブティカルフロー (BIO) アルゴリズムを使用した例示的なスナップショットを示す。

【図10】オブティカルフローを用いた予測微調整 (PROF) の例を示す。

【図11A】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図で

50

ある。

【図 1 1 B】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図である。

【図 1 1 C】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図である。

【図 1 1 D】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図である。

【図 1 1 E】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図である。

【図 1 1 F】本開示の技術による適応ループフィルタリング方法の一例を示すフロー図である。

10

【図 1 2】本特許明細書に記載されるビジュアルメディアの復号化又はビジュアルメディアの符号化技術を実現するためのハードウェアプラットフォームの一例を示すブロック図である。

【図 1 3】開示された技術を実装することができる例示的な映像処理システムを示すブロック図である。

【発明を実施するための形態】

【0017】

より高い解像度の映像の需要が増大しているため、近代技術において、映像符号化および技術は、遍在している。ビデオコーデックは、一般的に、デジタル映像を圧縮又は展開する電子回路又はソフトウェアを含み、より高い符号化効率を提供するように絶えず改良されている。ビデオコーデックは、非圧縮映像を圧縮フォーマットに変換する、又はその逆である。映像の品質、映像を表現するために使用されるデータの数（ビットレートで決まる）、エンコーディングおよびデコーディングアルゴリズムの複雑性、データの損失およびエラーに対する敏感さ、編集のしやすさ、ランダムアクセス、およびエンドツーエンドの遅延（待ち時間）の間には複雑な関係がある。この圧縮フォーマットは、通常、標準的な映像圧縮仕様、例えば、高効率映像符号化（HEVC）規格（H.265またはMPEG-H Part 2としても知られている）、完成させるべき汎用映像符号化（VVC）規格、または他の現在および/または将来の映像符号化基準に準拠する。

20

【0018】

いくつかの実施形態において、将来の映像符号化技術は、共同探索モデル（JEM）として知られる参照ソフトウェアを使用して探索される。JEMでは、サブブロックベースの予測は、アフィン予測、代替時間的動きベクトル予測（ATMVP）、空間的・時間的動きベクトル予測（STMVP）、双方向オブティカルフロー（BIO）、フレームレートアップ変換（FRUC）、ローカル適応動きベクトル解像度（LAMVR）、オーバーラップブロック動き補償（OBMC）、ローカル照明補償（LIC）、デコーダ側動きベクトル改良（DMVR）などの、いくつかの符号化ツールで適用されている。

30

【0019】

開示される技術の実施形態は、ランタイム性能を向上させるために、既存の映像符号化規格（例えば、HEVC、H.265）および将来の規格に適用されてもよい。本明細書では、説明の可読性を向上させるために章の見出しを使用しており、説明または実施形態（および/または実装形態）をそれぞれの章のみに限定するものではない。

40

【0020】

1. 色空間およびクロマサブサンプリングの実施例

色空間はカラーモデル（またはカラーシステム）としても知られ、色の範囲を数字のタプル（tuple）として簡単に記述する抽象的な数学モデルであり、一般的に、3または4つの値または色成分（例えばRGB）である。基本的には、色空間は座標系とサブ空間とを精緻化したものである。

【0021】

映像圧縮の場合、最も頻繁に使用される色空間は、YCbCrおよびRGBである。

50

【 0 0 2 2 】

$YCbCr$ 、 $Y'CbCr$ 、または $Y_Pb/CbPr/Cr$ は、 $YCBCR$ または $Y'CBCR$ と呼ばれ、映像およびデジタル写真システムのカラー画像パイプラインの一部として使用される色空間のファミリーである。 Y' は輝度成分であり、 Cb および Cr は青色差および赤色差クロマ成分である。 Y' （素数を有する）は Y とは区別され、 Y は輝度であり、ガンマ補正された RGB 原色に基づいて光強度が非線形に符号化されることを意味する。

【 0 0 2 3 】

クロマサブサンプリングは、人間の視覚システムが、輝度よりも色差の方が知覚が低いことを利用して、輝度情報よりもクロマ情報の方が解像度が低くなるように実装して画像を符号化する方法である。

10

【 0 0 2 4 】

1. 1 4 : 4 : 4 カラーフォーマット

3つの $Y'CbCr$ 成分の各々は、同じサンプルレートを有し、従って、クロマサブサンプリングは存在しない。このスキームは、ハイエンドのフィルムスキャナおよび映画のポストプロダクションに使用されることがある。

【 0 0 2 5 】

1. 2 4 : 2 : 2 カラーフォーマット

2つのクロマ成分は、輝度のサンプルレートの半分でサンプリングされ、例えば、水平クロマ解像度が半分にされる。これにより、視覚的にほとんどまたは全く差がなく、非圧縮の映像信号の帯域幅を $1/3$ に低減することができる。

20

【 0 0 2 6 】

1. 3 4 : 2 : 0 カラーフォーマット

4 : 2 : 0 では、水平サンプリングは $4 : 1 : 1$ に比べて2倍になるが、このスキームでは Cb および Cr チャネルを各1行おきのラインでのみサンプリングするので、垂直解像度は半分になる。従って、データレートは同じである。 Cb および Cr はそれぞれ水平および垂直方向の両方向に2倍にサブサンプリングされる。異なる水平および垂直位置を有する 4 : 2 : 0 スキームの3つの変形がある。

【 0 0 2 7 】

MPEG-2において、 Cb および Cr は水平方向に共座している。 Cb 、 Cr は垂直方向の画素間に位置する（格子間に位置する）。

30

【 0 0 2 8 】

JPEG/JFIFにおいて、H.261、およびMPEG-1、 Cb 、および Cr は、交互の輝度サンプルの中間の格子間に位置する。

【 0 0 2 9 】

4 : 2 : 0 DVにおいて、 Cb および Cr は、水平方向に共座している。垂直方向において、それらは交互に並ぶ線上に共座している。

【 0 0 3 0 】

2 典型的な映像コーデックの符号化フローの例

図1は、3つのインループフィルタリングブロック、すなわち非ブロック化フィルタ（DF）、サンプル適応オフセット（SAO）およびALFを含むVVCのエンコードブロック図の例を示す。DF（予め規定されたフィルタを使用する）とは異なり、SAOおよびALFは、現在の画像のオリジナルサンプルを利用し、オフセットを追加し、且つ有限インパルス応答（FIR）フィルタを適用することによって、オフセットおよびフィルタ係数を信号通知する符号化側情報とともに、元のサンプルと再構成サンプルとの間の平均二乗誤差をそれぞれ低減する。ALFは、各ピクチャの最後の処理ステージに位置し、前のステージで生成されたアーチファクトを捕捉し、修正しようとするツールと見なすことができる。

40

【 0 0 3 1 】

3 JEMにおける形状変換に基づく適応ループフィルタの例

50

JEMにおいて、ブロックに基づくフィルタ適応「3」を用いた形状変換に基づく適応ループフィルタ（GALF）が適用される。輝度成分は、局所勾配の方向および働きに基づいて、2×2ブロックごとに25個のフィルタのうち1つを選択する。

【0032】

3.1 フィルタ形状の例

本願において、輝度成分として、最大3つのダイヤモンドフィルタ形状（図2A、図2B、図2Cにそれぞれ示すように、5×5ダイヤモンド、7×7ダイヤモンド、9×9ダイヤモンド）を選択することができる。輝度成分に使用されるフィルタ形状を示すために、ピクチャレベルでインデックスが信号通知される。1つのピクチャにおけるクロマ成分に対して、5×5ダイヤモンド形状が常に使用される。

【0033】

3.1.1 ブロック区分

各2×2ブロックを25個のクラスのうち1つに分類する。分類インデックスCは、その方向性DおよびアクティビティA[^]の量子化値に基づいて、以下のように導出される。

【0034】

【数1】

$$C = 5D + \hat{A}. \quad (1)$$

【0035】

DおよびA[^]を計算するために、まず、1-Dラプラシアンを使用して、水平、垂直および2つの対角線方向の勾配を計算する。

【0036】

【数2】

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad (2)$$

【0037】

【数3】

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad (3)$$

【0038】

【数4】

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (4)$$

【0039】

【数5】

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (5)$$

10

20

30

40

50

【 0 0 4 0 】

i および j は、 2×2 ブロックの左上のサンプルの座標を表し、 $R(i, j)$ は、座標 (i, j) において再構成されたサンプルを示す。

そして、水平方向および垂直方向の勾配の D 最大値および D 最小値を以下のように設定する。

【 0 0 4 1 】

【 数 6 】

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v), \quad (6)$$

10

【 0 0 4 2 】

および 2 つの対角線方向の勾配の最大値および最小値は、以下のように設定される。

【 0 0 4 3 】

【 数 7 】

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}), \quad (7)$$

【 0 0 4 4 】

指向性 D の値を導出するために、これらの値を互いに且つ 2 つの閾値 t_1 および t_2 と比較する。

ステップ 1 .

【 0 0 4 5 】

【 数 8 】

$$g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min} \text{ と } g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$$

【 0 0 4 6 】

の両方が真である場合、0 に設定される。

ステップ 2 .

【 0 0 4 7 】

【 数 9 】

$$g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min},$$

【 0 0 4 8 】

の場合、ステップ 3 から続け、あるいは、ステップ 4 から続ける。

ステップ 3 .

【 0 0 4 9 】

【 数 10 】

$$g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$$

【 0 0 5 0 】

である場合、D は 2 に設定され、あるいは、D は 1 に設定される。

ステップ 4 .

50

【 0 0 5 1 】

【 数 1 1 】

$$g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$$

【 0 0 5 2 】

である場合、Dは4に設定され、あるいは、Dは3に設定される。

アクティビティ値Aは、以下のように計算される。

【 0 0 5 3 】

【 数 1 2 】

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (8)$$

10

【 0 0 5 4 】

Aをさらに0～4の範囲に量子化し、量子化された値をA^とする。

ピクチャにおける両クロマ成分に対して、分類方法は適用されず、即ち、単一のALF係数のセットが各クロマ成分に対して適用される。

【 0 0 5 5 】

20

3.1.2 フィルタ係数の幾何学的変換

各2×2輝度ブロックをフィルタリングする前に、そのブロックに対して計算された勾配値に基づいて、フィルタ係数f(k, l)に回転または対角線および垂直方向の反転等の幾何学的変換を施す。これは、これらの変換をフィルタ支持領域内のサンプルに適用することに等しい。その考えは、ALFが適用される異なるブロックを、それらの方向性を揃えることによって、より類似させることである。

【 0 0 5 6 】

対角線、垂直方向の反転および回転を含む3つの幾何学的変換を紹介する。

【 0 0 5 7 】

【 数 1 3 】

30

対角線: $f_D(k, l) = f(l, k),$

垂直方向の反転: $f_V(k, l) = f(k, K - l - 1), \quad (9)$

回転: $f_R(k, l) = f(K - l - 1, k).$

【 0 0 5 8 】

ここで、Kはフィルタのサイズであり、0 ≤ k, l ≤ K - 1が係数座標であり、位置(0, 0)は左上隅にあり、位置(K - 1, K - 1)は右下隅にある。この変換は、そのブロックに対して計算された勾配値に基づいて、フィルタ係数f(k, l)に適用される。変換と4方向の4つの勾配との関係を表1にまとめる。

40

【 0 0 5 9 】

【表 1】

表 1 : 1 つのブロックに対して計算された勾配と変換とのマッピング

勾配値	変換
$g_{d2} < g_{d1}$ および $g_h < g_v$	変換なし
$g_{d2} < g_{d1}$ および $g_v < g_h$	対角線
$g_{d1} < g_{d2}$ および $g_h < g_v$	垂直方向の反転
$g_{d1} < g_{d2}$ および $g_v < g_h$	回転

【0060】

10

3.1.3 フィルタパラメータの信号通知

JEMにおいて、GALFフィルタパラメータは、第1のCTUのために、すなわち、スライスヘッダの後且つ第1のCTUのSAOパラメータの前に信号通知される。最大25組の輝度フィルタ係数を信号通知することができる。ビットオーバーヘッドを低減するために、異なる分類のフィルタ係数をマージすることができる。また、参照ピクチャのGALF係数を記憶し、現在のピクチャのGALF係数として再利用することができる。現在のピクチャは、参照ピクチャのために記憶されたGALF係数を使用し、GALF係数信号通知を回避することを選択してもよい。この場合、1つの参照ピクチャへのインデックスのみが信号通知され、記憶されている示された参照ピクチャのGALF係数が現在のピクチャに継承される。

20

【0061】

GALF時間的予測をサポートするために、GALFフィルタセットの候補リストが保持される。新しいシーケンスを復号化する開始時は、候補リストは空である。1つのピクチャを復号化した後、対応するフィルタのセットを候補リストに加えてもよい。候補リストのサイズが最大許容値（すなわち、現在のJEMでは6）に達すると、新しい1組のフィルタが、最も古いセットを復号化の順序に上書きし、すなわち、先入れ先出し（FIFO）規則を適用して候補リストを更新する。重複を回避するために、対応するピクチャがGALF時間的予測を使用しない場合、1つのセットのみをリストに追加することができる。時間的スケーラビリティをサポートするために、複数のフィルタセットの候補リストがあり、各候補リストは1つの時間層に関連付けられる。具体的には、時間層インデックス（TempIdx）が割り当てられた各アレイは、TempIdxが小さい、前回復号化されたピクチャのフィルタセットを構成してもよい。例えば、k番目の配列は、kに等しいTempIdxに関連付けられるように割り当てられ、それは、TempIdxがk以下のピクチャからのフィルタセットのみを含む。特定のピクチャを符号化した後、このピクチャに関連付けられたフィルタセットを使用して、等しいまたはより高いTempIdxに関連付けられた配列を更新する。

30

【0062】

GALF係数の時間的予測は、信号通知オーバーヘッドを最小限に抑えるために、インター符号化されたフレームに使用される。イントラフレームの場合、時間的予測は利用できず、各クラスに1組の16個の固定フィルタが割り当てられる。固定フィルタの使用を示すために、各クラスのためのフラグが信号通知され、必要に応じて、選択された固定フィルタのインデックスが信号通知される。所与のクラスに対して固定フィルタを選択した場合でも、このクラスに対して適応フィルタ $f(k, l)$ の係数を送信することができ、この場合、再構成画像に適用されるフィルタの係数は両方の係数セットの合計となる。

40

【0063】

輝度成分のフィルタリング処理は、CUレベルで制御することができる。GALFがCUの輝度成分に適用されるかどうかを示すために、1つのフラグが信号通知される。クロマ成分の場合、GALFが適用されるかどうかは、ピクチャレベルでのみ示す。

【0064】

3.1.4 フィルタリング処理

50

デコーダ側において、1つのブロックに対してGALFが有効化されると、このブロック内の各サンプル $R(i, j)$ がフィルタリングされ、その結果、以下に示すように、サンプル値 $R'(i, j)$ が得られる。ここで、 L は、フィルタ長を表し、 f_m, n は、フィルタ係数を表し、 $f(k, l)$ は、復号化されたフィルタ係数を表す。

【0065】

【数14】

$$R'(i, j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k, l) \times R(i+k, j+l) \quad (10)$$

10

【0066】

3.1.5 エンコーダ側フィルタパラメータの判定処理

図3には、GALFのための全体的なエンコーダ決定処理が示されている。各CUの輝度サンプルに対して、エンコーダは、GALFが適用され、かつ適切な信号通知フラグがスライスヘッダに含まれているか否かを決定する。クロマサンプルの場合、フィルタを適用する決定は、CUレベルではなくピクチャレベルに基づいて行われる。さらに、ピクチャのためのクロマGALFは、このピクチャのために輝度GALFが有効化されている場合にのみチェックされる。

【0067】

4 VVCにおける形状変換に基づく適応ループフィルタの例

20

現在のVVCにおけるGALFの設計は、JEMにおける設計に比べ、以下のような大きな変化を有している。

1) 適応フィルタ形状を除去する。輝度成分に対しては 7×7 フィルタ形状のみが許可され、クロマ成分に対しては 5×5 フィルタ形状のみが許可される。

2) ALFパラメータの時間的予測および固定フィルタからの予測は、両方とも除去される。

3) 各CTUに対して、ALFが有効化されるかまたは無効化されるかどうかに関わらず、1ビットのフラグが信号通知される。

4) クラスインデックスの計算は、 2×2 の代わりに、 4×4 レベルで行われる。また、JVET-L0147で提案されているように、ALF分類のためのサブサンプリングされたラプラシアン計算方法が利用される。具体的には、1つのブロック内の各サンプルごとに水平/垂直/45対角線/135度勾配を計算する必要がない。その代わりに、1:2サブサンプリングが利用される。

30

【0068】

5 AVS2における領域に基づく適応ループフィルタの例

ALFは、インループフィルタリングの最終段階である。この処理には2つの段階がある。第1の段階は、フィルタ係数の導出である。フィルタ係数をトレーニングするために、エンコーダは、再構成された輝度成分の画素を16個の領域に分類し、wiener-hopf方程式を使用して、カテゴリ毎に1つのフィルタ係数のセットをトレーニングし、元のフレームと再構成されたフレームとの間の平均二乗誤差を最小限に抑える。これらの16個のフィルタ係数のセット間の冗長性を低減するために、エンコーダは、ひずみ率性能に基づいてそれらを適応的にマージする。その最大値において、16個の異なるフィルタセットを輝度成分に割り当てることができ、1つのフィルタセットのみをクロミナス成分に割り当てることができる。第2の段階は、フレームレベルおよびLCUレベルの両方を含むフィルタ決定である。まず、エンコーダは、フレームレベル適応ループフィルタリングを行うかどうかを決定する。フレームレベルALFがオンである場合、エンコーダは、LCUレベルALFを行うかどうかをさらに決定する。

40

【0069】

5.1 フィルタ形状

AVS-2に適応されたフィルタ形状は、 7×7 の十字形であり、輝度成分およびクロ

50

マ成分の両方について図5に示すように、 3×3 の正方形を重ね合わせたものである。図5中の四角はそれぞれサンプルに対応する。従って、合計17個のサンプルを使用して、位置C8のサンプルのためのフィルタリングされた値を導出する。係数を送信するオーバーヘッドを考慮して、点対称フィルタは、9つの係数だけを残して、 $\{C0, C1, \dots, C8\}$ を利用し、これにより、フィルタリングにおけるフィルタ係数の数を半分に減らすと共に乗算の数を減らす。この点対称フィルタは、1つのフィルタリングされたサンプルの計算の半分の減らすこともでき、例えば、1つのフィルタリングされたサンプルに対して9回の乗算および14回の加算演算のみを行う。

【0070】

5.2 領域別適応マージ

異なる符号化エラーに適応するために、AVS-2は、輝度成分のために、領域に基づく複数の適応ループフィルタを採用する。輝度成分は、図6に示すように、各基本領域が最大符号化ユニット(LCU)の境界に位置合わせされた16個の略等分したサイズの基本領域に分割され、領域毎に1つのウィナーフィルタを導出する。より多くのフィルタを使用するほど、より多くの歪みが低減されるが、これらの係数を符号化するために使用されるビットは、フィルタの数とともに増加する。最良のレートひずみ率を実現するために、これらの領域を、同じフィルタ係数を共有する、より少なく、より大きい領域にマージすることができる。マージ処理を簡単にするために、各領域には、画像の前置相関に基づいて修正されたヒルベルト順に従ってインデックスが割り当てられる。ひずみ率コストに基づいて、インデックスが連続する2つの領域をマージすることができる。

【0071】

領域間のマッピング情報は、デコーダに信号通知されるべきである。AVS-2において、基本領域の数はマージ結果を示すために使用され、フィルタ係数はその領域の順に従って順次圧縮される。例えば、 $\{0, 1\}$ 、 $\{2, 3, 4\}$ 、 $\{5, 6, 7, 8, 9\}$ および左側の基本領域をそれぞれ1つの領域にマージする場合、このマージマップを表すために3つの整数のみを符号化する(即ち、2, 3, 5)。

【0072】

5.3 副情報の信号通知

複数のスイッチフラグも使用される。シーケンス切替フラグ`adaptive_loop_filter_enable`は、シーケンス全体に適応ループフィルタを適用するかどうかを制御するために使用するフラグである。画像切り替えフラグ`picture_allf_enable[i]`は、対応するi番目の画像成分に対してALFを適用するかどうかを制御する。`picture_allf_enable[i]`が有効になっている場合にのみ、その色成分に対応するLCUレベルのフラグおよびフィルタ係数を送信する。LCUレベルフラグ、`lcu_allf_enable[k]`は、対応するk番目のLCUに対してALFが有効化されるかどうかを制御し、スライスデータにインターリーブされる。異なるレベルの調整済みフラグの決定はすべて、ひずみ率コストに基づく。柔軟性が高いので、ALFは、符号化効率をさらに有意に向上させる。

【0073】

いくつかの実施形態において、輝度成分のために、16個までのフィルタ係数のセットが存在可能である。

【0074】

いくつかの実施形態において、各クロマ成分(CbとCr)に対して1つのフィルタ係数セットを送信してもよい。

【0075】

6 VTM-4におけるGALF

VTM4.0において、適応ループフィルタのフィルタリング処理は、以下のように行う。

【0076】

【数15】

10

20

30

40

50

$$O(x, y) = \sum_{(i,j)} w(i, j) \cdot I(x + i, y + j) \quad (11)$$

【 0 0 7 7 】

ここで、サンプル $I(x + i, y + j)$ は入力サンプルであり、 $O(x, y)$ はフィルタリングされた出力サンプル（即ち、フィルタ結果）であり、 $w(i, j)$ はフィルタ係数を表す。実際において、V T M 4 . 0 は、固定小数点精度計算のために整数演算を使用して実装される。

【 0 0 7 8 】

【 数 1 6 】

10

$$O(x, y) = \left(\sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i, j) \cdot I(x + i, y + j) + 64 \right) \gg 7 \quad (12)$$

【 0 0 7 9 】

ここで、 L はフィルタ長を表し、 $w(i, j)$ は固定小数点精度におけるフィルタ係数である。

20

【 0 0 8 0 】

7 非線形適応ループフィルタリング (A L F)

7 . 1 フィルタリングの再形成

式 (1 1) は、符号化効率に影響を及ぼすことなく、以下の式で再定式化することができる。

【 0 0 8 1 】

【 数 1 7 】

$$O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x + i, y + j) - I(x, y)) \quad (13)$$

30

【 0 0 8 2 】

ここで、 $w(i, j)$ は、式 (1 1) におけるフィルタ係数と同じである [例外 $w(0, 0)$ 、式 (1 3) においては 1 に等しいが、式 (1 1) において、

【 0 0 8 3 】

【 数 1 8 】

$$1 - \sum_{(i,j) \neq (0,0)} w(i, j)$$

40

【 0 0 8 4 】

に等しい]。

【 0 0 8 5 】

7 . 2 修正されたフィルタ

上記 (1 3) のフィルタ式を使用することで、単純なクリッピング関数を使用して、近傍のサンプル値 ($I(x + i, y + j)$) が現在のサンプル値 ($I(x, y)$) のフィルタリングと異なり過ぎている場合に、その影響を低減することで、非線形性を容易に導入し、A L F をより効率的にする。

【 0 0 8 6 】

50

この提案において、A L Fフィルタは、以下のように修正される。

【 0 0 8 7 】

【 数 1 9 】

$$O'(x,y) = I(x,y) + \sum_{(i,j) \neq (0,0)} w(i,j) \cdot K(I(x+i,y+j) - I(x,y), k(i,j)) \quad (14)$$

【 0 0 8 8 】

ここで、 $K(d, b) = \min(b, \max(-b, d))$ はクリッピング関数であり、 $k(i, j)$ はクリッピングパラメータであり、これは (i, j) フィルタ係数に依存する。エンコーダは、最適化を行い、最良の $k(i, j)$ を見出す。

10

【 0 0 8 9 】

J V E T - N 0 2 4 2 の実装形態において、1つのA L Fフィルタごとにクリッピングパラメータ $k(i, j)$ を指定し、1つのフィルタ係数ごとに1つのクリッピング値を信号通知する。これは、1つの輝度フィルタ当たりビットストリームにおいて、最大12個のクリッピング値を信号通知することができ、クロマフィルタに対しては最大6個のクリッピング値を信号通知することができることを意味する。

【 0 0 9 0 】

信号通知コストおよびエンコーダの複雑性を制限するために、クリッピング値の評価は、小さなセットの可能な値に限定する。本提案では、I N T E R および I N T R A タイルグループに対して同じ4つの固定値のみを使用する。

20

【 0 0 9 1 】

局所的な差の分散は、輝度の場合、クロマの場合よりも大きいことが多いので、輝度フィルタおよびクロマフィルタの2つの異なるセットを使用する。各セットに最大サンプル値（ここでは、10ビットのビット深度の場合、1024）を含め、必要でない場合、クリッピングを無効にすることができる。

【 0 0 9 2 】

J V E T - N 0 2 4 2 試験に使用したクリッピング値のセットを表2に示す。4つの値は、対数ドメインにおいて、輝度についてのサンプル値（10ビットで符号化される）の全範囲を、およびクロマについての4～1024の範囲をほぼ等分することによって選択された。

30

【 0 0 9 3 】

より正確には、クリッピング値の輝度テーブルは、以下の式によって得られた。

【 0 0 9 4 】

【 数 2 0 】

$$AlfClip_L = \left\{ \text{round} \left(\left(\left(M \right)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{l_0} \text{ and } N=4.$$

40

【 0 0 9 5 】

同様に、クリッピング値のクロマテーブルは、以下の式に従って取得される。

【 0 0 9 6 】

【 数 2 1 】

$$AlfClip_C = \left\{ \text{round} \left(A \cdot \left(\left(\frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{l_0}, N=4 \text{ and } A=4.$$

【 0 0 9 7 】

50

【表 2】

表 2：認可されたクリッピング値

	イントラ／インタータイルグループ
輝度	{1024,181,32,6}
クロマ	{1024,161,25,4}

【0098】

選択されたクリッピング値は、上記表 2 のクリッピング値のインデックスに対応するゴロム符号化方式を使用して、「a l f _ d a t a」構文要素に符号化される。この符号化方式は、フィルタインデックスの符号化方式と同じである。

10

【0099】

8 J V E T - N 0 4 1 5 における C T U に基づく A L F

スライスレベルの時間的フィルタ。 V T M 4 には適応パラメータセット (A P S) が採用された。各 A P S は、1つのセットの信号通知された A L F フィルタを含み、最大 32 個の A P S がサポートされる。本提案では、スライスレベルの時間的フィルタを試験する。1つのタイルグループは、A P S からの A L F 情報を再利用することにより、オーバーヘッドを低減することができる。A P S は、先入れ先出し (F I F O) バッファとして更新される。

20

【0100】

C T B に基づく A L F。輝度成分のために、A L F が輝度 C T B に適用される場合、16 個の固定された、5 個の時間的な、または 1 つの信号通知されたフィルタセットの中からの選択が示される。フィルタセットインデックスのみが信号通知される。1つのスライスに対して、25 個のフィルタからなる 1 つの新しいセットのみを信号通知することができる。1つのスライスに対して新しいセットが信号通知された場合、同じスライス内のすべての輝度 C T B はそのセットを共有する。固定フィルタセットを使用して新しいスライスレベルフィルタセットを予測することができ、これを輝度 C T B の候補フィルタセットとして使用できる。フィルタの数は合計 64 個である。

【0101】

クロマ成分の場合、A L F をクロマ C T B に適用する時に、1つのスライスに対して新しいフィルタを信号通知する場合、C T B はこの新しいフィルタを使用し、そうでない場合、時間スケラビリティ制約を満たす最も新しい時間的クロマフィルタを適用する。

30

【0102】

スライスレベルの時間的フィルタとして、A P S は、先入れ先出し (F I F O) バッファとして更新される。

【0103】

仕様

J V E T - K 1 0 0 1 - v 6 に基づいて、{ { f i x e d f i l t e r } } , [[t e m p o r a l f i l t e r s]] , [[t e m p o r a l f i l t e r s]] および ((C T B - b a s e d f i l t e r i n d e x)) を使用して、すなわち、2 重大括弧、2 重角括弧、および 2 重丸括弧を使って修正する。

40

【0104】

7.3.3.2 適応ループフィルタデータ構文

【0105】

【表 3】

alf_data() {	デイスクリ フタ
alf_chroma_idc	tu(v)
((alf_signal_new_filter_luma	u(1)
if(alf_signal_new_filter_luma>0) {))	
{ { alf_luma_use_fixed_filter_flag	u(1)
if(alf_luma_use_fixed_filter_flag){	
alf_luma_fixed_filter_set_index	tb(v)
alf_luma_fixed_filter_usage_pattern	u(1)
if (alf_luma_fixed_filter_usage_pattern > 0)	
for (i = 0; i < NumAlfFilters; i++)	
alf_luma_fixed_filter_usage[i]	u(1) } }
((alf_num_available_temporal_filter_sets_luma	tb(1)))
alf_luma_num_filters_signalled_minus1	tb(v)
alf_luma_type_flag	u(1)
if(alf_luma_num_filters_signalled_minus1>0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	tu(v)
}	
alf_luma_coeff_delta_flag	u(1)
if (!alf_luma_coeff_delta_flag &&	
alf_luma_num_filters_signalled_minus1>0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	tu(v)
for(i = 0; i < (alf_luma_type_flag == 1) ? 2 : 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)
if (alf_luma_coeff_delta_flag) {	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++)	
alf_luma_coeff_flag[sigFiltIdx]	u(1)
}	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++){	
if (alf_luma_coeff_flag[sigFiltIdx]) {	
for (j = 0; j < (alf_luma_type_flag == 1) ? 6 : 12; j++) {	
alf_luma_coeff_delta_abs[sigFiltIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sigFiltIdx][j])	
alf_luma_coeff_delta_sign[sigFiltIdx][j]	u(1)
}	
}	
}	
((}))	
if (alf_chroma_idc > 0) {	
((alf_signal_new_filter_chroma	u(1)
if (alf_signal_new_filter_chroma){))	
alf_chroma_min_eg_order_minus1	tu(v)
for(i = 0; i < 2; i++)	
alf_chroma_eg_order_increase_flag[i]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[j]	uek(v)
if(alf_chroma_coeff_abs[j] > 0)	
alf_chroma_coeff_sign[j]	u(1)
}	
((}))	
}	
}	

10

20

30

40

【 0 1 0 6 】

7 . 3 . 4 . 2 符号化ツリーユニット構文

【 0 1 0 7 】

【表 4】

coding_tree_unit() {	デイスクリ プタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
if(slice_alf_enable_flag){	
alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if (alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	
{	
((coding_alf_ctu_filter_set_index(CtbAddrInRs)))	
}	
if(alf_chroma_idc==1 alf_chroma_idc==3)	
alf_ctb_flag[1][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if(alf_chroma_idc==2 alf_chroma_idc==3)	
alf_ctb_flag[2][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0,	
SINGLE_TREE)	
}	
((coding_alf_ctu_filter_set_index(CtbAddrInRs, slice_type){))	デイスクリ プタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
alf_use_new_filter	ae(1)
if (alf_use_new_filter == 0){	
alf_use_fixed_filter	ae(1)
}	
if (alf_use_new_filter){	
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16	
}	
else if (alf_use_fixed_filter){	
alf_fixed_filter_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = alf_fixed_filter_index	
}	
else{	
alf_temporal_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16 + alf_temporal_index	
}	

【 0 1 0 8 】

7 . 4 . 4 . 2 適応ループフィルタデータ意味論

((alf__signal__new__filter__luma)) が1である場合、新しい輝度フィルタセットが信号通知されることを示す。 alf__signal__new__filter__luma が0である場合、新しい輝度フィルタセットが信号通知されないことを示す。存在しない場合、0である。

{{ alf__luma__use__fixed__filter__flag }} が1である場合、適応ループフィルタに信号通知するのに固定フィルタセットを使用することを示す。 alf__luma__use__fixed__filter__flag が0である場合、適応

ループフィルタを信号通知するのに固定フィルタ使用しないことを示す。

{ { a l f _ l u m a _ f i x e d _ f i l t e r _ s e t _ i n d e x } } は、固定フィルタセットインデックスを示す。それは 0 . . . 15 であり得る。

{ { a l f _ l u m a _ f i x e d _ f i l t e r _ u s a g e _ p a t t e r n } } が 0 である場合、新しいフィルタすべてが固定フィルタを使用することを示す。a l f _ l u m a _ f i x e d _ f i l t e r _ u s a g e _ p a t t e r n が 1 である場合、新しいフィルタのうちいくつかは、固定フィルタを使用し、それ以外は使用しないことを示す。

{ { a l f _ l u m a _ f i x e d _ f i l t e r _ u s a g e [i] } } が 1 である場合、i 番目のフィルタは固定フィルタを使用することを示す。a l f _ l u m a _ f i x e d _ f i l t e r _ u s a g e [i] が 0 である場合、i 番目のフィルタは、固定フィルタを使用しないことを示す。存在しない場合、1 と推測される。

10

((a l f _ s i g n a l _ n e w _ f i l t e r _ c h r o m a)) が 1 である場合、新しいクロマフィルタが信号通知されることを示す。a l f _ s i g n a l _ n e w _ f i l t e r _ c h r o m a が 0 である場合、新しいクロマフィルタが信号通知されないことを示す。

(((a l f _ n u m _ a v a i l a b l e _ t e m p o r a l _ f i l t e r _ s e t s _ l u m a))) は、現在のスライスに使用できる利用可能な時間的フィルタセットの数を示す。この数は 0 . . . 5 であり得る。存在しない場合、0 である。

変数 a l f _ n u m _ a v a i l a b l e _ f i l t e r _ s e t s は、16 + a l f _ s i g n a l _ n e w _ f i l t e r _ l u m a + a l f _ n u m _ a v a i l a b l e _ t e m p o r a l _ f i l t e r _ s e t s _ l u m a として導出される。

20

((a l f _ s i g n a l _ n e w _ f i l t e r _ l u m a が 1 の場合は、下記の処理))

s i g F i l t I d x = 0 . . a l f _ l u m a _ n u m _ f i l t e r s _ s i g n a l l e d _ m i n u s 1 , j = 0 . . 11 を有する変数フィルタ C o e f f i c i e n t s [s i g F i l t I d x] [j] は、次のように初期化される。

f i l t e r C o e f f i c i e n t s [s i g F i l t I d x] [j] = a l f _ l u m a _ c o e f f _ d e l t a _ a b s [s i g F i l t I d x] [j] * (1 - 2 * a l f _ l u m a _ c o e f f _ d e l t a _ s i g n [s i g F i l t I d x] [j]) (7 - 50)

30

a l f _ l u m a _ c o e f f _ d e l t a _ p r e d i c t i o n _ f l a g が 1 である場合、s i g F i l t I d x = 1 . . a l f _ l u m a _ n u m _ f i l t e r s _ s i g n a l l e d _ m i n u s 1 および j = 0 . . 11 を有する f i l t e r C o e f f i c i e n t s [s i g F i l t I d x] [j] は、次のように修正される。

f i l t e r C o e f f i c i e n t s [s i g F i l t I d x] [j] + = f i l t e r C o e f f i c i e n t s [s i g F i l t I d x - 1] [j] (7 - 51)

f i l t I d x = 0 . . N u m A l f F i l t e r s - 1 , j = 0 . . 11 の要素 A l f C o e f f _ L [f i l t I d x] [j] を有する輝度フィルタ係数 A l f C o e f f _ L は、以下のように導出される。

A l f C o e f f _ L [f i l t I d x] [j] = f i l t e r C o e f f i c i e n t s [a l f _ l u m a _ c o e f f _ d e l t a _ i d x [f i l t I d x]] [j] (7 - 52)

40

{ { a l f _ l u m a _ u s e _ f i x e d _ f i l t e r _ f l a g が 1 で、a l f _ l u m a _ f i x e d _ f i l t e r _ u s a g e [f i l t i d x] が 1 の場合、以下が適用される。

A l f C o e f f _ L [f i l t I d x] [j] = A l f C o e f f _ L [f i l t I d x] [j] + A l f F i x e d F i l t e r C o e f f [A l f C l a s s T o F i l t e r M a p p i n g [a l f _ l u m a _ f i x e d _ f i l t e r _ i n d e x] [f i l t i d x]] [j] }

f i l t I d x = 0 . . N u m A l f F i l t e r s - 1 に対する最後のフィルタ係数 A

50

$l f C o e f f _ L [f i l t I d x] [1 2]$ は、以下のように導出される。

$A l f C o e f f _ L [f i l t I d x] [1 2] = 1 2 8 - k (A l f C o e f f _ L [f i l t I d x] [k] < 1)$, with $k = 0 \dots 11$ (7-53)
 $f i l t I d x = 0 \dots N u m A l f F i l t e r s - 1$, $j = 0 \dots 11$ の場合、 $A l f C o e f f _ L [f i l t I d x] [j]$ の値は、 $-2^7 \sim 2^7 - 1$ の範囲内とし、 $A l f C o e f f _ L [f i l t I d x] [1 2]$ の値は、 $0 \sim 2^8 - 1$ の範囲内とすることが、ビットストリーム適合性の要件である。

((輝度フィルタ係数)) $f i l t S e t I d x = 0 \dots 15$, $f i l t S e t I d x = 0 \dots N u m A l f F i l t e r s - 1$, $j = 0 \dots 12$ の場合、要素 $A l f C o e f f _ L u m a A l l [f i l t S e t I d x] [f i l t I d x] [j]$ を有する $A l f C o e f f _ L u m a A l l$ は、下記のようにして導出される。

$A l f C o e f f _ L u m a A l l [f i l t S e t I d x] [f i l t I d x] [j] = \{ \{ A l f F i x e d F i l t e r C o e f f [A l f C l a s s T o F i l t e r M a p p i n g] \} \} f i l t S e t I d x \{ \{ \} [f i l t I d x] \} [j] \}$

((輝度フィルタ係数)) $f i l t S e t I d x = 16$, $f i l t S e t I d x = 0 \dots N u m A l f F i l t e r s - 1$ および $j = 0 \dots 12$ を有する $A l f C o e f f _ L u m a A l l$ with elements $A l f C o e f f _ L u m a A l l [f i l t S e t I d x] [f i l t I d x] [j]$ は次のようにして導出される。

変数 $c l o s e s t _ t e m p o r a l _ i n d e x$ は、-1に初期化される。 $T i d$ は、現在のスライスの時間層インデックスである。

(($a l f _ s i g n a l _ n e w _ f i l t e r _ l u m a$ が1ならば))

$A l f C o e f f _ L u m a A l l [16] [f i l t I d x] [j] = A l f C o e f f _ L [f i l t I d x] [j]$

((そうでない場合は、以下の処理が呼び出される))。

for ($i = T i d$; $i \geq 0$; $i--$)

{

for ($k = 0$; $k < t e m p _ s i z e _ L$; $k++$)

{

if ($t e m p _ T i d _ L [k] == i$)

{

$c l o s e s t _ t e m p o r a l _ i n d e x$ is set as k ;

break;

}

}

}

$A l f C o e f f _ L u m a A l l [16] [f i l t I d x] [j] = T e m p _ L [c l o s e s t _ t e m p o r a l _ i n d e x] [f i l t I d x] [j]$

((輝度フィルタ係数)) $f i l t S e t I d x = 17 \dots a l f _ n u m _ a v a i l a b l e _ f i l t e r _ s e t s - 1$, $f i l t S e t I d x = 0 \dots N u m A l f F i l t e r s - 1$ and $j = 0 \dots 12$ の場合、要素 $A l f C o e f f _ L u m a A l l [f i l t S e t I d x] [f i l t I d x] [j]$, を有する $A l f C o e f f _ L u m a A l l$ は、次のように導出される。

$i = 17$;

for ($k = 0$; $k < t e m p _ s i z e _ L$ and $i < a l f _ n u m _ a v a i l a b l e _ f i l t e r _ s e t s$; $j++$)

{

((もし $t e m p _ T i d _ L [k] \leq T i d$ 、かつ k が $c l o s e s t _ t e m p o r a l _ i n d e x$ に等しくない場合)

{

$A l f C o e f f _ L u m a A l l [i] [f i l t I d x] [j] = T e m p _ L [$

```

k][filtIdx][j];
    i++;
}
}

```

```

{{AlfFixedFilterCoeff}}[64][13] =
{
{0, 0, 2, -3, 1, -4, 1, 7, -1, 1, -1, 5, 112},
{0, 0, 0, 0, 0, -1, 0, 1, 0, 0, -1, 2, 126},
{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 126},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 1, 128},
{2, 2, -7, -3, 0, -5, 13, 22, 12, -3, -3, 17, 34},
{-1, 0, 6, -8, 1, -5, 1, 23, 0, 2, -5, 10, 80},
{0, 0, -1, -1, 0, -1, 2, 1, 0, 0, -1, 4, 122},
{0, 0, 3, -11, 1, 0, -1, 35, 5, 2, -9, 9, 60},
{0, 0, 8, -8, -2, -7, 4, 4, 2, 1, -1, 25, 76},
{0, 0, 1, -1, 0, -3, 1, 3, -1, 1, -1, 3, 122},
{0, 0, 3, -3, 0, -6, 5, -1, 2, 1, -4, 21, 92},
{-7, 1, 5, 4, -3, 5, 11, 13, 12, -8, 11, 12, 16},
{-5, -3, 6, -2, -3, 8, 14, 15, 2, -7, 11, 16, 24},
{2, -1, -6, -5, -2, -2, 20, 14, -4, 0, -3, 25, 52},
{3, 1, -8, -4, 0, -8, 22, 5, -3, 2, -10, 29, 70},
{2, 1, -7, -1, 2, -11, 23, -5, 0, 2, -10, 29, 78},
{-6, -3, 8, 9, -4, 8, 9, 7, 14, -2, 8, 9, 14},
{2, 1, -4, -7, 0, -8, 17, 22, 1, -1, -4, 23, 44},
{3, 0, -5, -7, 0, -7, 15, 18, -5, 0, -5, 27, 60},
{2, 0, 0, -7, 1, -10, 13, 13, -4, 2, -7, 24, 74},
{3, 3, -13, 4, -2, -5, 9, 21, 25, -2, -3, 12, 24},
{-5, -2, 7, -3, -7, 9, 8, 9, 16, -2, 15, 12, 14},
{0, -1, 0, -7, -5, 4, 11, 11, 8, -6, 12, 21, 32},
{3, -2, -3, -8, -4, -1, 16, 15, -2, -3, 3, 26, 48},
{2, 1, -5, -4, -1, -8, 16, 4, -2, 1, -7, 33, 68},
{2, 1, -4, -2, 1, -10, 17, -2, 0, 2, -11, 33, 74},
{1, -2, 7, -15, -16, 10, 8, 8, 20, 11, 14, 11, 14},
{2, 2, 3, -13, -13, 4, 8, 12, 2, -3, 16, 24, 40},
{1, 4, 0, -7, -8, -4, 9, 9, -2, -2, 8, 29, 54},
{1, 1, 2, -4, -1, -6, 6, 3, -1, -1, -3, 30, 74},
{-7, 3, 2, 10, -2, 3, 7, 11, 19, -7, 8, 10, 14},
{0, -2, -5, -3, -2, 4, 20, 15, -1, -3, -1, 22, 40},
{3, -1, -8, -4, -1, -4, 22, 8, -4, 2, -8, 28, 62},
{0, 3, -14, 3, 0, 1, 19, 17, 8, -3, -7, 20, 34},
{0, 2, -1, -8, 3, -6, 5, 21, 1, 1, -9, 13, 84},
{-4, -2, 8, 20, -2, 2, 3, 5, 21, 4, 6, 1, 4},
{2, -2, -3, -9, -4, 2, 14, 16, 3, -6, 8, 24, 38},
{2, 1, 5, -16, -7, 2, 3, 11, 15, -3, 11, 22, 36},
{1, 2, 3, -11, -2, -5, 4, 8, 9, -3, -2, 26, 68},
{0, -1, 10, -9, -1, -8, 2, 3, 4, 0, 0, 29, 70},
{1, 2, 0, -5, 1, -9, 9, 3, 0, 1, -7, 20, 96},
{-2, 8, -6, -4, 3, -9, -8, 45, 14, 2, -13, 7, 54},
{1, -1, 16, -19, -8, -4, -3, 2, 19, 0, 4, 30, 54},

```

```

{ 1 , 1 , - 3 , 0 , 2 , - 1 1 , 1 5 , - 5 , 1 , 2 , - 9 , 2 4 , 9 2 } ,
{ 0 , 1 , - 2 , 0 , 1 , - 4 , 4 , 0 , 0 , 1 , - 4 , 7 , 1 2 0 } ,
{ 0 , 1 , 2 , - 5 , 1 , - 6 , 4 , 1 0 , - 2 , 1 , - 4 , 1 0 , 1 0 4 } ,
{ 3 , 0 , - 3 , - 6 , - 2 , - 6 , 1 4 , 8 , - 1 , - 1 , - 3 , 3 1 , 6 0 } ,
{ 0 , 1 , 0 , - 2 , 1 , - 6 , 5 , 1 , 0 , 1 , - 5 , 1 3 , 1 1 0 } ,
{ 3 , 1 , 9 , - 1 9 , - 2 1 , 9 , 7 , 6 , 1 3 , 5 , 1 5 , 2 1 , 3 0 } ,
{ 2 , 4 , 3 , - 1 2 , - 1 3 , 1 , 7 , 8 , 3 , 0 , 1 2 , 2 6 , 4 6 } ,
{ 3 , 1 , - 8 , - 2 , 0 , - 6 , 1 8 , 2 , - 2 , 3 , - 1 0 , 2 3 , 8 4 } ,
{ 1 , 1 , - 4 , - 1 , 1 , - 5 , 8 , 1 , - 1 , 2 , - 5 , 1 0 , 1 1 2 } ,
{ 0 , 1 , - 1 , 0 , 0 , - 2 , 2 , 0 , 0 , 1 , - 2 , 3 , 1 2 4 } ,
{ 1 , 1 , - 2 , - 7 , 1 , - 7 , 1 4 , 1 8 , 0 , 0 , - 7 , 2 1 , 6 2 } ,
{ 0 , 1 , 0 , - 2 , 0 , - 7 , 8 , 1 , - 2 , 0 , - 3 , 2 4 , 8 8 } ,
{ 0 , 1 , 1 , - 2 , 2 , - 1 0 , 1 0 , 0 , - 2 , 1 , - 7 , 2 3 , 9 4 } ,
{ 0 , 2 , 2 , - 1 1 , 2 , - 4 , - 3 , 3 9 , 7 , 1 , - 1 0 , 9 , 6 0 } ,
{ 1 , 0 , 1 3 , - 1 6 , - 5 , - 6 , - 1 , 8 , 6 , 0 , 6 , 2 9 , 5 8 } ,
{ 1 , 3 , 1 , - 6 , - 4 , - 7 , 9 , 6 , - 3 , - 2 , 3 , 3 3 , 6 0 } ,
{ 4 , 0 , - 1 7 , - 1 , - 1 , 5 , 2 6 , 8 , - 2 , 3 , - 1 5 , 3 0 , 4 8 } ,
{ 0 , 1 , - 2 , 0 , 2 , - 8 , 1 2 , - 6 , 1 , 1 , - 6 , 1 6 , 1 0 6 } ,
{ 0 , 0 , 0 , - 1 , 1 , - 4 , 4 , 0 , 0 , 0 , - 3 , 1 1 , 1 1 2 } ,
{ 0 , 1 , 2 , - 8 , 2 , - 6 , 5 , 1 5 , 0 , 2 , - 7 , 9 , 9 8 } ,
{ 1 , - 1 , 1 2 , - 1 5 , - 7 , - 2 , 3 , 6 , 6 , - 1 , 7 , 3 0 , 5 0 } ,
};
{ { A l f C l a s s T o F i l t e r M a p p i n g } } [ 1 6 ] [ 2 5 ] =
{
{ 8 , 2 , 2 , 2 , 3 , 4 , 5 3 , 9 , 9 , 5 2 , 4 , 4 , 5 , 9 , 2 , 8 , 1 0 , 9 ,
1 , 3 , 3 9 , 3 9 , 1 0 , 9 , 5 2 } ,
{ 1 1 , 1 2 , 1 3 , 1 4 , 1 5 , 3 0 , 1 1 , 1 7 , 1 8 , 1 9 , 1 6 , 2 0 , 2 0 ,
4 , 5 3 , 2 1 , 2 2 , 2 3 , 1 4 , 2 5 , 2 6 , 2 6 , 2 7 , 2 8 , 1 0 } ,
{ 1 6 , 1 2 , 3 1 , 3 2 , 1 4 , 1 6 , 3 0 , 3 3 , 5 3 , 3 4 , 3 5 , 1 6 , 2 0 ,
4 , 7 , 1 6 , 2 1 , 3 6 , 1 8 , 1 9 , 2 1 , 2 6 , 3 7 , 3 8 , 3 9 } ,
{ 3 5 , 1 1 , 1 3 , 1 4 , 4 3 , 3 5 , 1 6 , 4 , 3 4 , 6 2 , 3 5 , 3 5 , 3 0 , 5
6 , 7 , 3 5 , 2 1 , 3 8 , 2 4 , 4 0 , 1 6 , 2 1 , 4 8 , 5 7 , 3 9 } ,
{ 1 1 , 3 1 , 3 2 , 4 3 , 4 4 , 1 6 , 4 , 1 7 , 3 4 , 4 5 , 3 0 , 2 0 , 2 0 , 7
, 5 , 2 1 , 2 2 , 4 6 , 4 0 , 4 7 , 2 6 , 4 8 , 6 3 , 5 8 , 1 0 } ,
{ 1 2 , 1 3 , 5 0 , 5 1 , 5 2 , 1 1 , 1 7 , 5 3 , 4 5 , 9 , 3 0 , 4 , 5 3 , 1 9
, 0 , 2 2 , 2 3 , 2 5 , 4 3 , 4 4 , 3 7 , 2 7 , 2 8 , 1 0 , 5 5 } ,
{ 3 0 , 3 3 , 6 2 , 5 1 , 4 4 , 2 0 , 4 1 , 5 6 , 3 4 , 4 5 , 2 0 , 4 1 , 4 1 ,
5 6 , 5 , 3 0 , 5 6 , 3 8 , 4 0 , 4 7 , 1 1 , 3 7 , 4 2 , 5 7 , 8 } ,
{ 3 5 , 1 1 , 2 3 , 3 2 , 1 4 , 3 5 , 2 0 , 4 , 1 7 , 1 8 , 2 1 , 2 0 , 2 0 , 2
0 , 4 , 1 6 , 2 1 , 3 6 , 4 6 , 2 5 , 4 1 , 2 6 , 4 8 , 4 9 , 5 8 } ,
{ 1 2 , 3 1 , 5 9 , 5 9 , 3 , 3 3 , 3 3 , 5 9 , 5 9 , 5 2 , 4 , 3 3 , 1 7 , 5 9
, 5 5 , 2 2 , 3 6 , 5 9 , 5 9 , 6 0 , 2 2 , 3 6 , 5 9 , 2 5 , 5 5 } ,
{ 3 1 , 2 5 , 1 5 , 6 0 , 6 0 , 2 2 , 1 7 , 1 9 , 5 5 , 5 5 , 2 0 , 2 0 , 5 3 ,
1 9 , 5 5 , 2 2 , 4 6 , 2 5 , 4 3 , 6 0 , 3 7 , 2 8 , 1 0 , 5 5 , 5 2 } ,
{ 1 2 , 3 1 , 3 2 , 5 0 , 5 1 , 1 1 , 3 3 , 5 3 , 1 9 , 4 5 , 1 6 , 4 , 4 , 5 3
, 5 , 2 2 , 3 6 , 1 8 , 2 5 , 4 3 , 2 6 , 2 7 , 2 7 , 2 8 , 1 0 } ,
{ 5 , 2 , 4 4 , 5 2 , 3 , 4 , 5 3 , 4 5 , 9 , 3 , 4 , 5 6 , 5 , 0 , 2 , 5 , 1 0
, 4 7 , 5 2 , 3 , 6 3 , 3 9 , 1 0 , 9 , 5 2 } ,
{ 1 2 , 3 4 , 4 4 , 4 4 , 3 , 5 6 , 5 6 , 6 2 , 4 5 , 9 , 5 6 , 5 6 , 7 , 5 , 0
, 2 2 , 3 8 , 4 0 , 4 7 , 5 2 , 4 8 , 5 7 , 3 9 , 1 0 , 9 } ,

```

```
{ 3 5 , 1 1 , 2 3 , 1 4 , 5 1 , 3 5 , 2 0 , 4 1 , 5 6 , 6 2 , 1 6 , 2 0 , 4 1 ,
5 6 , 7 , 1 6 , 2 1 , 3 8 , 2 4 , 4 0 , 2 6 , 2 6 , 4 2 , 5 7 , 3 9 } ,
{ 3 3 , 3 4 , 5 1 , 5 1 , 5 2 , 4 1 , 4 1 , 3 4 , 6 2 , 0 , 4 1 , 4 1 , 5 6 , 7
, 5 , 5 6 , 3 8 , 3 8 , 4 0 , 4 4 , 3 7 , 4 2 , 5 7 , 3 9 , 1 0 } ,
{ 1 6 , 3 1 , 3 2 , 1 5 , 6 0 , 3 0 , 4 , 1 7 , 1 9 , 2 5 , 2 2 , 2 0 , 4 , 5 3
, 1 9 , 2 1 , 2 2 , 4 6 , 2 5 , 5 5 , 2 6 , 4 8 , 6 3 , 5 8 , 5 5 } , } ;
```

((alf__signal__new__filter__chromaが1である場合、次の処理)。

j = 0 . . 5 の場合、クロマフィルタ係数 AlfCoefFc[j] は、以下のように導出される。

```
AlfCoefFc[j] = alf__chroma__coeff__abs[j] * ( 1
- 2 * alf__chroma__coeff__sign[j] ) ( 7 - 5 7 )
```

j = 6 の場合の最後のフィルタ係数は、以下のように導出される。

```
AlfCoefFc[6] = 1 2 8 - k ( AlfCoefFc[k] < 1 ) , w i t h
k = 0 . . 5 ( 7 - 5 8 )
```

filterIdx = 0 . . NumAlfFilters - 1 , j = 0 . . 5 の場合、AlfCoefFc[j] の値は、 $-2^7 \sim 2^7 - 1$ の範囲内とし、AlfCoefFc[6] の値は、 $0 \sim 2^8 - 1$ の範囲内とすることが、ビットストリーム適合性の要件である。

そうでない場合、(((alf__signal__new__filter__chroma is 0))) 以下が呼び出される。

```
for ( i = Tid ; i >= 0 ; i - - )
{
    for ( k = 0 ; k < temp__size__C ; k + + )
    {
        if ( tempTid__c[k] == i )
        {
            closest__temporal__index is set as k ;
            break ;
        }
    }
}
```

j = 0 . . 6 の場合、クロマフィルタ係数 AlfCoefFc[j] は、以下のように導出される。

```
AlfCoefFc[j] = Tempc[closest__temporal__index][j]
```

7.4.5.2 符号化ツリーユニット構文

((alf__luma__ctb__filter__set__index[xCtb>>Log2CtbSize][yCtb>>Log2CtbSize])) は、位置(xCtb, yCtb) における輝度CTBのフィルタセットインデックスを指定する。

((alf__use__new__filter)) が1である場合、alf__luma__ctb__filter__set__index[xCtb>>Log2CtbSize][yCtb>>Log2CtbSize] は16であることを示す。alf__use__new__filter が0である場合、alf__luma__ctb__filter__set__index[xCtb>>Log2CtbSize][yCtb>>Log2CtbSize] は16に等しくないことを示す。

((alf__use__fixed__filter)) が1である場合、固定フィルタセットの1つを使用することを示し、alf__use__fixed__filter が0である場合、現在の輝度CTBは固定フィルタセットを使用しないことを示す。

((alf__fixed__filter__index)) は、固定フィルタセットのインデックスを示し、このインデックスは0 ~ 15までとすることができる。

10

20

30

40

50

(((alf__temporal__index)))は、時間的フィルタセットインデックスを示し、これは、0からalf__num__available__temporal__filter__sets__luma - 1とすることができる。

[[8 . 5 . 1 一般]]

1. sps__alf__enabled__flagが1である場合、以下が適用される。

- [[8 . 5 . 4 . 5 項に規定された時間的フィルタ更新処理が呼び出される。]]

- 8 . 5 . 4 . 1 項で規定された適応ループフィルタ処理は、再構成されたピクチャサンプルアレイス_L、S_{cb}およびS_{cr}を入力とし、サンプル適応オフセット後の修正された再構成されたピクチャサンプルアレイス_L、S_{cb}およびS_{cr}が、出力として呼び出される。

- 配列S_L、S_{cb}、S_{cr}は、それぞれ配列S_L、S_{cb}、S_{cr}（復号化されたピクチャを表す）に割り当てられている。

- [[8 . 5 . 4 . 6 項で規定された時間的フィルタ更新処理が呼び出される。]]

((8 . 5 . 4 . 2 輝度サンプルのための符号化ツリーブロックフィルタリング処理))

- filtIdx[x][y]で指定されたフィルタに対応する輝度フィルタ係数の配列f[j]は、j = 0 . . 12として以下のように導出される。

f[j] = ((AlfCoeffLumaAll)) [alf__luma__ctb__filter__set__index [xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]] [filtIdx[x][y]] [j] (8 - 732)

[[8 . 5 . 4 . 5 時間的フィルタ更新]]

以下の条件のいずれかが真である場合、

- 現在のピクチャはIDRピクチャである。

- 現在のピクチャはBLA画像である。

- 復号化の順序では、現在のピクチャは、POCが前回の復号化されたIRAPピクチャのPOCよりも大きい1つ目のピクチャ、すなわち先頭ピクチャの後ろかつ後続ピクチャの後ろにあるピクチャである。

次に、temp__size__Lおよびtemp__size__Cを0に設定する。

[[8 . 5 . 4 . 6 時間的フィルタの更新]]

slice__alf__enabled__flagが1であり、alf__signal__new__filter__lumaが1である場合、以下が適用される。

輝度時間的フィルタバッファサイズ、temp__size__L < 5の場合、temp__size__L = temp__size__L + 1である。

i = temp__size__L - 1 . . . 1, j = 0 . . . NumAlfFilters - 1, k = 0 . . . 12の場合、TempL[i][j][k]は、次のように更新される。

TempL[i][j][k] = TempL[i - 1][j][k]

j = 0 . . . NumAlfFilters - 1, k = 0 . . . 12の場合、TempL[0][j][k]は、次のように更新される。

TempL[0][j][k] = AlfCoeffL[j][k]

i = temp__size__L - 1 . . . 1の場合、TempTid_L[i]は、次のように更新される。

TempTid_L[i] = TempTid_L[i - 1]

TempTid_L[0]を現在のスライスの時間層インデックスTidとして設定する。

alf__chroma__idxが0でなく、alf__signal__new__filter__chromaが1の場合、以下が適用される。

i = temp__size__c - 1 . . . 1, j = 0 . . . 6のTempc[i][j]は、次のように更新される。

Tempc[i][j] = Tempc[i - 1][j]

j = 0 . . . 6の場合、Tempc[0][j]は、次のように更新される。

Tempc[0][j] = AlfCoeffc[j]

10

20

30

40

50

$i = \text{temp_size_C} - 1 \dots 1$ の場合、 $\text{TempTid_c}[i]$ は、次のように更新される。

$\text{TempTid_c}[i] = \text{TempTid_c}[i - 1]$
 $\text{TempTid_c}[0]$ を現在のスライスの Tid として設定する。
【 0 1 0 9 】

【表 5】

表 9-4 — 構文要素および関連する 2 値化

構文構造	構文要素	2 値化	
		処理	入力パラメータ
coding_tree_unit()	alf_ctb_flag[][][]	FL	cMax=1
	((alf_use_new_filter_set	FL	cMax=1
	alf_used_fixed_filter	FL	cMax=1
	alf_fixed_filter_index	TB	cMax=15
	alf_temporal_index	TB	cMax = alf_num_available_temporal_filter_sets_luma - 1))

10

【 0 1 1 0 】

【表 6】

表 9-10 — コンテキスト符号化ビンを有する構文要素への ctxInc の割り当て

構文要素	binIdx					
	0	1	2	3	4	>=5
((alf_use_new_filter_set	0	na	na	na	na	na
alf_used_fixed_filter	0	na	na	na	na	na))

20

【 0 1 1 1 】

9. J V E T - M 0 4 2 7 におけるインループリシェイプ (I L R)

インループリシェイプ (I L R) の基本的な考えは、元の (第 1 のドメインにおける) 信号 (予測 / 再構成信号) を第 2 のドメイン (リシェイプされたドメイン) に変換することである。

【 0 1 1 2 】

インループ輝度リシェイパは、1 対のルックアップテーブル (L U T) として実装されるが、信号通知された L U T から他方を計算することができるので、2 つの L U T のうちの一方のみを信号通知する必要がある。各 L U T は、1 次元 1 0 ビット 1 0 2 4 エントリマッピングテーブル (1 D - L U T) である。1 つの L U T は、フォワード L U T、F w d L U T であり、これは、入力輝度コード値 Y_i を変更された値 Y_r : $Y_r = \text{FwdLUT}[Y_i]$ にマッピングする。他方の L U T は、逆 L U T、I n v L U T であり、変更されたコード値 Y_r を Y^{\wedge}_i : $Y^{\wedge}_i = \text{InvLUT}[Y_r]$ にマッピングする。 (Y^{\wedge}_i は Y_i の再構成値を表す)。

【 0 1 1 3 】

9. 1 P W L モデル

概念的には、区分線形 (P W L) は、以下のように実装される。

【 0 1 1 4 】

x_1 、 x_2 を 2 つの入力支点とし、 y_1 、 y_2 を 1 つのピースに対応する出力支点とする。 x_1 と x_2 との間の任意の入力値 x に対する出力値 y は、以下の式によって補間することができる

30

40

50

【 0 1 1 5 】

$$y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$$

【 0 1 1 6 】

固定点実装では、この式は、以下のように書き換えることができる。

【 0 1 1 7 】

$$y = ((m * x + 2^{FP_PREC - 1}) >> FP_PREC) + c$$

【 0 1 1 8 】

ここで、 m はスカラーであり、 c はオフセットであり、 FP_PREC は精度を規定するための定数である。

【 0 1 1 9 】

なお、 $CE - 12$ ソフトウェアでは、 PWL モデルは、 1024 エントリの $FwdLUT$ マッピングテーブルおよび $InvLUT$ マッピングテーブルを予め計算するために使用されるが、 PWL モデルは、 LUT を予め計算することなく、実装において同一のマッピング値をオンザフライで計算することも可能にする。

【 0 1 2 0 】

9.2 第4回VVC協議における $CE - 12 - 2$ のテスト

9.2.1 輝度のリシェイプ

インループ輝度リシェイプのテスト2（すなわち、提案における $CE - 12 - 2$ ）は、より複雑性低いパイプラインを提供し、且つインタースライス再構成におけるブロック単位のイントラ予測のための復号化待ち時間を排除する。イントラ予測は、インタースライスおよびイントラスライスの両方のために、リシェイプされたドメインにおいて行われる。

【 0 1 2 1 】

イントラ予測は、スライスタイプにかかわらず、常にリシェイプされたドメインで行われる。このような構成によれば、前回の TU 再構成を行った直後にイントラ予測を開始することができる。このような構成は、スライスに依存する代わりに、イントラモードのための統一された処理を提供することもできる。図7は、 $CE - 12 - 2$ の方式に基づく復号化処理を示すブロック図である。

【 0 1 2 2 】

$CE - 12 - 2$ は、 16 個の区分線形（ PWL ）モデルを $CE - 12 - 1$ の 32 個の PWL モデルの代わりに使用して輝度およびクロマ残差スケーリングを行ってもよい。

【 0 1 2 3 】

$CE - 12 - 2$ におけるインループ輝度リシェイパを用いたインタースライス再構成（薄い影が付けられたブロックは、リシェイプされたドメインにおける信号を示す。輝度残差、輝度内予測、および輝度内再構成）

【 0 1 2 4 】

9.2.2 輝度依存クロマ残差スケーリング

輝度依存クロマ残差スケーリングは、固定小数点整数演算で実装される乗算処理である。クロマ残差スケーリングは、輝度信号のクロマ信号との相互作用を補償する。 TU レベルでクロマ残差スケーリングを適用する。具体的には、対応する輝度予測ブロックの平均値を利用する。

【 0 1 2 5 】

この平均値は、 PWL モデルにおけるインデックスを特定するために使用される。このインデックスは、スケーリング係数 $c_{ScaleInv}$ を特定する。クロマ残差にその数を乗算する。

【 0 1 2 6 】

なお、クロマスケール係数は、再構成された輝度値ではなく、フォワードマップされた予測輝度値から算出される。

【 0 1 2 7 】

9.2.3 I L R 副情報の信号通知

パラメータは（現在）タイルグループヘッダ（ ALF に類似）で送信される。これらは

10

20

30

40

50

、40～100ビットを要すると報告されている。

以下の規格は、J V E T - L 1 0 0 1 のバージョン 9 に基づいている。追加された構文は黄色で強調表示されている。

7 . 3 . 2 . 1 におけるシーケンスパラメータセット R B S P 構文

【 0 1 2 8 】

【表 7】

seq_parameter_set_rbsp() {	デイスクリ フタ
sps_seq_parameter_set_id	ue(v)
...	
sps_triangle_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if (sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2+1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
{{ sps_reshaper_enabled_flag	u(1) }}
rbps_trailing_bits()	
}	

10

20

【 0 1 2 9 】

7 . 3 . 3 . 1 におけるタイルグループヘッダ構文

【 0 1 3 0 】

【表 8】

tile_group_header() {	デイスクリ フタ
...	
if(num_tiles_in_tile_group_minus1 > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_tiles_in_tile_group_minus1; i++)	
entry_point_offset_minus1[i]	u(v)
}	
{{ if (sps_reshaper_enabled_flag) {	
tile_group_reshaper_model_present_flag	u(1)
if (tile_group_reshaper_model_present_flag)	
tile_group_reshaper_model ()	
tile_group_reshaper_enable_flag	u(1)
if (tile_group_reshaper_enable_flag && (!(qtbtt_dual_tree_intra_flag && tile_group_type == I)))	
tile_group_reshaper_chroma_residual_scale_flag	u(1)
} }}	
byte_alignment()	
}	

30

40

【 0 1 3 1 】

新しい構文表タイルグループリシェイパを追加する。

【 0 1 3 2 】

50

【表 9】

{{tile_group_reshaper_model() {	ディスプレイ フッタ
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW[i]	u(v)
if (reshaper_model_bin_delta_abs_CW[i] > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	
}}	

10

【0133】

{{一般的に、シーケンスパラメータセットR B S Pの意味論では、次の意味論を追加する。}}

s p s _ r e s h a p e r _ e n a b l e d _ f l a g が1と等しい場合は、符号化映像シーケンス(C V S)においてリシェイパが使用されていることを規定している。s p s _ r e s h a p e r _ e n a b l e d _ f l a g が0と等しい場合は、C V Sにおいてリシェイパが使用されていないことを規定している。

20

{{タイルグループヘッダ構文において、以下の意味論を追加する。}}

t i l e _ g r o u p _ r e s h a p e r _ m o d e l _ p r e s e n t _ f l a g が1と等しい場合は、t i l e _ g r o u p _ r e s h a p e r _ m o d e l () がタイルグループ内に存在することを規定している。t i l e _ g r o u p _ r e s h a p e r _ m o d e l _ p r e s e n t _ f l a g が0と等しい場合は、t i l e _ g r o u p _ r e s h a p e r _ m o d e l () がタイルグループヘッダに存在しないことを規定している。t i l e _ g r o u p _ r e s h a p e r _ m o d e l _ p r e s e n t _ f l a g が存在しない場合は、0に等しいと推論される。

t i l e _ g r o u p _ r e s h a p e r _ e n a b l e d _ f l a g が1と等しい場合は、現在のタイルグループに対してリシェイパが有効になっていることを規定している。t i l e _ g r o u p _ r e s h a p e r _ e n a b l e d _ f l a g が0と等しい場合は、現在のタイルグループに対してリシェイパが有効になっていないことを規定している。t i l e _ g r o u p _ r e s h a p e r _ e n a b l e _ f l a g が存在しない場合、0であると推論される。

30

t i l e _ g r o u p _ r e s h a p e r _ c h r o m a _ r e s i d u a l _ s c a l e _ f l a g が1と等しい場合は、現在のタイルグループに対してクロマ残差スケーリングを有効であることを規定している。t i l e _ g r o u p _ r e s h a p e r _ c h r o m a _ r e s i d u a l _ s c a l e _ f l a g が0と等しい場合は、現在のタイルグループに対してクロマ残差スケーリングが有効でないことを規定している。t i l e _ g r o u p _ r e s h a p e r _ c h r o m a _ r e s i d u a l _ s c a l e _ f l a g が存在しない場合は、0であると推論される。

40

{{t i l e _ g r o u p _ r e s h a p e r _ m o d e l () 構文を追加する}}

r e s h a p e _ m o d e l _ m i n _ b i n _ i d x は、最小のビン(またはピース)インデックスを、リシェイパ構築処理に使用するように規定している。r e s h a p e _ m o d e l _ m i n _ b i n _ i d x の値が0 ~ M a x B i n I d x の範囲内にあるものとする。M a x B i n I d x の値は15と等しいとする。

r e s h a p e _ m o d e l _ d e l t a _ m a x _ b i n _ i d x は、最大許容ビン(またはピース)インデックスM a x B i n I d x から最大ビンインデックスを引いたものが、リシェイパ構築処理で使用されることを規定している。r e s h a p e _ m o d e l

50

`__max__bin__idx`の値は、`MaxBinIdx - reshape__model__delta__max__bin__idx`に等しく設定される。

`reshaper__model__bin__delta__abs__cw__prec__min`
`us1 + 1`は、構文`reshape__model__bin__delta__abs__CW[i]`の表現に使用するビット数を規定している。

`reshape__model__bin__delta__abs__CW[i]`は、 i 番目のピンの絶対デルタコード名値を規定する。

`reshaper__model__bin__delta__sign__CW__flag[i]`は、`reshape__model__bin__delta__abs__CW[i]`のサインを次のように記述する。

- `reshape__model__bin__delta__sign__CW__flag[i]`が0と等しい場合、対応する変数`RspDeltaCW[i]`は正の値である。

- そうでない場合(`reshape__model__bin__delta__sign__CW__flag[i]`が0と等しくない)、対応する変数`RspDeltaCW[i]`は負の値である。

`reshape__model__bin__delta__sign__CW__flag[i]`が存在しない場合、0に等しいと推測される。

変数`RspDeltaCW[i] = (1 - 2 * reshape__model__bin__delta__sign__CW[i]) * reshape__model__bin__delta__abs__CW[i]`;

変数`RspCW[i]`は、以下のステップとして導出される。

変数`OrgCW`は、 $(1 < \text{BitDepthY}) / (\text{MaxBinIdx} + 1)$ に等しく設定される。

- `If reshaper__model__min__bin__idx <= i <= reshaper__model__max__bin__idx`

`RspCW[i] = OrgCW + RspDeltaCW[i]` .

- そうでない場合、`RspCW[i] = 0`である。

`BitDepthY`の値が10に等しい場合、`RspCW[i]`の値は、 $32 \sim 2 * \text{OrgCW} - 1$ の範囲内にくる。

i が $0 \sim \text{MaxBinIdx} + 1$ の範囲にある変数`InputPivot[i]`は、以下のように導出される。

`InputPivot[i] = i * OrgCW`

i が $0 \sim \text{MaxBinIdx} + 1$ の範囲にあり、変数`ScaleCoef[i]`および`InvScaleCoef[i]`がそれぞれ $0 \sim \text{MaxBinIdx}$ の範囲にあり、 i が $0 \sim \text{MaxBinIdx}$ の範囲にある変数`ReshapePivot[i]`は、以下のように導出される。

`shiftY = 14`

`ReshapePivot[0] = 0` ;

`for (i = 0 ; i <= MaxBinIdx ; i++) {`

`ReshapePivot[i + 1] = ReshapePivot[i] + RspCW[i]`

`ScaleCoef[i] = (RspCW[i] * (1 < < shiftY) + (1 < < (Log2(OrgCW) - 1))) > > (Log2(OrgCW))`

`if (RspCW[i] == 0)`

`InvScaleCoef[i] = 0`

`else`

`InvScaleCoef[i] = OrgCW * (1 < < shiftY) / RspCW[i]`

`}`

i が $0 \sim \text{MaxBinIdx}$ の範囲内にある変数`ChromaScaleCoef[i]`

10

20

30

40

50

は、以下のように導出される。

```
ChromaResidualScaleLut[64] = { 1 6 3 8 4 , 1 6 3 8 4
, 1 6 3 8 4 , 1 6 3 8 4 , 1 6 3 8 4 , 1 6 3 8 4 , 8 1 9 2 , 8 1 9 2
, 8 1 9 2 , 8 1 9 2 , 5 4 6 1 , 5 4 6 1 , 5 4 6 1 , 5 4 6 1 , 4 0 9 6 , 4 0 9 6
, 4 0 9 6 , 4 0 9 6 , 3 2 7 7 , 3 2 7 7 , 3 2 7 7 , 3 2 7 7 , 2 7 3 1 , 2 7 3 1
, 2 7 3 1 , 2 7 3 1 , 2 3 4 1 , 2 3 4 1 , 2 3 4 1 , 2 0 4 8 , 2 0 4 8 , 2 0 4 8
, 1 8 2 0 , 1 8 2 0 , 1 8 2 0 , 1 6 3 8 , 1 6 3 8 , 1 6 3 8 , 1 6 3 8 , 1 4 8 9
, 1 4 8 9 , 1 4 8 9 , 1 3 6 5 , 1 3 6 5 , 1 3 6 5 , 1 3 6 5 , 1 2 6 0
, 1 2 6 0 , 1 2 6 0 , 1 2 6 0 , 1 1 7 0 , 1 1 7 0 , 1 1 7 0 , 1 1 7 0 , 1 0 9 2
, 1 0 9 2 , 1 0 9 2 , 1 0 9 2 , 1 0 2 4 , 1 0 2 4 , 1 0 2 4 , 1 0 2 4 } ;
```

10

```
shiftC = 11
- if (RspCW[i] == 0)
    ChromaScaleCoef[i] = (1 << shiftC)
- Otherwise (RspCW[i] != 0), ChromaScaleCoef[i] = ChromaResidualScaleLut[RspCW[i] >> 1]
【0134】
```

9.2.4 ILRの使用方法

エンコーダ側では、まず、各ピクチャ（またはタイルグループ）をリシェイプされたドメインに変換する。そして、すべての符号化処理は、リシェイプされたドメインで行われる。イントラ予測の場合、近傍のブロックはリシェイプされたドメインにあり、インター予測の場合、まず、参照ブロック（復号化ピクチャバッファからの元のドメインから生成される）をリシェイプされたドメインに変換する。そして、残差を生成し、ビットストリームに符号化する。

20

【0135】

ピクチャ全体（またはタイルグループ）の符号化／復号化が終わった後、リシェイプされたドメインのサンプルを元のドメインに変換し、次いで、非ブロック化フィルタおよび他のフィルタを適用する。

【0136】

以下の場合、予測信号へのフォワードリシェイプは無効化される。

【0137】

30

現在のブロックはイントラ符号化されている

【0138】

現在のブロックはCPR（現在のピクチャを参照し、別名イントラブロックコピー、IBC）として符号化される。

【0139】

現在のブロックは、結合されたインターイントラモード（CIIP）として符号化され、イントラ予測ブロックのためにフォワードリシェイプは無効化される。

【0140】

10 双方向オプティカルフロー（BDOF）

10.1 BIOの概要と分析

40

BIOにおいて、まず、動き補償を行い、現在のブロックの（各予測方向における）第1の予測を生成する。第1の予測は、ブロック内の各サブブロックまたは画素の空間的勾配、時間的勾配、およびオプティカルフローを導出するために用いられ、これらを使用して第2の予測、例えば、サブブロックまたは画素の最終予測を生成する。以下、その詳細を説明する。

【0141】

双方向オプティカルフロー（BIO）方法は、双方向予測のためにブロック単位の動き補償の上で実行されるサンプル単位の動きの改良である。いくつかの実施例において、サンプルレベルの動きの改良は、シグナリングを用いない。

【0142】

50

ブロック動き補償後の基準 k ($k = 0, 1$) からの輝度を $I^{(k)}$ とし、
 【 0 1 4 3 】
 【数 2 2 】

$$\partial I^{(k)} / \partial x \text{ および } \partial I^{(k)} / \partial y$$

【 0 1 4 4 】

をそれぞれ $I^{(k)}$ 勾配の水平成分、垂直成分とする。オプティカルフローが有効であると仮定すると、動きベクトルフィールド (v_x, v_y) は、以下の式によって求められる。

10

【 0 1 4 5 】

【数 2 3 】

$$\partial I^{(k)} / \partial t + v_x \partial I^{(k)} / \partial x + v_y \partial I^{(k)} / \partial y = 0. \quad \text{式 (1 1)}$$

【 0 1 4 6 】

このオプティカルフロー方程式を各サンプルの動き軌跡に対してエルミート補間によって組み合わせることにより、両端にある両機能値 $I^{(k)}$ および導関数

20

【 0 1 4 7 】

【数 2 4 】

$$\partial I^{(k)} / \partial x \text{ および } \partial I^{(k)} / \partial y$$

【 0 1 4 8 】

に合致する固有の 3 次多項式が得られる。 $t = 0$ におけるこの多項式の値は、BIO 次式のような、BIO 予測となる。

30

【 0 1 4 9 】

【数 2 5 】

$$pred_{BIO} =$$

$$1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)} / \partial x - \tau_0 \partial I^{(0)} / \partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)} / \partial y - \tau_0 \partial I^{(0)} / \partial y)).$$

式 (1 2)

【 0 1 5 0 】

40

図 8 は、双方向オプティカルフロー (BIO) 法におけるオプティカルフローの軌跡の一例を示す。ここで、 d_0 および d_1 は、参照フレームまでの距離を示す。距離 d_0, d_1 は、 Ref_0 および Ref_1 の POC: $d_0 = POC(current) - POC(Ref_0)$, $d_1 = POC(current) - POC(Ref_1)$ に基づいて算出される。両方の予測が同じ時間方向から来たものである場合 (両方とも過去から来たものであるか、または両方とも将来から来たものである場合)、符号が異なる (即ち、 $d_0 \cdot d_1 < 0$)。このケースでは、予測が同じ時刻 (例えば、 t_0, t_1) からのものでない場合、BIO が適用される。

【 0 1 5 1 】

動きベクトルフィールド (v_x, v_y) は、点 A および B における値の差を最小化す

50

ることによって判定される。図 8 は、動き軌跡と参照フレーム平面の交差における例を示す。モデルは、に対するローカルテラー展開の第 1 の線形項のみを以下のように使用する。

【 0 1 5 2 】

【数 2 6 】

$$\Delta = \left(I^{(0)} - I^{(1)}_0 + v_x (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) + v_y (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \right)$$

式 (1 3)

10

【 0 1 5 3 】

上記式におけるすべての値は、(i ' , j ') として表されるサンプル位置に依存する。動きがローカル周辺エリアにおいて一貫していると仮定すると、は、現在の予測点 (i , j) を中心とする (2 M + 1) × (2 M + 1) 個の正方形窓 の内側で最小化することができる。式中、M は 2 に等しい。

【 0 1 5 4 】

【数 2 7 】

$$(v_x, v_y) = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{[i', j'] \in \Omega} \Delta^2[i', j'] \quad \text{式 (1 4)}$$

20

【 0 1 5 5 】

この最適化問題に対して、J E M は、まず垂直方向に最小化し、次に水平方向に最小化する簡単なアプローチを使用する。その結果、以下のようになる。

【 0 1 5 6 】

【数 2 8 】

$$v_x = (s_1 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_3}{(s_1 + r)} \right) : 0 \quad \text{式 (1 5)}$$

30

【 0 1 5 7 】

【数 2 9 】

$$v_y = (s_5 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5 + r)} \right) : 0 \quad \text{式 (1 6)}$$

40

【 0 1 5 8 】

ここで

【 0 1 5 9 】

【数 3 0 】

50

$$\begin{aligned}
s_1 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; \\
s_3 &= \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x); \\
s_2 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\
s_5 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; \\
s_6 &= \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)
\end{aligned}$$

式 (17)

【0160】

ゼロまたは非常に小さな数値での除算を回避するために、式(15)および式(16)において、正規化パラメータ r および m を導入する。式中、

【0161】

【数31】

$$r = 500 \cdot 4^{d-8} \quad \text{式 (18)}$$

【0162】

【数32】

$$m = 700 \cdot 4^{d-8} \quad \text{式 (19)}$$

【0163】

ここで、 d は映像サンプルのビット深度である。

【0164】

バイオメモリアクセスを通常の双方向予測動き補償と同じにするために、現在のブロック内の位置について、すべての予測値および勾配値

【0165】

【数33】

$$I^{(k)}, \partial I^{(k)} / \partial x, \partial I^{(k)} / \partial y$$

【0166】

を計算する。図9Aは、ステップ900の外部におけるアクセス位置を例示している。図9Aに示すとおり、式(17)において、予測ブロックの境界上の現在の予測点を中心とする $(2M+1) \times (2M+1)$ 個の正方形窓は、ブロックの外側の位置にアクセスする必要がある。JEMにおいて、ブロックの外部の

【0167】

【数34】

10

20

30

40

50

$$I^{(k)}, \partial I^{(k)}/\partial x, \partial I^{(k)}/\partial y$$

【 0 1 6 8 】

の値はブロックの内部で最も近い利用可能な数値に等しくなるように設定される。例えば、これは、図 9 B に示すように、パディング領域 9 0 1 として実装することができる。

【 0 1 6 9 】

B I O を用いることで、サンプル毎に動きフィールドを改良することができる。計算の複雑性を低減するために、J E M ではブロックに基づく B I O の設計が用いられている。動きの改良は、 4×4 ブロックに基づいて計算することができる。ブロックに基づく B I O において、 4×4 ブロックにおけるすべてのサンプルの、式 (1 7) における s_n の値を統合し、次いで、この統合した s_n の値を使用して、 4×4 ブロックのための B I O 動きベクトルオフセットを導出する。具体的には、ブロックに基づく B I O 導出には、以下の式を用いることができる。

【 0 1 7 0 】

【 数 3 5 】

$$\begin{aligned} s_{1,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j] \in \Omega(x,y)} (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x)^2; \\ s_{3,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x); \\ s_{2,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j] \in \Omega} (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x) (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y); \\ s_{5,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j] \in \Omega} (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y)^2; \\ s_{6,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y) \end{aligned}$$

式 (2 0)

【 0 1 7 1 】

式中、 b_k は、予測ブロックの k 番目の 4×4 ブロックに属するサンプルのセットを表し、式 (1 5) および式 (1 6) における s_n を ((s_n, b_k) > > 4) に置き換え、関連する動きベクトルオフセットを導出する。

【 0 1 7 2 】

シナリオによっては B I O の M V レジメンがノイズや不規則な動きで信頼できない場合がある。従って、B I O において、M V レジメンの大きさは閾値にクリップされる。閾値は、現在のピクチャの参照ピクチャがすべて一方向からのものであるか否かに基づいて判定される。例えば、現在のピクチャのすべての参照ピクチャが一方向からのものである場合、閾値を $1.2 \times 2^{14-d}$ に設定し、そうでない場合、閾値を $1.2 \times 2^{13-d}$ に設定する。

【 0 1 7 3 】

B I O の勾配は、H E V C 動き補償処理 (例えば、2 D 分離可能有限インパルス応答 (F I R)) に準拠した演算を使用して、動き補償補間と同時に計算してもよい。いくつかの実施形態において、前記 2 D 分離可能な F I R のための入力は、ブロック動きベクトルの端数部分に従って、動き補償処理および端数位置 ($\text{frac } X, \text{frac } Y$) のためのものと同じ参照フレームサンプルである。水平方向勾配

【 0 1 7 4 】

【 数 3 6 】

 $\partial I / \partial x$

【 0 1 7 5 】

の場合、まず、信号は、デスケーリングシフト $d - 8$ で、端数位置 $\text{frac } Y$ に対応する $\text{BIOfilter } S$ を使用して垂直方向に補間される。次に、 $18 - d$ によるデスケーリングシフトで端数位置 $\text{frac } X \text{ with}$ に対応する、水平方向に勾配フィルタ $\text{BIOfilter } G$ を適用する。垂直方向勾配

10

【 0 1 7 6 】

【 数 3 7 】

 $\partial I / \partial y$

【 0 1 7 7 】

の場合、デスケーリングシフト $d - 8$ で、端数位置 $\text{frac } Y$ に対応する $\text{BIOfilter } G$ を使用して勾配フィルタを垂直方向に適用する。そして、 $18 - d$ によるデスケーリングシフトで端数位置 $\text{frac } X$ に対応する水平方向の $\text{BIOfilter } S$ を使用して信号の移動を行う。適度な複雑性を保持するために、勾配計算 $\text{BIOfilter } G$ および信号変位 $\text{BIOfilter } F$ のための補間フィルタの長さはより短くてもよい（例えば 6 タップ）。表 2 は、 BIO におけるブロック動きベクトルの異なる端数位置の勾配計算に使用できる例示的なフィルタを示す。表 3 は、 BIO における予測信号の生成に使用できる例示的な補間フィルタを示す。

20

【 0 1 7 8 】

【 表 1 0 】

表 2 : BIO における勾配計算のための例示的なフィルタ

端数画素位置	グラジエント用補間フィルタ (BIOfilterG)
0	{8,-39,-3,46,-17,5}
1/16	{8,-32,-13,50,-18,5}
1/8	{7,-27,-20,54,-19,5}
3/16	{6,-21,-29,57,-18,5}
1/4	{4,-17,-36,60,-15,4}
5/16	{3,-9,-44,61,-15,4}
3/8	{1,-4,-48,61,-13,3}
7/16	{0,1,-54,60,-9,2}
1/2	{-1,4,-57,57,-4,1}

30

【 0 1 7 9 】

40

【表 1 1】

表 3：B I Oにおける予測信号生成のための例示的な補間フィルタ

端数画素位置	予測信号のための補間フィルタ(BIOfilterS)
0	{0,0,64,0,0,0}
1/16	{1,-3,64,4,-2,0}
1/8	{1,-6,62,9,-3,1}
3/16	{2,-8,60,14,-5,1}
1/4	{2,-9,57,19,-7,2}
5/16	{3,-10,53,24,-8,2}
3/8	{3,-11,50,29,-9,2}
7/16	{3,-11,44,35,-10,3}
1/2	{3,-10,35,44,-11,3}

10

【0 1 8 0】

本 J E Mにおいて、2つの予測が異なる参照ピクチャからのものである場合、B I Oをすべての双方向予測ブロックに適用することができる。C Uのローカル照明補償 (L I C)を有効にすると、B I Oを無効にすることができる。

【0 1 8 1】

いくつかの実施形態において、O B M Cは、通常のM C処理の後、1つのブロックに適用される。計算の複雑性を低減するために、O B M C処理中にB I Oを適用しなくてもよい。つまり、B I Oは、それ自身のM Vを使用する場合、1つのブロックのM C処理において適用され、O B M C処理において近傍のブロックのM Vを使用する場合、M C処理においては適用されない。

20

【0 1 8 2】

1 1 J V E T - N 0 2 3 6におけるオプティカルフロー (P R O F)による予測微調整

本寄稿では、オプティカルフローを用いたサブブロックに基づくアフィン動き補償予測を微調整する方法を提案している。サブブロックに基づくアフィン動き補償を行った後、オプティカルフロー方程式で導出された差を加算することで、予測サンプルを微調整し、これをオプティカルフロー予測微調整 (P R O F)と呼ぶ。提案した方法は、メモリアクセス帯域幅を増大させることなく、画素レベルの粒度におけるインター予測を実現することができる。

30

【0 1 8 3】

動き補償の粒度をより細かくするために、本寄稿では、オプティカルフローを用いたサブブロックに基づくアフィン動き補償予測を微調整する方法を提案している。サブブロックに基づくアフィン動き補償を行った後、オプティカルフロー方程式で導出された差を加算することで、輝度予測サンプルを微調整する。提案されたP R O Fは、以下の4つのステップとして説明する。

【0 1 8 4】

ステップ1) サブブロックに基づくアフィン動き補償を行い、サブブロック予測I (i , j)を生成する。

【0 1 8 5】

ステップ2) 3タップフィルタ [- 1 , 0 , 1]を使用して、個々のサンプル位置において、サブブロック予測の空間的勾配 $g_x(i, j)$ および $g_y(i, j)$ を算出する。

40

【0 1 8 6】

【数 3 8】

$$g_x(i, j) = I(i + 1, j) - I(i - 1, j)$$

【0 1 8 7】

【数 3 9】

50

$$g_y(i, j) = I(i, j + 1) - I(i, j - 1)$$

【 0 1 8 8 】

サブブロック予測は、勾配計算のために各側で 1 つの画素だけ拡張される。メモリの帯域幅および複雑性を低減するために、拡大された境界上の画素は、参照ピクチャにおける最も近い整数画素位置からコピーされる。従って、パディング領域のための追加の補間が回避される。

【 0 1 8 9 】

ステップ 3) オプティカルフロー方程式によって輝度予測の微調整を計算する。

10

【 0 1 9 0 】

【 数 4 0 】

$$\Delta I(i, j) = g_x(i, j) * \Delta v_x(i, j) + g_y(i, j) * \Delta v_y(i, j)$$

【 0 1 9 1 】

ここで、 $v(i, j)$ は、図 10 に示すように、 (i, j) によって表される、サンプル位置 $v(i, j)$ について算出された画素 MV と、画素 (i, j) が属するサブブロック MV のサブブロック MV との差分である。

20

【 0 1 9 2 】

サブブロック中心に対するアフィンモデルパラメータおよび画素位置は、サブブロックからサブブロックに変化しないので、第 1 のサブブロックについて $v(i, j)$ を計算し、同じ CU における他のサブブロックに再利用することができる。画素位置からサブブロックの中心までの水平および垂直オフセットを x 、 y とすると、 $v(x, y)$ は、以下の式で導出することができる。

【 0 1 9 3 】

【 数 4 1 】

$$\begin{cases} \Delta v_x(x, y) = c * x + d * y \\ \Delta v_y(x, y) = e * x + f * y \end{cases}$$

30

【 0 1 9 4 】

4 パラメータアフィンモデルの場合、

【 0 1 9 5 】

【 数 4 2 】

$$\begin{cases} c = f = \frac{v_{1x} - v_{0x}}{w} \\ e = -d = \frac{v_{1y} - v_{0y}}{w} \end{cases}$$

40

【 0 1 9 6 】

6 パラメータアフィンモデルの場合、

【 0 1 9 7 】

【 数 4 3 】

$$\begin{cases} c = \frac{v_{1x} - v_{0x}}{w} \\ d = \frac{v_{2x} - v_{0x}}{h} \\ e = \frac{v_{1y} - v_{0y}}{w} \\ f = \frac{v_{2y} - v_{0y}}{h} \end{cases}$$

【 0 1 9 8 】

10

ここで、 (v_{0x}, v_{0y}) 、 (v_{1x}, v_{1y}) 、 (v_{2x}, v_{2y}) は、左上、右上、左下の制御点動きベクトルであり、 w 、 h は、CUの幅および高さである。

【 0 1 9 9 】

ステップ4)最後に、輝度予測の微調整がサブブロック予測 $I(i, j)$ に加えられる。最終予測 I' は、次式のように生成される。

【 0 2 0 0 】

【数 4 4 】

$$I'(i, j) = I(i, j) + \Delta I(i, j)$$

20

【 0 2 0 1 】

1.2 既存の実装形態の欠点

J V E T - N 0 2 4 2 の設計における非線形 A L F (N L A L F) には、以下の問題がある。

【 0 2 0 2 】

(1) N L A L F では、多くのクリッピング演算が必要である。

【 0 2 0 3 】

(2) C T U に基づく A L F で、 $alf_num_available_temporal_filter_sets_luma$ が 0 である場合、利用可能な時間的輝度フィルタはない。しかしながら、 $alf_temporal_index$ は、引き続き信号通知してもよい。

30

【 0 2 0 4 】

(3) C T U に基づく A L F において、 $alf_signal_new_filter_chroma$ が 0 である場合、クロマ成分のために新しいフィルタは信号通知されず、時間的クロマフィルタが使用されると仮定される。しかしながら、時間的クロマフィルタが利用可能であることは保証されていない。

【 0 2 0 5 】

(4) C T U に基づく A L F において、 $alf_num_available_temporal_filter_sets_luma$ は、利用可能な時間的フィルタセットよりも大きくてもよい。

40

【 0 2 0 6 】

1.3 映像符号化のための適応ループフィルタリングのための例示的な方法

本開示の技術の実施形態は、既存の実装の欠点を克服し、それにより、より高い符号化効率を有する映像符号化を提供する。開示される技術に基づいた適応ループフィルタリングのための技術は、既存のおよび将来の映像符号化規格の両方を向上させることができ、様々な実施形態のために以下の例で解明される。以下に提供される開示される技術の例は、一般的な概念を説明するものであり、限定するものと解釈されるべきではない。一例において、明確に示されていない限り、逆に示されていない限り、これらの例に記載されている様々な特徴を組み合わせることができる。

1. サンプル差をクリッピングする代わりに、フィルタリング処理中の中間結果にクリ

50

ッピング演算を適用することが提案される。フィルタリング処理に利用される現在のサンプルの近傍のサンプル（隣接または非隣接）を、 $N(N-1)$ 個のグループに分類してもよい。

a. 一例において、1つのグループに対して1つ又は複数の中間結果を計算し、この1つまたは複数の中間結果に対してクリッピングを行ってもよい。

i. 例えば、1つのグループについて、まず各近傍の画素と現在の画素との間の差を計算し、次に、これらの差を、対応するALF係数を使用して重み平均してもよい（ $wAvgDiff$ と表記する）。グループに対して $wAvgDiff$ に対して1回クリッピングを行ってもよい。

b. 異なるグループに対して異なるクリッピングパラメータを使用してもよい。

c. 一例において、クリッピングは、サンプル差を乗算した、フィルタ係数の最終的な重み付けの合計に適用される。

i. 例えば、 $N=1$ であり、クリッピングは、 $K(d, b) = \min(b, \max(-b, d))$ がクリッピング関数であり、 k がクリッピングパラメータである場合、以下のように行ってもよい。

【0207】

【数45】

$$O(x, y) = I(x, y) + K\left(\sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x+i, y+j) - I(x, y)), k\right)$$

10

20

【0208】

1) さらに、代替的に、重み付けの合計は、次のようになる。

【0209】

【数46】

$$\sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x+i, y+j) - I(x, y))$$

【0210】

をさらに整数値に丸めてもよく、例えば、丸めて、あるいは丸めないでシフトしてもよい。

2. 1つのサンプルをフィルタリングするとき、 $N(N>1)$ 個の近傍のサンプルが1つのフィルタ係数を共有する場合、 N 個の近傍の画素すべてに対してクリッピング（例えば、非線形ALFによって必要とされる）を1回行ってもよい。

a. 例えば、 $I(x+i_1, y+j_1)$ と $I(x+i_2, y+j_2)$ が1つのフィルタ係数 $w(i_1, j_1)$ （または/および1つのクリッピングパラメータ $k(i_1, j_1)$ ）を共有する場合、クリッピングは以下のように1回行ってもよい。

【0211】

【数47】

$$clipValue(i1, j1) = K(I(x+i1, y+j1) + I(x+i2, y+j2) - 2 * I(x, y), k(i1, j1))$$

および、 $w(i1, j1) * clipValue(i1, j1)$

40

【0212】

を使用して、式(14)における

【0213】

【数48】

50

$$w(i1, j1) * K(I(x + i1, y + j1) - I(x, y), k(i1, j1)) + \\ w(i2, j2) * K(I(x + i2, y + j2) - I(x, y), k(i2, j2))$$

【 0 2 1 4 】

を置き換えてもよい。

【 0 2 1 5 】

i . 一例において、i 1 および i 2 は、対称的な位置にあってもよい。また、j 1 および j 2 は、対称的な位置にあってもよい。

1 . 一例において、i 1 は (- i 2) に等しく、j 1 は (- j 2) に等しい。

i i . 一例において、(x + i 1 , y + j 1) と (x , y) との間の距離と、(x + i 2 , j + j 2) と (x , y) との間の距離とは、同じであってもよい。

i i i . プレット 2 に開示された方法は、フィルタ形状が対称モードにある場合に有効化される。

i v . さらに、代替的に、I (x + i 1 , y + j 1) に関連付けられたクリッピングパラメータを、C l i p P a r a m と呼ばれるビットストリームから信号通知 / 導出してもよく、上述の k (i 1 , j 1) を、例えば 2 * C l i p P a r a m のような信号通知されたクリッピングパラメータから導出する。

b . 例えば、(i , j) C が 1 つのフィルタ係数 w 1 (または / および 1 つのクリッピングパラメータ k 1) を共有し、C が N 個の要素を含む場合、クリッピングは以下のように 1 回行われてもよい。

【 0 2 1 6 】

【数 4 9 】

$$clipValue = K \left(\left(\sum_{(i,j) \in C} I(x + i, y + j) \right) - N * I(x, y), k1 \right)$$

【 0 2 1 7 】

ここで、k 1 は、C に関連付けられたクリッピングパラメータであり、c l i p V a l u e * w 1 は、式 (1 4) における以下の項目を置き換えるために使用してもよい。

【 0 2 1 8 】

【数 5 0 】

$$\sum_{(i,j) \in C} w(i, j) * K(I(x + i, y + j) - I(x, y), k(i, j))$$

【 0 2 1 9 】

i . さらに、代替的に、I (x + i , y + j) に関連付けられたクリッピングパラメータを、C l i p P a r a m と呼ばれるビットストリームから信号通知 / 導出してもよく、k 1 を、例えば N * C l i p P a r a m のような信号通知されたクリッピングパラメータから導出する。

i i . 代替的に、

【 0 2 2 0 】

【数 5 1 】

10

20

30

40

50

$$\sum_{(i,j) \in C} I(x+i, y+j) \text{ または } (\sum_{(i,j) \in C} I(x+i, y+j)) - N * I(x, y)$$

【 0 2 2 1 】

を右シフトしてからクリップしてもよい。

c . 一例において、クリッピングは、N個の近傍のサンプルのM1 (M1 < = N) に対して1回行ってよい。

d . 一例において、N個の近傍のサンプルをM2個のグループに分類し、各グループに対して1回クリッピングを行ってもよい。

e . 一例において、この方法は、特定のまたはすべての色成分に適用してもよい。

i . 例えば、輝度成分に適用してもよい。

ii . 例えば、Cbおよび/またはCr成分に適用してもよい。

3 . 本開示では、入力をminおよびmaxを含む[min , max] の範囲にクリップするクリッピング関数K (min , max , input) を使用してもよい。

a . 一例において、入力を、minおよびmaxを除く範囲 (min , max) にクリップするクリッピング関数K (min , max , input) を、上記黒丸に使用してもよい。

b . 一例において、上記黒丸において、maxを含むがminを含まない範囲 (min , max) にあるように入力をクリップするクリッピング関数K (min , max , input) を使用してもよい。

c . 一例において、上記黒丸において、minを含むがmaxを含まない範囲 [min , max] にあるように入力をクリップするクリッピング関数K (min , max , input) を使用してもよい。

4 . 時間的ALF係数セットが利用できない (例えば、前にALF係数が符号化/復号化されていない、または符号化/復号化されたALF係数が「利用不可能」とマークされている) 場合、どの時間的ALF係数セットが使用されるかを示す信号通知を省略してもよい。

a . 一例において、時間的ALF係数セットが利用不可能な場合、CTB/ブロック/タイルグループ/タイル/スライス/ピクチャに対して新しいALF係数も固定ALF係数も使用されない場合、CTB/ブロック/タイルグループ/タイル/スライス/ピクチャに対してALFが許可されないと推測される。

i . さらに、代替的に、この場合、CTB/ブロック/タイルグループ/タイル/スライス/ピクチャに対してALFが適用されることが示されてもよいとしても (例えば、CTU/ブロックに対してalf__ctb__flagが真) 、ALFは、CTB/ブロック/タイルグループ/タイル/スライス/ピクチャに対して最終的に許可されないと推測されてもよい。

b . 一例において、時間的ALF係数セットが利用不可能な場合、適合ビットストリームにおけるCTB/ブロック/タイルグループ/タイル/スライス/ピクチャのために、新しいALF係数または固定ALF係数等のみを使用するように指示されてもよい。

i . 例えば、alf__use__new__filterまたはalf__use__fixed__filterのいずれかが真であるとする。

c . 一例において、時間的ALF係数セットが利用不可能な場合、ALFの採用が指示されているCTB/ブロック/タイルグループ/タイル/スライス/ピクチャに対して、新規のALF係数も固定のALF係数も使用することが示されていない、という条件が満たされた場合、そのビットストリームは不適合ビットストリームと見なされる。

i . 例えば、alf__use__new__filterおよびalf__use__fixed__filterの両方が偽であるビットストリームは、不適合ビットストリームと見なされる。

d . 一例において、alf__num__available__temporal__fi

10

20

30

40

50

l t e r _ s e t s _ l u m a が 0 である場合、a l f _ t e m p o r a l _ i n d e x は信号通知されなくてもよい。

e . 提案した方法は、異なる色成分に対して異なる方法で適用されてもよい。

5 . 1つのタイルグループ/タイル/スライス/ピクチャ/C T B /ブロック/映像ユニットに対して何個の時間的 A L F 係数セットを使用できるかは、使用可能な時間的 A L F 係数セット (A L F _ a v a i と示す)、例えば、「利用可能」とマークされた前に符号化/復号化された A L F 係数セットに依存してもよい。

a . 一例において、1つのタイルグループ/タイル/スライス/ピクチャ/C T B /ブロックに対して A L F _ a v a i 以下の時間的 A L F 係数セットを使用できる。

b . 1つのタイルグループ/タイル/スライス/ピクチャ/C T B /ブロックに対して、 $N \geq 0$ として、 $\min(N, A L F _ a v a i)$ 個以下の時間的 A L F 係数セットを使用できる。例えば、 $N = 5$ である。

6 . 新しい A L F 係数セットは、符号化/復号化された後、“利用可能”とマークしてもよい。一方、すべての“利用可能な” A L F 係数セットは、イントラランダムアクセスポイント (I R A P) アクセスユニットまたは/および I R A P ピクチャ、または/および瞬時復号化リフレッシュ (I D R) アクセスユニットまたは/および I D R ピクチャに遭遇したときに、すべて“利用不可能”とマークしてもよい。

a . “利用可能な” A L F 係数セットは、後続の符号化ピクチャ/タイル/タイルグループ/スライス/C T B /ブロックのための時間的 A L F 係数セットとして使用してもよい。

b . “利用可能な” A L F 係数セットは、 N に等しい最大サイズ ($N > 0$) の 1 つの A L F 係数セットリストに保持してもよい。

i . A L F 係数セットリストは、先入れ先出し順に保持してもよい。

c . “利用不可能”とマークされている場合、関連する A L F _ A P S 情報をビットストリームから削除するか、または他の A L F _ A P S 情報に置き換える。

7 . 各時間層に 1 つの A L F 係数セットリストを保持してもよい。

8 . K 個の近傍の時間層に対して 1 つの A L F 係数セットリストを保持してもよい。

9 . 前のピクチャからの予測 (表示順) のみを行うかどうかに基づいて、異なるピクチャごとに異なる A L F 係数セットリストを保持してもよい。

a . 例えば、1つの A L F 係数セットリストは、前のピクチャのみから予測されたピクチャに対して保持してもよい。

b . 例えば、前のピクチャおよび後のピクチャの両方から予測されたピクチャに対して、1つの A L F 係数セットリストを保持してもよい。

10 . A L F 係数セットリストは、I R A P アクセスユニットおよび/または I R A P ピクチャおよび/または I D R アクセスユニットおよび/または I D R ピクチャに遭遇した後、空にしてもよい。

11 . 異なる色成分に対して異なる A L F 係数セットリストを保持してもよい。

a . 一例において、1つの A L F 係数セットリストは、輝度成分のために保持される。

b . 一例において、C b および/または C r 成分のために、1つの A L F 係数セットリストが保持される。

12 . 1つの A L F 係数セットリストを保持してもよいが、異なるピクチャ/タイルグループ/タイル/スライス/C T U に対して、リストにおけるエントリに異なるインデックス (または優先順位) を割り当ててもよい。

a . 一例において、A L F 係数セットには、現在のピクチャ/タイルグループ/タイル/スライス/C T U との間の上り絶対時間層差のための上りインデックスが割り当てられてもよい。

b . 一例において、A L F 係数セットは、現在のピクチャ/タイルグループ/タイル/スライス/C T U との間に昇順絶対 P O C (ピクチャオーダカウント) 差の昇順インデックスを割り当てられてもよい。

c . 一例において、現在のピクチャ/タイルグループ/タイル/スライス/C T U に

よって許可されるK個のALF係数セットが存在し、これらのK個のALF係数セットは、最小のインデックスを有するK個のALF係数セットであってもよい。

d. 一例において、現在のピクチャ/タイルグループ/タイル/スライス/CTUがどの時間的ALF係数セットを使用するかは、リストにおける元のエントリインデックスの代わりに、割り当てられたインデックスに依存してもよい。

13. ALFで使用される近傍のサンプルは、K ($K \geq 1$) 個のグループに分類されてもよく、各グループに1つのクリッピングパラメータのセットが信号通知されてもよい。

14. クリッピングパラメータは、特定のまたは全ての固定ALFフィルタセットに対して予め規定されてもよい。

a. 代替的に、現在のタイルグループ/スライス/ピクチャ/タイルによって使用される特定のまたはすべての固定フィルタセットに対して、クリッピングパラメータを信号通知してもよい。

10

i. 一例において、クリッピングパラメータは、特定の色成分（例えば、輝度成分）についてのみ信号通知してもよい。

b. 代替的に、固定ALFフィルタセットを使用する場合、クリッピングを行わなくてもよい。

i. 一例において、クリッピングは、特定の色成分に対して行われてもよく、他の色成分に対して行われなくてもよい。

15. クリッピングパラメータはALF係数と共に記憶されてもよく、符号化されたCTU/CU/タイル/タイルグループ/スライス/ピクチャをフォローすることで継承してもよい。

20

a. 一例において、CTU/CU/タイル/タイルグループ/スライス/ピクチャが時間的ALF係数セットを使用する場合、対応するALFクリッピングパラメータを使用してもよい。

i. 一例において、クリッピングパラメータは、特定の色成分（例えば、輝度成分）についてのみ継承してもよい。

b. 代替的に、CTU/CU/タイル/タイルグループ/スライス/ピクチャが時間的ALF係数セットを使用する場合、クリッピングパラメータを信号通知してもよい。

i. 一例において、クリッピングパラメータは、特定の色成分（例えば、輝度成分）についてのみ信号通知してもよい。

30

c. 一例において、クリッピングパラメータは、特定の色成分に対して継承されてもよく、他の色成分に対して信号通知してもよい。

d. 一例において、時間的ALF係数セットが使用される場合、クリッピングは行われない。

i. 一例において、クリッピングは、特定の色成分に対して行われてもよく、他の色成分に対して行われなくてもよい。

16. 非線形ALFを使用するかどうかは、ALFフィルタセットのタイプ（例えば、固定ALFフィルタセット、時間的ALFフィルタセット、または信号通知ALF係数セット）に依存してもよい。

a. 一例において、現在のCTUが固定ALFフィルタセットまたは時間的ALFフィルタセット（別名、前に信号通知されたフィルタセットが使用される）を使用する場合、非線形ALFは、現在のCTUに使用されなくてもよい。

40

b. 一例において、`alf_luma_use_fixed_filter_flag`が1である場合、現在のスライス/タイルグループ/タイル/CTUに対して非線形ALFを使用してもよい。

17. 非線形ALFクリッピングパラメータは、ALFフィルタセットの種類（例えば、固定ALFフィルタセット、時間的ALFフィルタセット、または信号通知されたALF係数セット）に基づいて、条件付きで信号通知してもよい。

a. 一例において、非線形ALFクリッピングパラメータは、すべてのALFフィルタセットに対して信号通知してもよい。

50

b. 一例において、非線形 A L F クリッピングパラメータは、信号通知された A L F フィルタ係数セットに対してのみ信号通知してもよい。

c. 一例において、非線形 A L F クリッピングパラメータは、固定 A L F フィルタ係数セットに対してのみ信号通知してもよい。

【0222】

上述した例は、以下に説明する方法、例えば、方法 1 1 1 0 , 1 1 2 0 , 1 1 3 0 , 1 1 4 0 , 1 1 5 0 および 1 1 6 0 のコンテキストに含まれてもよく、これらの方法は、映像デコーダおよび / または映像エンコーダにおいて実装されてもよい。

【0223】

図 1 1 A は、例示的な映像処理方法のフローチャートを示す。方法 1 1 1 0 は、動作 1 1 1 2 において、映像の現在の映像ブロックに対して、フィルタ係数を使用し、少なくとも 1 つの中間結果を伴う 2 つ以上の演算を含むフィルタリング処理を行うことを含む。

10

【0224】

方法 1 1 1 0 は、動作 1 1 1 4 において、前記少なくとも 1 つの中間結果にクリッピング演算を適用することを含む。

【0225】

方法 1 1 1 0 は、動作 1 1 1 6 において、少なくとも 1 つの中間結果に基づいて、前記現在の映像ブロックと前記映像のビットストリーム表現との間での変換を行うことを含む。いくつかの実施形態において、前記少なくとも 1 つの中間結果は、前記フィルタ係数の重み付けの合計と、前記現在の映像ブロックの現在のサンプルと前記現在のサンプルの近傍のサンプルとの間の差とに基づく。

20

【0226】

図 1 1 B は、例示的な映像処理方法のフローチャートを示す。方法 1 1 2 0 は、動作 1 1 2 2 において、映像の現在の映像ブロックを映像のビットストリーム表現に符号化することを含み、現在の映像ブロックは、適応ループフィルタ (A L F) で符号化される。

【0227】

方法 1 1 2 0 は、動作 1 1 2 4 において、時間的適応フィルタの 1 つ以上の集合の可用性または使用に基づいて、前記ビットストリーム表現における前記時間的適応フィルタの前記 1 つ以上の集合内の時間的適応フィルタの集合の指示を選択的に含むことと、を含む。

【0228】

30

図 1 1 C は、例示的な映像処理方法のフローチャートを示す。この方法 1 1 3 0 は、動作 1 1 3 2 において、映像のビットストリーム表現における時間的適応フィルタのセットの指示に基づいて、適応ループフィルタ (A L F) で符号化される映像の現在の映像ブロックに適用可能な時間的適応フィルタの前記セットを備える 1 つ以上のセットの前記時間的適応フィルタの可用性または使用を判定することを含む。

【0229】

方法 1 1 3 0 は、動作 1 1 3 4 において、前記判定することに基づいて、時間的適応フィルタの前記セットを選択的に適用することによって、前記ビットストリーム表現から復号化された現在の映像ブロックを生成することを含む。

【0230】

40

図 1 1 D は、例示的な映像処理方法のフローチャートを示す。この方法 1 1 4 0 は、動作 1 1 4 2 において、適応ループフィルタで符号化された現在の映像ブロックに対して、利用可能な時間 A L F 係数セットに基づいて、時間的適応ループフィルタリング (A L F) 係数セットの数を判定することを含み、前記利用可能な時間 A L F 係数セットは、前記判定する前に符号化または復号化されており、前記 A L F 係数セットの数は、前記現在の映像ブロックを構成するタイルグループ、タイル、スライス、ピクチャ、符号化ツリーブロック (C T B) 、または映像ユニットに使用される。

【0231】

方法 1 1 4 0 は、動作 1 1 4 4 において、時間的 A L F 係数セットの前記数に基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現の間での変

50

換を行うことを含む。

【0232】

図11Eは、例示的な映像処理方法のフローチャートを示す。この方法1150は、動作1152において、映像の現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域のヘッダにおける適応ループフィルタリング（ALF）の指示が、前記ビットストリーム表現に関連付けられた適応パラメータセット（APS）ネットワーク抽象化層（NAL）ユニットにおけるALFの指示と等しいことを判定することを含む。

【0233】

方法1150は、動作1154において、変換を行うことを含む。

10

【0234】

図11Fは、例示的な映像処理方法のフローチャートを示す。この方法1160は、動作1162において、現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域で使用される適応ループフィルタのタイプに基づいて、非線形適応ループフィルタリング（ALF）動作を選択的に有効化することを含む。

【0235】

この方法1160は、動作1164において、前記選択的に有効化した後に、前記変換を行うことを含む。

【0236】

10 開示される技術の例示的な実装形態

20

10.1 実施形態#1

輝度およびクロマそれぞれについて1つのALF係数セットリストが保持され、2つのリストのサイズがそれぞれlumaALFSetSizeおよびchromaALFSetSizeであるとする。ALF係数セットリストの最大サイズは、それぞれlumaALFSetMax（例えば、lumaALFSetMaxが5に等しい）およびchromaALFSetMax（例えば、chromaALFSetMaxが5に等しい）である。

【0237】

新規に追加された部品は、二重の太い大括弧で囲まれ、即ち、{ { a } }は、“a”が追加されており、且つ、削除された部品は、二重の角括弧で囲まれ、即ち、[[a]]は、“a”が削除されていることを意味する。

30

7.3.3.2 適応ループフィルタデータ構文

【0238】

40

50

【表 1 2】

alf_data() {	デイスクリ プタ
alf_chroma_idc	tu(v)
alf_signal_new_filter_luma	u(1)
if(alf_signal_new_filter_luma>0) {	
alf_luma_use_fixed_filter_flag	u(1)
if(alf_luma_use_fixed_filter_flag){	
alf_luma_fixed_filter_set_index	tb(v)
alf_luma_fixed_filter_usage_pattern	u(1)
if (alf_luma_fixed_filter_usage_pattern > 0)	
for (i = 0; i < NumAlfFilters; i++)	
alf_luma_fixed_filter_usage[i]	u(1)
alf_num_available_temporal_filter_sets_luma	tb(1)
alf_luma_num_filters_signalled_minus1	tb(v)
alf_luma_type_flag	u(1)
if(alf_luma_num_filters_signalled_minus1>0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	tu(v)
}	
alf_luma_coeff_delta_flag	u(1)
if (!alf_luma_coeff_delta_flag && alf_luma_num_filters_signalled_minus1>0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	tu(v)
for(i = 0; i < (alf_luma_type_flag == 1) ? 2 : 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)
if (alf_luma_coeff_delta_flag) {	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++)	
alf_luma_coeff_flag[sigFiltIdx]	u(1)
}	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++){	
if (alf_luma_coeff_flag[sigFiltIdx]) {	
for (j = 0; j < (alf_luma_type_flag == 1) ? 6 : 12; j++) {	
alf_luma_coeff_delta_abs[sigFiltIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sigFiltIdx][j])	
alf_luma_coeff_delta_sign[sigFiltIdx][j]	u(1)
}	
}	
}	
}	
if (alf_chroma_idc > 0) {	
alf_signal_new_filter_chroma	u(1)
if (alf_signal_new_filter_chroma){	
alf_chroma_min_eg_order_minus1	tu(v)
for(i = 0; i < 2; i++)	
alf_chroma_eg_order_increase_flag[i]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[j]	uek(v)
if(alf_chroma_coeff_abs[j] > 0)	
alf_chroma_coeff_sign[j]	u(1)
}	
}	
}	
}	

10

20

30

40

【 0 2 3 9】

7 . 3 . 4 . 2 符号化ツリーユニット構文

【 0 2 4 0】

【表 1 3】

coding_tree_unit() {	デイスクリ フタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
if(slice_alf_enable_flag){	
alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if (alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	
{	
coding_alf_ctu_filter_set_index(CtbAddrInRs)	
}	
if(alf_chroma_idc==1 alf_chroma_idc==3)	
alf_ctb_flag[1][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if(alf_chroma_idc==2 alf_chroma_idc==3)	
alf_ctb_flag[2][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0,	
SINGLE_TREE)	
}	

10

coding_alf_ctu_filter_set_index(CtbAddrInRs, slice_type){	デイスクリ フタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
alf_use_new_filter	ae(1)
if (alf_use_new_filter == 0){	
alf_use_fixed_filter	ae(1)
}	
if (alf_use_new_filter){	
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16	
}	
else if (alf_use_fixed_filter){	
alf_fixed_filter_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = alf_fixed_filter_index	
}	
Else {{ if (alf_num_available_temporal_filter_sets_luma > 0) }}	
{	
alf_temporal_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16 + alf_temporal_index	
}	

20

30

40

【 0 2 4 1】

alf_signal_new_filter_lumaが1に等しい場合、新しい輝度フィルタセットが信号通知されることを指定する。alf_signal_new_filter_lumaが0に等しい場合、新しい輝度フィルタセットが信号通知されないことを示す。存在しない場合、0である。

alf_luma_use_fixed_filter_flagが1に等しい場合、適応ループフィルタに信号通知するのに固定フィルタセットを使用することを示す。alf_luma_use_fixed_filter_flagが0に等しい場合、適応ループフィルタを信号通知するのに固定フィルタ使用しないことを示す。

50

`alf_num_available_temporal_filter_sets_luma` は、現在のスライスに使用できる利用可能な時間的フィルタセットの数を示す。この数は `0 [[5]] { { lumaALFSetSize } }` であり得る。存在しない場合、`0` である。

{ { `alf_num_available_temporal_filter_sets_luma` が `0` のとき、`alf_signal_new_filter_luma` または `alf_luma_use_fixed_filter_flag` のどちらかが `1` にならなければならないという制約がある。 } }

`alf_signal_new_filter_chroma` が `1` に等しい場合、新しいクロマフィルタが信号通知されることを指定する。`alf_signal_new_filter_chroma` が `0` に等しい場合、新しいクロマフィルタが信号通知されないことを示す。

{ { `chromaALFSetSize` が `0` の時に `alf_signal_new_filter_chroma` が `1` にならないといけないという制約がある。 } }

【 0 2 4 2 】

10.2 実施形態 # 2

輝度およびクロマそれぞれについて1つのALF係数セットリストが保持され、2つのリストのサイズがそれぞれ `lumaALFSetSize` および `chromaALFSetSize` であるとする。ALF係数セットリストの最大サイズは、それぞれ `lumaALFSetMax` (例えば、`lumaALFSetMax` が `5` に等しい) および `chromaALFSetMax` (例えば、`chromaALFSetMax` が `5` に等しい) である。

【 0 2 4 3 】

新規に追加された部品は、二重の太い大括弧で囲まれ、即ち、{ { `a` } } は、“`a`” が追加されており、且つ、削除された部品は、二重の角括弧で囲まれ、即ち、[[`a`]] は、“`a`” が削除されていることを意味する。

7.3.3.2 適応ループフィルタデータ構文

【 0 2 4 4 】

10

20

30

40

50

【表 1 4】

alf_data() {	デイスクリ プタ
alf_chroma_idc	tu(v)
alf_signal_new_filter_luma	u(1)
if(alf_signal_new_filter_luma>0) {	
alf_luma_use_fixed_filter_flag	u(1)
if(alf_luma_use_fixed_filter_flag){	
alf_luma_fixed_filter_set_index	tb(v)
alf_luma_fixed_filter_usage_pattern	u(1)
if(alf_luma_fixed_filter_usage_pattern > 0)	
for(i = 0; i < NumAlfFilters; i++)	
alf_luma_fixed_filter_usage[i]	u(1)
{{ if(lumaALFSetSize > 0) }}	
alf_num_available_temporal_filter_sets_luma	tb(1)
alf_luma_num_filters_signalled_minus1	tb(v)
alf_luma_type_flag	u(1)
if(alf_luma_num_filters_signalled_minus1>0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	tu(v)
}	
alf_luma_coeff_delta_flag	u(1)
if (!alf_luma_coeff_delta_flag && alf_luma_num_filters_signalled_minus1>0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	tu(v)
for(i = 0; i < (alf_luma_type_flag == 1) ? 2 : 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)
if (alf_luma_coeff_delta_flag) {	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++)	
alf_luma_coeff_flag[sigFiltIdx]	u(1)
}	
for(sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++){	
if (alf_luma_coeff_flag[sigFiltIdx]) {	
for (j = 0; j < (alf_luma_type_flag == 1) ? 6 : 12; j++) {	
alf_luma_coeff_delta_abs[sigFiltIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sigFiltIdx][j])	
alf_luma_coeff_delta_sign[sigFiltIdx][j]	u(1)
}	
}	
}	
if (alf_chroma_idc > 0) {	
alf_signal_new_filter_chroma	u(1)
if(alf_signal_new_filter_chroma){	
alf_chroma_min_eg_order_minus1	tu(v)
for(i = 0; i < 2; i++)	
alf_chroma_eg_order_increase_flag[i]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[j]	uek(v)
if(alf_chroma_coeff_abs[j] > 0)	
alf_chroma_coeff_sign[j]	u(1)
}	
}	
}	
}	

10

20

30

40

【 0 2 4 5】

7 . 3 . 4 . 2 符号化ツリーユニット構文

【 0 2 4 6】

【表 1 5】

coding_tree_unit() {	デイスクリ フタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
if(slice_alf_enable_flag){	
alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if (alf_ctb_flag[0][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	
{	
coding_alf_ctu_filter_set_index(CtbAddrInRs)	
}	
if(alf_chroma_idc==1 alf_chroma_idc==3)	
alf_ctb_flag[1][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
if(alf_chroma_idc==2 alf_chroma_idc==3)	
alf_ctb_flag[2][xCtb>>Log2CtbSize][yCtb>>Log2CtbSize]	ae(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0,	
SINGLE_TREE)	
}	

10

coding_alf_ctu_filter_set_index(CtbAddrInRs, slice_type){	デイスクリ フタ
xCtb = (CtbAddrInRs%PicWidthInCtbsY)<<CtbLog2SizeY	
yCtb = (CtbAddrInRs/PicWidthInCtbsY)<<CtbLog2SizeY	
alf_use_new_filter	ae(1)
if (alf_use_new_filter == 0){	
alf_use_fixed_filter	ae(1)
}	
if (alf_use_new_filter){	
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16	
}	
else if (alf_use_fixed_filter){	
alf_fixed_filter_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = alf_fixed_filter_index	
}	
Else {{ if (alf_num_available_temporal_filter_sets_luma > 0) }}	
{	
alf_temporal_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb>>Log2CtbSize][yCtb>>Log2Ct	
bSize] = 16 + alf_temporal_index	
}	

20

30

40

【 0 2 4 7】

alf_signal_new_filter_lumaが1に等しい場合、新しい輝度フィルタセットが信号通知されることを指定する。alf_signal_new_filter_lumaが0に等しい場合、新しい輝度フィルタセットが信号通知されないことを示す。存在しない場合、0である。

alf_luma_use_fixed_filter_flagが1に等しい場合、適応ループフィルタに信号通知するのに固定フィルタセットを使用することを示す。alf_luma_use_fixed_filter_flagが0に等しい場合、適応ループフィルタを信号通知するのに固定フィルタ使用ないことを示す。

50

`alf_num_available_temporal_filter_sets_luma` は、現在のスライスに使用できる利用可能な時間的フィルタセットの数を示し、この数は $0 \leq \text{alf_num_available_temporal_filter_sets_luma} \leq \text{chroma_alf_set_size}$ であり得る。存在しない場合、0である。

{ { `alf_num_available_temporal_filter_sets_luma` が 0 のとき、`alf_signal_new_filter_luma` または `alf_luma_use_fixed_filter_flag` のどちらかが 1 にならなければならないという制約がある。 } }

`alf_signal_new_filter_chroma` が 1 に等しい場合、新しいクロマフィルタが信号通知されることを指定する。`alf_signal_new_filter_chroma` が 0 に等しい場合、新しいクロマフィルタが信号通知されないことを示す。

{ { `chroma_alf_set_size` が 0 の時に `alf_signal_new_filter_chroma` が 1 にならないといけないという制約がある。 } }

【0248】

いくつかの実施形態において、下記のような技術的解決策を実装することができる。

【0249】

A1．映像の現在の映像ブロックに対して、フィルタ係数を使用し、少なくとも1つの中間結果を伴う2つ以上の演算を含むフィルタリング処理を行うことと、前記少なくとも1つの中間結果にクリッピング演算を適用することと、前記少なくとも1つの中間結果に基づいて、前記現在の映像ブロックと前記映像のビットストリーム表現との間での変換を行うこととを含み、前記少なくとも1つの中間結果は、前記フィルタ係数の重み付けの合計と、前記現在の映像ブロックの現在のサンプルと前記現在のサンプルの近傍のサンプルとの間の差とに基づく、映像処理方法。

【0250】

A2．前記現在のサンプルに対して、前記現在のサンプルの近傍のサンプルを複数のグループに分類することをさらに含み、前記複数のグループの各々における中間結果に異なるパラメータを使用して前記クリッピング演算を適用する、解決策 A1 に記載の方法。

【0251】

A3．前記少なくとも1つの中間結果は、前記現在のサンプルと前記複数のグループの各々における前記近傍のサンプルとの間の差の重み付け平均を含む、解決策 A2 に記載の方法。

【0252】

A4．前記現在の映像ブロックのサンプルの複数の近傍のサンプルが1つのフィルタ係数を共有し、前記複数の近傍のサンプルの各々に1回の前記クリッピング演算を適用する、解決策 A1 に記載の方法。

【0253】

A5．前記複数の近傍のサンプルのうちの少なくとも2つのサンプルの位置が、前記現在の映像ブロックの前記サンプルに対して対称である、解決策 A4 に記載の方法。

【0254】

A6．前記フィルタリング処理に関連付けられたフィルタ形状が対称モードである、解決策 A4 または A5 に記載の方法。

【0255】

A7．前記クリッピング演算の1つ以上のパラメータが前記ビットストリーム表現で信号通知される解決策 A4 ~ A6 のいずれかに記載の方法。

【0256】

A8．前記現在の映像ブロックのサンプルがN個の近傍のサンプルを含み、前記N個の近傍のサンプルのM1個の近傍のサンプルにクリッピング演算を1回適用し、M1およびNは正の整数であり、 $M1 \leq N$ である、解決策 A1 に記載の方法。

【0257】

10

20

30

40

50

A 9 . 前記現在の映像ブロックのサンプルに対して、前記サンプルのN個の近傍のサンプルをM 2 個のグループに分類することをさらに含み、前記クリッピング演算は、前記M 2 個のグループの各々に1回適用され、M 2 およびNは、正の整数である、解決策 A 1 に記載の方法。

【 0 2 5 8 】

A 1 0 . 前記クリッピング演算は、前記現在の映像ブロックに関連付けられた輝度成分に適用される、解決策 A 1 に記載の方法。

【 0 2 5 9 】

A 1 1 . 前記クリッピング演算は、現在の映像ブロックに関連付けられた C b 成分または C r 成分に適用される、解決策 A 1 に記載の方法。

10

【 0 2 6 0 】

A 1 2 . 前記クリッピング演算は、 $K(\min, \max, input)$ として定義され、式中、 $input$ は、前記クリッピング演算への入力であり、 \min は、前記クリッピング演算の出力の公称最小値であり、 \max は、前記クリッピング演算の出力の公称最大値である、解決策 A 1 ~ A 1 1 のいずれかに記載の方法。

【 0 2 6 1 】

A 1 3 . 前記クリッピング演算の出力の実際の最大値が前記公称最大値よりも小さく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値よりも大きい、解決策 A 1 2 に記載の方法。

【 0 2 6 2 】

20

A 1 4 . 前記クリッピング演算の出力の実際の最大値が前記公称最大値と等しく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値よりも大きい、解決策 A 1 2 に記載の方法。

【 0 2 6 3 】

A 1 5 . 前記クリッピング演算の出力の実際の最大値が前記公称最大値よりも小さく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値と等しい、解決策 A 1 2 に記載の方法。

【 0 2 6 4 】

A 1 6 . 前記クリッピング演算の出力の実際の最大値が前記公称最大値と等しく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値と等しい、解決策 A 1 2 に記載の方法。

30

【 0 2 6 5 】

A 1 7 . 前記フィルタリング処理は、複数の A L F フィルタ係数セットで構成される適応ループフィルタリング (A L F) 処理を含む、解決策 A 1 に記載の方法。

【 0 2 6 6 】

A 1 8 . 前記複数の A L F フィルタ係数セットのうちの1つ以上の A L F フィルタ係数セットに対して、前記クリッピング演算のための少なくとも1つのパラメータを予め規定する、解決策 A 1 7 に記載の方法。

【 0 2 6 7 】

A 1 9 . 前記現在の映像ブロックを構成するタイルグループ、スライス、ピクチャまたはタイルのビットストリーム表現において、前記クリッピング演算のための少なくとも1つのパラメータを信号通知する、解決策 A 1 7 に記載の方法。

40

【 0 2 6 8 】

A 2 0 . 前記少なくとも1つのパラメータは、現在の映像ブロックに関連付けられた1つ以上の色成分のみに対して信号通知される、解決策 A 1 9 に記載の方法。

【 0 2 6 9 】

A 2 1 . 前記複数の A L F フィルタ係数セットのうちの少なくとも1つと、前記クリッピング演算のための1つ以上のパラメータとが、同じ記憶位置に記憶され、かつ、前記複数の A L F フィルタ係数セットのうちの前記少なくとも1つ以上または、前記1つ以上のパラメータが、前記現在の映像ブロックを含む符号化ツリーユニット (C T U)、符号化

50

ユニット（CU）、タイル、タイルグループ、スライス、またはピクチャによって継承される、解決策 A 1 7 に記載の方法。

【0270】

A 2 2 . 前記クリッピング演算は、現在の映像ブロックを構成するCTU、CU、タイル、タイルグループ、スライス、またはピクチャのフィルタリング処理に時間的ALF係数セットを使用すると判定された場合、複数のALFフィルタ係数セットの時間的ALF係数セットに対応する1つ以上のパラメータを使用するように構成される、解決策 A 2 1 に記載の方法。

【0271】

A 2 3 . 時間的ALF係数セットに対応する1つ以上のパラメータは、現在の映像ブロックに関連付けられた1つ以上の色成分に対してのみ使用される、解決策 A 2 2 に記載の方法。

10

【0272】

A 2 4 . 前記複数のALFフィルタ係数セットの時間的ALF係数セットに対応する1つ以上のパラメータは、時間的ALF係数セットが、現在の映像ブロックを構成する前記CTU、前記CU、前記タイル、前記タイルグループ、前記スライス、または前記ピクチャのフィルタリング処理で使用されていると判定されたときに、前記ビットストリーム表現で信号通知される、解決策 A 2 1 に記載の方法。

【0273】

A 2 5 . 前記時間的ALF係数セットに対応する前記1つ以上のパラメータは、前記現在の映像ブロックに関連付けられた1つ以上の色成分に対してのみ信号通知される、解決策 A 2 4 に記載の方法。

20

【0274】

A 2 6 . 前記現在の映像ブロックに関連付けられた第1の色成分のための前記1つ以上のパラメータの第1のセットのパラメータを信号通知し、前記現在の映像ブロックに関連付けられた第2の色成分のための前記1つ以上のパラメータの第2のセットのパラメータを継承する、解決策 A 2 1 に記載の方法。

【0275】

A 2 7 . 前記変換は、前記ビットストリーム表現から前記現在のブロックを生成する、解決策 A 1 ~ A 2 6 のいずれかに記載の方法。

30

【0276】

A 2 8 . 前記変換は、前記現在の映像ブロック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 A 1 ~ A 2 6 のいずれかに記載の方法。

【0277】

A 2 9 . 処理装置と、その処理装置に命令が記憶された非一時的メモリとを備える装置であって、命令が処理装置によって実装されることにより、処理装置に、解決策 A 1 ~ A 2 8 のいずれかに記載の方法を実施させる映像システムの装置。

【0278】

A 3 0 . 非一時的なコンピュータ可読媒体に記憶されたコンピュータプログラム製品であって、解決策 A 1 ~ A 2 8 のいずれかに記載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製品。

40

【0279】

いくつかの実施形態において、下記のような技術的解決策を実装することができる。

【0280】

B 1 . 映像の現在の映像ブロックを前記映像のビットストリーム表現に符号化することであって、前記現在の映像ブロックは、適応ループフィルタ（ALF）で符号化される、符号化することと、時間的適応フィルタの1つ以上のセットの可用性または使用に基づいて、前記ビットストリーム表現における前記時間的適応フィルタの前記1つ以上のセット内の時間的適応フィルタのセットの指示を選択的に含むことと、を含む、映像処理方法。

【0281】

50

B 2 . 前記時間的適応フィルタのセットが利用不可能な場合、前記セットの指示を前記ビットストリーム表現から排除する、解決策 B 1 に記載の方法。

【 0 2 8 2 】

B 3 . 前記時間的適応フィルタのセットが利用不可能な場合、前記セットの指示を前記ビットストリーム表現に含める、解決策 B 1 または B 2 に記載の方法。前記 1 つ以上の時間的適応フィルタのセットのいずれも利用可能でない場合、この指示はビットストリーム表現から除外される、解決策 B 1 ~ B 3 のいずれかに記載の方法。

【 0 2 8 3 】

B 4 . 前記 1 つ以上の時間的適応フィルタのセットの各々が、1 つのフィルタインデックスに関連付けられる、解決策 B 1 ~ B 3 のいずれかに記載の方法。

10

【 0 2 8 4 】

B 5 . 前記 1 つ以上の時間的適応フィルタのセットのいずれも利用可能でない場合、固定フィルタを使用する指示は真に等しいものとする、解決策 B 1 ~ B 3 のいずれかに記載の方法。

【 0 2 8 5 】

B 6 . 前記 1 つ以上の時間的適応フィルタのセットのいずれも利用可能でない場合、時間的適応フィルタを使用する指示は偽に等しいものとする、解決策 B 1 ~ B 3 のいずれかに記載の方法。

【 0 2 8 6 】

B 7 . 前記 1 つ以上の時間的適応フィルタのセットのいずれも利用可能でない場合、固定フィルタのインデックスの指示がビットストリーム表現に含まれる、解決策 B 1 ~ B 3 のいずれかに記載の方法。

20

【 0 2 8 7 】

B 8 . 映像のビットストリーム表現における時間的適応フィルタのセットの指示に基づいて、適応ループフィルタ (A L F) で符号化される映像の現在の映像ブロックに適用可能な時間的適応フィルタの前記セットを備える 1 つ以上のセットの前記時間的適応フィルタの可用性または使用を判定することと、前記判定することに基づいて、時間的適応フィルタの前記セットを選択的に適用することによって、前記ビットストリーム表現から復号化された現在の映像ブロックを生成することと、を含む、映像処理方法。

【 0 2 8 8 】

30

B 9 . 前記時間的適応フィルタのセットが利用不可能な場合、前記生成は、前記時間的適応フィルタのセットを適用せずに行われる、解決策 B 8 に記載の方法。

【 0 2 8 9 】

B 1 0 . 前記時間的適応フィルタのセットが利用不可能な場合、前記生成を行うことは、前記時間的適応フィルタのセットを適用することを含む、解決策 B 8 または B 9 に記載の方法。

【 0 2 9 0 】

B 1 1 . 前記 1 つ以上の時間的適応フィルタのセットが 1 つの適応パラメータセット (A P S) に含まれ、前記指示が 1 つの A P S インデックスである、解決策 B 1 ~ B 1 0 のいずれかに記載の方法。

40

【 0 2 9 1 】

B 1 2 . 異なる方向の勾配計算に基づいて、1 つ以上の時間的適応フィルタのセットのうちの少なくとも 1 つの時間的適応フィルタのためのフィルタインデックスを判定することとをさらに含む、解決策 B 1 ~ B 1 0 のいずれかに記載の方法。

【 0 2 9 2 】

B 1 3 . 前記時間的適応フィルタの 1 つ以上のセットのいずれも利用可能でなく、かつ新しい A L F 係数セットおよび固定 A L F 係数セットが、前記現在の映像ブロックを構成する符号化ツリーブロック (C T B)、ブロック、タイルグループ、タイル、スライス、またはピクチャにおいて使用されないことを判定することと、前記判定に基づいて、適応ループフィルタリングが無効であると推論することと、をさらに含む、解決策 B 1 ~ B 1

50

1 のいずれかに記載の方法。

【0293】

B14．前記ビットストリーム表現は、前記1つ以上のセットの時間的適応フィルタの少なくとも1つが利用不可能であることに呼応して、新しいALF係数セットの使用の第1の指示と、固定ALF係数セットの使用の第2の指示とを含み、第1の指示と第2の指示のうちの1つが正確に前記ビットストリーム表現において真である、解決策B1～B11のいずれかに記載の方法。

【0294】

B15．前記ビットストリーム表現は、前記ALFの演算に関連付けられたフォーマット規則に準拠する、解決策B14に記載の方法。

10

【0295】

B16．前記1つ以上の時間的適応フィルタのセットのいずれも利用可能でないことに呼応して、前記ビットストリーム表現は、前記ALFが有効化されていること、および新しいALF係数セットおよび固定ALF係数セットが、前記現在の映像ブロックを構成する符号化ツリーブロック(CTB)、ブロック、タイルグループ、タイル、スライス、またはピクチャにおいて使用されないことを示す指示を含む、解決策B1～B11のいずれかに記載の方法。

【0296】

B17．前記ビットストリーム表現は、前記ALFの動作に関連付けられたフォーマット規則に準拠しない、解決策B16に記載の方法。

20

【0297】

B18．前記ALFが、現在の映像ブロックに関連付けられた1つ以上の色成分に適用される、解決策B1～B17のいずれかに記載の方法。

【0298】

B19．適応ループフィルタで符号化された現在の映像ブロックに対して、利用可能な時間的ALF係数セットに基づいて、時間的適応ループフィルタリング(ALF)係数セットの数を判定することであって、前記利用可能な時間的ALF係数セットは、前記判定する前に符号化または復号化されており、前記ALF係数セットの数は、前記現在の映像ブロックを構成するタイルグループ、タイル、スライス、ピクチャ、符号化ツリーブロック(CTB)、または映像ユニットに使用される、判定することと、時間的ALF係数セットの前記数に基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現の間での変換を行うことと、を含む、映像処理方法。

30

【0299】

B20．時間的ALF係数セットの数の最大数が、利用可能な時間的ALF係数セットの数に等しく設定される、解決策B19に記載の方法。

【0300】

B21．前記時間的ALF係数セットの数は、利用可能な時間的ALF係数セットの数と予め規定された数Nとのうちの小さい方に等しく設定され、ここで、Nは整数であり、N>0である、解決策B20に記載の方法。

【0301】

B22．N=5である、解決策B21に記載の方法。

40

【0302】

B23．映像の現在の映像ブロックと映像のビットストリーム表現との間での変換の一部として、1つ以上の新しい適応ループフィルタリング(ALF)係数セットを処理することであって、現在の映像ブロックが適応ループフィルタで符号化される、処理することと、前記処理の後に、この1つ以上の新しいALF係数セットを利用可能なALF係数セットとして指定することと、を含む、映像処理方法。

【0303】

B24．イントラランダムアクセスポイント(IRAP)アクセスユニット、IRAPピクチャ、瞬時復号化リフレッシュ(IDR)アクセスユニット、またはIDRピクチャ

50

に遭遇することと、前記遭遇したことに基づいて、前記利用可能な A L F 係数セットを、利用不可能な A L F 係数セットとして指定することと、をさらに含む解決策 B 2 3 に記載の方法。

【 0 3 0 4 】

B 2 5 . 前記利用可能な A L F 係数セットのうちの少なくとも 1 つは、前記現在の映像ブロックの後続の映像ブロックのための時間的 A L F 係数セットである、解決策 B 2 3 または B 2 4 に記載の方法。

【 0 3 0 5 】

B 2 6 . 前記利用可能な A L F 係数セットが、最大サイズが N である A L F 係数セットリストに保持され、ここで、N が整数である、解決策 B 2 3 ~ B 2 5 のいずれかに記載の方法。

10

【 0 3 0 6 】

B 2 7 . 前記 A L F 係数セットリストは、先入れ先出し (F I F O) の順序で保持される、解決策 B 2 6 に記載の方法であって、方法。

【 0 3 0 7 】

B 2 8 . 前記現在の映像ブロックに関連付けられた各時間層に 1 つの A L F 係数セットリストを保持する、解決策 B 1 ~ B 2 7 のいずれかに記載の方法。

【 0 3 0 8 】

B 2 9 . 前記現在の映像ブロックに関連付けられた K 個の近傍の時間層ごとに 1 つの A L F 係数セットリストを保持する、解決策 B 1 ~ B 2 7 のいずれかに記載の方法。

20

【 0 3 0 9 】

B 3 0 . 前記現在の映像ブロックを含む現在のピクチャのために第 1 の A L F 係数セットリストを保持し、前記現在のピクチャの後続のピクチャのために第 2 の A L F 係数セットリストを保持する、解決策 B 1 ~ B 2 7 のいずれかに記載の方法。

【 0 3 1 0 】

B 3 1 . 前記現在のピクチャに基づいて、前記現在のピクチャに続く前記ピクチャ画像を予測し、前記第 1 の A L F 係数セットリストが前記第 2 の A L F 係数セットリストと同じである、解決策 B 3 0 に記載の方法。

【 0 3 1 1 】

B 3 2 . 前記現在のピクチャに続く前記ピクチャおよび前記現在のピクチャの前のピクチャに基づいて、現在のピクチャを予測し、前記第 1 の A L F 係数セットリストが前記第 2 の A L F 係数セットリストと同じである、解決策 B 3 0 に記載の方法。

30

【 0 3 1 2 】

B 3 3 . イントラランダムアクセスポイント (I R A P) アクセスユニット、I R A P ピクチャ、瞬時復号化リフレッシュ (I D R) アクセスユニット、または I D R ピクチャに遭遇することと、前記遭遇した後に、1 つ以上の A L F 係数セットリストを空にすること、をさらに含む解決策 B 2 3 に記載の方法。

【 0 3 1 3 】

B 3 4 . 現在の映像ブロックに関連付けられた異なる色成分について、異なる A L F 係数セットリストを保持する、解決策 B 2 3 に記載の方法。

40

【 0 3 1 4 】

B 3 5 . 前記異なる色成分は、輝度成分、C r 成分、および C b 成分のうちの 1 つ以上を含む、解決策 B 3 4 に記載の方法。

【 0 3 1 5 】

B 3 6 . 複数のピクチャ、タイルグループ、タイル、スライス、または符号化ツリーユニット (C T U) に対して 1 つの A L F 係数セットリストを保持し、前記 1 つの A L F 係数セットリストのインデックス付けは、前記複数のピクチャ、タイルグループ、タイル、スライス、または符号化ツリーユニット (C T U) の各々に対して異なる、解決策 B 2 3 に記載の方法。

【 0 3 1 6 】

50

B 3 7 . 前記インデックス付けは、昇順であり、前記現在の映像ブロックに関連付けられた第 1 の時間層インデックスと、前記現在の映像ブロックを構成する現在のピクチャ、タイルグループ、タイル、スライス、または符号化ツリーユニット (C T U) に関連付けられた第 2 の時間層インデックスとに基づいている、解決策 B 3 6 に記載の方法。

【 0 3 1 7 】

B 3 8 . 前記インデックス付けは、昇順であり、前記現在の映像ブロックに関連付けられたピクチャオーダカウント (P O C) と、前記現在の映像ブロックを構成する現在のピクチャ、タイルグループ、タイル、スライス、または符号化ツリーユニット (C T U) に関連付けられた第 2 の P O C とに基づいている、解決策 B 3 6 に記載の方法。

【 0 3 1 8 】

B 3 9 . 前記インデックス付けは、前記利用可能な A L F 係数セットに割り当てられる最小のインデックスを含む、解決策 B 3 6 に記載の方法。

【 0 3 1 9 】

B 4 0 . 前記変換は、クリッピング演算を含み、前記方法は、現在の映像ブロックのサンプルに対して、前記サンプルの近傍のサンプルを複数のグループに分類することと、ビットストリーム表現で信号通知された単一のパラメータのセットを使用して、前記複数のグループの各々についてのクリッピング演算を行うことと、をさらに含む、解決策 B 2 3 に記載の方法。

【 0 3 2 0 】

B 4 1 . 前記変換は、クリッピング演算を含み、前記クリッピング演算のための 1 つのパラメータのセットが、前記 1 つ以上の新しい A L F 係数セットに対して予め規定される、解決策 B 2 3 に記載の方法。

【 0 3 2 1 】

B 4 2 . 前記変換は、クリッピング演算を含み、前記クリッピング演算のための 1 つのパラメータのセットが、前記 1 つ以上の新しい A L F 係数セットに対して前記ビットストリーム表現で信号通知される、解決策 B 2 3 に記載の方法。

【 0 3 2 2 】

B 4 3 . 映像の現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域のヘッダにおける適応ループフィルタリング (A L F) の指示が、前記ビットストリーム表現に関連付けられた適応パラメータセット (A P S) ネットワーク抽象化層 (N A L) ユニットにおける A L F の指示と等しいことを判定することと、前記変換を行うことと、を含む、映像処理方法。

【 0 3 2 3 】

B 4 4 . 前記映像領域は 1 つのピクチャである、解決策 B 4 3 に記載の方法。

【 0 3 2 4 】

B 4 5 . 前記映像領域は 1 つのスライスである、解決策 B 4 3 に記載の方法。

【 0 3 2 5 】

B 4 6 . 現在の映像ブロックと前記映像のビットストリーム表現との間での変換のために、前記映像の映像領域で使用される適応ループフィルタのタイプに基づいて、非線形適応ループフィルタリング (A L F) 動作を選択的に有効化することと、前記選択的に有効化した後に、前記変換を行うこととを含む、映像処理方法。

【 0 3 2 6 】

B 4 7 . 前記映像領域は符号化ツリーユニット (C T U) であり、前記適応ループフィルタのタイプが固定 A L F セットまたは時間的 A L F セットを含むと判定されると、前記非線形 A L F 動作は無効化される、解決策 B 4 6 に記載の方法。

【 0 3 2 7 】

B 4 8 . 前記映像領域は、スライス、タイルグループ、タイルまたは符号化ツリーユニット (C T U) であり、前記非線形 A L F 動作は、前記適応ループフィルタのタイプが固定 A L F セットを含むと判定されると、有効化される、解決策 B 4 6 に記載の方法。

【 0 3 2 8 】

10

20

30

40

50

B 4 9 . 前記非線形 A L F 動作のためのビットストリーム表現において、1つ以上のクリッピングパラメータを選択的に信号通知することをさらに含む、解決策 B 4 6 に記載の方法。

【 0 3 2 9 】

B 5 0 . 前記1つ以上のクリッピングパラメータが信号通知される、解決策 B 4 9 に記載の方法。

【 0 3 3 0 】

B 5 1 . 前記1つ以上のクリッピングパラメータは、前記ビットストリーム表現で信号通知される A L F フィルタ係数セットに対して信号通知される、解決策 B 4 9 に記載の方法。

【 0 3 3 1 】

B 5 2 . 前記適応ループフィルタのタイプが固定 A L F セットを含むと判定されると、前記1つ以上のクリッピングパラメータが信号通知される、解決策 B 4 9 に記載の方法。

【 0 3 3 2 】

B 5 3 . 前記変換は、前記ビットストリーム表現から前記現在のブロックを生成する、解決策 B 1 9 ~ B 5 2 のいずれかに記載の方法。

【 0 3 3 3 】

B 5 4 . 前記変換は、前記現在の映像ブロック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 B 1 9 ~ B 5 2 のいずれかに記載の方法。

【 0 3 3 4 】

B 5 5 . 処理装置と、その処理装置に命令が記憶された非一時的メモリとを備える装置であって、命令が処理装置によって実装されることにより、処理装置に、解決策 B 1 ~ A 5 4 のいずれかに記載の方法を実施させる映像システムの装置。

【 0 3 3 5 】

B 5 6 . 非一時的なコンピュータ可読媒体に記憶されたコンピュータプログラム製品であって、解決策 B 1 ~ B 5 4 のいずれかに記載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製品。

【 0 3 3 6 】

いくつかの実施形態において、下記のような技術的解決策を実装することができる。

【 0 3 3 7 】

C 1 . 現在の映像ブロックに対して、少なくとも1つの中間結果を伴う2つ以上の演算を含むフィルタリング処理を行うことと、前記少なくとも1つの中間結果にクリッピング演算を適用することと、前記フィルタリング演算に基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換を行うこと、を含む映像処理方法。

【 0 3 3 8 】

C 2 . 前記現在の映像ブロックのサンプルに対して、前記サンプルのうちの近傍のサンプルを複数のグループに分類することをさらに含み、前記複数のグループの各々における中間結果に異なるパラメータを使用して前記クリッピング演算を適用する、解決策 C 1 に記載の方法。

【 0 3 3 9 】

C 3 . 前記少なくとも1つの中間結果は、前記現在のサンプルと前記複数のグループの各々における前記近傍のサンプルとの間の差の重み付け平均を含む、解決策 C 2 に記載の方法。

【 0 3 4 0 】

C 4 . 前記フィルタリング処理は、フィルタ係数を使用し、前記少なくとも1つの中間結果は、前記フィルタ係数と、前記現在のサンプルと前記近傍のサンプルとの間の差との重み付けの合計を含む、解決策 C 2 に記載の方法。

【 0 3 4 1 】

C 5 . 前記現在の映像ブロックのサンプルのうちの複数の近傍のサンプルがフィルタ係

10

20

30

40

50

数を共有し、前記複数の近傍のサンプルの各々に1回のクリッピング演算を適用する、解決策C1に記載の方法。

【0342】

C6．前記フィルタリング演算に関連付けられたフィルタ形状が対称モードである、解決策C5に記載の方法。

【0343】

C7．前記クリッピング演算の1つ以上のパラメータが前記ビットストリーム表現で信号通知される解決策C5またはC6に記載の方法。

【0344】

C8．前記クリッピング演算は、 $K(\min, \max, \text{input})$ として定義され、式中、 input は、前記クリッピング演算への入力であり、 \min は、前記クリッピング演算の出力の公称最小値であり、 \max は、前記クリッピング演算の出力の公称最大値である、解決策C1～C7のいずれかに記載の方法。

10

【0345】

C9．前記クリッピング演算の出力の実際の最大値が前記公称最大値よりも小さく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値よりも大きい、解決策C8に記載の方法。

【0346】

C10．前記クリッピング演算の出力の実際の最大値が前記公称最大値と等しく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値よりも大きい、解決策C8に記載の方法。

20

【0347】

C11．前記クリッピング演算の出力の実際の最大値が前記公称最大値よりも小さく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値と等しい、解決策C8に記載の方法。

【0348】

C12．前記クリッピング演算の出力の実際の最大値が前記公称最大値と等しく、かつ前記クリッピング演算の出力の実際の最小値が前記公称最小値と等しい、解決策C8に記載の方法。

【0349】

C13．時間的適応ループフィルタリング係数セットが利用不可能であることに基づいて、前記ビットストリーム表現が前記時間的適応ループフィルタリング係数セットの指示を省略するような前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換を行うことを含む、映像処理方法。

30

【0350】

C14．現在の映像ブロックを構成する符号化ツリーブロック(CTB)、ブロック、タイルグループ、タイル、スライス、またはピクチャにおいて、新しい適応ループフィルタリング(ALF)係数および固定ALF係数を使用しないことを判定することと、適応ループフィルタリングが無効化されていることを推測することをさらに含む、解決策C13に記載の方法。

40

【0351】

C15．適合ビットストリームは、新しい適応ループフィルタリング(ALF)係数の指示または固定ALF係数の指示を含む、解決策C13に記載の方法。

【0352】

C16．現在の映像ブロックに対して、利用可能な時間的ALF係数セットに基づいて、1つ以上の時間的適応ループフィルタリング(ALF)係数セットの数を判定することであって、前記利用可能な時間的ALF係数セットは、前記判定する前に符号化または復号化されている、判定することと、前記1つ以上の時間的ALF係数セットに基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現の間での変換を行うことと、を含む、映像処理方法。

50

【0353】

C17. 前記1つ以上の時間的ALF係数セットの最大数がALF_{available}である、解決策16に記載の方法。

【0354】

C18. 前記1つ以上の時間的ALF係数セットの数は、 $\min(N, ALF_{available})$ であり、ここで、Nは整数であり、 $N \geq 0$ である、解決策C17に記載の方法。

【0355】

C19. $N = 5$ である、解決策C18に記載の方法。

【0356】

C20. 現在の映像ブロックに対して、1つ以上の新しい適応ループフィルタリング(ALF)係数セットを処理することと、この処理の後に、前記1つ以上の新しいALF係数セットを利用可能なALF係数セットに指定することと、利用可能なALF係数セットに基づいて、前記現在の映像ブロックと前記現在の映像ブロックのビットストリーム表現との間での変換を行うステップとを含む、映像処理方法。

10

【0357】

C21. イントラランダムアクセスポイント(IRAP)アクセスユニット、IRAPピクチャ、瞬時復号化リフレッシュ(IDR)アクセスユニット、またはIDRピクチャに遭遇することと、前記利用可能なALF係数セットを、利用不可能なALF係数セットとして指定することと、をさらに含む解決策C20に記載の方法。

【0358】

C22. 前記利用可能なALF係数セットは、前記現在の映像ブロックの後続の映像ブロックのための時間的ALF係数セットである、解決策C20またはC21に記載の方法。

20

【0359】

C23. 前記利用可能なALF係数セットが、最大サイズがNであるALF係数セットリストに保持され、ここで、Nが整数である、解決策C20～C22のいずれかに記載の方法。

【0360】

C24. 前記ALF係数セットリストは、先入れ先出し(FIFO)の順序で保持される、解決策C23に記載の方法であって、方法。

【0361】

C25. 前記現在の映像ブロックに関連付けられた各時間層に1つのALF係数セットリストを保持する、解決策C13～C24のいずれかに記載の方法。

30

【0362】

C26. 前記現在の映像ブロックに関連付けられたK個の近傍の時間層ごとに1つのALF係数セットリストを保持する、解決策C13～C24のいずれかに記載の方法。

【0363】

C27. 前記現在の映像ブロックを含む現在のピクチャのために第1のALF係数セットリストを保持し、前記現在のピクチャの後続のピクチャのために第2のALF係数セットリストを保持する、解決策C13～C24のいずれかに記載の方法。

【0364】

C28. 前記現在のピクチャに基づいて、前記現在のピクチャに続く前記ピクチャ画像を予測し、前記第1のALF係数セットリストが前記第2のALF係数セットリストと同じである、解決策C27に記載の方法。

40

【0365】

C29. イントラランダムアクセスポイント(IRAP)アクセスユニット、IRAPピクチャ、瞬時復号化リフレッシュ(IDR)アクセスユニット、またはIDRピクチャに遭遇することと、前記遭遇した後に、1つ以上のALF係数セットリストを空にすること、をさらに含む解決策C20に記載の方法。

【0366】

C30. 現在の映像ブロックの異なる色成分について、異なるALF係数セットリスト

50

を保持する、解決策 C 2 0 に記載の方法。

【 0 3 6 7 】

C 3 1 . 処理装置と、その処理装置に命令が記憶された非一時的メモリとを備える装置であって、命令が処理装置によって実装されることにより、処理装置に、解決策 C 1 ~ C 3 0 のいずれかに記載の方法を実施させる映像システムの装置。

【 0 3 6 8 】

C 3 2 . 非一時的なコンピュータ可読媒体に記憶されたコンピュータプログラム製品であって、解決策 C 1 ~ C 3 0 のいずれかに記載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製品。

【 0 3 6 9 】

図 1 2 は、映像処理装置 1 2 0 0 のブロック図である。装置 1 2 0 0 は、本明細書に記載の方法の 1 つ以上を実装するために使用してもよい。装置 1 2 0 0 は、スマートフォン、タブレット、コンピュータ、I o T (モノのインターネット) 受信機等により実施されてもよい。装置 1 2 0 0 は、1 つ以上の処理装置 1 2 0 2 と、1 つ以上のメモリ 1 2 0 4 と、映像処理ハードウェア 1 2 0 6 と、を含んでもよい。1 つまたは複数の処理装置 1 2 0 2 は、本明細書に記載される 1 つ以上の方法 (方法 1 1 0 0 および 1 1 5 0 を含むが、これに限定されない) を実装するように構成されてもよい。メモリ (複数可) 1 2 0 4 は、本明細書で説明される方法および技術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してもよい。映像処理ハードウェア 1 2 0 6 は、本明細書に記載される技術をハードウェア回路にて実装するために使用してもよい。

【 0 3 7 0 】

いくつかの実施形態において、映像符号化法は、図 1 2 を参照して説明したように、ハードウェアプラットフォームに実装される装置を使用して実施してもよい。

【 0 3 7 1 】

開示される技術のいくつかの実施形態は、映像処理ツールまたはモードを有効化するように決定または判定することを含む。一例において、映像処理ツールまたはモードが有効化される場合、エンコーダは、1 つの映像ブロックを処理する際にこのツールまたはモードを使用するまたは実装するが、このツールまたはモードの使用に基づいて、結果として得られるビットストリームを必ずしも修正しなくてもよい。すなわち、映像のブロックから映像のビットストリーム表現への変換は、決定または判定に基づいて映像処理ツールまたはモードが有効化される場合に、この映像処理ツールまたはモードを使用する。別の例において、映像処理ツールまたはモードが有効化される場合、デコーダは、ビットストリームが映像処理ツールまたはモードに基づいて修正されたことを知って、ビットストリームを処理する。すなわち、決定または判定に基づいて有効化された映像処理ツールまたはモードを使用して、映像のビットストリーム表現から映像のブロックへの変換を行う。

【 0 3 7 2 】

開示される技術のいくつかの実施形態は、映像処理ツールまたはモードを無効化するように決定または判定することを含む。一例において、映像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、映像のブロックを映像のビットストリーム表現に変換する際に、このツールまたはモードを使用しない。別の例において、映像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、決定または判定に基づいて有効化された映像処理ツールまたはモードを使用してビットストリームが修正されていないことを知って、ビットストリームを処理する。

【 0 3 7 3 】

図 1 3 は、本明細書で開示される様々な技術が実装され得る例示的な映像処理システム 1 3 0 0 を示すブロック図である。様々な実装形態は、システム 1 3 0 0 のモジュールの一部又は全部を含んでもよい。システム 1 3 0 0 は、映像コンテンツを受信するための入力ユニット 1 3 0 2 を含んでもよい。映像コンテンツは、未加工又は非圧縮フォーマット、例えば、8 又は 1 0 ビットのマルチモジュール画素値で受信されてもよく、又は圧縮又は符号化フォーマットで受信されてもよい。入力ユニット 1 3 0 2 は、ネットワークイン

10

20

30

40

50

ターフェース、周辺バスインターフェース、又は記憶インターフェースを表してもよい。ネットワークインターフェースの例は、イーサネット（登録商標）、パッシブ光ネットワーク（PON）等の有線インターフェース、およびWi-Fi（登録商標）またはセルラーインターフェース等の無線インターフェースを含む。

【0374】

システム1300は、本明細書に記載される様々な符号化又は符号化方法を実装することができる符号化モジュール1304を含んでもよい。符号化モジュール1304は、入力ユニット1302からの映像の平均ビットレートを符号化モジュール1304の出力に低減し、映像の符号化表現を生成してもよい。従って、この符号化技術は、映像圧縮または映像コード変換技術と呼ばれることがある。符号化モジュール1304の出力は、モジュール1306によって表されるように、記憶されてもよいし、接続された通信を介して送信されてもよい。入力ユニット1302において受信された、記憶された又は通信された映像のビットストリーム（又は符号化）表現は、モジュール1308によって使用されて、表示インターフェースユニット1310に送信される画素値又は表示可能な映像を生成してもよい。ビットストリーム表現からユーザが見ることができる映像を生成する処理は、映像伸張（映像展開）と呼ばれることがある。さらに、特定の映像処理動作を“符号化”動作又はツールと呼ぶが、符号化ツール又は動作は、エンコーダ及びそれに対応する、復号化の結果を逆にする復号化ツール又は動作が、デコーダによって行われることが理解されよう。

【0375】

周辺バスインターフェースユニットまたは表示インターフェースユニットの例は、ユニバーサルシリアルバス（USB）または高精細マルチメディアインターフェース（HDMI（登録商標））またはディスプレイポート等を含んでもよい。ストレージインターフェースの例は、シリアルアドバンスドテクノロジーアタッチメント（SATA）、PCI、IDEインターフェース等を含む。本明細書に記載される技術は、携帯電話、ノートパソコン、スマートフォン、又はデジタルデータ処理及び/又は映像表示を実施可能な他のデバイス等の様々な電子デバイスに実施されてもよい。

【0376】

以上、説明の目的で本開示の技術の特定の実施形態を説明したが、本発明の範囲から逸脱することなく様々な修正が可能であることは、理解されるであろう。従って、本開示の技術は、添付の特許請求の範囲による場合を除き、限定されない。

【0377】

本特許明細書に記載された主題および機能操作の実装形態は、本明細書に開示された構造およびその構造的等価物を含め、様々なシステム、デジタル電子回路、またはコンピュータソフトウェア、ファームウェア、若しくはハードウェアで実施されてもよく、またはそれらの1つ以上の組み合わせで実施してもよい。本明細書に記載された主題の実装形態は、1つ以上のコンピュータプログラム製品、すなわち、データ処理装置によって実行されるため、又はデータ処理装置の操作を制御するために、有形で非可搬性のコンピュータ可読媒体上に符号化されたコンピュータプログラム命令の1つ以上のモジュールとして実装することができる。このコンピュータ可読媒体は、機械可読記憶装置、機械可読記憶基板、メモリデバイス、機械可読伝播信号をもたらす物質の組成物、又はこれらの1つ以上の組み合わせであってもよい。“データ処理ユニット”又は“データ処理装置”という用語は、例えば、プログラマブル処理装置、コンピュータ、又は複数の処理装置若しくはコンピュータを含め、データを処理するためのすべての装置、デバイス、および機械を含む。この装置は、ハードウェアの他に、当該コンピュータプログラムの実行環境を作るコード、例えば、処理装置ファームウェア、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、またはこれらの1つ以上の組み合わせを構成するコードを含むことができる。

【0378】

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、またはコードとも呼ばれる)は、コンパイルされた言語または解釈された言語を含む任意の形式のプログラミング言語で記述することができ、また、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使用するのに適したモジュール、成分、サブルーチン、または他のユニットとして含む任意の形式で展開することができる。コンピュータプログラムは、必ずしもファイルシステムにおけるファイルに対応するとは限らない。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(例えば、マークアップ言語文書に格納された1つ以上のスクリプト)に記録されていてもよいし、当該プログラム専用の単一のファイルに記憶されていてもよいし、複数の調整ファイル(例えば、1つ以上のモジュール、サブプログラム、またはコードの一部を格納するファイル)に記憶されていてもよい。1つのコンピュータプログラムを、1つのサイトに位置する1つのコンピュータ、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で実行させるように展開することも可能である。

10

【0379】

本明細書に記載された処理およびロジックフローは、入力データ上で動作し、出力を生成することによって機能を実行するための1つ以上のコンピュータプログラムを実行する1つ以上のプログラマブル処理装置によって行うことができる。処理およびロジックフローはまた、特定用途のロジック回路、例えば、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)によって行うことができ、装置はまた、特別目的のロジック回路として実装することができる。

20

【0380】

コンピュータプログラムの実行に適した処理装置は、例えば、汎用および専用マイクロ処理装置の両方、並びに任意の種類のデジタルコンピュータの任意の1つ以上の処理装置を含む。一般的に、処理装置は、読み出し専用メモリ又はランダムアクセスメモリ又はその両方から命令及びデータを受信する。コンピュータの本質的な要素は、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1つ以上の記憶装置とである。一般的に、コンピュータは、データを記憶するための1つ以上の大容量記憶デバイス、例えば、磁気、光磁気ディスク、または光ディスクを含んでもよく、またはこれらの大容量記憶デバイスからデータを受信するか、またはこれらにデータを転送するように動作可能に結合されてもよい。しかしながら、コンピュータは、このようなデバイス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適したコンピュータ可読媒体は、あらゆる形式の不揮発性メモリ、媒体、およびメモリデバイスを含み、例えば、EPROM、EEPROM、フラッシュメモリデバイス等の半導体メモリデバイスを含む。処理装置およびメモリは、特定用途のロジック回路によって補完されてもよく、または特定用途のロジック回路に組み込まれてもよい。

30

【0381】

本特許明細書は多くの詳細を含むが、これらは、任意の発明の範囲又は特許請求の範囲を限定するものと解釈されるべきではなく、むしろ、特定の発明の特定の実施形態に特有であり得る特徴の説明と解釈されるべきである。本特許文献において別個の実施形態の文脈で説明されている特定の特徵は、1つの例において組み合わせて実装してもよい。逆に、1つの例のコンテキストで説明された様々な特徴は、複数の実施形態において別個にまたは任意の適切なサブコンビネーションで実装してもよい。さらに、特徴は、特定の組み合わせで作用するものとして上記に記載され、最初にそのように主張されていてもよいが、主張された組み合わせからの1つ以上の特徴は、場合によっては、組み合わせから抜粋されることができ、主張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バリエーションに向けられてもよい。

40

【0382】

同様に、動作は図面において特定の順番で示されているが、これは、所望の結果を達成するために、このような動作が示された特定の順番でまたは連続した順番で行われること、または示された全ての動作が行われることを必要とするものと理解されるべきではない

50

。また、本特許明細書に記載されている例における様々なシステムの構成要素の分離は、全ての実施形態においてこのような分離を必要とするものと理解されるべきではない。

【 0 3 8 3 】

いくつかの実装形態および例のみが記載されており、この特許文献に記載され図示されているコンテンツに基づいて、他の実施形態、拡張および変形が可能である。

10

20

30

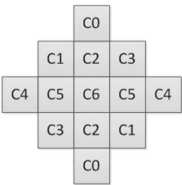
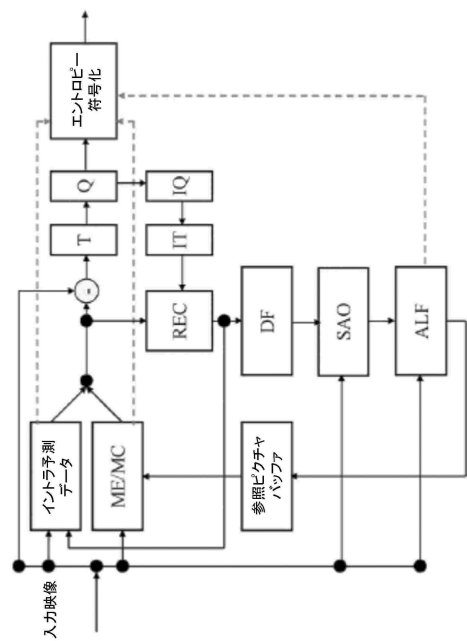
40

50

【図面】

【図 1】

【図 2 A】

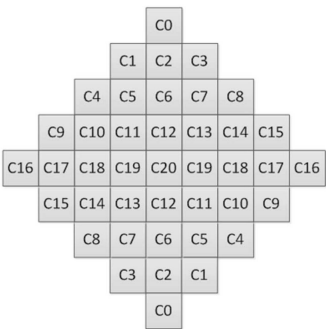
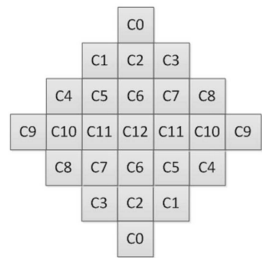


10

20

【図 2 B】

【図 2 C】

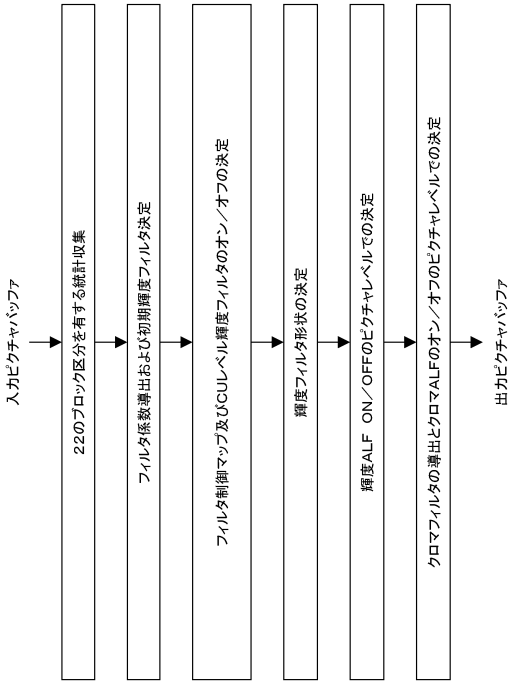


30

40

50

【図 3】



【図 4 A】

✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓

10

20

【図 4 B】

H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H

【図 4 C】

D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1

30

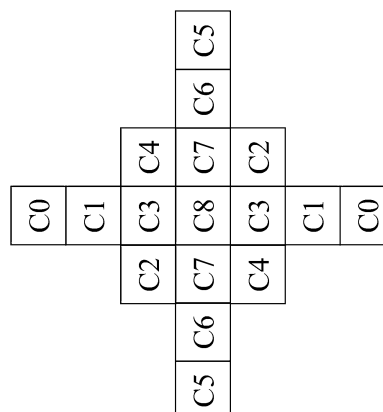
40

50

【 図 4 D 】

D_2		D_2		D_2		D_2	
	D_2		D_2		D_2		D_2
D_2		D_2		D_2		D_2	
	D_2		D_2		D_2		D_2
D_2		D_2		D_2		D_2	
	D_2		D_2		D_2		D_2
D_2		D_2		D_2		D_2	
	D_2		D_2		D_2		D_2

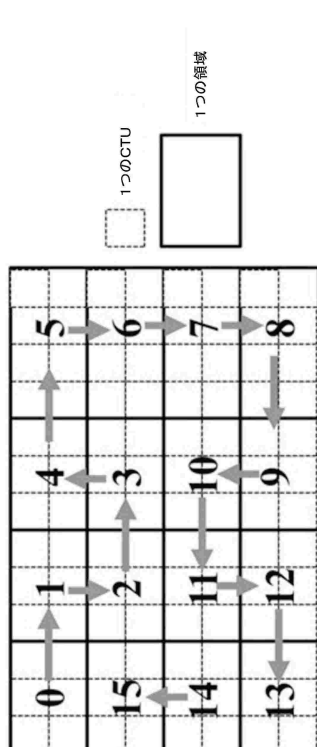
【 図 5 】



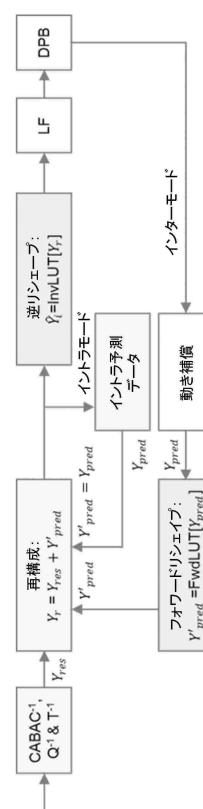
10

20

【圖 6】



【圖 7】

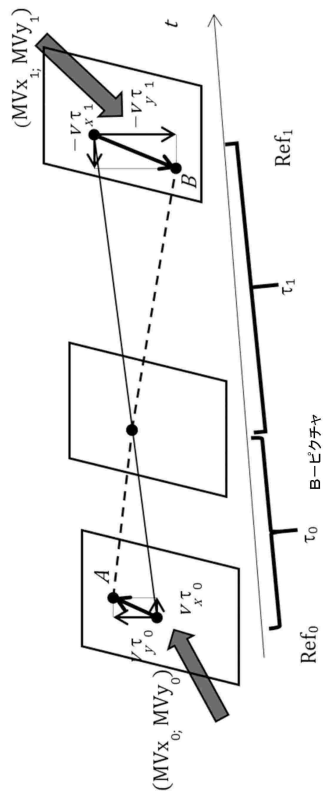


30

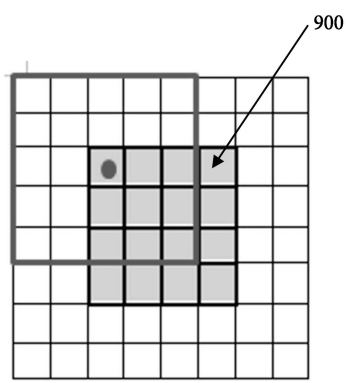
40

50

【図 8】



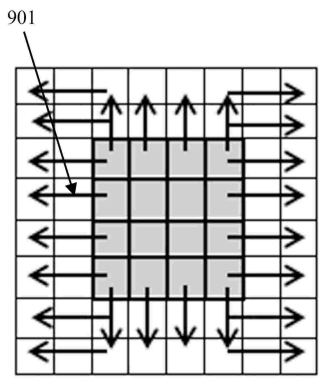
【図 9 A】



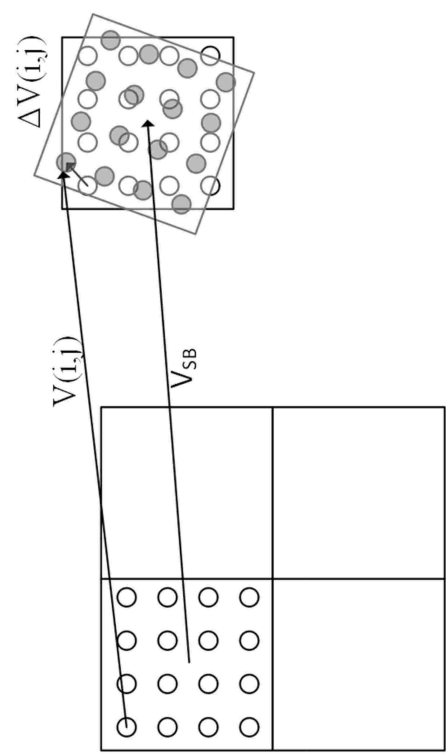
10

20

【図 9 B】



【図 10】

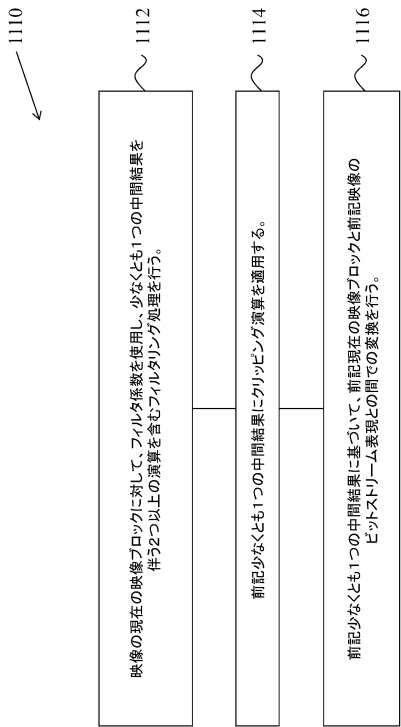


30

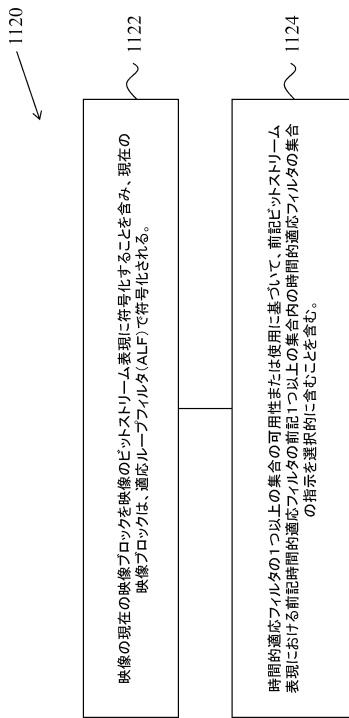
40

50

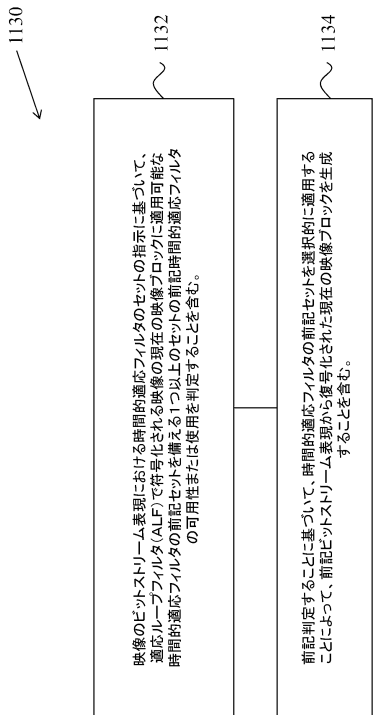
【図 1 1 A】



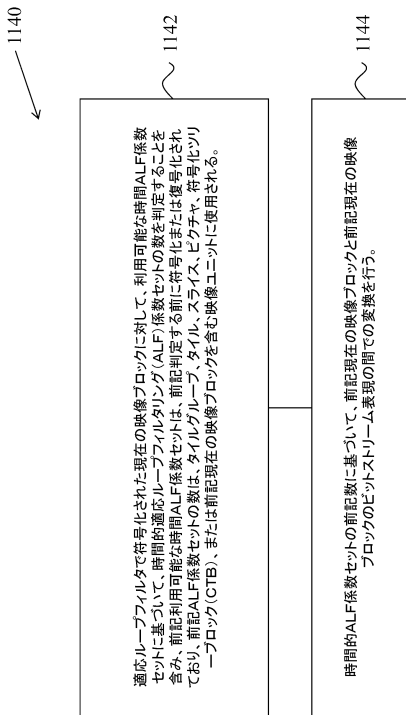
【図 1 1 B】



【図 1 1 C】



【図 1 1 D】



10

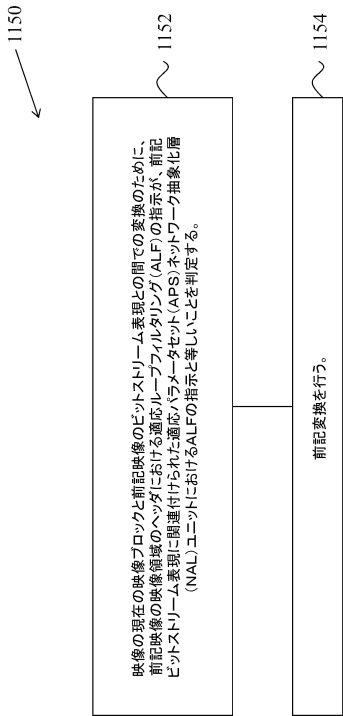
20

30

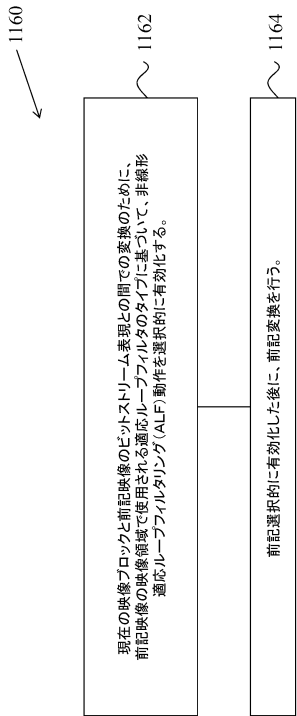
40

50

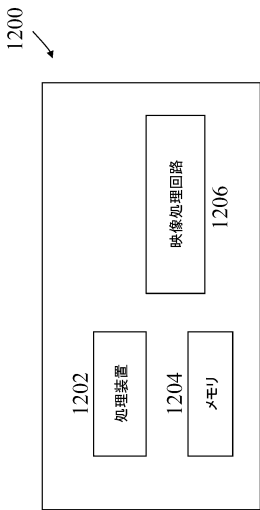
【図 1 1 E】



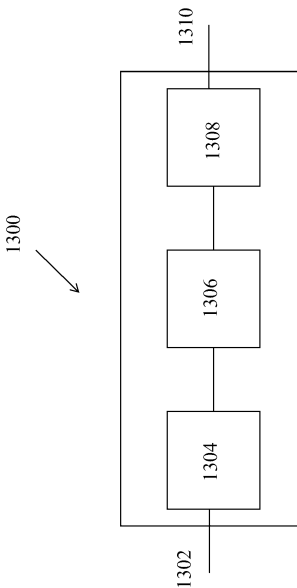
【図 1 1 F】



【図 1 2】



【図 1 3】



10

20

30

40

50

フロントページの続き

- 中国(CN)
BYTEDANCE INC.
アメリカ合衆国 カリフォルニア州 90066, ロサンゼルス, ウェスト ジェファーソン ブ
ールヴァード 12655, シックス フロア, スイート ナンバー・137
12655 West Jefferson Boulevard, Sixth Floor, S
uite No. 137 Los Angeles, California 90066 Unit
ed States of America
- (74)代理人 110002000
弁理士法人栄光事務所
- (72)発明者 リウ ホンビン
中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, チーチュン ロード,
チャイナ サテライト コミュニケーションズ タワー, ナンバー 63, ジンリトウティアオ ポ
ストオフィス
- (72)発明者 ジャン リー
アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブ
ールバード 12655, シックス フロア, スイート ナンバー 137
- (72)発明者 ジャン カイ
アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブ
ールバード 12655, シックス フロア, 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 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブ
ールバード 12655, シックス フロア, スイート ナンバー 137
- (72)発明者 ドン ジピン
中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, チーチュン ロード,
チャイナ サテライト コミュニケーションズ タワー, ナンバー 63, ジンリトウティアオ ポ
ストオフィス
- 合議体
審判長 畑中 高行
審判官 高橋 宣博
審判官 圓道 浩史
- (56)参考文献 Jianle Chen, et al., "Algorithm Description of Joint Exploration Test Model 4", Document: JV
ET-D1001_v3, [online], Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/
IEC JTC 1/SC 29/WG 11, 2016年11月19日, 発行日, JVET-D1001 (version 3), Pages 28-33,
[令和5年5月12日検索], インターネット, URL: https://jvet-experts.org/doc_end_user/current_document.php?id=2904 and URL: https://jvet-experts.org/doc_end_user/documents/4_Chengdu/wg11/JVET-D1001-v3.zip .
Sung-Chang Lim, et al., "CE2: Subsampled Laplacian calculation (Test 6.1, 6.2, 6.3, and 6.4)",
Document: JVET-L0147, [online], Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and
ISO/IEC JTC 1/SC 29/WG 11, 2018年09月24日, 発行日, JVET-L0147 (version 1), Pages 1-8,
[令和5年5月12日検索], インターネット, URL: https://jvet-experts.org/doc_end_user/current_document.php?id=4228 and URL: https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0147-v1.zip .
Jung Won Kang, et al., "Description of SDR video coding technology proposal by ETRI and S
ejong University", Document: JVET-J0013-v2, [online], Joint Video Experts Team (JVET) of I
TU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2018年04月11日, 発行日, JVET-J0013
(version 3), Pages 1 and 8-10, [令和5年5月12日検索], インターネット, URL: https://jvet-experts.org/doc_end_user/current_document.php?id=3446 and URL: https://jvet-experts.org/doc_end_user/documents/10_San%20Diego/wg11/JVET-J0013-v3.zip .
Jianle Chen, Yan Ye, Seung Hwan Kim, "Algorithm description for Versatile Video Coding an

d Test Model 5 (VTM 5)", Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting:Geneva, CH, 2019年05月21日, 掲載日, pp. 1, 53-58, JVET-N1002-v1 (version 1), [online], Internet https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N1002-v1.zip

Benjamin Bross, Jianle Chen, Shan Liu, "Versatile Video Coding (Draft 5)", Joint Video Experts team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting:Geneva, CH, 2019年04月12日, JVET-N1001-v2, [online], Internet https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N1001-v2.zip

(58)調査した分野 (Int.Cl. , D B 名)

H04N 19/00-19/98

CSDB (日本国特許庁)

IEEEXplore (IEEE)