

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7401666号
(P7401666)

(45)発行日 令和5年12月19日(2023.12.19)

(24)登録日 令和5年12月11日(2023.12.11)

(51)国際特許分類		F I	
H O 4 N	19/117(2014.01)	H O 4 N	19/117
H O 4 N	19/167(2014.01)	H O 4 N	19/167
H O 4 N	19/174(2014.01)	H O 4 N	19/174
H O 4 N	19/182(2014.01)	H O 4 N	19/182
H O 4 N	19/186(2014.01)	H O 4 N	19/186
請求項の数 11 (全77頁) 最終頁に続く			
(21)出願番号	特願2022-525117(P2022-525117)	(73)特許権者	520476341
(86)(22)出願日	令和2年11月4日(2020.11.4)		北京字節跳動網絡技術有限公司
(65)公表番号	特表2023-501192(P2023-501192 A)		Beijing Bytedance Network Technology Co., Ltd.
(43)公表日	令和5年1月18日(2023.1.18)		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/CN2020/126332		Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan District Beijing 100041 China
(87)国際公開番号	WO2021/088835	(73)特許権者	520477474
(87)国際公開日	令和3年5月14日(2021.5.14)		バイトダンス インコーポレイテッド
審査請求日	令和4年5月16日(2022.5.16)		最終頁に続く
(31)優先権主張番号	PCT/CN2019/115321		
(32)優先日	令和1年11月4日(2019.11.4)		
(33)優先権主張国・地域又は機関	中国(CN)		

(54)【発明の名称】 クロス成分適応ループフィルタ

(57)【特許請求の範囲】
【請求項1】

1つ以上のサンプルを含む映像の現在の映像ユニットと前記映像のビットストリームとの変換のために、クロス成分適応ループフィルタ（CC-A LF）ツールを前記現在の映像ユニットの少なくともいくつかのサンプルに適用する間に、利用不可のサンプルをパディングするために使用されるパディング処理を決定することであって、前記現在の映像ユニットは現在のコーディングツリーユニットに対応する、決定することと、

前記決定することに基づいて、前記変換を行うことと、を含む映像データを処理する方法であって、

前記CC-A LFツールは、前記現在の映像ユニットの第1の色成分に適用され、前記現在の映像ユニットの第2の色成分のサンプル値を使用して、前記現在の映像ユニットの前記第1の色成分のサンプル値を補正することを含み、

第1の利用不可のサンプルが前記現在の映像ユニットの右下側のコーディングツリーユニットに位置しており、かつ、前記現在の映像ユニットの下側のコーディングツリーユニットおよび前記現在の映像ユニットの右側のコーディングツリーユニットが共に利用可能である場合、前記第1の利用不可のサンプルを、前記下側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングし、

第2の利用不可のサンプルが、前記現在の映像ユニットの左上側のコーディングツリーユニットに位置し、かつ、前記現在の映像ユニットの上側のコーディングツリーユニットおよび前記現在の映像ユニットの左側のコーディングツリーユニットが共に利用可能である

場合、前記第2の利用不可のサンプルを、前記上側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングする、方法。

【請求項2】

前記右下側のコーディングツリーユニットは、前記現在の映像ユニットが位置する第1のスライスとは異なる第2のスライスに位置し、前記下側のコーディングツリーユニットおよび前記右側のコーディングツリーユニットは、共に前記第1のスライスに位置する、請求項1に記載の方法。

【請求項3】

前記左上側のコーディングツリーユニットは、前記第1のスライスおよび前記第2のスライスとは異なる第3のスライスに位置し、前記上側のコーディングツリーユニットおよび前記左側のコーディングツリーユニットは、共に前記第1のスライスに位置する、請求項2に記載の方法。

10

【請求項4】

前記パディング処理は、前記CC-ALFツールの前記使用に関連付けられた1つ以上のALF仮想境界において、前記利用不可のサンプルに適用されるミラーリングパディング(mirrored padding)に対応する、請求項1～3のいずれか1項に記載の方法。

【請求項5】

前記パディング処理は、スライス、タイル、ピクチャ、サブピクチャ、または前記CC-ALFの前記適用に関連付けられた360度仮想境界のうちの1つにおいて前記利用不可のサンプルに適用される片側パディングに対応する、請求項1～3のいずれか1項に記載の方法。

20

【請求項6】

前記パディング処理は、前記CC-ALFの前記適用に関連付けられた1つ以上のALF仮想境界を除き、1つ以上の境界に適用される片側パディングに対応する、請求項1～3のいずれか1項に記載の方法。

【請求項7】

前記変換は、前記映像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1～6のいずれか1項に記載の方法。

【請求項8】

前記変換は、前記映像を前記ビットストリームから復号することを含む、請求項1～6のいずれか1項に記載の方法。

30

【請求項9】

処理装置と、命令を備えた非一時的メモリとを含む映像データ処理装置であって、前記命令は、前記処理装置によって実行されると、前記処理装置に、

1つ以上のサンプルを含む映像の現在の映像ユニットと前記映像のビットストリームとの変換のために、クロス成分適応ループフィルタ(CC-ALF)ツールを前記現在の映像ユニットの少なくともいくつかのサンプルに適用する間に、利用不可のサンプルをパディングするために使用されるパディング処理を決定することであって、前記現在の映像ユニットは現在のコーディングツリーユニットに対応する、決定することと、

40

前記決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CC-ALFツールは、前記現在の映像ユニットの第1の色成分に適用され、前記現在の映像ユニットの第2の色成分のサンプル値を使用して、前記現在の映像ユニットの前記第1の色成分のサンプル値を補正することを含み、

第1の利用不可のサンプルが前記現在の映像ユニットの右下側のコーディングツリーユニットに位置しており、かつ、前記現在の映像ユニットの下側のコーディングツリーユニットおよび前記現在の映像ユニットの右側のコーディングツリーユニットが共に利用可能である場合、前記第1の利用不可のサンプルを、前記下側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングし、

第2の利用不可のサンプルが、前記現在の映像ユニットの左上側のコーディングツリーユ

50

ニットに位置し、かつ、前記現在の映像ユニットの上側のコーディングツリーユニットおよび前記現在の映像ユニットの左側のコーディングツリーユニットが共に利用可能である場合、前記第2の利用不可のサンプルを、前記上側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングする、映像データ処理装置。

【請求項10】

命令を記憶した非一時的なコンピュータ可読記憶媒体であって、前記命令は、処理装置に、

1つ以上のサンプルを含む映像の現在の映像ユニットと前記映像のビットストリームとの変換のために、クロス成分適応ループフィルタ（CC-A LF）ツールを前記現在の映像ユニットの少なくともいくつかのサンプルに適用する間に、利用不可のサンプルをパディングするために使用されるパディング処理を決定することであって、前記現在の映像ユニットは現在のコーディングツリーユニットに対応する、決定することと、

前記決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CC-A LFツールは、前記現在の映像ユニットの第1の色成分に適用され、前記現在の映像ユニットの第2の色成分のサンプル値を使用して、前記現在の映像ユニットの前記第1の色成分のサンプル値を補正することを含み、

第1の利用不可のサンプルが前記現在の映像ユニットの右下側のコーディングツリーユニットに位置しており、かつ、前記現在の映像ユニットの下側のコーディングツリーユニットおよび前記現在の映像ユニットの右側のコーディングツリーユニットが共に利用可能である場合、前記第1の利用不可のサンプルを、前記下側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングし、

第2の利用不可のサンプルが、前記現在の映像ユニットの左上側のコーディングツリーユニットに位置し、かつ、前記現在の映像ユニットの上側のコーディングツリーユニットおよび前記現在の映像ユニットの左側のコーディングツリーユニットが共に利用可能である場合、前記第2の利用不可のサンプルを、前記上側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングする、非一時的なコンピュータ可読記憶媒体。

【請求項11】

映像のビットストリームを記憶する方法であって、

1つ以上のサンプルを含む映像の現在の映像ユニットのために、クロス成分適応ループフィルタ（CC-A LF）ツールを前記現在の映像ユニットの少なくともいくつかのサンプルに適用する間に、利用不可のサンプルをパディングするために使用されるパディング処理を決定することであって、前記現在の映像ユニットは現在のコーディングツリーユニットに対応する、決定することと、

前記決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CC-A LFツールは、前記現在の映像ユニットの第1の色成分に適用され、前記現在の映像ユニットの第2の色成分のサンプル値を使用して、前記現在の映像ユニットの前記第1の色成分のサンプル値を補正することを含み、

第1の利用不可のサンプルが前記現在の映像ユニットの右下側のコーディングツリーユニットに位置しており、かつ、前記現在の映像ユニットの下側のコーディングツリーユニットおよび前記現在の映像ユニットの右側のコーディングツリーユニットが共に利用可能である場合、前記第1の利用不可のサンプルを、前記下側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングし、

第2の利用不可のサンプルが、前記現在の映像ユニットの左上側のコーディングツリーユニットに位置し、かつ、前記現在の映像ユニットの上側のコーディングツリーユニットおよび前記現在の映像ユニットの左側のコーディングツリーユニットが共に利用可能である場合、前記第2の利用不可のサンプルを、前記上側のコーディングツリーユニットに位置するその最も近い近傍のサンプルでパディングする、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11月4日出願の国際特許出願第PCT/CN2019/115321号の優先権と利益を主張する、2020年11月4日出願の国際特許出願第PCT/CN2020/126332号に基づく。前述のすべての特許出願は、その全体が参照により本明細書に組み込まれる。

【0002】

この特許明細書は、映像符号化および復号化の技術、デバイスおよびシステムに関する。

【背景技術】

【0003】

現在、現在の映像コーデック技術の性能を向上させ、より優れた圧縮率を実現するか、又はより低い複雑度又は並列化された実装を可能にする映像符号化及び復号化方式を提供する努力が行われている。産業界の専門家は、最近、いくつかの新しい映像符号化ツールを提案し、それらの有効性を判定するための試験が現在進行中である。

【発明の概要】

【0004】

デジタル映像コーディングに関し、具体的には、動きベクトルの管理に関するデバイス、システム、および方法が記載される。説明される方法は、既存の映像コーディング規格（例えば、高効率映像コーディング（High Efficiency Video Coding、HEVC）又は多様な映像コーディング）及び将来の映像コーディング規格又は映像コーデックに適用され得る。

【0005】

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、1つ以上の映像ブロックを含む映像の現在の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、現在の映像ユニットの少なくともいくつかの映像ブロックにクロス成分適応ループフィルタリング（CC-ALF）ツールを適用する間に、利用不可のサンプルをパディングするために用いられるパディング処理を決定することと、この決定に基づいて変換を行うことを含み、前記規則は現在の映像ユニットの1つ以上の映像ブロックに適応ループフィルタリング（ALF）ツールを適用する間に利用不可のサンプルをパディングするためにも用いられることを規定する。

【0006】

別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の映像ユニットとこの映像のビットストリーム表現との変換を行うことを含み、この変換中、映像ユニットの利用不可のサンプルは、適応ループフィルタリング（ALF）処理またはクロス成分適応ループフィルタリング（CC-ALF）処理の適用における規則に従って、予め定義されたパディング順序でパディングされる。

【0007】

さらに別の代表的な態様において、開示される技術は、映像処理の別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（ALF）の適用が有効化された映像の映像領域について、その映像領域が映像ユニットの境界で交差することを決定することと、映像と映像のビットストリーム表現との変換を行うことを含み、変換のために、映像領域が映像ユニットの境界と交差していることによる規則に従って、映像領域を複数のパーティションに分割する。

【0008】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、映像ユニットの境界に位置するサンプルに対して適応ループフィルタリング（ALF）および／またはクロス成分適応ループ

10

20

30

40

50

フィルタリング（CC-A L F）の適用が、境界をまたぐフィルタリング処理が許可されない場合、許可されないことを規定する。

【0009】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、ビットストリーム表現はフォーマット規則に準拠し、映像領域はコーディングツリーブロックとは異なり、フォーマット規則は適応ループフィルタリング（A L F）ツールおよび／またはクロス成分適応ループフィルタリング（CC-A L F）ツールの映像領域への適用可能性を示すビットストリーム表現に構文要素が含まれるかどうかを規定する。

10

【0010】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、映像ユニットのサンプルへのクロス成分適応ループフィルタリング（CC-A L F）ツールの適用可能性を決定することと、この決定に基づいて変換を行うこととを含み、ビットストリーム表現は、CC-A L Fが映像ユニットに利用可能である指示を含み、規則は指示をオーバーライドする1つ以上の条件を規定する。

【0011】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、この規則は、変換中に使用される演算が、クロマ適応ループフィルタリング（A L F）に対応する第1のクリッピング演算、クロス成分適応ループフィルタリング（CC-A L F）オフセット導出に対応する第2のクリッピング演算、および最終的なクロマサンプル値を導出するためのクロマフィルタリングされたサンプルの微調整に対応する第3のクリッピング演算を含む3つのクリッピング演算のうちの少なくとも1つを省略することを規定する。

20

【0012】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の第1の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、クロス成分適応ループフィルタリング（CC-A L F）オフセットを決定することと、この決定に基づいてこの変換を行うこととを含み、この規則は、CC-A L Fオフセットを、 $[-(1 << (BitDepthC - 1)), (1 << (BitDepthC - 1)) - 1]$ と表される第2の範囲と異なる第1の範囲にクリッピングすることを規定し、 $BitDepthC$ はビット深度値である。

30

【0013】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の第1の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、クロス成分適応ループフィルタリング（CC-A L F）オフセットを導出することと、CC-A L Fオフセットを使用して変換を行うこととを含み、規則は、固定値の代わりに、ビット深度値に基づいて丸めオフセットを用いて、CC-A L Fオフセットを丸めることを規定する。

40

【0014】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、映像ユニットのサンプルに適用されるクロマ適応ループフィルタリング（A L F）処理および／またはクロス成分適応ループフィルタリング（CC-A L F）処理中に使用される1つ以上の処理ステップが同じであることを規定する。

【0015】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供する

50

ために使用してもよい。この方法は、規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、変換中に、映像ユニットの第2の色成分の情報を使用して修正を適用することによって、映像ユニットの第1の色成分のサンプルの値を修正することを規定し、この修正は、映像ユニットのための適応ループフィルタリング（ALF）処理で使用される1つ以上のパラメータに基づく。

【0016】

さらに別の代表的な態様において、開示される技術は、映像処理の別の方法を提供するために使用してもよい。この方法は、映像の第1のサブピクチャと映像のビットストリーム表現との変換のために、規則に従って、クロス成分ループフィルタリング（CC-ALF）が第1のサブピクチャのサンプルに適用可能であるかどうかを決定することと、この決定に基づいた変換を行うことを含み、サンプルに対するCC-ALFは第2のサブピクチャからのサンプルを使用し、規則は第1のサブピクチャおよび／または第2のサブピクチャに対して、サブピクチャの境界をまたぐループフィルタリングが許可されているかどうかに基づいている。

10

【0017】

さらに、代表的な態様において、プロセッサと、命令を搭載した非一時的メモリとを備える、映像システムにおける装置が開示される。命令は、プロセッサによって実行されると、プロセッサに、開示された方法のいずれか1つ以上を実装させる。

【0018】

また、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、開示された方法のいずれか1つ以上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が開示される。

20

【0019】

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により詳細に説明される。

【図面の簡単な説明】

【0020】

【図1】エンコーダブロック図の例を示す。

【図2】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3】1つの輝度成分に関連付けられたループフィルタラインバッファの例を示す。

30

【図4】クロマ成分に関連付けられたループフィルタラインバッファの例を示す。

【図5A】 $N=4$ の場合の仮想境界における修正ALFブロック分類の例を示す。

【図5B】 $N=4$ の場合の仮想境界における修正ALFブロック分類の例を示す。

【図6A】修正輝度ALFフィルタリングに関連して、仮想境界（VB）付近の1つのライン、2つのライン、および3つのラインの例を示す。

【図6B】修正輝度ALFフィルタリングに関連して、仮想境界（VB）付近の1つのライン、2つのライン、および3つのラインの例を示す。

【図6C】修正輝度ALFフィルタリングに関連して、仮想境界（VB）付近の1つのライン、2つのライン、および3つのラインの例を示す。

【図7A】修正クロマALFフィルタリングに関連して、仮想境界（VB）に近い1つのラインおよび2つのラインの例を示す。

40

【図7B】修正クロマALFフィルタリングに関連して、仮想境界（VB）に近い1つのラインおよび2つのラインの例を示す。

【図8】修正係数に基づくALF（MALF）の例を示す。

【図9A】サブサンプリングされたラプラシアン計算の例を示す。

【図9B】サブサンプリングされたラプラシアン計算の例を示す。

【図9C】サブサンプリングされたラプラシアン計算の例を示す。

【図9D】サブサンプリングされたラプラシアン計算の例を示す。

【図10A】他のループフィルタに対するCC-ALFの配置の例を示す。

【図10B】菱形フィルタの例を示す。

50

【図 1 1】 3×4 菱形のフィルタの例を示す。

【図 1 2】ピクチャのラスタスキャンスライス分割の例を示す。

【図 1 3】ピクチャの矩形スライス分割の例を示す。

【図 1 4】ピクチャの矩形スライス分割の他の例を示す。

【図 1 5】ピクチャのサブピクチャ分割の例を示す。

【図 1 6】ALF 処理ユニットの例および狭 ALF 処理ユニットの例を示す。

【図 1 7】ALF 処理ユニットに繰り返しパディングを適用する例を示す。

【図 1 8】利用不可のサンプルを埋める例を示す。

【図 1 9】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。

10

【図 2 0】動画処理のフローチャートを示す。

【図 2 1】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。

【図 2 2】例示的な映像コーディングシステムを示すブロック図である。

【図 2 3】開示された技術のいくつかの実装形態によるエンコーダを示すブロック図である。

【図 2 4】開示された技術のいくつかの実装形態によるデコーダを示すブロック図である。

【図 2 5 A】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

20

【図 2 5 B】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

【図 2 5 C】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

【図 2 5 D】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

【図 2 5 E】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

【図 2 5 F】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

30

【図 2 5 G】開示された技術のいくつかの実装形態に基づく、映像処理のための例示的な方法のフローチャートを示す。

【発明を実施するための形態】

【0021】

1. HEVC/H. 265 における映像コーディング

映像コーディング規格は、主に周知のITU-TおよびISO/IEC規格の開発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、映像コーディング規格は、時間予測と変換コーディングが利用されるハイブリッド映像コーディング構造に基づく。HEVCを超えた将来の映像コーディング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VVC規格の策定に取り組んでいる。

40

【0022】

50

2. 1. 色空間及びクロマサブサンプリング

色空間は、カラーモデル（又はカラーシステム）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、典型的には3又は4つの値又は色成分（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合成したものである。

【0023】

映像圧縮の場合、最も頻繁に使用される色空間は、YCbCr及びRGBである。

【0024】

YCbCr、Y'CbCr、またはY Pb/Cb Pr/Crは、YCBCRまたはY'CBCRとも呼ばれ、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。

10

【0025】

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。

【0026】

2. 1. 1. 4 : 4 : 4

20

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。この方式は、ハイエンドフィルムスキャナ及び映画のポストプロダクションに用いられることがある。

【0027】

2. 1. 2. 4 : 2 : 2

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、水平クロマ解像度は半分にされる。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。

【0028】

2. 1. 3. 4 : 2 : 0

30

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及びCrチャンネルを各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及びCrはそれぞれ水平及び垂直方向の両方に2倍ずつサブサンプリングされる。異なる水平及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。

● MPEG-2において、Cb及び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）。

● JPEG/JFIF、H. 261、及びMPEG-1において、Cb及びCrは、交互の輝度サンプルの中間に間欠的に位置する。

● 4 : 2 : 0 DVにおいて、Cb及びCrは、水平方向に共座している。垂直方向には、それらは交互に共座している。

40

【0029】

2. 1. 4. 異なる色成分のコーディング

`separate_colour_plane_flag`の値に基づいて、変数`ChromaArrayType`の値は、以下のように割り当てられる。

— `separate_colour_plane_flag`が0に等しい場合、`ChromaArrayType`は`chroma_format_idc`に等しく設定される。

— そうでない場合（`separate_colour_plane_flag`が1に等しい）、`ChromaArrayType`は0に設定される。

【0030】

50

2.2 典型的な映像コーデックのコーディングフロー

図1は、3つのインループフィルタリングブロック、すなわちデブロッキングフィルタ（DF）、サンプル適応オフセット（SAO）およびALFを含むVVCのエンコーダブロック図の例を示す。DF（予め定義されたフィルタを使用する）とは異なり、SAOおよびALFは、現在のピクチャのオリジナルサンプルを利用し、それぞれ、オフセットを追加することにより、および、有限インパルス応答（FIR）フィルタを適用することにより、オフセットおよびフィルタ係数を信号通知するコード化側情報を用いて、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。

10

【0031】

2.3 JEMにおける形状変換に基づく適応ループフィルタ

JEMにおいて、ブロックに基づくフィルタ適応を用いたジオメトリ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】

2.3.1. フィルタ形状

JEMにおいて、輝度成分として、最大3つの菱形フィルタ形状（図2に示す）を選択することができる。図2は、3つのGALFフィルタ形状、すなわち 5×5 菱形、 7×7 菱形、 9×9 菱形（左側から右側へ）を示す。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。

20

【0033】

2.3.1.1. 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 $\hat{A}$ の量子化値に基づいて、以下のように導出される。
 $C = 5D + \hat{A} \quad (1)$

【0034】

Dおよび $\hat{A}$ 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。

【0035】

30

【数1】

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

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

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

40

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

【0036】

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。

【0037】

50

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 3 8 】

【数 2】

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

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 3 9 】

【数 3】

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

【 0 0 4 0 】

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

【 0 0 4 1 】

【数 4】

ステップ 1. $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ と $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ の両方が TRUE である場合、D が 0 に設定される。

ステップ 2. $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$ 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. $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ 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. $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ 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。

【 0 0 4 2 】

アクティビティ値 A は、以下のように計算される。

【 0 0 4 3 】

【数 5】

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

【 0 0 4 4 】

A をさらに 0 ～ 4 の範囲に量子化し、量子化された値を $\hat{A}$ とする。

【 0 0 4 5 】

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 A L F 係数のセットが各クロマ成分に対して適用される。

【 0 0 4 6 】

2. 3. 1. 2. フィルタ係数の幾何学的変換

それぞれの 2×2 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 $f(k, l)$ に対して、回転、又は対角線及び垂直方

10

20

30

40

50

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サポート領域内のサンプルに適用することに等しい。その考えは、A L F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。

【0047】

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。

対角線： $f_D(k, l) = f(l, k)$,

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

回転： $f_R(k, l) = f(K - l - 1, k)$

この場合、Kはフィルタのサイズであり、 $0 \leq k, l \leq K - 1$ 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。

【0048】

【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

【0049】

2. 3. 1. 3. フィルタパラメータ信号通知

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係数の信号通知を迂回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通知され、記憶されている指定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。

【0050】

GALF時間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維持される。新しいシーケンスの復号開始時に、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つの時間レイヤに関連付けられる。具体的に、時間レイヤインデックス（TempIdx）により割り当てられた各配列は、下位のTempIdxに等しい、前回復号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。特定のピクチャをコーディングした後、このピクチャに関連付けられたフィルタセットを用いて、同等又は上位のTempIdxに関連付けられた配列を更新する。

【0051】

GALF係数の時間予測は、信号通知のオーバーヘッドを最小限に抑えるために、イン

ター符号化フレームに使用される。イントラフレームの場合、時間予測は利用できず、16個の固定フィルタのセットが各クラスに割り当てられる。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。

【0052】

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C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、画像レベルでのみ示される。

10

【0053】

2.3.1.4. フィルタリング処理

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、このブロック内のそれぞれのサンプル $R(i, j)$ 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。

【0054】

【数6】

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

20

【0055】

あるいは、適応ループフィルタのフィルタリング処理は、以下のように表すことができる。

【0056】

【数7】

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

【0057】

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、VTM4.0では、固定小数点精度計算のために整数演算を用いて実施されている。

30

【0058】

【数8】

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

40

【0059】

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。

【0060】

2.4. 非線形ALF

2.4.1. フィルタリング再定式化

式(11)は、コーディング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。

【0061】

【数9】

50

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

【0062】

ここで、 $w(i, j)$ は、式(11)と同じフィルタ係数である[但し、 $w(0, 0)$ は式(13)の1に等しいが、式(11)の $1 - \sum_{(i, j) \neq (0, 0)} w(i, j)$ に等しい]。

【0063】

2. 4. 2. 修正されたフィルタ

上記(13)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、ALFをより効率的にする。

【0064】

この提案において、ALFフィルタは、以下のように修正される。

【0065】

【数10】

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

【0066】

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。

【0067】

実現を容易にするために、フィルタ係数 $w(i, j)$ を記憶し、整数精度で用いる。上記式は、以下のように書き換えることができる。

【0068】

【数11】

$$O'(i, j) = I(i, j) + ((\sum_{k \neq 0} \sum_{l \neq 0} w(k, l) \times K(I(i + k, j + l) - I(i, j), c(k, l)) + 64) \gg 7) \quad (15)$$

【0069】

この場合、 $w(k, l)$ は復号化されたフィルタ係数を表し、 $K(x, y)$ はクリッピング関数であり、 $c(k, l)$ は復号化されたクリッピングパラメータを表す。変数 k および l は、 $(-L/2)$ と $(L/2)$ との間で変化し、ここで、 L はフィルタ長を表す。関数 $\text{Clip3}(-y, y, x)$ に対応するクリッピング関数 $K(x, y) = \min(y, \max(-y, x))$

【0070】

J V E T - N 0 2 4 2 の実装形態において、1つのALFフィルタごとにクリッピングパラメータ $k(i, j)$ を規定し、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。

【0071】

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、可能な値の小さなセットに限定する。本提案では、インターおよびイントラ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。

【0072】

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化することができる。

【0073】

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、対数領域において、輝度についてはサンプル値（10ビットで符号化される）の全範囲、クロマについて4～1024の範囲をほぼ均等に分割して選択されている。

【0074】

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。

【0075】

【数12】

$$\text{AlfClip}_L = \left\{ \text{round} \left(\left(\left(M \right)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{10} \text{ and } N=4. \quad (16)$$

【0076】

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。

【0077】

【数13】

$$\text{AlfClip}_C = \left\{ \text{round} \left(A \cdot \left(\left(\frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{10}, N=4 \text{ and } A=4. \quad (17)$$

【0078】

【表2】

表2：許可クリッピング値

	イントラ／インタースライス
輝度	{ 1024, 181, 32, 6 }
クロマ	{ 1024, 161, 25, 4 }

【0079】

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用いて、「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。

【0080】

2.5. 仮想境界

ハードウェアおよび組み込みソフトウェアにおいて、ピクチャに基づく処理は、その高いピクチャバッファ要件のために、実際には受け入れられない。オンチップピクチャバッファの使用は非常に高価であり、オフチップピクチャバッファの使用は、外部メモリアクセス、電力消費、およびデータアクセス待ち時間を大幅に増加させる。そのため、実際の製品において、D F、S A O、A L Fを画像に基づく復号からL C Uに基づく復号に変更することになる。D F、S A O、A L FにL C Uに基づく処理を使用する場合、複数のL C Uを並列処理するために、L C Uパイプライン方式を用いたラスタスキャンでL C Uごとに全体の復号処理を行うことができる。この場合、1つのL C U行を処理するには上側L C U行からのピクセルが必要であるので、D F、S A O、およびA L Fのためにライン

10

20

30

40

50

バッファが必要である。オフチップラインバッファ（例えば、DRAM）を使用する場合、外部メモリの帯域幅および消費電力が増大し、オンチップラインバッファ（例えば、SRAM）を使用する場合、チップ面積が増大する。従って、ラインバッファは既にピクチャバッファよりも遥かに小さいが、ラインバッファを低減することが依然として望ましい。

【0081】

VTM-4.0において、図3に示すように、輝度成分に必要なラインバッファの総数は11.25ラインである。ラインバッファ要件の説明は以下のとおりである。CTUEッジと重複する水平エッジのデブロッキングは、決定およびフィルタリングには、第1のCTUからのラインK, L, M, M、および最下CTUからのラインO, Pを必要とするので、行うことができない。そのため、CTU境界と重なる水平エッジのデブロッキングは、下側CTUが来るまで延期される。従って、ラインK, L, M, Nに対して、再構成された輝度サンプルをラインバッファ（4ライン）に記憶しなければならない。次に、ラインA~Jに対してSAOフィルタリングを行うことができる。ラインJはSAOフィルタリングすることができ、これは、デブロッキングによってラインKにおけるサンプルが変化しないためである。ラインKのSAOフィルタリングの場合、エッジオフセット分類決定はラインバッファにのみ記憶される（これは0.25輝度ラインである）。ALFフィルタリングは、ラインA~Fに対してのみ行い得る。図3に示すように、4×4ブロックごとにALF分類を行う。各4×4ブロック分類は、8×8のサイズのアクティビティ窓を必要とし、このアクティビティ窓は、1Dラプラシアンを計算して勾配を決定するために9×9の窓を必要とする。

【0082】

従って、ラインG, H, I, Jと重なる4×4ブロックのブロック分類のために、Jは、仮想境界より下のSAOフィルタリングされたサンプルを必要とする。さらに、ALF分類のために、ラインD, E, FのSAOフィルタリングされたサンプルが必要である。さらに、ラインGのALFフィルタリングは、上側ラインから3つのSAOフィルタリングされたラインD, E, Fを必要とする。従って、総ラインバッファ要件は、以下のとおりである。

- ー ラインK-N（水平DFピクセル）：4ライン
- ー ラインD-J（SAOフィルタリングされたピクセル）：7ライン
- ー ラインJとラインKとの間のSAOエッジオフセット分類子値：0.25ライン

【0083】

従って、必要とされる輝度ラインの総数は、 $7 + 4 + 0.25 = 11.25$ である。

【0084】

同様に、クロマ成分のラインバッファ要求は図4に例示されている。クロマ成分のためのラインバッファ要件は、6.25ラインであると評価される。

【0085】

SAOおよびALFのラインバッファ要件を排除するために、最近のVVCにおいて、仮想境界（VB）の概念が導入されている。図3に示すように、VBは、水平LCU境界においてN個のピクセルだけ上向きにシフトされている。LCUごとに、SAOおよびALFは、下側LCUが来る前に、VBより上のピクセルを処理することができるが、下側LCUが来るまで、VBより下のピクセルを処理することはできず、これは、DFによってもたらされる。ハードウェアの実施費用を勘案し、提案したVBと水平LCU境界との間の空間を、輝度が4ピクセル（即ち、図3における $N=4$ ）、クロマが2ピクセル（即ち、図9における $N=2$ ）として設定する。

【0086】

2.5.1. VBサイズNが4である場合の修正ALFブロック分類

図5Aおよび図5Bは、仮想境界がCTU境界の4ライン上にある場合（ $N=4$ ）について、修正されたブロック分類を示す。図5Aは、ラインGで始まる4×4ブロックの分類を示し、図5Bは、ラインKで始まる4×4ブロックの分類を示す。図5Aに示すように、ラインGで始まる4×4ブロックの場合、ブロック分類は、ラインE~Jのみを使用

する。しかし、ライン J に属するサンプルのラプラシアン勾配計算は、下にもう 1 つのライン（ライン K）を必要とする。従って、ライン K はライン J とパディングされる。

【0087】

同様に、図 5 B に示すように、ライン K で始まる 4×4 ブロックの場合、ブロック分類はライン K ～ P のみを使用する。しかし、ライン K に属するサンプルのラプラシアン勾配計算には、上にもう 1 つのライン（ライン J）を必要とする。従って、ライン J はライン K とパディングされる。

【0088】

2. 5. 2. 仮想境界をまたぐサンプルのための二面パディング

図 6 A ～ 6 C に示すように、フィルタを切り捨てたバージョンは、仮想境界に近いラインに属する輝度サンプルをフィルタリングするために用いられる。図 6 A は、仮想境界（V B）の上／下にある（辺ごとに）パディングされるべき 1 つの必要なラインを示し、図 6 B は、V B の上／下にある（辺ごとに）パディングされるべき 2 つの必要なラインを示し、図 6 C は、V B の上／下にある（辺ごとに）パディングされるべき 3 つの必要なラインを示す。図 6 A から 6 C において、V B は灰色のラインで表される。図 6 A について例えば、図 3 で示されたライン M をフィルタリングする場合、即ち、 7×7 菱形サポートの中心サンプルがライン M にある場合、V B より上の 1 つのラインに接近することが必要である。この場合、V B より上のサンプルは、V B より下の右下サンプルからコピーされ、例えば、実線の P 0 サンプルは、上の破線位置にコピーされる。対称的に、実線の P 3 サンプルもまた、その位置に対するサンプルが利用可能でも、右下の破線の位置にコピーされる。コピーされたサンプルは、輝度フィルタリング処理にのみ用いられる。

【0089】

A L F 仮想境界のために使用されるパディング法は、「二面パディング」と呼ばれてもよく、 (i, j) に位置する 1 つのサンプル（例えば、図 6 B に点線を有する P 0 A）をパディングする場合、図 6 A ～ 6 C および図 7 A ～ 7 C に示すように、同じフィルタ係数を共有する (m, n) に位置する対応するサンプル（例えば、図 6 C に点線を有する P 3 B）もまた、サンプルが利用可能でもパディングを行う。

【0090】

同様に、図 7 A ～ 7 C に示すように、クロマ A L F フィルタリングにも両面パディング法が用いられる。図 7 A は、仮想境界（V B）の上／下にある 1 つの必要なラインをパディングする必要があることを示し（一辺）、図 7 B は、V B の上／下にある 2 つの必要なラインをパディングする必要があることを示す（一辺）。図 7 A および 7 B において、V B は灰色のラインで表される。

【0091】

2. 5. 3. 非線形 A L F が無効にされている場合の二面パディングの代替実装方法

C T B に対して非線形 A L F が無効である場合、例えば、式 (14) におけるクリッピングパラメータ $k(i, j)$ が $(1 < \text{Bit depth})$ に等しい場合、パディング処理は、フィルタ係数を修正（別称、*modified-coefficient-based A L F*, M A L F）によって置き換えることができる。例えば、ライン L / I におけるサンプルをフィルタリングするとき、フィルタ係数 c_5 が c_5' に修正され、このケースでは、図 8 で実線 P 0 A から破線 P 0 A へ、実線 P 3 B から破線 P 3 B への輝度サンプルをコピーする必要がない。この場合、二面パディングおよび M A L F は、フィルタリングされる現在のサンプルが (x, y) に位置すると仮定して、同じ結果を生成する。

【0092】

【数 14】

$$c_5 \cdot K(I(x-1, y-1) - I(x, y), k(-1, -1)) + c_1 \cdot K(I(x-1, y-2) - I(x, y), k(-1, -2)) \\ = (c_5 + c_1) \cdot K(I(x-1, y-1) - I(x, y), k(-1, -1))$$

(18)

10

20

30

40

【0093】

なぜなら、パディングによって、 $K(d, b) = d$ であり、 $I(x-1, y-1) = I(x-1, y-2)$ であるからである。

【0094】

しかしながら、非線形ALFが有効化される場合、MALFおよび二面パディングは、異なるフィルタリング結果を生成することができる。その理由は、非線形パラメータが、例えば、フィルタ係数 c_5 および c_1 の場合などの各係数に関連付けられ、クリッピングパラメータが異なるためである。そのため、

【数15】

$$c_5 \cdot K(I(x-1, y-1) - I(x, y), k(-1, -1)) + c_1 \cdot K(I(x-1, y-2) - I(x, y), k(-1, -2)) \\ \neq (c_5 + c_1) \cdot K(I(x-1, y-1) - I(x, y), k(-1, -1))$$

(19)

$K(d, b) \neq d$ なので、パディングによって、 $I(x-1, y-1) = I(x-1, y-2)$ であってもよい。

【0095】

2.6. VVCにおける形状変換に基づく適応ループフィルタ

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) 適応フィルタ形状は除去される。輝度成分に対しては 7×7 のフィルタ形状のみが許容され、色度成分に対しては 5×5 のフィルタ形状のみが許容される。

2) ALFフィルタ係数は、ALF適応パラメータ集合(ALF Adaptation Parameter Set、APS)において信号通知される。

3) 非線形ALFを適用してもよい。

4) CTUごとに、色成分ごとに1つのビットフラグが、ALFが有効であるか無効であるかを信号通知される。

5) クラスインデックスの計算は、 2×2 ではなく、 4×4 のレベルで行われる。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。

【0096】

2.7. 適応パラメータ集合におけるALFパラメータの信号通知

VVC草案の最新バージョンにおいて、ALFパラメータは、適応パラメータ集合(Adaptation Parameter Set、APS)において信号通知されてもよく、各CTUによって適応的に選択されてもよい。1つのAPSにおいて、最大25組の輝度フィルタ係数およびクリッピング値インデックス、並びに最大8組のクロマフィルタ係数およびクリッピング値インデックス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、輝度成分の異なる分類のフィルタ係数をマージすることができる。スライスヘッダにおいて、現在のスライスに使用されるAPSのインデックスが信号通知される。

【0097】

フィルタ係数は、128に等しいノルムで量子化される。乗算の複雑性を抑えるために、非中心位置の係数値が $-2^7 \sim 2^7 - 1$ の範囲内に含まれるように、ビットストリーム適合性が適用される。中心位置係数はビットストリームにおいて信号通知されず、128に等しいと見なされる。

【0098】

ALFの詳細なシグナル伝達(JVET-P2001-v9)は以下のとおりである。

【0099】

10

20

30

40

50

【表 3】

7. 3. 2. 5 適応パラメータセット構文

	記述子
adaptation_parameter_set_rbsp() {	
adaptation_parameter_set_id	u(5)
aps_params_type	u(3)
if(aps_params_type == ALF_APS)	
alf_data()	
else if(aps_params_type == LMCS_APS)	
lmcs_data()	
else if(aps_params_type == SCALING_APS)	
scaling_list_data()	
aps_extension_flag	u(1)
if(aps_extension_flag)	
while(more_rbsp_data())	
aps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

10

【 0 1 0 0 】

20

30

40

50

【表 4】

7. 3. 2. 16 適応ループフィルタデータ構文

alf_data() {	記述子
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	ue(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	u(v)
}	
alf_luma_coeff_signalled_flag	u(1)
if(alf_luma_coeff_signalled_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++)	
alf_luma_coeff_flag[sIdx]	u(1)
}	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for (j = 0; j < 12; j++) {	
alf_luma_coeff_abs[sIdx][j]	uek(v)
if(alf_luma_coeff_abs[sIdx][j])	
alf_luma_coeff_sign[sIdx][j]	u(1)
}	
}	
}	
if(alf_luma_clip_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signalled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for (j = 0; j < 12; j++)	
alf_luma_clip_idx[sIdx][j]	u(2)
}	
}	
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_num_alt_filters_minus1	ue(v)
for(altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++) {	
alf_chroma_clip_flag[altIdx]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[altIdx][j]	uek(v)
if(alf_chroma_coeff_abs[altIdx][j] > 0)	
alf_chroma_coeff_sign[altIdx][j]	u(1)
}	
if(alf_chroma_clip_flag[altIdx]) {	
for(j = 0; j < 6; j++)	
alf_chroma_clip_idx[altIdx][j]	u(2)
}	
}	
}	
}	

【0101】

7. 4. 3. 5 適応パラメータセット意味論

各APS RBS Pは、それが参照される前に復号化処理で利用できるか、それを参照する符号化スライスNALユニットのTemporal Id以下のTemporal Idを持つ少なくとも一つのアクセスユニット内に含まれるか、外部手段を通じて提供されるものとする。

as p L a y e r I dをAPS NALユニットのn u h _ l a y e r _ i dとする。as p L a y e r I dに等しいn u h _ l a y e r _ i dを持つレイヤが独立レイヤである（即ち、v p s _ i n d e p e n d e n t _ l a y e r _ f l a g [G e n e r a l L

ayerIdx[aspLayerId]))が1に等しい) 場合、APS RBSPを含むAPS NALユニットは、それを参照するコーディングスライスNALユニットのnuh_layer_idに等しいnuh_layer_idを持つものとする。それ以外の場合、APS RBSPを含むAPS NALユニットは、それを参照する符号化スライスNALユニットのnuh_layer_idに等しいか、またはそれを参照するコーディングスライスNALユニットを含むレイヤの直接依存レイヤのnuh_layer_idに等しいnuh_layer_idを有しているものとする。

アクセスユニット内において、adaptation_parameter_set_idの特定の値と、aps_params_typeの特定の値を持つすべてのAPS NALユニットは、同じコンテンツを有している。

adaptation_parameter_set_idは、他の構文要素が参照するAPSの識別子を提供する。

aps_params_typeがALF_APSまたはSCALING_APSに等しい場合、adaptation_parameter_set_idの値は0～7の範囲に含まれるものとする。

aps_params_typeがLMCS_APSと等しい場合、adaptation_parameter_set_idの値は0～3の範囲に含まれるものとする。

aps_params_typeは、表3に示されるように、APSにおいて実行されるAPSパラメータのタイプを指定する。aps_params_typeが1(LMCS_APS)である場合、adaptation_parameter_set_idの値は、0以上3以下の範囲内にあるものとする。

【0102】

【表5】

表3－APSパラメータのタイプコード及びAPSパラメータのタイプ

aps_params_type	aps_params_typeの名称	APS パラメータの種類
0	ALF_APS	ALF パラメータ
1	LMCS_APS	LMCS パラメータ
2	SCALING_APS	スケーリングリストパラメータ
3..7	予約済み	予約済み

【0103】

注1－各タイプのAPSは、adaptation_parameter_set_idに別個の値空間を使用する。

注2－APS NALユニット(adaptation_parameter_set_idの特定の値とaps_params_typeの特定の値を有する)は、ピクチャ間で共有することができ、ピクチャ内の異なるスライスは、異なるALF_APSを参照することができる。

aps_extension_flag=0の場合、APS RBSP構文構造にaps_extension_data_flag構文要素が存在しないことを示す。aps_extension_flag=1の場合、APS RBSP構文構造にaps_extension_data_flag構文要素が存在することを示す。

aps_extension_data_flagは任意の値を有することができる。その存在および値は、本明細書バージョンで特定された特徴に対するデコーダの適合性に影響を与えない。本明細書バージョンに準拠するデコーダは、すべてのaps_extension_data_flag構文要素を無視しなければならない。

【0104】

7.4.3.14 適応ループフィルタデータ意味論

alf_luma_filter_signal_flag=1は、輝度フィルタセットが信号通知されることを示す。alf_luma_filter_signal_flg

$ag = 0$ は、輝度フィルタセットは信号通知されないことを示す。

$alf_chroma_filter_signal_flag$ が 1 に等しい場合、クロマフィルタが信号通知されることを示す。 $alf_chroma_filter_signal_flag$ が 0 に等しい場合、クロマフィルタが信号通知されないことを示す。 $ChromaArrayType$ が 0 に等しい場合、 $alf_chroma_filter_signal_flag$ は 0 に等しい。

異なる適応ループフィルタの数を規定する変数 $NumAlfFilters$ は、25 に設定される。

$alf_luma_clip_flag = 0$ は、線形適応ループフィルタリングが輝度成分に適用されることを規定する。 $alf_luma_clip_flag = 1$ は、非線形適応ループフィルタリングが輝度成分に適用され得ることを規定する。

$alf_luma_num_filters_signalled_minus1$ プラス 1 は、輝度係数を信号通知することができる適応ループフィルタクラスの数进行規定する。 $alf_luma_num_filters_signalled_minus1$ の値は、 $0 \sim NumAlfFilters - 1$ の範囲内にあるものとする。

$alf_luma_coeff_delta_idx[filtIdx]$ は、 $filtIdx$ が示すフィルタクラスのための、0 から $NumAlfFilters - 1$ までの範囲にわたる、信号通知される適応ループフィルタ輝度係数デルタのインデックスを規定する。 $alf_luma_coeff_delta_idx[filtIdx]$ が存在しない場合、それは 0 であると推論される。 $alf_luma_coeff_delta_idx[filtIdx]$ の長さは、 $Ceil(\log_2(alf_luma_num_filters_signalled_minus1 + 1))$ ビットである。

1 に等しい $alf_luma_coeff_signalled_flag_equal$ は、 $alf_luma_coeff_flag[sfIdx]$ が信号通知されることを示す。0 に等しい $alf_luma_coeff_signalled_flag$ は、 $alf_luma_coeff_flag[sfIdx]$ が信号通知されないことを示す。

$alf_luma_coeff_flag[sfIdx] = 1$ は、 $sfIdx$ で示される輝度フィルタの係数が信号通知されることを規定する。 $alf_luma_coeff_flag[sfIdx] = 0$ は、 $sfIdx$ で示される輝度フィルタのすべてのフィルタ係数が 0 に等しく定する。存在しない場合、 $alf_luma_coeff_flag[sfIdx]$ は 1 に等しく設定される。

$alf_luma_coeff_abs[sfIdx][j]$ は、 $sfIdx$ で示される信号輝度フィルタの j 番目の係数の絶対値を規定する。 $alf_luma_coeff_abs[sfIdx][j]$ が存在しない場合、それは 0 に等しいと推論される。

exp_Golomb 二値化 $uek(v)$ の次数 k は 3 に等しく設定される。

$alf_luma_coeff_sign[sfIdx][j]$ は、 $sfIdx$ が示すフィルタの j 番目の輝度係数の符号を以下のように規定する。

— $alf_luma_coeff_sign[sfIdx][j]$ が 0 に等しい場合、対応する輝度フィルタ係数は正の値を有する。

— そうでない場合 ($alf_luma_coeff_sign[sfIdx][j] = 1$)、対応する輝度フィルタ係数は負の値を有する。

$alf_luma_coeff_sign[sfIdx][j]$ が存在しない場合、それは 0 に等しいと推論される。

$sfIdx = 0 \dots alf_luma_num_filters_signalled_minus1$, $j = 0 \dots 11$ の変数 $filtCoeff[sfIdx][j]$ は、次のように初期化される。

$$filtCoeff[sfIdx][j] = alf_luma_coeff_abs[sfIdx][j] * (1 - 2 * alf_luma_coeff_sign[sfIdx][j]) \quad (7-47)$$

要素 $AlfCoeffL[adaptation_parameter_set_id]$

10

20

30

40

50

]`[filtIdx][j]`を有する輝度フィルタ係数`AlfCoeffL[adaptation_parameter_set_id]`、(但し、`filtIdx=0..NumAlfFilters-1`、`j=0..11`)は以下のように導出される。

`AlfCoeffL[adaptation_parameter_set_id][filtIdx][j]=filtCoeff[alf_luma_coeff_delta_idx[filtIdx]][j]` (7-48)

固定フィルタ係数`AlfFixFiltCoeff[i][j]` (但し`i=0..64`、`j=0..11`)、およびの`AlfClassToFiltMap[m][n]` (但し`m=0..15`、`n=0..24`)をフィルタマッピングするクラスは、以下のように導出される。

【0105】

`AlfFixFiltCoeff=`

```
{
{0, 0, 2, -3, 1, -4, 1, 7, -1, 1, -1, 5}
{0, 0, 0, 0, 0, -1, 0, 1, 0, 0, -1, 2}
{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 1}
{2, 2, -7, -3, 0, -5, 13, 22, 12, -3, -3, 17}
{-1, 0, 6, -8, 1, -5, 1, 23, 0, 2, -5, 10}
{0, 0, -1, -1, 0, -1, 2, 1, 0, 0, -1, 4}
{0, 0, 3, -11, 1, 0, -1, 35, 5, 2, -9, 9}
{0, 0, 8, -8, -2, -7, 4, 4, 2, 1, -1, 25}
{0, 0, 1, -1, 0, -3, 1, 3, -1, 1, -1, 3}
{0, 0, 3, -3, 0, -6, 5, -1, 2, 1, -4, 21}
{-7, 1, 5, 4, -3, 5, 11, 13, 12, -8, 11, 12}
{-5, -3, 6, -2, -3, 8, 14, 15, 2, -7, 11, 16}
{2, -1, -6, -5, -2, -2, 20, 14, -4, 0, -3, 25}
{3, 1, -8, -4, 0, -8, 22, 5, -3, 2, -10, 29}
{2, 1, -7, -1, 2, -11, 23, -5, 0, 2, -10, 29}
{-6, -3, 8, 9, -4, 8, 9, 7, 14, -2, 8, 9}
{2, 1, -4, -7, 0, -8, 17, 22, 1, -1, -4, 23}
{3, 0, -5, -7, 0, -7, 15, 18, -5, 0, -5, 27}
{2, 0, 0, -7, 1, -10, 13, 13, -4, 2, -7, 24}
{3, 3, -13, 4, -2, -5, 9, 21, 25, -2, -3, 12}
{-5, -2, 7, -3, -7, 9, 8, 9, 16, -2, 15, 12}
{0, -1, 0, -7, -5, 4, 11, 11, 8, -6, 12, 21}
{3, -2, -3, -8, -4, -1, 16, 15, -2, -3, 3, 26}
{2, 1, -5, -4, -1, -8, 16, 4, -2, 1, -7, 33}
{2, 1, -4, -2, 1, -10, 17, -2, 0, 2, -11, 33}
{1, -2, 7, -15, -16, 10, 8, 8, 20, 11, 14, 11}
{2, 2, 3, -13, -13, 4, 8, 12, 2, -3, 16, 24}
{1, 4, 0, -7, -8, -4, 9, 9, -2, -2, 8, 29}
{1, 1, 2, -4, -1, -6, 6, 3, -1, -1, -3, 30}
{-7, 3, 2, 10, -2, 3, 7, 11, 19, -7, 8, 10}
{0, -2, -5, -3, -2, 4, 20, 15, -1, -3, -1, 22}
{3, -1, -8, -4, -1, -4, 22, 8, -4, 2, -8, 28}
{0, 3, -14, 3, 0, 1, 19, 17, 8, -3, -7, 20}
{0, 2, -1, -8, 3, -6, 5, 21, 1, 1, -9, 13}
{-4, -2, 8, 20, -2, 2, 3, 5, 21, 4, 6, 1}
{2, -2, -3, -9, -4, 2, 14, 16, 3, -6, 8, 24}
```

10

20

30

40

50

```

{2, 1, 5, -16, -7, 2, 3, 11, 15, -3, 11, 22}
{1, 2, 3, -11, -2, -5, 4, 8, 9, -3, -2, 26}
{0, -1, 10, -9, -1, -8, 2, 3, 4, 0, 0, 29}
{1, 2, 0, -5, 1, -9, 9, 3, 0, 1, -7, 20}
{-2, 8, -6, -4, 3, -9, -8, 45, 14, 2, -13, 7}
{1, -1, 16, -19, -8, -4, -3, 2, 19, 0, 4, 30}
{1, 1, -3, 0, 2, -11, 15, -5, 1, 2, -9, 24}
{0, 1, -2, 0, 1, -4, 4, 0, 0, 1, -4, 7}
{0, 1, 2, -5, 1, -6, 4, 10, -2, 1, -4, 10}
{3, 0, -3, -6, -2, -6, 14, 8, -1, -1, -3, 31}
{0, 1, 0, -2, 1, -6, 5, 1, 0, 1, -5, 13}
{3, 1, 9, -19, -21, 9, 7, 6, 13, 5, 15, 21}
{2, 4, 3, -12, -13, 1, 7, 8, 3, 0, 12, 26}
{3, 1, -8, -2, 0, -6, 18, 2, -2, 3, -10, 23}
{1, 1, -4, -1, 1, -5, 8, 1, -1, 2, -5, 10}
{0, 1, -1, 0, 0, -2, 2, 0, 0, 1, -2, 3}
{1, 1, -2, -7, 1, -7, 14, 18, 0, 0, -7, 21}
{0, 1, 0, -2, 0, -7, 8, 1, -2, 0, -3, 24}
{0, 1, 1, -2, 2, -10, 10, 0, -2, 1, -7, 23}
{0, 2, 2, -11, 2, -4, -3, 39, 7, 1, -10, 9}
{1, 0, 13, -16, -5, -6, -1, 8, 6, 0, 6, 29}
{1, 3, 1, -6, -4, -7, 9, 6, -3, -2, 3, 33}
{4, 0, -17, -1, -1, 5, 26, 8, -2, 3, -15, 30}
{0, 1, -2, 0, 2, -8, 12, -6, 1, 1, -6, 16}
{0, 0, 0, -1, 1, -4, 4, 0, 0, 0, -3, 11}
{0, 1, 2, -8, 2, -6, 5, 15, 0, 2, -7, 9}
{1, -1, 12, -15, -7, -2, 3, 6, 6, -1, 7, 30}
}, (7-49)
[0106]
AlfClassToFilterMap=
{
{8, 2, 2, 2, 3, 4, 53, 9, 9, 52, 4, 4, 5, 9, 2, 8, 10, 9,
1, 3, 39, 39, 10, 9, 52}
{11, 12, 13, 14, 15, 30, 11, 17, 18, 19, 16, 20, 20,
4, 53, 21, 22, 23, 14, 25, 26, 26, 27, 28, 10}
{16, 12, 31, 32, 14, 16, 30, 33, 53, 34, 35, 16, 20,
4, 7, 16, 21, 36, 18, 19, 21, 26, 37, 38, 39}
{35, 11, 13, 14, 43, 35, 16, 4, 34, 62, 35, 35, 30, 5
6, 7, 35, 21, 38, 24, 40, 16, 21, 48, 57, 39}
{11, 31, 32, 43, 44, 16, 4, 17, 34, 45, 30, 20, 20, 7
, 5, 21, 22, 46, 40, 47, 26, 48, 63, 58, 10}
{12, 13, 50, 51, 52, 11, 17, 53, 45, 9, 30, 4, 53, 19
, 0, 22, 23, 25, 43, 44, 37, 27, 28, 10, 55}
{30, 33, 62, 51, 44, 20, 41, 56, 34, 45, 20, 41, 41,
56, 5, 30, 56, 38, 40, 47, 11, 37, 42, 57, 8}
{35, 11, 23, 32, 14, 35, 20, 4, 17, 18, 21, 20, 20, 2
0, 4, 16, 21, 36, 46, 25, 41, 26, 48, 49, 58}
{12, 31, 59, 59, 3, 33, 33, 59, 59, 52, 4, 33, 17, 59
, 55, 22, 36, 59, 59, 60, 22, 36, 59, 25, 55}
{31, 25, 15, 60, 60, 22, 17, 19, 55, 55, 20, 20, 53,

```

```

19, 55, 22, 46, 25, 43, 60, 37, 28, 10, 55, 52}
{12, 31, 32, 50, 51, 11, 33, 53, 19, 45, 16, 4, 4, 53
, 5, 22, 36, 18, 25, 43, 26, 27, 27, 28, 10}
{5, 2, 44, 52, 3, 4, 53, 45, 9, 3, 4, 56, 5, 0, 2, 5, 10
, 47, 52, 3, 63, 39, 10, 9, 52}
{12, 34, 44, 44, 3, 56, 56, 62, 45, 9, 56, 56, 7, 5, 0
, 22, 38, 40, 47, 52, 48, 57, 39, 10, 9}
{35, 11, 23, 14, 51, 35, 20, 41, 56, 62, 16, 20, 41,
56, 7, 16, 21, 38, 24, 40, 26, 26, 42, 57, 39}
{33, 34, 51, 51, 52, 41, 41, 34, 62, 0, 41, 41, 56, 7
, 5, 56, 38, 38, 40, 44, 37, 42, 57, 39, 10}
{16, 31, 32, 15, 60, 30, 4, 17, 19, 25, 22, 20, 4, 53
, 19, 21, 22, 46, 25, 55, 26, 48, 63, 58, 55}
}, (7-50)
[0107]

```

`AlfCoeffL[adaptation_parameter_set_id][filtIdx][j]` (但し `filtIdx=0..NumAlfFilters-1`, `j=0..11`) の値が、 $-2^7 \sim 2^7 - 1$ の範囲内とすることは、ビットストリーム適合性の要件である。

`alf_luma_clip_idx[sfIdx][j]` は、`sfIdx` が示す信号通知された輝度フィルタの `j` 番目の係数を乗じる前に使用するクリッピング値のクリッピングインデックスを規定する。`alf_luma_clip_idx[sfIdx][j]` (但し `sfIdx=0..alf_luma_num_filters_signalled_minus1` かつ `j=0..11`) の値は、 $0 \sim 3$ 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。

輝度フィルタクリッピング値 `AlfClipL[adaptation_parameter_set_id]` (但し要素 `AlfClipL[adaptation_parameter_set_id][filtIdx][j]` であり、`filtIdx=0..NumAlfFilters-1` および `j=0..11`) は、`BitDepthY` に等しく設定されている `bitDepth` および `alf_luma_clip_idx[alf_luma_coeff_delta_idx[filtIdx]][j]` に等しく設定されている `clipIdx` に依存して表 4 で規定されているように導出される。

`alf_chroma_num_alt_filters_minus1` プラス 1 は、クロマ成分のための代替フィルタの数を規定する。

`alf_chroma_clip_flag[altIdx]=0` は、インデックス `altIdx` を有するクロマフィルタを使用する場合、クロマ成分に対して線形適応ループフィルタリングを適用することを規定し、`alf_chroma_clip_flag[altIdx]=1` は、インデックス `altIdx` を有するクロマフィルタを使用する場合、クロマ成分に対して非線形適応ループフィルタリングを適用することを規定する。存在しない場合、`alf_chroma_clip_flag[altIdx]` は 0 に等しいと推論される。

`alf_chroma_coeff_abs[altIdx][j]` は、インデックス `altIdx` を有する代替クロマフィルタに対する `j` 番目のクロマフィルタ係数の絶対値を規定する。`alf_chroma_coeff_abs[altIdx][j]` が存在しない場合、それは 0 に等しいと推論される。`alf_chroma_coeff_abs[altIdx][j]` の値が $0 \sim 2^7 - 1$ の範囲内 (両端を含む) であることが、ビットストリーム適合性の要件である。

`exp-Golomb` 二値化 `uek(v)` の次数 `k` は 3 に等しく設定される。

`alf_chroma_coeff_sign[altIdx][j]` は、インデックス `altIdx` を有する代替クロマフィルタの `j` 番目のクロマフィルタ係数の符号を以下

10

20

30

40

50

のように規定する。

－ $\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ が 0 に等しい場合、対応するクロマフィルタ係数は正の値を有する。

－ そうでない場合 ($\text{alf_chroma_coeff_sign}[\text{altIdx}][j] = 1$)、対応するクロマフィルタ係数は負の値を有する。

$\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ が存在しない場合、それは 0 に等しいと推論される。

$\text{AlfCoeffc}[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ を要素とするクロマフィルタ係数 $\text{AlfCoeffc}[\text{adaptation_parameter_set_id}][\text{altIdx}]$ (但し $\text{altIdx} = 0 \dots \text{alf_chroma_num_alt_filters_minus1}$, $j = 0 \dots 5$ とする) は、以下のように導出する。

$\text{AlfCoeffc}[\text{adaptation_parameter_set_id}][\text{altIdx}][j] = \text{alf_chroma_coeff_abs}[\text{altIdx}][j] * (1 - 2 * \text{alf_chroma_coeff_sign}[\text{altIdx}][j])$
(7-51)

$\text{AlfCoeffc}[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ の値 (但し、 $\text{altIdx} = 0 \dots \text{alf_chroma_num_alt_filters_minus1}$, $j = 0 \dots 5$) が $-2^7 - 1 \sim 2^7 - 1$ の範囲内にあることは、ビットストリーム適合性の要件である。

$\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ は、インデックス altIdx を有する代替クロマフィルタの j 番目の係数を乗じる前に使用するクリッピング値のクリッピングインデックスを規定する。 $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ の値 (但し、 $\text{altIdx} = 0 \dots \text{alf_chroma_num_alt_filters_minus1}$, $j = 0 \dots 5$ である) が、0～3 の範囲内にあることは、ビットストリーム適合性の要件である。

要素 $\text{AlfClipc}[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ を有するクロマフィルタクリッピング値 $\text{AlfClipc}[\text{adaptation_parameter_set_id}][\text{altIdx}]$ (但し、 $\text{altIdx} = 0 \dots \text{alf_chroma_num_alt_filters_minus1}$, $j = 0 \dots 5$) は、 BitDepthc と等しく設定されている bitDepth および $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ と等しく設定されている clipIdx に依存して、表 4 で規定されているように導出される。

【0108】

【表 6】

表 4－ビット深度と clipIdx に依存する AlfClip 仕様

bitDepth	clipIdx			
	0	1	2	3
8	255	64	16	4
9	511	108	23	5
10	1023	181	32	6
11	2047	304	45	7
12	4095	512	64	8
13	8191	861	91	10
14	16383	1448	128	11
15	32767	2435	181	13
16	65535	4096	256	16

【0109】

2. 8. CTUのためのALFパラメータの信号通知

VTM6において、ALFフィルタパラメータは、適応パラメータセット (APS) に

において信号通知される。1つのAPSにおいて、最大25組の輝度フィルタ係数およびクリッピング値インデックス、並びに最大8組のクロマフィルタ係数およびクリッピング値インデックス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、輝度成分の異なる分類のフィルタ係数をマージすることができる。スライスヘッダにおいて、現在のスライスに使用されるAPSのインデックスが信号通知される。

【0110】

APSから復号化されたクリッピング値インデックスは、クリッピング値の輝度テーブル(Luma table)およびクリッピング値のクロマテーブル(Chroma table)を使用してクリッピング値を判定することができるようになる。これらのクリッピング値は、内部ビット深度に依存する。

10

【0111】

スライスヘッダにおいて、現在のスライスに使用される輝度フィルタ集合を規定するように、最大7つのAPSインデックスを信号通知することができる。フィルタリング処理はCTBレベルでさらに制御されてもよい。ALFが輝度CTBに適用されるかどうかを示すために、常に1つのフラグが信号通知される。1つの輝度CTBは、16個の固定フィルタセットのうち1つのフィルタセットを選択し、複数のAPSから1つのフィルタセットを選択することができる。どのフィルタセットが適用されるかを示すように、輝度CTBのためにフィルタセットインデックスが信号通知される。エンコーダおよびデコーダの両方において、16個の固定フィルタセットを予め規定し、ハードコードする。

【0112】

20

クロマ成分の場合、現在のスライスに使用されているクロマフィルタ集合を示すように、スライスヘッダにAPSインデックスを信号通知する。CTBレベルにおいて、APSにおいて複数のクロマフィルタ集合が存在する場合、クロマCTBごとにフィルタインデックスを信号通知する。

【0113】

具体的には、以下が適用される。

【0114】

スライス内の少なくとも1つのCTUがALFを適用するかどうかを示すために、まずスライスオン／オフ制御フラグをコード化する。真である場合、CTUごとに、以下をチェックし、順に信号通知する。

30

【0115】

太字の下線付きテキストは、既存の標準への変更案を示す。

【0116】

輝度部分に関して：

1. ALFが輝度CTBに適用されるかどうか。そうである場合、ステップ2に進む。そうでない場合、さらなる信号通知は必要とされない。
2. 現在のスライスに使用されているALF APSの数をチェックし、それをnumALFAPSで表す。
3. numALFAPSが0に等しい場合、固定フィルタのインデックス(例えば、alf_luma_fixed_filter_idx)が信号通知される。そうでない場合、以下が適用される。
 - － 1つ目のALF APSから予測されたものであるかどうかを示すためのフラグを信号通知する。
 - － そうでない場合、ステップ4に進む。そうでない場合には、輝度CTBのためのALFパラメータの信号通知は停止される。
4. numALFAPSが1よりも大きい場合、それがALF APSから予測されたものであるかどうかを示すためのフラグを信号通知する。
 - － 大きくない場合、固定フィルタのインデックスを信号通知する。
 - － 大きい場合、かつnumALFAPSが2よりも大きい場合、ALF APSのインデックスから1を引いたものを、端数を切り捨てた単項信号通知する。

40

50

【0117】

クロマ部分に関して：

1. Cb/Cr CTBにALFを適用するかどうか。そうである場合、ステップ2に進む。そうでない場合、さらなる信号通知は必要とされない。

2. i 番目のALF APSに関連付けられたフィルタのインデックスを信号通知する。ここで、APSインデックスは、スライスヘッダにおいて信号通知される。

【0118】

【表7】

7. 3. 8. 2 符号化ツリーユニット構文

記述子	
<code>coding_tree_unit() {</code>	
<code> xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY</code>	
<code> yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY</code>	
<code> if(slice_sao_luma_flag slice_sao_chroma_flag)</code>	
<code> sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)</code>	
<code> if(slice_alf_enabled_flag){</code>	
<code> alf_ctb_flag[0 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	<u>ae(v)</u>
<code> if(alf_ctb_flag[0 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]) {</code>	
<code> if(slice_num_alf_aps_ids_luma > 0)</code>	
<code> alf_ctb_use_first_aps_flag</code>	<u>ae(v)</u>
<code> if(!alf_ctb_use_first_aps_flag) {</code>	
<code> if(slice_num_alf_aps_ids_luma > 1)</code>	
<code> alf_use_aps_flag</code>	<u>ae(v)</u>
<code> if(alf_use_aps_flag) {</code>	
<code> if(slice_num_alf_aps_ids_luma > 2)</code>	
<code> alf_luma_prev_filter_idx_minus1</code>	<u>ae(v)</u>
<code> } else</code>	
<code> alf_luma_fixed_filter_idx</code>	<u>ae(v)</u>
<code> }</code>	
<code> }</code>	
<code> if(slice_alf_chroma_idc == 1 slice_alf_chroma_idc == 3) {</code>	
<code> alf_ctb_flag[1 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	<u>ae(v)</u>
<code> if(alf_ctb_flag[1 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	
<code> && aps_alf_chroma_num_alt_filters_minus1 > 0)</code>	
<code> alf_ctb_filter_alt_idx[0 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	<u>ae(v)</u>
<code> }</code>	
<code> if(slice_alf_chroma_idc == 2 slice_alf_chroma_idc == 3) {</code>	
<code> alf_ctb_flag[2 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	<u>ae(v)</u>
<code> if(alf_ctb_flag[2 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	
<code> && aps_alf_chroma_num_alt_filters_minus1 > 0)</code>	
<code> alf_ctb_filter_alt_idx[1 xCtb >> CtbLog2SizeY yCtb >> CtbLog2SizeY]</code>	<u>ae(v)</u>
<code> }</code>	
<code> }</code>	
<code> if(slice_type == I && qtbtt_dual_tree_intra_flag)</code>	
<code> dual_tree_implicit_qt_split(xCtb, yCtb, CtbSizeY, 0)</code>	
<code> Else</code>	
<code> coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 1, 0, 0, 0, 0, 0,</code>	
<code> SINGLE_TREE, MODE_TYPE_ALL)</code>	
<code> }</code>	

【0119】

2. 9. クロス成分適応ループフィルタ (CC-ALF)

クロス成分適応ループフィルタ (CC-ALF) は、輝度サンプル値を用いて各クロマ成分を微調整する。基本的に、CC-ALFは、CC-ALFが適用される場合、輝度サ

サンプルをフィルタリングすることによりクロマサンプルごとの補正を行う。それはループフィルタステップとして適用される。このツールは、ビットストリームにおける情報によって制御され、この情報は、(a) クロマ成分ごとのフィルタ係数と、(b) サンプルブロックのためのフィルタの適用を制御するマスクとを含む。

【0120】

以下の図10Aは、他のループフィルタに対するCC-A LFの配置を示す。CC-A LFは、クロマ成分ごとに輝度チャンネルに線形菱形フィルタ（図10Bに示す）を適用することによって動作し、以下のように表される。

【0121】

【数16】

$$\Delta I_i(x, y) = \sum_{(x_0, y_0) \in S_i} I_0(x_c + x_0, y_c + y_0) c_i(x_0, y_0), \quad (20)$$

10

ここで、

(x, y) は、微調整されたクロマ成分 i の位置である。

(x_c, y_c) は、(x, y) に基づく輝度位置である。

S_i は、クロマ成分 i のための輝度におけるフィルタサポートである。

c_i(x₀, y₀) は、フィルタ係数を表す。

【0122】

CC-A LF 処理は、JVET-00636 にさらに記載されている。主な特徴には以下が含まれる。

20

- 輝度平面とクロマ平面との間の空間拡大縮小率に基づいて、サポート領域の中心となる輝度位置 (x_c, y_c) を計算する。
- すべてのフィルタ係数はAPSで送信され、8ビットのダイナミックレンジを有する。
- スライスヘッダにおいて、APSが参照されてもよい。
- スライスのクロマ成分ごとに使用されるCC-A LF係数も、時間的サブレイヤに対応するバッファに記憶される。これらの時間的サブレイヤフィルタ係数セットの再利用は、スライスレベルフラグを使用して容易になる。
- CC-A LFフィルタの適用は、可変ブロックサイズで制御され、サンプルのブロックごとに受信されたコンテキスト符号化フラグによって信号通知される。ブロックサイズは、CC-A LF有効化フラグと共に、クロマ成分ごとにスライスレベルで受信される。
- 水平方向の仮想境界のための境界パディングは、繰り返しを利用する。残りの境界については、通常のA LFと同じタイプのパディングを使用する。

30

【0123】

2. 9. 1. JVET-P1008におけるCC-A LFの更なる簡素化

JVET-P0080に比べ、CC-A LFの設計を簡単にするために、以下のような新しい態様を提案する。

ー 複雑性低減

○ 図11に示すように、フィルタ形状を3×4菱形に変化させ、フィルタ演算における乗数を低減する。

40

○ CC-A LF係数の6ビットまでのダイナミックレンジを制限

○ クロマA LFと乗数の共用を許可する。

ー A LFとの位置合わせ

○ フィルタの選択をCTUレベルでの信号通知に制限

○ 時間レイヤ係数バッファの除去

○ A LF仮想境界における対称ライン選択の使用

【0124】

また、簡略化により符号化効率が低下するため、クロマ成分当たり4つまでのフィルタを適用することができるように制限される。

【0125】

50

2.9.1.1. CC-A L Fの構文および意味論

太字の下線付きテキストは、既存の標準への変更案を示す。

【0126】

【表8】

7.3.6 スライスヘッダ構文

7.3.6.1 一般スライスセグメントヘッダ構文

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
non_reference_picture_flag	u(1)
slice_type	ue(v)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
slice_pic_order_cnt_lsb	u(v)
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	ue(v)
if(nal_unit_type == IDR_W_RADL nal_unit_type == IDR_N_LP nal_unit_type == CRA_NUT NalUnitType == GDR_NUT)	
no_output_of_prior_pics_flag	u(1)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 (i == 1 && rpl1_idx_present_flag)))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if((slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	

10

20

30

40

50

【表 9】

if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
slice_log2_diff_min_qt_min_cb_luma	ue(v)
slice_max_mtt_hierarchy_depth_luma	ue(v)
if(slice_max_mtt_hierarchy_depth_luma != 0)	
slice_log2_diff_max_bt_min_qt_luma	ue(v)
slice_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(slice_type == I && qtbtt_dual_tree_intra_flag) {	
slice_log2_diff_min_qt_min_cb_chroma	ue(v)
slice_max_mtt_hierarchy_depth_chroma	ue(v)
if(slice_max_mtt_hierarchy_depth_chroma != 0)	
slice_log2_diff_max_bt_min_qt_chroma	ue(v)
slice_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if(slice_type != I) {	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
slice_temporal_mvp_enabled_flag	u(1)
if(slice_type == B && !pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(slice_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table()	
if(!pps_six_minus_max_num_merge_cand_plus1)	
six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1) [Ed.(YK):There is a syntax element name subsetting issue here.]	
five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
slice_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
slice_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1) [Ed.(YK):There is a syntax element name subsetting issue here.]	

10

20

30

40

50

【表 1 0】

max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if(sps_ibc_enabled_flag)	
slice_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_sign_flag	u(1)
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
<u>if(ChromaArrayType != 0)</u>	
<u>slice cross component alf cb enabled flag</u>	<u>u(1)</u>
<u>if(slice cross component alf cb enabled flag) {</u>	
<u>slice cross component alf cb aps id</u>	<u>u(3)</u>
}	
<u>if(ChromaArrayType != 0)</u>	
<u>slice cross component alf cr enabled flag</u>	<u>u(1)</u>
<u>if(slice cross component alf cr enabled flag) {</u>	
<u>slice cross component alf cr aps id</u>	<u>u(3)</u>
}	
...	
}	
byte_alignment()	
}	

【 0 1 2 7 】

10

20

30

40

50

【表 1 1】

適応ループフィルタデータ構文

alf_data() {	記述子
alf_luma_filter_signal_flag	u(1)
alf_chroma_filter_signal_flag	u(1)
alf_cross_component_cb_filter_signal_flag	u(1)
alf_cross_component_cr_filter_signal_flag	u(1)
if(alf_luma_filter_signal_flag) {	
alf_luma_clip_flag	u(1)
alf_luma_num_filters_signalled_minus1	ue(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	u(v)
}	
alf_luma_coeff_signalled_flag	u(1)
if(alf_luma_coeff_signalled_flag) {	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++)	
alf_luma_coeff_flag[sfIdx]	u(1)
}	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++) {	
if(alf_luma_coeff_flag[sfIdx]) {	
for (j = 0; j < 12; j++) {	
alf_luma_coeff_abs[sfIdx][j]	uek(v)
if(alf_luma_coeff_abs[sfIdx][j])	
alf_luma_coeff_sign[sfIdx][j]	u(1)
}	
}	
}	
if(alf_luma_clip_flag) {	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++) {	
if(alf_luma_coeff_flag[sfIdx]) {	
for (j = 0; j < 12; j++)	
alf_luma_clip_idx[sfIdx][j]	u(2)
}	
}	
}	
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_num_alt_filters_minus1	ue(v)
for(altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++) {	
alf_chroma_clip_flag[altIdx]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[altIdx][j]	uek(v)
if(alf_chroma_coeff_abs[altIdx][j] > 0)	
alf_chroma_coeff_sign[altIdx][j]	u(1)
}	
}	

10

20

30

40

50

【表 1 2】

if(alf_chroma_clip_flag[altIdx]) {	
for(j = 0; j < 6; j++)	
alf_chroma_clip_idx[altIdx][j]	u(2)
}	
}	
}	
if (alf_cross_component_cb_filter_signal_flag) {	
<u>alf_cross_component_cb_filters_signalled_minus1</u>	<u>ue(v)</u>
for(k = 0; k < (alf_cross_component_cb_filters_signalled_minus1+1); k++) {	
for (j = 0; j < 8; j++)	
<u>alf_cross_component_cb_coeff_plus32[k][j]</u>	<u>u(6)</u>
}	
}	
if (alf_cross_component_cr_filter_signal_flag) {	
<u>alf_cross_component_cr_filters_signalled_minus1</u>	<u>ue(v)</u>
for(k = 0; k < (alf_cross_component_cr_filters_signalled_minus1+1); k++) {	
for (j = 0; j < 8; j++)	
<u>alf_cross_component_cr_coeff_plus32[k][j]</u>	<u>u(6)</u>
}	
}	
}	

10

20

【 0 1 2 8 】

30

40

50

【表 1 3】

7. 3. 8. 2 符号化ツリーユニット構文

記述子	
	<code>coding_tree_unit() {</code>
	<code>xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY</code>
	<code>yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY</code>
	<code>if(slice_sao_luma_flag slice_sao_chroma_flag)</code>
	<code>sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)</code>
	<code>if(slice_alf_enabled_flag){</code>
ae(v)	<code>alf_ctb_flag[0][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>if(alf_ctb_flag[0][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]) {</code>
	<code>if(slice_num_alf_aps_ids_luma > 0)</code>
ae(v)	<code>alf_ctb_use_first_aps_flag</code>
	<code>if(!alf_ctb_use_first_aps_flag) {</code>
	<code>if(slice_num_alf_aps_ids_luma > 1)</code>
ae(v)	<code>alf_use_aps_flag</code>
	<code>if(alf_use_aps_flag) {</code>
	<code>if(slice_num_alf_aps_ids_luma > 2)</code>
ae(v)	<code>alf_luma_prev_filter_idx_minus1</code>
	<code>} else</code>
ae(v)	<code>alf_luma_fixed_filter_idx</code>
	<code>}</code>
	<code>}</code>
	<code>}</code>
	<code>if(slice_cross_component_alf_cb_enabled_flag)</code>
ae(v)	<code>alf_ctb_cross_component_cb_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>if(slice_alf_chroma_idc == 1 slice_alf_chroma_idc == 3) {</code>
ae(v)	<code>alf_ctb_flag[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>if(alf_ctb_flag[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>&& aps_alf_chroma_num_alt_filters_minus1 > 0)</code>
ae(v)	<code>alf_ctb_filter_alt_idx[0][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>}</code>
	<code>if(slice_cross_component_alf_cr_enabled_flag)</code>
ae(v)	<code>alf_ctb_cross_component_cr_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>if(slice_alf_chroma_idc == 2 slice_alf_chroma_idc == 3) {</code>
ae(v)	<code>alf_ctb_flag[2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>if(alf_ctb_flag[2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>&& aps_alf_chroma_num_alt_filters_minus1 > 0)</code>
ae(v)	<code>alf_ctb_filter_alt_idx[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]</code>
	<code>}</code>
	<code>if(slice_type == I && qtbtt_dual_tree_intra_flag)</code>
	<code>dual_tree_implicit_qt_split (xCtb, yCtb, CtbSizeY, 0)</code>
	<code>else</code>
	<code>coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 1, 0, 0, 0, 0,</code>
	<code>SINGLE_TREE, MODE_TYPE_ALL)</code>
	<code>}</code>

【0 1 2 9】

`alf_ctb_cross_component_cb_idc[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]` が 0 に等しい場合、クロス成分 Cb フィルタが Cb 色成分サンプルのブロックの輝度位置 (xCtb, yCtb) に適用されていないことを示す。`alf_cross_component_cb_idc[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]` が 0 に等しくない場合、`alf_cross_component_cb_idc[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]` - t h クロス成分 Cb フィルタが Cb 色成分サンプルのブロックの輝度位置 (xCtb

、 y_{Ctb})に適用されていることを示す。

$alf_cross_component_cr_idc[x_{Ctb} > CtbLog2SizeY][y_{Ctb} > CtbLog2SizeY]$ が0に等しい場合、クロス成分CrフィルタがCr色成分サンプルのブロックの輝度位置(x_{Ctb} , y_{Ctb})に適用されていないことを示す。 $alf_cross_component_cr_idc[x_{Ctb} > CtbLog2SizeY][y_{Ctb} > CtbLog2SizeY]$ が0に等しくない場合、 $alf_cross_component_cr_idc[x_{Ctb} > CtbLog2SizeY][y_{Ctb} > CtbLog2SizeY] - th$ クロス成分CrフィルタがCr色成分サンプルのブロックの輝度位置(x_{Ctb} , y_{Ctb})に適用されていることを示す。

10

【0130】

2.10. ピクチャ、サブピクチャ、スライス、タイル、CTUの分割

2.10.1. ピクチャをサブピクチャ、スライス、タイルに分割

この従属節では、ピクチャをサブピクチャ、スライス、およびタイルにどのように分割するかを規定する。

【0131】

1つのピクチャは、1つ以上のタイル行および1つ以上のタイル列に分割される。1つのタイルは、1つの画像の1つの矩形領域を覆う1つのCTUのシーケンスである。1つのタイルにおけるCTUは、そのタイル内でラスタスキャン順にスキャンされる。

【0132】

1つのスライスは、1つのピクチャのタイル内において、整数個の完全なタイルまたは整数個の連続した完全なCTU行からなる。

20

【0133】

2つのモードのスライス、即ちラスタスキャンスライスモードおよび矩形スライスモードがサポートされる。ラスタスキャンスライスモードにおいて、1つのスライスは、1つのピクチャのタイルラスタスキャンにおける1つの完全なタイルのシーケンスを含む。矩形スライスモードにおいて、1つのスライスは、ピクチャの矩形領域を集合的に形成する複数の完全なタイル、またはピクチャの矩形領域を集合的に形成する1つのタイルの複数の連続した完全なCTU行のいずれかを含む。矩形スライス内のタイルを、そのスライスに対応する矩形領域内で、タイルラスタスキャンの順にスキャンする。

30

【0134】

1つのサブピクチャは、1つのピクチャの矩形領域を集合的に覆う1つ以上のスライスを含む。

【0135】

図12は、ピクチャのラスタスキャンスライス分割の例を示し、ピクチャは、12個のタイルおよび3個のラスタスキャンスライスに分割される。

【0136】

図13は、ピクチャの矩形スライス分割の例を示し、ピクチャは、24個のタイル(6個のタイル列および4個のタイル行)と9個の矩形スライスとに分割される。

【0137】

図14は、タイルおよび矩形のスライスに分割されたピクチャの例を示し、このピクチャは、4つのタイル(2つのタイルの列および2つのタイルの行)と4つの矩形スライスとに分割される。

40

【0138】

図15は、ピクチャのサブピクチャ分割の例を示し、ピクチャは、様々な寸法の28個のサブピクチャに分割される。[Ed. (RS): 図15を図14に統合することで、コンセプトの整合性を説明するのに役立つ。]

【0139】

ピクチャが3つの別個の色平面を使用してコード化される($separate_colour_plane_flag$ が1に等しい)場合、スライスは、 $colour_plane$

50

ne_idの対応する値によって識別される1つの色成分のCTUのみを含み、ピクチャの各色成分配列は、同じcolour_plane_idを有するスライスから成る。ピクチャ内でcolour_plane_idが異なる値を有する符号化スライスは、colour_plane_idの各値について、colour_plane_idのその値を有する符号化スライスNALユニットは、各符号化スライスNALユニットの最初のCTUに対してタイルスキャン順でCTUアドレスを増加する順になるという制約の下で、互いにインターリーブすることができる。

【0140】

separate_colour_plane_flagが0に等しい場合、ピクチャの各CTUが、ちょうど1つのスライスに含まれる。separate_colour_plane_flagが1に等しい場合、色成分の各CTUはちょうど1つのスライスに含まれる（つまり、ピクチャの各CTUに対する情報がちょうど3つのスライスに存在し、これら3つのスライスはcolour_plane_idの値が異なる）。

【0141】

3. 既存の実装形態の欠点

ALF仮想境界に使用されるパディング法は、「二面パディング」または「ミラーリングパディング(mirrored padding)」と呼ばれてもよく、ここで、(i, j)に位置する1つのサンプルにパディングを行う場合、図6および図7に示すように、サンプルが利用可能でも、同じフィルタ係数を共有する(m, n)に位置する対応するサンプルにもパディングを行う。

【0142】

ピクチャ境界／360度映像仮想境界、通常の境界（例えば、最上および最下境界）に使用されるパディング方法は、「片側パディング」または「繰り返しパディング」と呼ばれてもよく、使用される1つのサンプルが境界の外側にある場合、そのサンプルは、ピクチャの内側の利用可能サンプル（例えば、最も近い利用可能サンプル）からコピーされる。

【0143】

現在のCC-ALF設計において、以下のような問題がある。

1. CC-ALFで使用されるパディング方法は、非線形ALFで使用されるパディング方法と完全には整合していない。
2. CC-ALFのオン／オフ制御はCTBレベルにある。すなわち、それが有効化されている場合、CTB内のすべてのクロマサンプルは、1つのクロマサンプルに対応するフィルタリングされた輝度サンプルから導出オフセットを加えることによって修正されるものとする。
3. CC-ALFおよびクロマALFの両方を適用したクロマサンプルごとに、3段階（クロマALFフィルタリング、CC-ALFオフセット導出、および最終的なクロマサンプルを導出するためのクロマフィルタサンプルの微調整）から少なくとも3つのクリッピング演算が必要である。具体的には、以下の3つのステージを説明する。

a. クロマALF処理において、フィルタリングされたクロマサンプルをクリップする。すなわち、

$$\text{sum}_{\text{chromaALF}} = \text{curr} + (\text{sum}_{\text{chromaALF}} + 64) >> 7 \quad (8-1290)$$

$$\text{alfPicture}[x_{\text{CTB}} + x][y_{\text{CTB}} + y] = \text{Clip3}(0, (1 << \text{BitDepthC}) - 1, \text{sum}_{\text{chromaALF}}) \quad (8-1291)$$

ここで、「curr」は、フィルタリングされる前のクロマサンプルを表し、「sum_chromaALF」は、クロマALFフィルタサポートでクリッピングされたクロマサンプルの差分にクロマALFフィルタ係数を掛けた合計を表す。

b. 導出オフセット(sumで示される)を輝度ビット深度に応じた範囲にクリップする、すなわち、

$$\text{sum} = \text{Clip3}(-(1 << (\text{BitDepthC} - 1)), (1 << (\text{BitDepthC} - 1)) - 1, \text{sum}) \quad (8-1292)$$

10

20

30

40

50

ここで、「sum」は、CC-A LFフィルタサポートにおける輝度サンプルにCC-A LFフィルタ係数を掛けた合計を表す。

c. フィルタリングされたクロマサンプルに導出オフセットを追加した後、最終的に微調整されたクロマサンプルをクリップする。

$$\text{sum} = \text{alfPicture}[x\text{CtbC}+x][y\text{CtbC}+y] + (\text{sum} + 64) >> (7 + (\text{BitDepthY} - \text{BitDepthC})) \quad (8-1293)$$

$$\text{ccAlfPicture}[x\text{CtbC}+x][y\text{CtbC}+y] = \text{Clip3}(0, (1 << \text{BitDepthC}) - 1, \text{sum}) \quad (8-1294)$$

ここで、「alfPicture[xCtbC+x][yCtbC+y]」は、chroma A LFでフィルタされた後のクロマサンプルを表し（ステージaの出力）、「sum」はCC-A LFフィルタサポートにおける輝度サンプルにCC-A LFフィルタ係数を掛けた合計を表し（ステージbの出力）、ccAlfPictureは最終的に微調整されたクロマサンプルを表している（ステージcの出力、最終ステージ）。

4. 式(8-1293)に示すように、ステージbの出力、即ち、可変の合計は、64と等しいオフセットで丸められるが、(BitDepthY-BitDepthC)を考慮しない。(BitDepthY-BitDepthC)が0に等しくない場合、丸めオフセットは最適ではない。

【0144】

4. 例示的な技術および実施形態

以下に詳述する実施形態の詳細は、一般的な概念を説明するための例であると考えられるべきである。これらの実施形態は狭義に解釈されるべきではない。さらに、これらの実施形態は、任意の方法で組み合わせることができる。

【0145】

本開示において、A LFフィルタは、所与の色成分の情報を用いて所与の色成分に適用されるフィルタを表してもよく（例えば、輝度A LFフィルタ（線形または非線形）は、輝度情報を用いて輝度に適用され、クロマA LFフィルタは、クロマ情報を用いてクロマに適用され、例えば、CbクロマA LFフィルタは、Cbサンプルをフィルタリングするためのものであり、CrクロマA LFフィルタは、Crサンプルをフィルタリングするためのものである）、一方、CC-A LFフィルタは、第1の色成分情報と異なる少なくとも第2の色成分情報を用いて第1の色成分に適用されるフィルタを表してもよい（例えば、第1の色成分は、CbまたはCrでもよく、第2の色成分は、輝度でもよい）。

【0146】

CC-A LFにおいて、「サンプルの補正」を導出し、これをA LFで処理する前にサンプルに加えることで、「微調整サンプル」を生成することができ、または「微調整サンプル」を直接導出することができる。

【0147】

映像ユニットは、スライス／タイル／ブリック／サブピクチャ／ピクチャ／360度仮想ピクチャ（360度仮想境界で囲まれる）であってもよく、または複数のサンプル／ピクセルを含む他の種類の映像領域であってもよい。

【0148】

以下の説明において、サンプルが「映像ユニットの境界にある」とは、サンプルと映像ユニットの境界との距離が閾値未満である、または閾値以下であることを意味する場合がある。「ライン」は、1つの同じ水平位置でのサンプルまたは1つの同じ垂直位置でのサンプル（即ち、同じ行のサンプルおよび／または同じ列のサンプル）を含む場合がある。一例において、関数Abs(x)は、以下のように定義されてもよい。

【0149】

【数17】

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

10

20

30

40

50

【0150】

「A L F 処理ユニット」とは、2つの水平方向の境界および2つの垂直方向の境界で囲まれたユニットを指す。2つの水平境界は、2つのA L F 仮想境界または1つのA L F 仮想境界および1つのピクチャ境界（またはスライス境界／タイル境界／ブリック境界／サブピクチャ境界／360度仮想境界）を含むことができる。2つの垂直境界は、2つの垂直C T U境界または1つの垂直C T U境界および1つのピクチャ境界（またはスライス境界／タイル境界／ブリック境界／サブピクチャ境界／360度仮想境界）を含むことができる。一例を図16に示す。

【0151】

「狭A L F 処理ユニット」とは、2つの水平方向の境界および2つの垂直方向の境界で囲まれたユニットを指す。1つの水平境界は、1つのA L F 仮想境界または1つの360度仮想境界を含んでもよく、他の水平境界は、1つのスライス／ブリック／タイル／サブピクチャ境界または1つの360度仮想境界または1つのピクチャ境界を含んでもよい。垂直境界は、C T U境界、ピクチャ境界、360度仮想境界またはスライス／ブリック／タイル／サブピクチャ境界であることができる。一例を図16に示す。

【0152】

本開示において、隣接サンプルは、現在のサンプルを含む現在の映像ユニットの外にあり（例えば、現在のピクチャ／サブピクチャ／タイル／スライス／ブリック／C T Uの外にある）、そのような映像ユニットの境界をまたぐフィルタリングが許可されていない場合、または隣接サンプルと現在のサンプルが360度の仮想境界の異なる側にあり、360度仮想境界をまたぐフィルタリングが許可されていない場合に、「利用不可」とマークすることができる。さらに別の例において、隣接サンプルは、異なる映像ユニットにおいて、現在の処理ユニット（例えば、A L F 処理ユニットまたは狭A L F 処理ユニット）の外にあり、映像ユニットをまたぐフィルタリングが許可されていない場合に、「利用不可」とマークされてもよい。

【0153】

同様に、C C - A L F でフィルタリングされるサンプル（例えば、現在のC C - A L F 設計における輝度サンプル）の場合、サンプルが、現在のクロマサンプルを含むものとは異なる映像ユニットに位置している場合、このサンプルにも「利用不可」とマークすることができる。

1. C C - A L F において、「利用不可」のサンプルをパディングする方法は、A L F に使用されるものと同じ規則に従うことができる。

a. 1つの例において、「ミラーリングパディング」は、C C - A L F におけるA L F 仮想境界における「利用不可」のサンプルのパディングのために適用され得る。

b. 1つの例において、「繰り返しパディング」は、C C - A L F におけるスライス／タイル／ブリック／サブピクチャ／ピクチャ／360度仮想境界における「利用不可」のサンプルをパディングするために適用され得る。

c. 一例において、「繰り返しパディング」は、C C - A L F におけるA L F 仮想境界を除くすべての境界に適用されてもよい。

i. あるいは、すべての境界に「繰り返しパディング」を適用してもよい。

i i. あるいは、すべての水平境界に「繰り返しパディング」を適用してもよい。

i i i. あるいは、すべての垂直境界に「繰り返しパディング」を適用してもよい。

i v. あるいは、すべての水平境界に「ミラーリングパディング」を適用してもよい。

v. あるいは、すべての垂直境界に「ミラーリングパディング」を適用してもよい。

v i. あるいは、すべての境界に「ミラーリングパディング」を適用してもよい。

d. 一例において、「繰り返しパディング」は、「A L F 処理ユニット」若しくはC T U、または「狭A L F 処理ユニット」若しくは「基本A L F 処理ユニット」（以下で定義される）またはこれらのすべての処理ユニットとは異なる映像領域に適用されてもよい。

e. 繰り返しパディングをどのように適用するかは、隣接サンプルが異なる隣接スラ

10

20

30

40

50

イス内にあるかどうかには依存し、ここで、現在のスライスはラスト走査スライスであってもよく、および／または隣接するスライスはラスト走査スライスであってもよい。

i. 一例において、ラスト走査スライスにおけるサンプルの場合、「繰り返しパディング」は、「A L F 処理ユニット」に適用されてもよい。例えば、隣接する「利用不可」のサンプルは、現在の「A L F 処理ユニット」中でその最も近いサンプルでパディングされる。例えば、図 17 に示すように、現在の「A L F 処理ユニット」の左側の隣接サンプルは「利用不可」であり、かつ、現在の「A L F 処理ユニット」におけるその最も近いサンプルから「利用不可」のサンプルをそれぞれパディングする。

2. 処理ユニット（例えば、A L F 処理ユニットまたは／および狭 A L F 処理ユニットまたは／および C T U）の利用不可の隣接サンプルは、A L F または／および C C - A L F において、以下のように、予め定義された順にパディングされてもよい。

10

a. 処理ユニットの上側の隣接サンプルが利用不可の場合、これらサンプルは処理ユニットの最上行でパディングされてもよい。

i. あるいは、さらに、左上の隣接サンプルは、処理ユニットの左最上サンプルでパディングされてもよい。

i i. あるいは、さらに、右上の隣接サンプルは、処理ユニットの右最上サンプルでパディングされてもよい。

b. 処理ユニットの下側の隣接サンプルが利用不可の場合、処理ユニットの最下行でパディングすることができる。

i. あるいは、さらに、左最下の隣接サンプルは、処理ユニットの左下サンプルでパディングされてもよい。

20

i i. あるいは、さらに、右下の隣接サンプルは、処理ユニットの右最下サンプルでパディングされてもよい。

c. 処理ユニットの左側の隣接サンプルが利用不可の場合、処理ユニットの左側の列でパディングしてもよい。

d. 処理ユニットの右側の隣接サンプルが利用不可の場合、処理ユニットの右側の列でパディングしてもよい。

e. 処理ユニットの左側の隣接サンプルおよび上側の隣接サンプルが利用可能であり、処理ユニットの左上の隣接サンプルが利用不可の場合、上側の隣接サンプルを使用して、処理ユニットの左上の隣接サンプルをパディングすることができる。

30

i. 代替的に、左側の隣接サンプルを使用して、処理ユニットの左上の隣接サンプルをパディングしてもよい。

f. 処理ユニットの左上および上側の隣接サンプルが利用不可で、処理ユニットの左側の隣接サンプルが利用可能の場合、左側の隣接サンプルを使用して、処理ユニットの左上の隣接サンプルをパディングすることができる。

g. 処理ユニットの左上および左側の隣接サンプルが利用不可で、処理ユニットの上側の隣接サンプルが利用可能の場合、上側の隣接サンプルを使用して処理ユニットの左上の隣接サンプルをパディングすることができる。

h. 処理ユニットの右側の隣接サンプルおよび下側の隣接サンプルが利用可能で、処理ユニットの右下の隣接サンプルが利用不可の場合、下側の隣接サンプルを使用して、処理ユニットの右下の隣接サンプルをパディングすることができる。

40

i. あるいは、右側の隣接サンプルを使用して、処理ユニットの右下の隣接サンプルをパディングすることができる。

i. 処理ユニットの右下および下側の隣接サンプルが利用不可で、処理ユニットの右側の隣接サンプルが利用可能の場合、右側の隣接サンプルを用いて処理ユニットの右下の隣接サンプルをパディングすることができる。

j. 処理ユニットの右下および右側の隣接サンプルが利用不可で、処理ユニットの下側の隣接サンプルが利用可能の場合、下側の隣接サンプルを使用して、処理ユニットの右下の隣接サンプルをパディングすることができる。

k. 処理ユニットの右上および右側の隣接サンプルが利用不可で、処理ユニットの上

50

側の隣接サンプルが利用可能の場合、上側の隣接サンプルを使用して処理ユニットの右上の隣接サンプルをパディングすることができる。

l. 処理ユニットの右上および上側の隣接サンプルが利用不可で、処理ユニットの右側の隣接サンプルが利用可能の場合、右側の隣接サンプルを使用して、処理ユニットの右上の隣接サンプルをパディングすることができる。

m. 処理ユニットの左下および左側の隣接サンプルが利用不可で、処理ユニットの下側の隣接サンプルが利用可能の場合、下側の隣接サンプルを使用して、処理ユニットの左下の隣接サンプルをパディングすることができる。

n. 処理ユニットの左下および下側の隣接サンプルが利用不可で、処理ユニットの左側の隣接サンプルが利用可能の場合、左側の隣接サンプルを使用して、処理ユニットの左下の隣接サンプルをパディングすることができる。

o. 処理ユニットは、`ctuUp`で表されるCTUの N (N は整数であり、例えば、 $N=4$) 行と、`ctuUp`の下にある`ctuDown`で表されるCTUの $CtbSize-M$ (M は整数であり、例えば、 $M=N$) 行とを含んでもよい。処理ユニットの隣接サンプルが利用可能かどうかをチェックするとき、隣接サンプルと`ctuDown`との関係を考慮してもよい。

i. 一例において、隣接サンプルが`ctuDown`と異なる映像ユニット内にあり (例えば、隣接サンプルおよび`ctuDown`は、異なるブリック／タイル／スライス／サブピクチャに属する、またはそれらが360度の仮想境界の異なる側にある)、そのような映像ユニットをまたぐフィルタリングが許可されない場合、隣接サンプルは「利用不可」と見なされる。

ii. あるいは、隣接サンプルの可用性をチェックするために、隣接サンプルと`ctuUp`との関係を考慮してもよい。

3. ラスタ走査スライスにおける「利用不可」のサンプルは、`CC-ALF`または／および`ALF`において予め定義された順にパディングされてもよい。現在のCTUは、現在のスライスにある。

a. 一例において、上側CTUおよび左側CTUが現在のスライスと同じスライス内にあり、左上CTUが現在のスライスと異なるスライス内にある場合、左上CTUにおける「利用不可」の隣接サンプルは、現在のCTUの上側CTUにおいてその最も近い隣接サンプルでパディングされる。一例を図18に示す。

i. あるいは、左上CTUにおける「利用不可」の隣接サンプルは、現在のCTUの左側CTUにおいてその最も近い隣接サンプルでパディングされる。

b. 一例において、下側CTUおよび右側CTUが現在のスライスと同じスライス内にあり、右下CTUが現在のスライスとは異なるスライス内にある場合、右下CTUにおける「利用不可」の隣接サンプルが、現在のCTUの下側CTUにおいてその最も近い隣接サンプルとパディングされる。一例を図18に示す。

i. あるいは、右下CTUの「利用不可」の隣接サンプルは、現在のCTUの右側CTUにおいてその最も近い隣接サンプルでパディングされる。

4. 処理ユニット (例えば、`ALF`処理ユニット、または／および狭`ALF`処理ユニット、または／およびCTU) は、1つ以上のブリック／スライス／タイル／サブピクチャ境界または／および360度仮想境界が交差する場合であって、そのような境界をまたぐフィルタリングが許可されていない場合、複数の処理ユニットに (水平または／および垂直に) 分割されてもよい。

a. あるいは、さらに、分割処理は、処理ユニットが任意のブリック／スライス／タイル／サブピクチャ境界または／および360度仮想境界または／および`ALF`仮想境界と交差しなくなるまで、再帰的に行われてもよく、そのような境界をまたぐフィルタリング処理が許可されず、例えば、そのような境界のみを処理ユニットの境界とすることができる。以下、このような処理ユニットを「基本`ALF`処理ユニット」と呼ぶ。「基本`ALF`処理ユニット」は、「`ALF`処理ユニット」がそのような境界のいずれかで交差しなければ、「`ALF`処理ユニット」であってもよい。

b. あるいは、さらに、このような分割処理が完成した後に、A L F 処理または／および C C - A L F 処理を行い、例えば、「基本 A L F 処理ユニット」において A L F 処理または／および C C - A L F を行う。

c. あるいは、上記のパディング方法は、「基本 A L F 処理ユニット」上で行われてもよい。

d. 「基本 A L F 処理ユニット」を生成する際に、異なる水平方向の境界を予め定義された順序でチェックしてもよい。

i. 一例において、チェック順は、A L F 仮想境界、360度仮想境界、そしてブリック／スライス／タイル／サブピクチャ境界である C T U 境界である。

i i. 一例において、チェック順は、360度仮想境界、A L F 仮想境界、そしてブリック／スライス／タイル／サブピクチャ境界である C T U 境界である。

i i i. 一例において、「基本 A L F 処理ユニット」を形成する際に、最上境界および最下境界に異なるチェック順を用いてもよい。

a) 例えば、最上境界の場合、チェック順は、A L F 仮想境界、360度仮想境界、そしてブリック／スライス／タイル／サブピクチャ境界である C T U 境界である。

b) 例えば、最下境界の場合、チェック順は、360度仮想境界、A L F 仮想境界、そしてブリック／スライス／タイル／サブピクチャ境界である C T U 境界である。

e. 「基本 A L F 処理ユニット」を生成する際に、異なる垂直方向の境界を予め定義された順にチェックしてもよい。

i. 一例において、チェック順は、360度仮想境界、そしてブリック／スライス／タイル／サブピクチャ境界である C T U 境界である。

i i. 一例において、チェック順は、ブリック／スライス／タイル／サブピクチャ境界、そして360度仮想境界である C T U 境界である。

i i i. 一例において、「基本 A L F 処理ユニット」を生成する際に、左境界および右境界に対して異なるチェック順を用いてもよい。

5. そのような境界をまたぐフィルタリング処理が許可されていない場合、A L F または／および C C - A L F 処理を映像ユニット境界でのサンプルに対して、無効とすることが提案されている。

a. 一例において、サンプルの A L F または／および C C - A L F 処理に少なくとも N (N は整数) 個の「利用不可」のサンプルが含まれている場合、このサンプルに対して A L F または／および C C - A L F 処理はスキップされる。例えば、N = 1 または 2 または 3 である。

b. 一例において、サンプルが水平映像ユニット境界にあり、そのような境界をまたぐフィルタリング処理は許可されず、サンプルと映像ユニット境界との垂直距離が M 個のサンプル以下である場合、このサンプルに対して A L F または／および C C - A L F はスキップされる。例えば、M = 0 または 1 または 2 または 3 である。

c. 一例において、サンプルが垂直映像ユニット境界にあり、そのような境界をまたぐフィルタリング処理が許可されず、サンプルと映像ユニット境界との水平距離が M 個のサンプル以下である場合、サンプルに対して A L F または／および C C - A L F はスキップされる。例えば、M = 0 または 1 または 2 または 3 である。

6. C T B と異なる映像領域に対して、A L F または／および C C - A L F のオン／オフ制御を決定してもよいことが提案される。

a. 一例において、映像領域は、「A L F 処理ユニット」または「狭 A L F 処理ユニット」または「ベーシック A L F 処理ユニット」である。

b. 1つの例において、オン／オフ制御は、映像領域に対して明確に信号通知されてもよい。

c. 一例において、オン／オフ制御は、映像領域に対して暗黙的に導出されてもよい。

7. C C - A L F の信号通知されたオン／オフ制御フラグは、第1の映像ユニット（例えば、C T B）に関連付けられ得るが、第1の映像ユニット内のサンプルについては、信号通知されたフラグが C C - A L F が有効とされていることを伝える場合でも、C C - A

10

20

30

40

50

L F は依然として無効にされてもよい。

a. 1つの例において、第1の映像ユニット内のサンプルのために C C - A L F を有効にするかまたは無効にするかは、サンプルの位置に依存し得る。

b. 一例において、第1の映像ユニット内のサンプルのために C C - A L F を有効にするかまたは無効にするかは、C C - A L F 処理に含まれる輝度サンプルに依存し得る。

c. 1つの例において、第1の映像ユニット内のサンプルのために C C - A L F を有効にするかまたは無効にするかは、現在のサンプルおよびその隣接サンプル、例えば、それらの差に依存し得る。

d. 一例において、第1の映像ユニット内のサンプルのために C C - A L F を有効にするかまたは無効にするかは、導出オフセットに依存し得る。

i. 一例において、導出オフセットが閾値よりも大きい（または小さくない）場合、オフセットは廃棄されてもよく、即ち、C C - A L F は無効にされる（オフセットが0にリセットされる場合に等しい）。

i i. 一例において、導出オフセットが閾値よりも小さい（または超えない）場合、オフセットは廃棄されてもよく、即ち、C C - A L F が無効にされる（オフセットが0にリセットされる場合に等しい）。

e. 一例において、第1の映像ユニット内のサンプルのために C C - A L F を有効にするかまたは無効にするかは、クロマサンプルおよび／または隣接ブロックを含むブロックのコード化情報に依存しうる。

8. そこで、クリッピング処理の段階を3（セクション3の3項の箇条書きで記載）から1または2に減らすことが提案されている。

a. 一例において、C C - A L F が有効にされる場合、クロマ A L F フィルタリング処理におけるクリッピング処理は省略される。

b. 一例において、C C - A L F 処理における導出オフセットのクリッピング処理は省略される。

c. 一例において、上記2つの方法は両方とも適用される。

9. C C - A L F 処理における導出オフセットのクリッピング範囲は、 $[-(1 < (BitDepthC - 1)) , (1 < (BitDepthC - 1)) - 1]$ （両端を含む）の範囲とは異なって設定されてもよい。

a. 一例において、より小さな範囲を利用してもよい。

b. 一例において、この範囲はビット深度から独立してもよい。

c. 一例において、この範囲は、第1の映像領域から別の映像領域に適応的に変更されてもよい。

d. 一例において、この範囲は、クロマ A L F 処理に用いられる非線形クリッピング範囲（例えば、式（14）における $k(i, j)$ ）から導出されてもよい。

10. 導出オフセット（例えば、セクション3の箇条書き中の sum ）は、固定値（即ち、64）ではなく、ビット深度に応じた値に丸めオフセットにより丸められてもよい。

a. 一例において、丸めオフセットは、第1の色成分（例えば、Y）と第2の色成分（例えば、Cb r または Cr）とのビット深度差に依存してもよい。

i. 代替的に、それはまた、C C - A L F フィルタ係数のビット深さに依存してもよい。

i i. 一例において、丸めオフセットは、 $(1 < (K - 1 + (BitDepthY - BitDepthC)))$ に設定されてもよく、K は C C - A L F フィルタ係数のビット深度を表し、 $BitDepthY$ および $BitDepthC$ はそれぞれ輝度成分およびクロマ成分のビット深度を表す。

b. 一例において、式（8-1293）を置き換えるために、以下を適用してもよい。削除されたテキストには二重括弧で囲んだ印が付けられている（例えば、 $[[a]]$ は文字「a」の削除を意味する）。

$sum = alfPicture[xCbC + x][yCbC + y] + (sum + Off[[64]]) >> (7 + (BitDepthY - BitDepthC))$ (8

10

20

30

40

50

− 1 2 9 3)

$ccAlfPicture[xCtbC+x][yCtbC+y] = Clip3(0, (1 \ll BitDepth_C) - 1, sum')$ (8-1294)

ここで、Offは、 $(1 \ll (6 + (BitDepth_Y - BitDepth_C)))$ に等しく設定される。

11. クロマALFおよびCC-ALFを共同して最適化することが提案されている。

a. 一例において、CTBまたは映像領域に対してクロマALFおよびCC-ALFの両方が有効にされる場合、CC-ALFがクロマALFの後に適用される場合、CC-ALF処理においてのみ、クリッピング演算（例えば、クロマALFまたは／およびCC-ALFにより生成されたクロマサンプルを、クロマビット深度に依存し得る所定の範囲内にクリップする）が適用されてもよい。

10

b. 一例において、CTBまたは映像領域に対してクロマALFおよびCC-ALFの両方が有効にされる場合、CC-ALFがクロマALFの前に適用される場合、ALF処理においてのみ、クリッピング演算（例えば、クロマALFまたは／およびCC-ALFにより生成されたクロマサンプルを、クロマビット深度に依存し得る所定の範囲内にクリップする）が適用されてもよい。

12. 第1の色成分におけるサンプルSを $S = S + D$ として修正し、ここで、Dは、例えばCC-ALFのような、第2の色成分情報を用いた適応ループフィルタリング方法により導出オフセットであり、適応的にクリッピングされ得る。例えば、Sは、 $S = S + Clip3(Dlow, Dhigh, D)$ として修正されてもよく、Dlowおよび／またはDhighは、異なるサンプルごとに異なってもよい。

20

a. 例えば、Dlowおよび／またはDhighは、エンコーダからデコーダに信号通知されてもよい。

b. 例えば、Sの値に基づいてDlowおよび／またはDhighを導出してよい。

c. 例えば、Sの位置に基づいてDlowおよび／またはDhighを導出してよい。

d. 例えば、Dlowおよび／またはDhighの複数の候補に信号通知してもよく、または予め定義してもよい。選択された候補は、エンコーダから信号通知されてもよいし、デコーダにおいて導出されてもよい。

13. 第1サブピクチャの第1サンプルSに対するCC-ALFを、第2サブピクチャのサンプルに依存し得るものとして、次に、第1サブピクチャおよび／または第2のサブピクチャに対してサブピクチャをまたぐループフィルタリングが許可されるかどうかによって、SがCC-ALFによってフィルタリングされるかどうかを決定する。（例えば、第1のサブピクチャおよび／または第2のサブピクチャの`loop_filter_across_subpic_enabled_flag`）。

30

a. 一例において、サブピクチャをまたぐループフィルタが第1のサブピクチャに対して許可されない場合、SはCC-ALFによってフィルタリングされない。

b. 一例において、サブピクチャをまたぐループフィルタが第2のサブピクチャに対して許可されない場合、SはCC-ALFによってフィルタリングされない。

c. 一例において、第1のサブピクチャおよび第2のサブピクチャに対してサブピクチャをまたぐループフィルタが許可されない場合、SはCC-ALFによってフィルタリングされない。

40

d. 一例において、第1のサブピクチャまたは第2のサブピクチャに対してサブピクチャをまたぐループフィルタが許可されない場合、SはCC-ALFによってフィルタリングされない。

14. 上記方法を適用するかどうかおよび／またはどのように適用するかは、以下に列挙される1つ以上の条件に基づくことができる。

a. 映像コンテンツ（例えば、スクリーンコンテンツまたは自然コンテンツ）

b. 一例において、デフォルトモードは、DPS／SPS／VPS／PPS／APS／ピクチャヘッダ／スライスヘッダ／タイルグループヘッダ／最大コーディングユニット

50

(LCU)／コーディングユニット(CU)／LCU行／LCUグループ／TU／PUブロック／映像コーディングユニットにおいて信号通知されてもよい。

c. CU／PU／TU／ブロック／映像コーディングユニットの位置

d. 現在のブロックおよび／またはその近傍のブロックの復号化情報

i. 現在のブロックおよび／またはその近傍のブロックのブロック寸法／ブロック形状

e. カラーフォーマットの表示(例えば、4:2:0、4:4:4、RGB、YUV等)

f. コーディングツリー構造(例えば、デュアルツリーまたはシングルツリー)

g. スライス／タイルグループのタイプおよび／またはピクチャのタイプ

h. 色成分(例えば、輝度成分および／またはクロマ成分にのみ適用されてもよい)。

i. 時間レイヤID

j. 標準のプロファイル／レベル／ティア

【0154】

5. 追加の実施形態

本明細書では、VVC草案におけるテキストの変更を、下線を引いた太いフォントで示す。

【0155】

6. 開示される技術の例示的な実装形態

図19Aは、映像処理装置1900のブロック図である。装置1900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1900は、スマートフォン、タブレット、コンピュータ、モノのインターネット(IoT)受信機等にも実施されてもよい。装置1900は、1つ以上のプロセッサ1902と、1つ以上のメモリ1904と、映像処理ハードウェア1906と、を含んでもよい。1つまたは複数のプロセッサ1902は、本明細書に記載される1つ以上の方法を実装するように構成されてもよい。メモリ(複数可)1904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1906を使用して、ハードウェア回路において、本明細書に記載されるいくつかの技術を実装してもよく、一部又は全部がプロセッサ1902の一部(例えば、グラフィックプロセッサコアGPU又は他の信号処理回路)であってもよい。

【0156】

図21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2100を示すブロック図である。様々な実装形態は、システム2100のコンポーネントの一部又は全部を含んでもよい。システム2100は、映像コンテンツを受信するための入力ユニット2102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10ビットのマルチコンポーネント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2102は、ネットワークインターフェース、周辺バスインターフェース、又はストレージ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。

【0157】

システム2100は、本明細書に記載される様々なコーディング又は符号化方法を実装することができるコーディングコンポーネント2104を含んでもよい。コーディングコンポーネント2104は、入力ユニット2102からの映像の平均ビットレートをコーディングコンポーネント2104の出力に低減し、映像のコーディング表現を生成してもよい。従って、このコーディング技術は、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。コーディングコンポーネント2104の出力は、コンポーネント2106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2102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

10

20

30

40

50

リーム（又はコーディング）表現は、コンポーネント 2108 によって使用されて、表示インターフェースユニット 2110 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、特定の映像処理操作を「符号化」操作又はツールと呼ぶが、符号化ツール又は操作は、エンコーダで使用され、符号化の結果を逆転させる対応する復号化ツール又は操作は、デコーダで実行されることが理解されよう。

【0158】

周辺バスインターフェースまたは表示インターフェースの例は、ユニバーサルシリアルバス（USB）または高精細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。

【0159】

図 22 は、本開示の技法を利用し得る例示的な映像コーディングシステム 100 を示すブロック図である。

【0160】

図 22 に示すように、映像コーディングシステム 100 は、送信元デバイス 110 と、送信先デバイス 120 と、を備えてもよい。送信元デバイス 110 は、映像符号化デバイスとも称され得る符号化映像データを生成する。送信先デバイス 120 は、映像復号化デバイスと呼ばれ得る送信元デバイス 110 によって生成された符号化映像データを復号化し得る。

【0161】

送信元デバイス 110 は、映像ソース 112 と、映像エンコーダ 114 と、入出力（I/O）インターフェース 116 と、を備えてもよい。

【0162】

映像ソース 112 は、映像キャプチャデバイスなどのソース、映像コンテンツプロバイダからの映像データを受信するためのインターフェース、および／または映像データを生成するためのコンピュータグラフィックスシステム、またはこれらのソースの組み合わせを含んでもよい。映像データは、1 つ以上のピクチャを含んでもよい。映像エンコーダ 114 は、映像ソース 112 からの映像データを符号化し、ビットストリームを生成する。ビットストリームは、映像データのコーディング表現を形成するビットシーケンスを含んでもよい。ビットストリームは、コーディングピクチャおよび関連付けられたデータを含んでもよい。コーディングピクチャは、ピクチャのコーディング表現である。関連付けられたデータは、シーケンスパラメータセット、ピクチャパラメータセット、および他の構文構造を含んでもよい。I/O インターフェース 116 は、変復調器（モデム）および／または送信機を含んでもよい。符号化された映像データは、ネットワーク 130a を介して、I/O インターフェース 116 を介して直接送信先デバイス 120 に送信されることができる。符号化された映像データは、送信先デバイス 120 がアクセスするために、記憶媒体／サーバ 130b に記憶してもよい。

【0163】

送信先デバイス 120 は、I/O インターフェース 126、映像デコーダ 124、および表示装置 122 を含んでもよい。

【0164】

I/O インターフェース 126 は、受信機および／またはモデムを含んでもよい。I/O インターフェース 126 は、送信元デバイス 110 または記憶媒体／サーバ 130b から符号化映像データを取得してもよい。映像デコーダ 124 は、符号化された映像データを復号化してもよい。表示装置 122 は、復号化した映像データをユーザに表示してもよい。表示デバイス 122 は、送信先デバイス 120 と一体化されてもよく、または外部表

10

20

30

40

50

示デバイスとインターフェースで接続するように構成される送信先デバイス 120 の外部にあってもよい。

【0165】

映像エンコーダ 114 および映像デコーダ 124 は、高効率映像コーディング (HEVC) 規格、汎用映像コーディング (VVC) 規格、および他の現在のおよび／または更なる規格等の映像圧縮規格に従って動作してもよい。

【0166】

図 23 は、映像エンコーダ 200 の一例を示すブロック図であり、この映像エンコーダ 200 は、図 22 に示されるシステム 100 における映像エンコーダ 114 であってもよい。

【0167】

映像エンコーダ 200 は、本開示の技術のいずれか又は全部を実行するように構成されてもよい。図 23 の実施例において、映像エンコーダ 200 は、複数の機能性モジュールを含む。本開示で説明される技法は、映像エンコーダ 200 の様々なモジュール間で共有されてもよい。いくつかの例では、プロセッサは、本開示で説明される技術のいずれかまたはすべてを行うように構成してもよい。

【0168】

映像エンコーダ 200 の機能モジュールは、パーティションユニット 201 と、モード選択ユニット 203、動き推定ユニット 204、動き補償ユニット 205 及びイントラ予測ユニット 206 を含んでもよい予測ユニット 202 と、残差生成ユニット 207 と、変換ユニット 208 と、量子化ユニット 209 と、逆量子化ユニット 210 と、逆変換ユニット 211 と、再構成ユニット 212 と、バッファ 213 と、エントロピーコーディングユニット 214 とを含んでもよい。

【0169】

他の例において、映像エンコーダ 200 は、より多くの、より少ない、又は異なる機能コンポーネントを含んでもよい。一例において、予測ユニット 202 は、イントラブロックコピー (IBC) ユニットの含んでもよい。IBC ユニットの、少なくとも 1 つの参照ピクチャが現在の映像ブロックが位置するピクチャである IBC モードにおいて予測 (prediction) を行うことができる。

【0170】

さらに、動き推定ユニット 204 および動き補償ユニット 205 などのいくつかのコンポーネントは、高度に統合されてもよいが、説明のために、図 23 の例においては別個に表現されている。

【0171】

分割ユニット 201 は、1 つのピクチャを 1 つ以上の映像ブロックに分割することができる。映像エンコーダ 200 及び映像デコーダ 300 は、様々な映像ブロックサイズをサポートしてもよい。

【0172】

モード選択ユニット 203 は、例えば、エラー結果に基づいて、イントラ又はインターのいずれかのコーディングモードの 1 つを選択し、得られたイントラ又はインターコーディングブロックを、残差生成ユニット 207 に供給して残差ブロックデータを生成し、また再構成ユニット 212 に供給して参照ピクチャとして使用するために符号化ブロックを再構成してもよい。本発明の実施例において、モード選択ユニット 203 は、インター予測信号およびイントラ予測信号に基づいて予測を行うイントラおよびインター予測 (CIIP) モードの組み合わせを選択してもよい。また、モード選択ユニット 203 は、インター予測の場合、ブロックのために動きベクトルの解像度 (例えば、サブピクセル又は整数ピクセル精度) を選択してもよい。

【0173】

現在の映像ブロックに対してインター予測を実行するために、動き推定ユニット 204 は、バッファ 213 からの 1 つ以上の参照フレームと現在の映像ブロックとを比較するこ

10

20

30

40

50

とで、現在の映像ブロックのために動き情報を生成してもよい。動き補償ユニット205は、現在の映像ブロックに関連付けられたピクチャ以外のバッファ213からのピクチャの動き情報及び復号化サンプルに基づいて、現在の映像ブロックのために予測映像ブロックを判定してもよい。

【0174】

動き推定ユニット204及び動き補償ユニット205は、例えば、現在の映像ブロックがIスライスであるか、Pスライスであるか、又はBスライスであるかに基づいて、現在の映像ブロックに対して異なる演算を実行してもよい。

【0175】

いくつかの例において、動き推定ユニット204は、現在の映像ブロックに対して単方向予測を実行し、動き推定ユニット204は、現在の映像ブロックに対して、リスト0又はリスト1の参照ピクチャを検索して、参照映像ブロックを求めてもよい。そして、動き推定ユニット204は、参照映像ブロックと、現在の映像ブロックと参照映像ブロックとの間の空間的変位を示す動きベクトルとを含む、リスト0またはリスト1における参照ピクチャを示す参照インデックスを生成してもよい。動き推定ユニット204は、参照インデックス、予測方向インジケータ、および動きベクトルを、現在の映像ブロックの動き情報として出力してもよい。動き補償ユニット205は、現在の映像ブロックの動き情報が示す参照映像ブロックに基づいて、現在のブロックの予測映像ブロックを生成してもよい。

【0176】

他の例において、動き推定ユニット204は、現在の映像ブロックを双方向予測してもよく、動き推定ユニット204は、リスト0における参照ピクチャの中から現在の映像ブロックのために参照映像ブロックを検索してもよく、また、リスト1における参照ピクチャの中から現在の映像ブロックのために別の参照映像ブロックを検索してもよい。そして、動き推定ユニット204は、参照映像ブロックを含むリスト0およびリスト1における参照ピクチャを示す参照インデックスと、参照映像ブロックと現在の映像ブロックとの間の空間的変位を示す動きベクトルとを生成してもよい。動き推定ユニット204は、現在の映像ブロックの参照インデックスおよび動きベクトルを、現在の映像ブロックの動き情報として出力してもよい。動き補償ユニット205は、現在の映像ブロックの動き情報が示す参照映像ブロックに基づいて、現在の映像ブロックの予測映像ブロックを生成してもよい。

【0177】

いくつかの例において、動き推定ユニット204は、デコーダの復号化処理のために、動き情報のフルセットを出力してもよい。

【0178】

いくつかの例では、動き推定ユニット204は、現在の映像のために動き情報のフルセットを出力しなくてもよい。むしろ、動き推定ユニット204は、別の映像ブロックの動き情報を参照して、現在の映像ブロックの動き情報を信号通知してもよい。例えば、動き推定ユニット204は、現在の映像ブロックの動き情報が近傍の映像ブロックの動き情報に十分に類似していると判定してもよい。

【0179】

一例において、動き推定ユニット204は、現在の映像ブロックに関連付けられた構文構造において、現在の映像ブロックが別の映像ブロックと同じ動き情報を有することを映像デコーダ300に示す値を示してもよい。

【0180】

別の例において、動き推定ユニット204は、現在の映像ブロックに関連付けられた構文構造において、別の映像ブロックと、動きベクトル差(MVD)とを識別してもよい。動きベクトルの差分は、現在の映像ブロックの動きベクトルと、示された映像ブロックの動きベクトルとの差分を示す。映像デコーダ300は、指示された映像ブロックの動きベクトルと、動きベクトルの差分を用いて、現在の映像ブロックの動きベクトルを判定してもよい。

【0181】

上述したように、映像エンコーダ200は、動きベクトルを予測的に信号通知してもよい。映像エンコーダ200によって実装され得る予測信号通知技法の2つの例は、高度動きベクトル予測（AMVP）およびマージモード信号通知を含む。

【0182】

イントラ予測ユニット206は、現在の映像ブロックに対してイントラ予測を行ってもよい。イントラ予測ユニット206が現在の映像ブロックをイントラ予測する場合、イントラ予測ユニット206は、同じピクチャにおける他の映像ブロックの復号化されたサンプルに基づいて、現在の映像ブロックのための予測データを生成してもよい。現在の映像ブロックのための予測データは、予測された映像ブロック及び様々な構文要素を含んでもよい。

10

【0183】

残差生成ユニット207は、現在の映像ブロックから現在の映像ブロックの予測された映像ブロックを減算することによって（例えば、マイナス符号によって示されている）、現在の映像ブロックのために残差データを生成してもよい。現在の映像ブロックの残差データは、現在の映像ブロックにおけるサンプルの異なるサンプル成分に対応する残差映像ブロックを含んでもよい。

【0184】

他の例において、例えば、スキップモードにおいて、現在の映像ブロックのための残差データがなくてもよく、残差生成ユニット207は、減算演算を実行しなくてもよい。

20

【0185】

変換処理ユニット208は、現在の映像ブロックに関連付けられた残差映像ブロックに1つ以上の変換を適用することによって、現在の映像ブロックのために1つ以上の変換係数映像ブロックを生成してもよい。

【0186】

変換処理ユニット208が現在の映像ブロックに関連付けられた変換係数映像ブロックを生成した後、量子化ユニット209は、現在の映像ブロックに関連付けられた1つ以上の量子化パラメータ（QP）値に基づいて、現在の映像ブロックに関連付けられた変換係数映像ブロックを量子化してもよい。

【0187】

逆量子化ユニット210および逆変換ユニット211は、変換係数映像ブロックに逆量子化および逆変換をそれぞれ適用し、変換係数映像ブロックから残差映像ブロックを再構成してもよい。再構成ユニット212は、予測ユニット202が生成した1つ以上の予測映像ブロックから対応するサンプルに再構成された残差映像ブロックを加え、現在のブロックに関連付けられた再構成映像ブロックを生成し、バッファ213に記憶することができる。

30

【0188】

再構成ユニット212が映像ブロックを再構成した後、映像ブロックにおける映像ブロックキングアーチファクトを縮小するために、ループフィルタリング動作を行ってもよい。

【0189】

エントロピー符号化ユニット214は、映像エンコーダ200の他の機能コンポーネントからデータを受信してもよい。エントロピー符号化ユニット214は、データを受信すると、1つ以上のエントロピー符号化演算を行い、エントロピー符号化データを生成し、エントロピー符号化データを含むビットストリームを出力してもよい。

40

【0190】

図24は、映像デコーダ300の一例を示すブロック図であり、この映像デコーダ300は、図22に示すシステム100における映像デコーダ114であってもよい。

【0191】

映像デコーダ300は、本開示の技術のいずれか又は全部を実行するように構成されてもよい。図24の実施例において、映像デコーダ300は、複数の機能コンポーネントを

50

備える。本開示で説明される技法は、映像デコーダ300の様々なコンポーネント間で共有されてもよい。いくつかの例では、プロセッサは、本開示で説明される技術のいずれかまたはすべてを行うように構成してもよい。

【0192】

図24の実施例において、映像デコーダ300は、エントロピー復号化ユニット301、動き補正ユニット302、イントラ予測ユニット303、逆量子化ユニット304、逆変換ユニット305、及び再構成ユニット306、並びにバッファ307を備える。映像デコーダ300は、いくつかの例では、映像エンコーダ200（例えば、図23）に関して説明した符号化パスとほぼ逆の復号化パスを行ってもよい。

【0193】

エントロピー復号化ユニット301は、符号化ビットストリームを取り出す。符号化ビットストリームは、エントロピー符号化された映像データ（例えば、映像データの符号化ブロック）を含んでもよい。エントロピー復号化ユニット301は、エントロピー符号化された映像データを復号化し、エントロピー復号された映像データから、動き補償ユニット302は、動きベクトル、動きベクトル精度、参照ピクチャリストインデックス、および他の動き情報を含む動き情報を決定してもよい。動き補償ユニット302は、例えば、AMVP及びマージモードを実行することで、このような情報を判定してもよい。

【0194】

動き補償ユニット302は、動き補償されたブロックを生成してもよく、場合によっては、補間フィルタに基づいて補間を実行する。構文要素には、サブピクセルの精度で使用される補間フィルタのための識別子が含まれてもよい。

【0195】

動き補償ユニット302は、映像ブロックの符号化中に映像エンコーダ20によって使用されるような補間フィルタを使用して、参照ブロックのサブ整数ピクセルのための補間値を計算してもよい。動き補償ユニット302は、受信した構文情報に基づいて、映像エンコーダ200が使用する補間フィルタを決定し、この補間フィルタを使用して予測ブロックを生成してもよい。

【0196】

動き補償ユニット302は、構文情報の一部を用いて、符号化された映像シーケンスのフレーム（複数可）および／またはスライス（複数可）を符号化するために使用されるブロックのサイズ、符号化された映像シーケンスのピクチャの各マクロブロックがどのように分割されるかを記述する分割情報、各分割がどのように符号化されるかを示すモード、インター符号化ブロック間の各1つ以上の参照フレーム（および参照フレームリスト）、および符号化された映像シーケンスを復号化するための他の情報を決定してもよい。

【0197】

イントラ予測ユニット303は、例えば、ビットストリームにおいて受信したイントラ予測モードを使用して、空間的に隣接するブロックから予測ブロックを形成してもよい。逆量子化ユニット303は、ビットストリームに提供され、エントロピー復号化ユニット301によって復号された量子化された映像ブロック係数を逆量子化（すなわち、逆量子化）する。逆変換ユニット303は、逆変換を適用する。

【0198】

再構成ユニット306は、残差ブロックと、動き補償ユニット202又はイントラ予測ユニット303によって生成された対応する予測ブロックとを合計し、復号化されたブロックを形成してもよい。所望であれば、ブロックアーチファクトを除去するために、復号化されたブロックをフィルタリングするためにデブロックフィルタを適用してもよい。復号された映像ブロックはバッファ307に記憶され、バッファ307はその後の動き補償のための参照ブロックを提供する。

【0199】

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

10

20

30

40

50

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。

【0200】

本明細書では、「映像処理」という用語は、映像符号化、映像復号化、映像圧縮、または映像展開を指すことができる。例えば、映像圧縮アルゴリズムは、映像の画素表現から対応するビットストリーム表現への変換、またはその逆の変換中に適用されてもよい。現在の映像ブロックのビットストリーム表現は、例えば、構文によって規定されるように、ビットストリーム内の同じ場所または異なる場所に拡散されるビットに対応していてもよい。例えば、1つのマクロブロックは、変換およびコーディングされた誤り残差値の観点から、且つビットストリームにおけるヘッダおよび他のフィールドにおけるビットを使用して符号化されてもよい。

【0201】

開示された方法及び技法は、本明細書に開示された技法を使用できるようにすることで、スマートフォン、ノートパソコン、卓上パソコン、及び類似した機器等の映像処理デバイスに組み込まれる映像エンコーダおよび／またはデコーダの実施形態に有益となることが理解される。

【0202】

図20は、映像処理の方法2000の一例を示すフローチャートである。方法2000は、2010において、視覚メディアの映像ユニットと視覚メディアのコーディング表現との変換を実行することを含み、変換中、適応ループフィルタ（ALF）を使用する1つ以上の規則が、クロス成分適応ループフィルタ（CC-ALF）の使用に関連付けられた利用不可のサンプルをパディングするために適用される。

【0203】

いくつかの実施形態を、以下の項目に基づくフォーマットを使用して説明することができる。以下の項目は、前章に記載された技術の例示的な実施形態を示す。

【0204】

1. 視覚メディアの映像ユニットと視覚メディアのコーディング表現との変換を実行することを含み、変換中に、適応ループフィルタ（ALF）を使用する1つ以上の規則が、クロス成分適応ループフィルタ（CC-ALF）の使用に関連付けられた利用不可のサンプルをパディングするために適用される、映像処理方法。

【0205】

2. 前記映像ユニットの色成分に適用されるALFフィルタは、前記映像ユニットの色成分によって提供される情報を使用して定義される、項目1に記載の方法。

【0206】

3. 前記映像ユニットの第1の色成分に適用される前記CC-ALFフィルタは、前記映像ユニットの少なくとも第2の色成分によって提供される情報を使用して定義される、項目1～2のいずれか1つ以上に記載の方法。

【0207】

4. 前記映像ユニットは、スライス、タイル、ブリック、サブピクチャ、ピクチャまたは360度の仮想境界で囲まれた360度の仮想ピクチャに対応する、項目1～3のいずれか1つ以上に記載の方法。

【0208】

5. CC-ALFの使用に関連付けられた1つ以上のALF仮想境界において、利用

10

20

30

40

50

不可のサンプルに対してミラーリングパディングの規則を適用する、項目 1～4 のいずれか 1 つ以上に記載の方法。

【0209】

6. CC-A L F の使用に関連付けられた、スライス、タイル、ブリック、ピクチャ、サブピクチャ、または 360 度仮想境界のうちの 1 つにおける、利用不可のサンプルに対して繰り返しパディングのための規則を適用する、項目 1～5 のいずれか 1 つ以上に記載の方法。

【0210】

7. CC-A L F の使用に関連付けられた 1 つ以上の A L F 仮想境界を除き、1 つ以上の境界に繰り返しパディングのための規則が適用される、項目 1～6 のいずれか 1 つ以上に記載の方法。

10

【0211】

8. 前記 1 つ以上の境界は、すべての境界、水平方向の境界、または垂直方向の境界のうちの 1 つを含む、項目 7 に記載の方法。

【0212】

9. CC-A L F の使用に関連付けられた 1 つ以上の A L F 仮想境界を除き、1 つ以上の境界にミラーリングパディングのための規則を適用する、項目 1～6 のいずれか 1 つ以上の方法。

【0213】

10. 前記 1 つ以上の境界は、すべての境界、水平境界、または垂直境界のうちの 1 つを含む、項目 9 に記載の方法。

20

【0214】

11. 映像ユニットは、A L F 処理ユニット、コーディングツリーユニット (C T U) 、または基本 A L F 処理ユニットに対応する、項目 1～10 のいずれか 1 つ以上に記載の方法。

【0215】

12. A L F 処理ユニットは、2 つの水平境界および 2 つの垂直境界によって境界付けられるユニットを定義する、項目 11 に記載の方法。

【0216】

13. 2 つの水平境界は、2 つの A L F 仮想境界、または 1 つの A L F 仮想境界のうちの 1 つと、ピクチャ／スライス／タイル／ブリック／サブピクチャまたは 360 度仮想境界の 1 つの境界とを含む、項目 11 に記載の方法。

30

【0217】

14. 基本 A L F 処理ユニットは、A L F 処理ユニット、C T U、または狭 A L F 処理ユニットを再帰的に分割することに基づいて生成され、基本 A L F 処理ユニットの境界線に、ブリック／スライス／タイル／サブピクチャ境界線、および／または 360 度仮想境界線、および／または境界線をまたぐフィルタリングが許可されない A L F 仮想境界線で交差しない、項目 11 に記載の方法。

【0218】

15. 繰り返しパディングの規則は、映像ユニットの隣接サンプルが異なる隣接するスライス内にあるかどうかに基づく、項目 1～14 のいずれか 1 つに記載の方法。

40

【0219】

16. 映像ユニットの現在のスライスはラスタスキャンスライスであり、隣接するスライスもラスタスキャンスライスである、項目 15 に記載の方法。

【0220】

17. 現在の映像ユニットが A L F 処理ユニットであり、A L F 処理ユニットにおけるその最も近いサンプルを使用して、利用不可の隣接サンプルをパディングする、項目 15～16 のいずれか 1 つ以上に記載の方法。

【0221】

18. 映像処理方法であって、視覚メディアの映像ユニットと視覚メディアの符号化

50

表現との変換を行い、変換中に、映像ユニットの利用不可のサンプルをパディングする1つ以上の規則が、A L F 処理またはC C - A L F 処理の使用に関連付けられた予め定義されたパディング順序を規定する。

【0222】

19. 予め定義されたパディング順序は、映像ユニットの少なくとも1つの最上行を使用して利用不可のサンプルをパディングすることを含む、項目18に記載の方法。

【0223】

20. 予め定義されたパディング順序は、映像ユニットの少なくとも1つの最下行を使用して利用不可のサンプルをパディングすることを含む、項目18に記載の方法。

【0224】

21. 予め定義されたパディング順序は、映像ユニットの少なくとも1つの左列を使用して利用不可のサンプルをパディングすることを含む、項目18に記載の方法。

【0225】

22. 予め定義されたパディング順序は、映像ユニットの少なくとも1つの右列を使用して利用不可のサンプルをパディングすることを含む、項目18に記載の方法。

【0226】

23. 映像処理方法であって、視覚メディアの映像ユニットと視覚メディアの符号化表現との変換を行い、変換中に、映像ユニットの1つ以上の隣接サンプルが利用可能かどうかを決定する1つ以上の規則は、1つ以上の隣接サンプルと映像ユニットに関連付けられたパラメータとの関係を含む。

【0227】

24. 映像ユニットに関連付けられたパラメータ ($c t u D o w n$ と表される) は、 $c t u D o w n = C t b S i z e - M$ (但し、 M は整数) として算出される、項目23に記載の方法。

【0228】

25. 隣接サンプルが $c t u D o w n$ とは異なる値を有する第2の映像ユニットに含まれている場合、1つ以上の規則は、第2の映像ユニットをまたぐフィルタリングが許可されていないことを規定する、項目24に記載の方法。

【0229】

26. 1つ以上の規則は、1つ以上の隣接サンプルと映像ユニットの行数に基づく $c t u U p$ パラメータとの関係をさらに含む、項目23に記載の方法。

【0230】

27. 利用不可のサンプルが隣接サンプルである、項目18～26のいずれか1つ以上に記載の方法。

【0231】

28. 利用不可のサンプルがラスタスキャンスライスに含まれている、項目18～26のいずれか1つ以上に記載の方法。

【0232】

29. 映像ユニットに対して左上C T Uに空間的に位置する利用不可の隣接サンプルが、映像ユニットに対して上側C T Uに空間的に位置するその最も近い隣接サンプルでパディングする、項目28に記載の方法。

【0233】

30. 映像ユニットに対して右下C T Uに空間的に位置する利用不可の隣接サンプルを、映像ユニットに対して下側C T Uに空間的に位置するその最も近い隣接サンプルでパディングする、項目28に記載の方法。

【0234】

31. 映像処理方法であって、映像メディアの映像ユニットと映像メディアのコーディング表現との変換を行い、変換中に、A L F および／またはC C - A L F を適用する1つ以上の規則は、映像ユニットの境界に位置するサンプルに対するA L F および／またはC C - A L F の適用を、境界をまたぐフィルタリングが許可されていない場合、無効とす

10

20

30

40

50

ることを規定する。

【0235】

32. ALFおよび／またはCC-ALFに関連付けられた少なくともN個の隣接サンプルが利用不可と決定されると、ALFおよび／またはCC-ALFの適用が無効にされる、項目31に記載の方法。

【0236】

33. 映像ユニットの境界は水平境界であり、サンプルと映像ユニットの水平境界との垂直距離がM個のサンプル以下であると決定されると、ALFおよび／またはCC-ALFの適用が無効にされる、項目31に記載の方法。

【0237】

34. 映像ユニットの境界は垂直境界であり、サンプルと映像ユニットの垂直境界との水平距離がM個のサンプル以下であると決定されると、ALFおよび／またはCC-ALFを適用することが無効にされる、項目31に記載の方法。

【0238】

35. 映像処理方法であって、視覚メディアの第1の映像ユニットと視覚メディアのコーディング表現との変換を行い、変換中、ALFおよび／またはCC-ALFを適用する1つ以上の規則は、第1の映像ユニットとは異なる第2の映像ユニットのために、ALFおよび／またはCC-ALFを選択的にオンまたはオフにすることを規定する。

【0239】

36. 第2の映像ユニットのためにALFおよび／またはCC-ALFがオンまたはオフにされるかどうかは、第2の映像ユニットのコーディング表現において明確に示される、項目35に記載の方法。

【0240】

37. 第2の映像ユニットのためにALFおよび／またはCC-ALFがオンまたはオフにされるかどうかは、第2の映像ユニットのコーディング表現から導出される、項目35に記載の方法。

【0241】

38. 第1の映像ユニットに関連付けられたフラグが、ALFおよび／またはCC-ALFを有効にすることを示し、変換中、1つ以上の条件に基づいて、第1の映像ユニットに位置するサンプルに対してALFおよび／またはCC-ALFを適用することを選択的に無効にする、項目35～37のいずれか1つ以上に記載の方法。

【0242】

39. 1つ以上の条件は、第1の映像ユニットに対するサンプルの空間位置を含む、項目38に記載の方法。

【0243】

40. 1つ以上の条件が、CC-ALFにおいて使用される輝度サンプル値を含む、項目38に記載の方法。

【0244】

41. 1つ以上の条件が、サンプルとその隣接サンプルとのサンプル値の差を含む、項目38に記載の方法。

【0245】

42. 1つ以上の条件が、クロマサンプルに対応するフィルタリングされた輝度サンプルを使用して導出オフセットの値と閾値との関係を含む、項目38に記載の方法。

【0246】

43. 映像処理方法であって、視覚メディアの第1の映像ユニットと視覚メディアのコーディング表現との変換を行い、変換中、ALFおよび／またはCC-ALFを選択的に適用する1つ以上の規則は、3つのクリッピング演算のうちの少なくとも1つを省略することを規定し、第1のクリッピング演算はクロマALFフィルタリングに対応し、第2のクリッピング演算はCC-ALFオフセット導出に対応し、第3のクリッピング演算は最終的なクロマサンプル値を導出するためのクロマフィルタサンプルの微調整に対応する。

10

20

30

40

50

【0247】

44. ALFおよび／またはCC-ALFを適用する1つ以上の規則は、CC-ALFを適用することが有効である場合、第1のクリッピング動作を省略することを規定する、項目43に記載の方法。

【0248】

45. ALFおよび／またはCC-ALFを適用する1つ以上の規則は、CC-ALFを適用することが無効である場合、第2のクリッピング演算を省略することを規定する、項目43に記載の方法。

【0249】

46. ALFおよび／またはCC-ALFを適用する1つ以上の規則は、第1のクリッピング演算および第2のクリッピング演算を省略することを規定する、項目43に記載の方法。

10

【0250】

47. 第2のクリッピング演算中、導出CC-ALFオフセットは、 $[-(1 < (\text{BitDepthC} - 1)) , (1 < (\text{BitDepthC} - 1)) - 1]$ として表される第2の範囲とは異なる第1の範囲にあり、第2の範囲の端点が第2の範囲に含まれ、 BitDepthC がビット深度値である、項目43に記載の方法。

【0251】

48. 第1の範囲は、ビット深度値への依存性を欠く、項目47に記載の方法。

【0252】

49. 第2の映像ユニットに関連付けられた第3の範囲が、第1の映像ユニットに関連付けられた第1の範囲とは異なるように定義される、項目47に記載の方法。

20

【0253】

50. 第1の範囲は、クロマALFフィルタリングに対応する第1のクリッピング演算に使用される非線形クリッピング範囲から導出される、項目47に記載の方法。

【0254】

51. 第2のクリッピング演算の間、固定値の代わりにビット深度値を使用して、導出CC-ALFオフセットを丸める、項目47に記載の方法。

【0255】

52. 第2のクリッピング演算中に、導出CC-ALFオフセットは、以下の式を用いて丸められる、項目47に記載の方法。

30

$$\text{sum} = \text{alfPicture}[x \text{Ct}bC + x][y \text{Ct}bC + y] + (\text{sum} + \text{Off}) >> (7 + (\text{BitDepthY} - \text{BitDepthC}))$$

$$\text{ccAlfPicture}[x \text{Ct}bC + x][y \text{Ct}bC + y] = \text{Clip3}(0, (1 < \text{BitDepthC}) - 1, \text{sum})$$
において、Offは、 $(1 < (6 + (\text{BitDepthY} - \text{BitDepthC})))$ に等しく設定され、 $\text{alfPicture}[x \text{Ct}bC + x][y \text{Ct}bC + y]$ は、第1のクリッピング演算を適用した後のクロマサンプルを表し、sumは、第2のクリッピング演算を適用した後のCC-ALFフィルタサポートの輝度サンプルにCC-ALFフィルタ係数を掛けた合計を表し、ccAlfPictureは第3のクリッピング演算を適用した後の最終的に微調整されたクロマサンプルを表す。

40

【0256】

53. 映像処理方法であって、視覚メディアの映像ユニットと視覚メディアの符号化表現との変換を行い、変換中、映像ユニットに対してクロマALF処理および／またはCC-ALF処理を共同して最適化する。

【0257】

54. 変換は、予め定義された範囲内にあるクロマサンプルをもたらすクロマALF処理および／またはCC-ALF処理によって生成されたクロマサンプルをクリッピングするためのクリッピング演算を含む、項目53に記載の方法。

【0258】

55. 予め定義された範囲は、クロマビット深度値に少なくとも部分的に基づいてい

50

る、項目 5 4 に記載の方法。

【0259】

56. 共同最適化は、CC-A L F 処理がクロマ A L F 処理の後に適用される場合、CC-A L F 処理中にのみクリッピング演算を適用することを含む、項目 5 4 に記載の方法。

【0260】

57. 共同最適化は、クロマ A L F 処理の前に CC-A L F 処理が適用される場合、クロマ A L F 処理中にのみクリッピング演算を適用することを含む、項目 5 4 に記載の方法。

【0261】

58. 映像処理方法であって、視覚メディアの映像ユニットと視覚メディアのコーディング表現との変換を行い、変換中に、映像ユニットの第 2 の色成分の情報を用いて、映像ユニットの第 1 の色成分のサンプル値を修正し、映像ユニットの第 2 の色成分の情報は、適応ループフィルタリング処理に用いられる 1 つ以上のパラメータに基づく。

【0262】

59. 適応ループフィルタリング処理で使用される 1 つ以上のパラメータは、クリッピングパラメータであり、第 1 のサンプル値に使用されるクリッピングパラメータは、第 2 のサンプル値に使用されるクリッピングパラメータとは異なり、第 1 のサンプル値および第 2 のサンプル値は映像ユニット中に含まれる、項目 5 8 に記載の方法。

【0263】

60. クリッピングパラメータは、最小クリッピング値および最大クリッピング値を有する範囲内にあるように定義される、項目 5 9 に記載の方法。

【0264】

61. 最小クリッピング値および／または最大クリッピング値は、コーディング表現において明確に信号通知される、項目 6 0 に記載の方法。

【0265】

62. 最小クリッピング値および／または最大クリッピング値は、サンプル値に部分的に基づいている、項目 6 0 に記載の方法。

【0266】

63. 最小クリッピング値および／または最大クリッピング値は、映像ユニットに対するサンプル値の空間位置に部分的に基づいている、項目 6 0 に記載の方法。

【0267】

64. 最小クリッピング値および／または最大クリッピング値は、予め定義された値である、項目 6 0 に記載の方法。

【0268】

65. 最小クリッピング値および／または最大クリッピング値は、他のパラメータから導出される、項目 6 0 に記載の方法。

【0269】

66. 映像処理方法であって、視覚メディアの第 1 の映像ユニットと視覚メディアの符号化表現との変換を行い、変換中に、第 1 の映像ユニットの第 1 のサンプルに対する CC-A L F 処理の適用は、視覚メディアの第 2 の映像ユニットの第 2 のサンプルに基づいて選択的に行われ、選択的に基づくことは、映像ユニットをまたぐループフィルタが有効であるか無効であるかの決定を含む。

【0270】

67. 第 1 の映像ユニットの第 1 のサンプルに対する CC-A L F 処理の適用は、第 1 の映像ユニットおよび／または第 2 の映像ユニットに対して映像ユニットをまたぐループフィルタが無効であるという決定に基づいてスキップされる、項目 6 6 に記載の方法。

【0271】

68. 第 1 の映像ユニットの第 1 のサンプルに対する CC-A L F 処理の適用は、映像ユニットをまたぐループフィルタは第 1 の映像ユニットに対して無効であるとの決定に

10

20

30

40

50

基づいてスキップされる、項目 6 6 に記載の方法。

【0272】

69. 第1の映像ユニットの第1のサンプルに対するCC-A LF処理の適用は、複数の映像ユニットをまたぐループフィルタは第2の映像ユニットに対して無効であるとの決定に基づいてスキップされる、項目 6 6 に記載の方法。

【0273】

70. 第1の映像ユニットおよび／または第2の映像ユニットに関連付けられたフラグを読み取ることに基づいて、映像ユニットをまたぐループフィルタが有効であるかまたは無効であるかを決定する、項目 6 6 ～ 6 9 のいずれか1つ以上に記載の方法。

【0274】

71. CC-A LF処理および／または前記A LF処理が、i. 映像ユニットのコンテンツ、ii. DPS/SPS/VPS/PPS/APS/ピクチャヘッダ/スライスヘッダ/タイルグループヘッダ/最大コーディングユニット(LCU)/コーディングユニット(CU)/LCU行/LCU群/TU/PUブロック/映像コーディングユニットで信号通知されたメッセージ、iii. CU/PU/TU/ブロック/映像コーディングユニットの位置、iv. 映像ユニットの形状または寸法および／または隣接する映像ユニットの形状または寸法、vii. カラーフォーマットの表示、viii. コーディングツリー構造、ix. スライス/タイルグループタイプおよび／またはピクチャタイプ、x. 映像ユニットの色成分、xi. 時間層IDまたはxii. 規格のプロファイル/レベル/ティア、の少なくとも1つに関連付けられる、項目1～70のいずれか1つ以上に記載の方法。

【0275】

72. 項目1～71の1つ以上に記載の方法を実装するように構成されたプロセッサを備える、映像復号化装置。

【0276】

73. 項目1～71の1つ以上に記載の方法を実装するように構成されたプロセッサを備える映像符号化装置。

【0277】

74. コンピュータコードが記憶されたコンピュータプログラム製品において、前記のコードがプロセッサにより実行されると、前記プロセッサは、項目1～71のいずれかに記載の方法を実装する。

【0278】

75. 本明細書に記載の方法、装置またはシステム。

【0279】

第2の節は、前節で論じた技法の例示的な実施例を示す。

【0280】

1. 1つ以上の映像ブロックからなる映像の現在の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、現在の映像ユニットの少なくともいくつかの映像ブロックに対してクロス成分適応ループフィルタリング(CC-A LF)を適用する間に、利用不可のサンプルをパディングするために用いられるパディング処理を決定すること(2512)、およびその決定に基づき変換を行うこと(2514)を含み、規則は、パディング処理が、現在の映像ユニットの1つ以上の映像ブロックに適応ループフィルタリング(ALF)を適用する間に利用不可のサンプルをパディングするためにも使用されることを規定する、映像処理方法(例えば、図25Aに示す方法2510)。

【0281】

2. ALFツールは、現在の映像ユニットの色成分に適用され、ALFツールは、現在の映像ユニットの色成分によって提供される情報を使用して、色成分のサンプル値を補正することを含む、項目1に記載の方法。

【0282】

3. CC-A LFツールは、現在の映像ユニットの第1の色成分に適用され、現在の

10

20

30

40

50

映像ユニットの第2の成分のサンプル値を使用して、現在の映像ユニットの第1の成分のサンプル値を補正することを含む、項目1～2のいずれか1つ以上に記載の方法。

【0283】

4. 現在の映像ユニットは、360度の仮想境界で囲まれたスライス、タイル、ブリック、サブピクチャ、ピクチャまたは360度の仮想ピクチャに対応する、項目1～3のいずれか1つ以上に記載の方法。

【0284】

5. パディング処理は、CC-A LFツールの使用に関連付けられた1つ以上のA LF仮想境界において、利用不可のサンプルに適用されるミラーリングパディングに対応する、項目1～4のいずれか1つに記載の方法。

10

【0285】

6. パディング処理は、CC-A LFの使用に関連付けられた、スライス、タイル、ブリック、ピクチャ、サブピクチャ、または360度仮想境界のうちの1つにおいて利用不可のサンプルに適用される繰り返しパディングに対応する、項目1～4のいずれか1つ以上に記載の方法。

【0286】

7. パディング処理は、CC-A LFの使用に関連付けられた1つ以上のA LF仮想境界を除き、1つ以上の境界に適用される繰り返しパディングに対応する、項目1～4のいずれか1つ以上に記載の方法。

【0287】

8. 前記1つ以上の境界は、すべての境界、水平方向の境界、または垂直方向の境界のうちの1つを含む、項目7に記載の方法。

20

【0288】

9. パディング処理は、CC-A LFの使用に関連付けられた1つ以上のA LF仮想境界を除き、1つ以上の境界にミラーリングパディングを適用することに対応する、項目1～4のいずれか1つ以上に記載の方法。

【0289】

10. 前記1つ以上の境界は、すべての境界、水平境界、または垂直境界のうちの1つを含む、項目9に記載の方法。

【0290】

11. 現在の映像ユニットは、A LF処理ユニット、コーディングツリーユニット（CTU）、または基本A LF処理ユニットに対応する、項目1～10のいずれか1つ以上に記載の方法。

30

【0291】

12. A LF処理ユニットは、2つの水平境界および2つの垂直境界によって境界付けられるユニットを定義する、項目11に記載の方法。

【0292】

13. 2つの水平境界は、2つのA LF仮想境界、または1つのA LF仮想境界うちの1つと、ピクチャ／スライス／タイル／ブリック／サブピクチャまたは360度仮想境界の1つの境界とを含む、項目11に記載の方法。

40

【0293】

14. 基本A LF処理ユニットは、A LF処理ユニット、CTU、または狭A LF処理ユニットを再帰的に分割することに基づいて生成され、基本A LF処理ユニットの境界線に、ブリック／スライス／タイル／サブピクチャ境界線、および／または360度仮想境界線、および／または境界線をまたぐフィルタリングが許可されないA LF仮想境界線で交差しない、項目11に記載の方法。

【0294】

15. パディング処理は、現在の映像ユニットの隣接サンプルが異なる隣接スライス内にあるかどうかに基づく繰り返しパディングに対応する、項目1～4のいずれか1つ以上に記載の方法。

50

【0295】

16. 現在の映像ユニットの現在のスライスがラストスキャンスライスであり、現在のスライスの隣接するスライスもラストスキャンスライスである、項目15に記載の方法。

【0296】

17. 現在の映像ユニットがALF処理ユニットであり、ALF処理ユニットにおけるその最も近いサンプルを使用して利用不可のサンプルをパディングする、項目15～16のいずれか1つ以上に記載の方法。

【0297】

18. 映像の映像ユニットと映像のビットストリーム表現との変換を行うこと(2522)を含み、変換中、適応ループフィルタリング(ALF)処理またはクロス成分適応ループフィルタリング(CC-ALF)処理の適用において、規則に従って、映像ユニットの利用不可のサンプルを予め定義されたパディング順序でパディングする、映像処理方法(例えば、図25Bに示す方法2520)。

10

【0298】

19. 映像ユニットは処理ユニットに対応し、規則は、映像ユニットの利用不可のサンプルをパディングするための予め定義されたパディング順序を規定する、項目18に記載の方法。

【0299】

20. 予め定義されたパディング順序は、映像ユニットの少なくとも1つの行または少なくとも1つの列を使用して、利用不可のサンプルをパディングすることを含み、少なくとも1つの行または少なくとも1つの列の位置が、利用不可サンプルの位置に基づいて決定される、項目18に記載の方法。

20

【0300】

21. 予め定義されたパディング順序は、映像ユニットの少なくとも1つのサンプルを使用して利用不可のサンプルをパディングすることを含み、少なくとも1つのサンプルの位置が、利用不可のサンプルの位置に基づいて決定される、項目18に記載の方法。

【0301】

22. 映像ユニットの少なくとも1つの隣接サンプルの可用性は、1つ以上の隣接サンプルと映像領域に関連付けられたパラメータ(ctuDownまたはctuUp)との関係に基づいて決定され、映像領域は、1つのコーディングツリーユニット(CTU)のN個の行に対応するctuUpと、CTUの「CtbSize-M」行に対応するctuDownとを含む処理ユニットに対応し、このときMは整数である、項目18に記載の方法。

30

【0302】

23. 隣接サンプルがctuDownと異なる第2の映像ユニットにある場合、この規則は、第2の映像ユニットをまたぐフィルタリングを許可しないことを規定する、項目22に記載の方法。

【0303】

24. 映像ユニットは、ラスト走査スライスに含まれる現在のコーディングツリーユニットに対応し、規則は、映像ユニットの利用不可のサンプルをパディングするための予め定義されたパディング順序を規定する、項目18に記載の方法。

40

【0304】

25. 予め定義されたパディング順序は、ラストスキャンスライス内の上側CTUに位置するその最も近い隣接サンプルを使用してラストスキャンスライスとは異なるスライス内の左上CTUに位置する利用不可のサンプルをパディングすることを含む、項目18に記載の方法。

【0305】

26. 予め定義されたパディング順序は、ラストスキャンスライス内の左側CTUに位置するその最も近い隣接サンプルを使用してラストスキャンスライスとは異なるスライス内の左上CTUに位置する利用不可のサンプルをパディングすることを含む、項目18

50

に記載の方法。

【0306】

27. 予め定義されたパディング順序は、ラスタスキャンスライスとは異なるスライス内の右下CTUに位置する利用不可のサンプルを、ラスタスキャンスライス内の下側CTUに位置するその最も近い隣接サンプルを使用してパディングすることを含む、項目18に記載の方法。

【0307】

28. 予め定義されたパディング順序は、ラスタスキャンスライスとは異なるスライス内の右下CTUに位置する利用不可のサンプルを、ラスタスキャンスライス内の右側CTUに位置するその最も近い隣接サンプルを使用してパディングすることを含む、項目18に記載の方法。

10

【0308】

29. 適応ループフィルタ（ALF）の適用が有効にされた映像の映像領域について、その映像領域が映像ユニット境界と交差しているかを決定すること（2532）と、映像とその映像のビットストリーム表現との変換を行うこと（2534）を含み、変換のために、映像領域が映像ユニットの境界と交差していることに起因し、映像領域は規則に従って複数の区分に分割される、映像処理方法（例えば、図25Cに示す方法2530）。

【0309】

30. 境界をまたぐフィルタリング処理は許可されていない、項目29に記載の方法。

【0310】

31. 映像領域は、ALF処理ユニット、狭ALF処理ユニット、またはコーディングツリーユニットに対応する、項目29に記載の方法。

20

【0311】

32. 映像ユニットは、ブリック、スライス、タイル、サブピクチャ、または360度仮想ピクチャに対応する、項目29に記載の方法。

【0312】

33. 規則は、映像領域が映像ユニットの境界をまたがなくなるまで、映像領域を再帰的に分割することをさらに規定する、項目29に記載の方法。

【0313】

34. フィルタリング処理は、ALF処理またはCC-ALF処理に対応し、規則はさらに、映像領域の分割処理の完了後にフィルタリング処理を行うことを規定する、項目29又は30に記載の方法。

30

【0314】

35. 少なくとも1つのサンプルをパディングするために使用されるパディング処理は、映像領域が映像ユニットの境界をまたがなくなるまで、映像領域を分割した後に取得される基本ALF処理ユニットに対して行われる、項目29に記載の方法。

【0315】

36. 規則は、所定の順序で異なる水平境界をチェックし、映像領域が映像ユニットの境界をまたがなくなるまで映像領域を分割した後に得られる基本ALF処理ユニットを生成することを規定する、項目29に記載の方法。

40

【0316】

37. 予め定義された順序は、ALF仮想境界、360度仮想境界、および映像ユニットの境界であるコーディングツリーユニット（CTU）境界である、項目36に記載の方法。

【0317】

38. 予め定義された順序は、360度仮想境界、ALF仮想境界、および映像ユニットの境界であるコーディングツリーユニット（CTU）境界である、項目36に記載の方法。

【0318】

39. 規則は、映像ユニットの最上境界および最下境界に対して、予め定義された異

50

なる順序を規定する、項目 36 に記載の方法。

【0319】

40. 項目 29 に記載の方法において、規則は、所定の順序で異なる垂直境界をチェックし、映像領域が映像ユニットの境界をまたがなくなるまで映像領域を分割した後に得られる基本 ALF 処理ユニットを生成することを規定する。

【0320】

41. 予め定義された順序は、360 度仮想境界および映像ユニットの境界であるコーディングツリーユニット (CTU) 境界である、項目 40 に記載の方法。

【0321】

42. 予め定義された順序は、映像ユニットのコーディングツリーユニット (CTU) 境界および 360 度仮想境界である、項目 40 に記載の方法。

【0322】

43. 規則は、映像ユニットの左境界および右境界に対して予め定義された異なる順序を規定する、項目 40 に記載の方法。

【0323】

44. 規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、映像ユニットの境界に位置するサンプルに対する適応ループフィルタリング (ALF) および／またはクロス成分適応ループフィルタリング (CC-ALF) の適用は、境界をまたぐフィルタリング処理が許可されていない場合、許可されないことを規定する、映像処理方法。

【0324】

45. ALF および／または CC-ALF に関連付けられた少なくとも N 個の隣接サンプルが利用不可と決定されると、ALF および／または CC-ALF の適用が無効にされ、このとき N は整数である、項目 44 に記載の方法。

【0325】

46. 映像ユニットの境界は水平境界であり、サンプルと映像ユニットの水平境界との垂直距離が M 個のサンプル以下であると決定されると、ALF および／または CC-ALF を適用することが無効にされ、このとき M は整数である、項目 44 に記載の方法。

【0326】

47. 映像ユニットの境界は垂直境界であり、サンプルと映像ユニットの垂直境界との水平距離が M 個のサンプル以下であると決定されると、ALF および／または CC-ALF を適用することが無効にされ、このとき M は整数である、項目 44 に記載の方法。

【0327】

48. 映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、ビットストリーム表現はフォーマット規則に準拠し、映像領域はコーディングツリーブロックとは異なり、フォーマット規則は適応ループフィルタリング (ALF) ツールおよび／またはクロス成分適応ループフィルタリング (CC-ALF) ツールの映像領域に対する適用可能性を示すビットストリーム表現に構文要素が含まれるかどうかを規定する、映像処理方法。

【0328】

49. 映像領域は、ALF 処理ユニットに対応する、項目 48 に記載の方法。

【0329】

50. ALF ツールおよび／または CC-ALF ツールの適用可能性が、ビットストリーム表現において明確に示される、項目 48 に記載の方法。

【0330】

51. ALF ツールおよび／または CC-ALF ツールの適用可能性が導出される、項目 48 に記載の方法。

【0331】

52. 映像の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、クロス成分適応ループフィルタリング (CC-ALF) ツールを映像ユニットの

10

20

30

40

50

サンプルに適用可能であることを決定することと（2542）、決定に基づいて変換を行うこと（2544）を含み、ビットストリーム表現は、CC-A L Fが映像ユニットに利用可能である指示を含み、規則は指示をオーバーライドする1つ以上の状態を規定する、映像処理方法（例えば、図25Dに示す方法2540）。

【0332】

53. 1つ以上の条件が、映像ユニットに対するサンプルの空間位置を含む、項目52に記載の方法。

【0333】

54. 1つ以上の条件が、CC-A L Fにおいて使用される輝度サンプル値を含む、項目52に記載の方法。

【0334】

55. 1つ以上の条件が、サンプルとその隣接サンプルとのサンプル値の差を含む、項目52に記載の方法。

【0335】

56. 1つ以上の条件が、クロマサンプルに対応するフィルタリングされた輝度サンプルを使用して導出されたオフセットの値と閾値との関係を含む、項目52に記載の方法。

【0336】

57. 1つ以上の条件が、クロマサンプルを含むブロックおよび／またはブロックの隣接ブロックのコード化情報を含む、項目52に記載の方法。

【0337】

58. 規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、変換の際に使用される演算が、クロマ適応ループフィルタリング（A L F）に対応する第1のクリッピング演算、クロス成分適応ループフィルタリング（CC-A L F）オフセット導出に対応する第2のクリッピング演算、および最終的なクロマサンプル値を導出するためのクロマフィルタリングされたサンプルの微調整に対応する第3のクリッピング演算を含む3つのクリッピング演算のうちの少なくとも1つを省略することを規定する、映像処理方法。

【0338】

59. 規則は、CC-A L Fを適用することが有効である場合、第1のクリッピング演算を省略することを規定する、項目58に記載の方法。

【0339】

60. 規則は、CC-A L Fを適用することが無効である場合、第2のクリッピング演算を省略することを規定する、項目58に記載の方法。

【0340】

61. 規則は、第1のクリッピング演算および第2のクリッピング演算を省略することを規定する、項目58に記載の方法。

【0341】

62. 映像の第1の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、クロス成分適応ループフィルタリング（CC-A L F）オフセットを決定すること（2552）と、この決定に基づいて変換を行うこと（2554）とを含み、この規則は、CC-A L Fオフセットを、 $[-(1 << (BitDepthC - 1)) , (1 << (BitDepthC - 1)) - 1]$ と表される第2の範囲と異なる第1の範囲にクリッピングすることを規定し、 $BitDepthC$ はビット深度値である、映像処理方法（例えば、図25Eに示す方法2550）。

【0342】

63. 第1の範囲は、ビット深度値への依存性を欠く、項目62に記載の方法。

【0343】

64. 第2の映像ユニットに関連付けられた第3の範囲が、第1の映像ユニットに関連付けられた第1の範囲とは異なるように定義される、項目62に記載の方法。

【0344】

10

20

30

40

50

65. 第1の範囲は、クロマA L Fフィルタリングに対応する別のクリッピング演算に使用される非線形クリッピング範囲から導出される、項目62に記載の方法。

【0345】

66. 映像の第1の映像ユニットと映像のビットストリーム表現との変換のために、規則に従って、1つのクロス成分適応ループフィルタリング(C C - A L F)オフセットを導出すること(2562)と、C C - A L Fオフセットを使用してこの変換を行うこと(2564)とを含み、規則は、C C - A L Fオフセットを固定値の代わりにビット深度値に基づいて丸めオフセットを用いて丸めることを規定する、映像処理の方法(例えば、図25Fに示す方法2560)。

【0346】

67. 丸めオフセットは、第1の色成分と第2の色成分とのビット深度差に依存する、項目66に記載の方法。

【0347】

68. C C - A L Fオフセットは、次の式を用いて丸められる：

$$\text{sum} = \text{alfPicture}[x_{\text{CtbC}} + x][y_{\text{CtbC}} + y] + (\text{sum} + \text{Off}) \gg (7 + (\text{BitDepthY} - \text{BitDepthC}))$$
 および
$$\text{ccAlfPicture}[x_{\text{CtbC}} + x][y_{\text{CtbC}} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum})$$
 であり、ここでOffは、 $(1 \ll (6 + (\text{BitDepthY} - \text{BitDepthC})))$ に等しく設定され、 $\text{alfPicture}[x_{\text{CtbC}} + x][y_{\text{CtbC}} + y]$ は、第1のクリッピング演算を適用した後のクロマサンプルを表し、 sum は、第2のクリッピング演算を適用した後のC C - A L Fフィルタサポートの輝度サンプルにC C - A L Fフィルタ係数を掛けた合計を表し、 ccAlfPicture は第3のクリッピング演算を適用した後の最終的に微調整されたクロマサンプルを表し、 BitDepthC と BitDepthY はそれぞれ輝度成分とクロマ成分のビット深度を表す、項目66に記載の方法。

【0348】

69. 規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、映像ユニットのサンプルに適用されるクロマ適応ループフィルタリング(A L F)処理および／またはクロス成分適応ループフィルタリング(C C - A L F)処理中に使用される1つ以上の処理ステップが同じであることを規定する、映像処理方法。

【0349】

70. 変換は、予め定義された範囲内にあるクロマサンプルをもたらしクロマA L F処理および／またはC C - A L F処理によって生成されたクロマサンプルをクリッピングするためのクリッピング演算を含む、項目69に記載の方法。

【0350】

71. 予め定義された範囲は、クロマビット深度値に少なくとも部分的に基づいている、項目70に記載の方法。

【0351】

72. C C - A L F処理がクロマA L F処理の後に適用される場合、クリッピング演算はC C - A L F処理中にのみ適用される、項目70に記載の方法。

【0352】

73. C C - A L F処理がクロマA L F処理の前に適用される場合、クリッピング演算はクロマA L F処理中にのみ適用される、項目70に記載の方法。

【0353】

74. 規則に従って、映像の映像ユニットと映像のビットストリーム表現との変換を行うことを含み、規則は、変換中に、映像ユニットの第2の色成分の情報を使用して修正を適用することによって、映像ユニットの第1の色成分のサンプルの値を修正することを規定し、修正は、映像ユニットのための適応ループフィルタリング(A L F)処理で使用される1つ以上のパラメータに基づく、映像処理方法。

10

20

30

40

50

【0354】

75. 適応ループフィルタリング処理で使用される1つ以上のパラメータは、クリッピングパラメータであり、第1のサンプル値に使用されるクリッピングパラメータは、第2のサンプル値に使用されるクリッピングパラメータとは異なり、第1のサンプル値および第2のサンプル値は映像ユニット中に含まれている、項目74に記載の方法。

【0355】

76. クリッピングパラメータは、最小クリッピング値および最大クリッピング値を有する範囲内にあるように定義される、項目75に記載の方法。

【0356】

77. 最小クリッピング値および／または最大クリッピング値は、ビットストリーム表現において明確に信号通知される、項目76に記載の方法。

10

【0357】

78. 最小クリッピング値および／または最大クリッピング値は、サンプルの値に部分的に基づいている、項目76に記載の方法。

【0358】

79. 最小クリッピング値および／または最大クリッピング値は、映像ユニットに対するサンプルの値の空間的位置に部分的に基づいている、項目76に記載の方法。

【0359】

80. 最小クリッピング値および／または最大クリッピング値は、信号通知されるかまたは予め定義される、項目76に記載の方法。

20

【0360】

81. 最小クリッピング値および／または最大クリッピング値は、他のパラメータから導出される、項目76に記載の方法。

【0361】

82. 映像の第1のサブピクチャと映像のビットストリーム表現との変換のために、規則に従って、クロス成分ループフィルタリング(CC-ALF)が第1のサブピクチャのサンプルに適用可能であるかどうかを決定すること(2572)と、この決定に基づいた変換を行うこと(2574)を含み、サンプルのためのCC-ALFは第2のサブピクチャからのサンプルを使用し、規則は第1のサブピクチャおよび／または第2のサブピクチャに対して、サブピクチャの境界をまたぐループフィルタリングが許可されているかどうかに基づく、映像処理方法(例えば、図25Gに示す方法2570)。

30

【0362】

83. サブピクチャ境界をまたぐループフィルタリングが第1のサブピクチャに対して許可されない場合、サンプルに対してCC-ALFをスキップする、項目82に記載の方法。

【0363】

84. サブピクチャ境界をまたぐループフィルタリングが第2のサブピクチャに対して許可されない場合、第1のサンプルに対してCC-ALFをスキップする、項目82に記載の方法。

【0364】

85. 第1のサブピクチャおよび／または第2のサブピクチャのためにサブピクチャ境界をまたぐループフィルタリングが許可されない場合、第1のサンプルに対してCC-ALFをスキップする、項目82の方法。

40

【0365】

86. サブピクチャ境界をまたぐループフィルタリングが許可されるかどうかは、第1のサブピクチャおよび／または第2のサブピクチャに関連付けられたフラグに基づく、項目82～84のいずれかの1つ以上に記載の方法。

【0366】

87. 前項のいずれか1つに記載の方法であって、方法は、さらに、以下のうちの少なくとも1つに基づく1)映像コンテンツのタイプ、2)シーケンスパラメータセット(S

50

P S)、映像パラメータセット(V P S)、ピクチャパラメータセット(P P S)、依存性パラメータセット(D P S)、適応パラメータセット(A P S)、ピクチャヘッダ、スライスヘッダ、タイルグループヘッダ、最大コーディングユニット(L C U)、コーディングユニット(C U)、L C U行、L C Uグループ、変換ユニット(T U)、予測ユニット(P U)ブロック、または映像コーディングユニットで信号通知されたメッセージ、3) C U、P U、T U、ブロック、または映像コーディングユニットの位置、4) 現在のブロックおよび／または隣接ブロックの復号化情報、5) 現在のブロックおよび／または隣接ブロックの寸法または形状、6) 色フォーマットの指示、7) コーディングツリー構造、8) スライスタイプ、タイルグループタイプ、および／またはピクチャタイプ、9) 色成分のタイプ、10) 時間層識別子、11) 標準のプロファイル／レベル／ティア。

10

【0367】

88. 変換は、前記映像を前記ビットストリーム表現に符号化することを含む、項目1～87のいずれかに記載の方法。

【0368】

89. 変換は、ビットストリーム表現から映像を復号することを含む、項目1～87のいずれか1つに記載の方法。

【0369】

90. 項目1から89のいずれか1つまたは複数に記載された方法を実施するように構成されたプロセッサを含む映像処理装置。

【0370】

91. 実行されると、項目1から89までのいずれか1つ以上に記載された方法をプロセッサに実施させるプログラムコードを格納したコンピュータ可読媒体。

20

【0371】

92. 上述した方法のいずれかに従って生成されたコーディング表現またはビットストリーム表現を記憶する、コンピュータ可読媒体。

【0372】

本明細書に記載された開示された、およびその他の解決策、実施例、実施形態、モジュール、および機能動作の実装形態は、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、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。開示された、およびその他の実施形態は、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装されるため、またはデータ処理装置の動作を制御するために、コンピュータ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施することができる。このコンピュータ可読媒体は、機械可読記憶デバイス、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、またはこれらの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、または複数のプロセッサ、若しくはコンピュータを含む、データを処理するためのすべての装置、デバイス、および機械を含む。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、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはこれらの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。伝播信号は、人工的に生成した信号、例えば、機械で生成した電気、光、または電磁信号であり、適切な受信装置に送信するための情報を符号化するために生成される。

30

【0373】

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、コンポーネント、サブルーチン、または他のユニットとして含む任意の形式で展

40

50

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。

【0374】

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プロセッサ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。

【0375】

コンピュータプログラムの実行に適したプロセッサは、例えば、汎用および専用マイクロプロセッサの両方、並びに任意の種類のデジタルコンピュータの任意の1つ以上のプロセッサを含む。一般的に、プロセッサは、リードオンリーメモリまたはランダムアクセスメモリまたはその両方から命令およびデータを受信する。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つ以上のメモリデバイスとである。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、フラッシュメモリ装置、磁気ディスク、例えば内部ハードディスクまたはリムーバブルディスク、光磁気ディスク、およびCD-ROMおよびDVD-ROMディスク等の半導体記憶装置を含む。プロセッサおよびメモリは、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。

【0376】

本特許明細書は多くの詳細を含むが、これらは、任意の主題の範囲また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、特定の技術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コンテキストで説明されている特定の技術的特徴は、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つ以上の技術的特徴は、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。

【0377】

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。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。

10

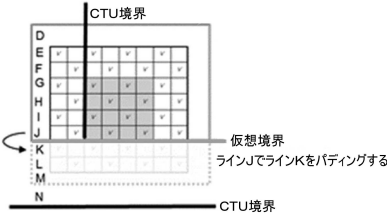
20

30

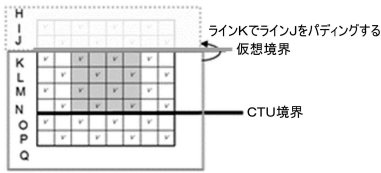
40

50

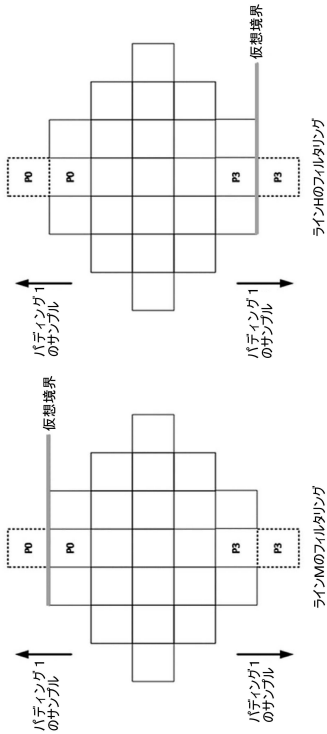
【図 5 A】



【図 5 B】

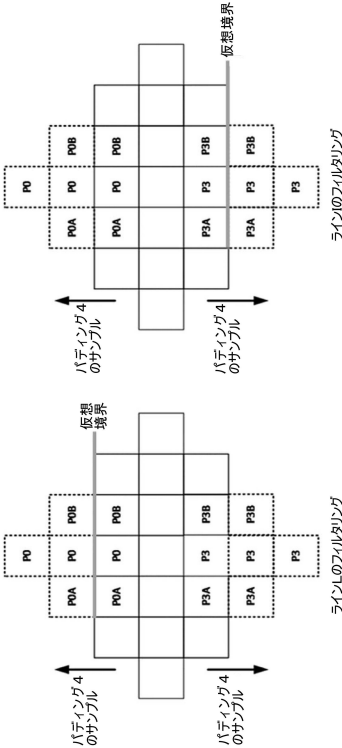


【図 6 A】



仮想境界における修正輝度ALF (7 × 7 菱形フィルタサポート、中心はフィルタリングされる現在のサンプル)

【図 6 B】



仮想境界における修正輝度ALF (7 × 7 菱形フィルタサポート、中心はフィルタリングされる現在のサンプル)

10

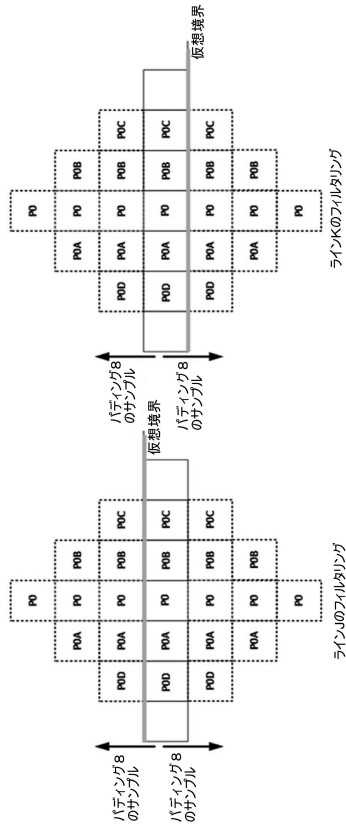
20

30

40

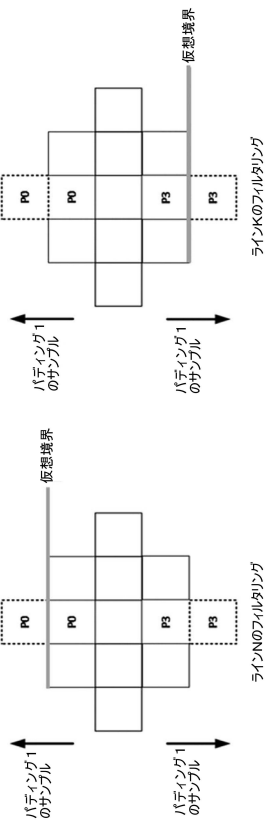
50

【図 6 C】



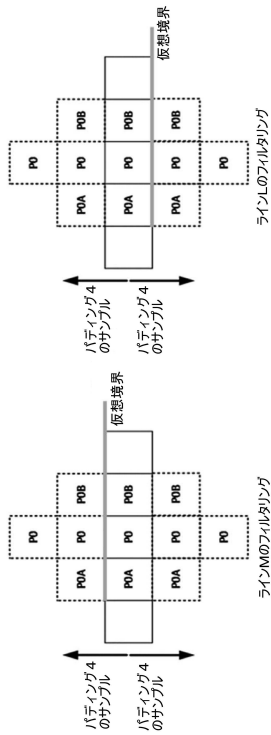
仮想境界での修正補度ALFフィルタリング(7×7菱形フィルタサポート、中心はフィルタリングされる現在のサンプル)

【図 7 A】



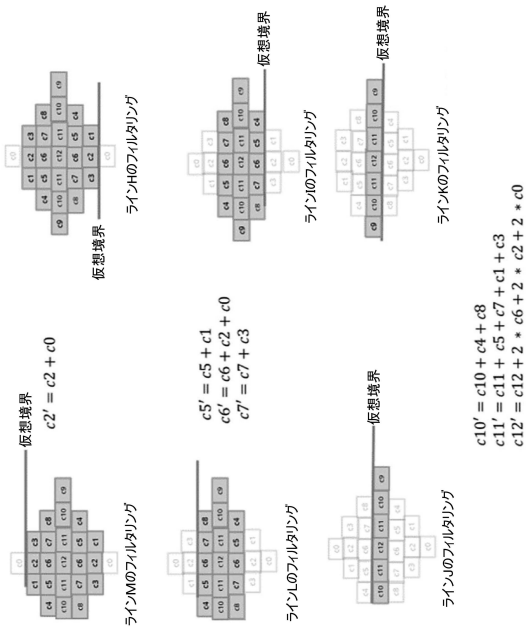
仮想境界での修正クロマALFフィルタリング(5×5菱形フィルタサポート、中心はフィルタリングされる現在のサンプル)

【図 7 B】



仮想境界での修正クロマALFフィルタリング(5×5菱形フィルタサポート、中心はフィルタリングされる現在のサンプル)

【図 8】



【図 9 A】

V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V

垂直勾配のサブサンプリングされたラプラシアン計算位置

【図 9 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

水平勾配のサブサンプリングされたラプラシアン計算位置

10

【図 9 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

対角勾配のサブサンプリングされたラプラシアン計算位置

【図 9 D】

D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2

対角勾配のサブサンプリングされたラプラシアン計算位置

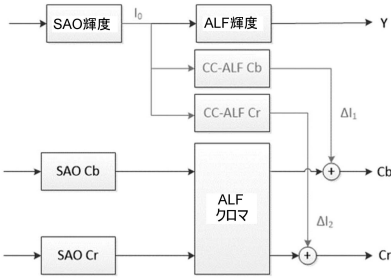
20

30

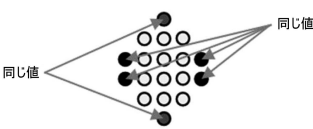
40

50

【図 1 0 A】



【図 1 0 B】

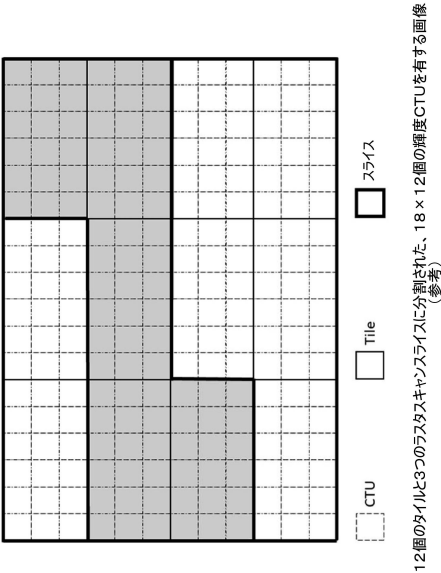


10

【図 1 1】



【図 1 2】



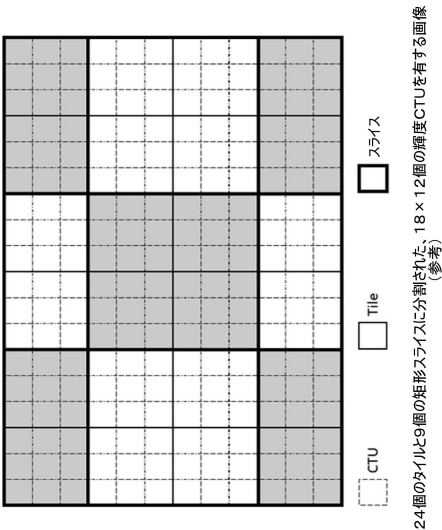
20

30

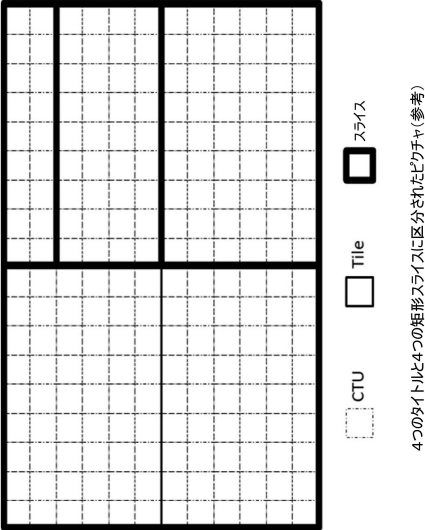
40

50

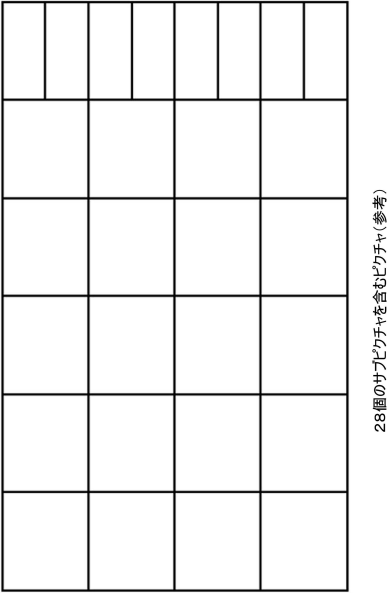
【図 1 3】



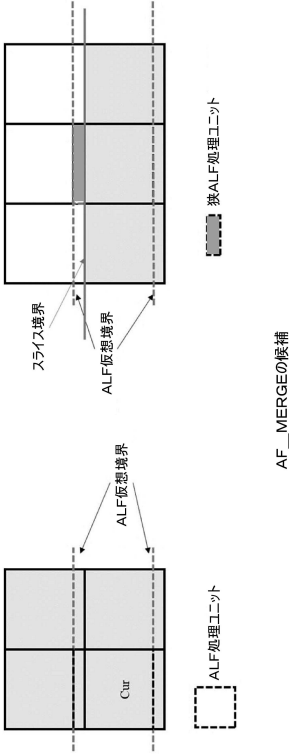
【図 1 4】



【図 1 5】



【図 1 6】



10

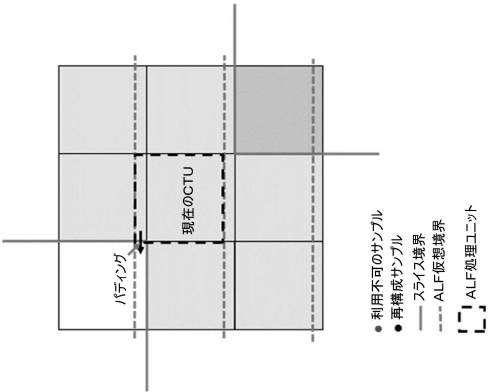
20

30

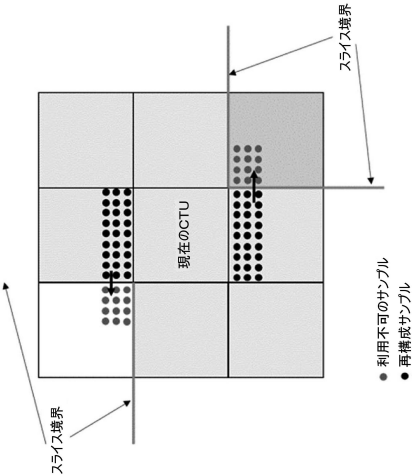
40

50

【図 17】

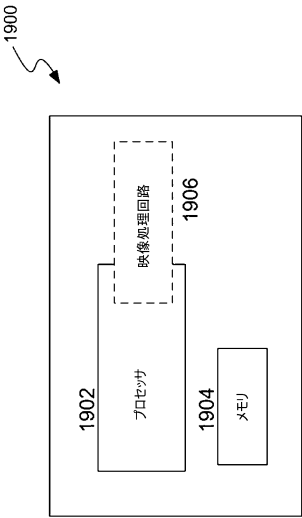


【図 18】

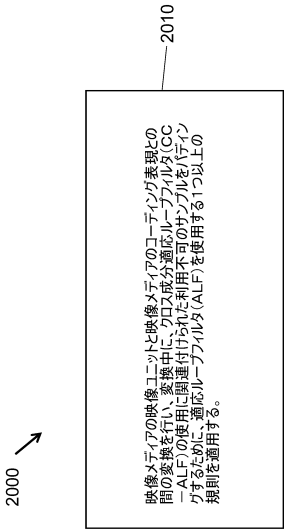


10

【図 19】



【図 20】



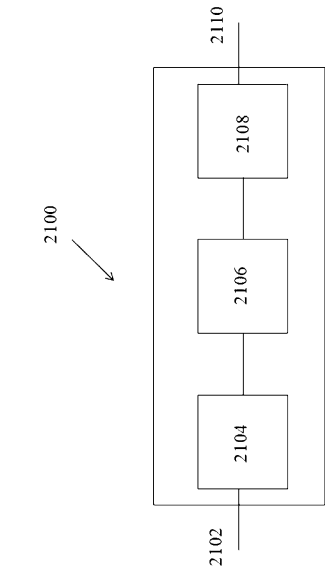
20

30

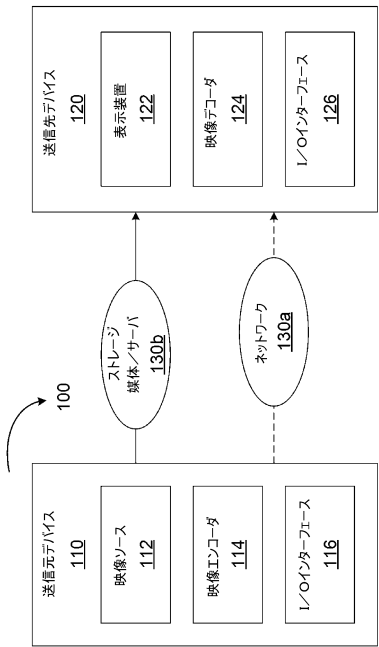
40

50

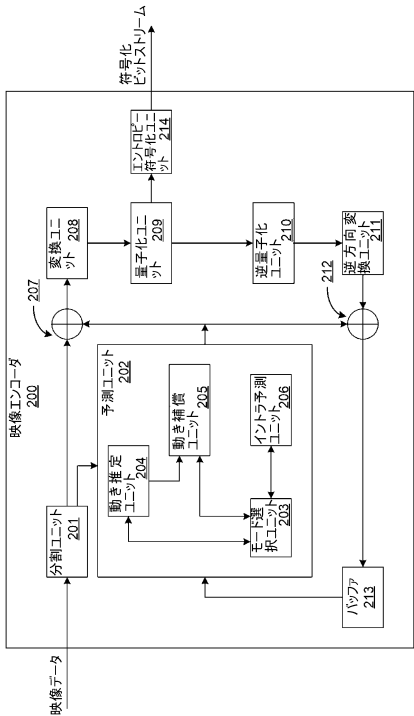
【図 2 1】



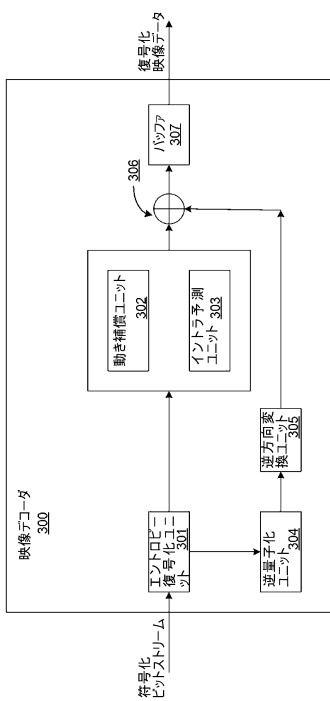
【図 2 2】



【図 2 3】



【図 2 4】



10

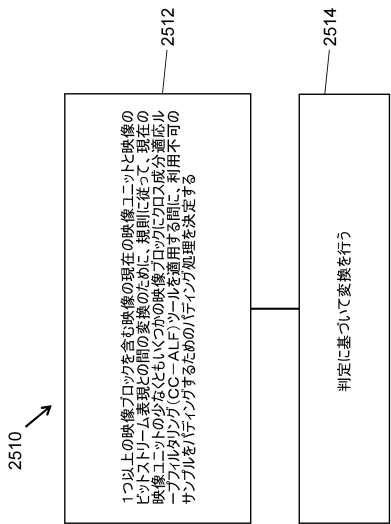
20

30

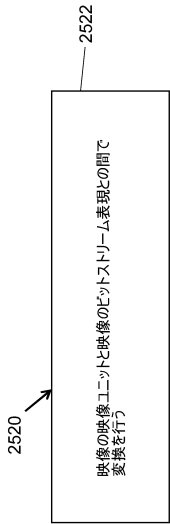
40

50

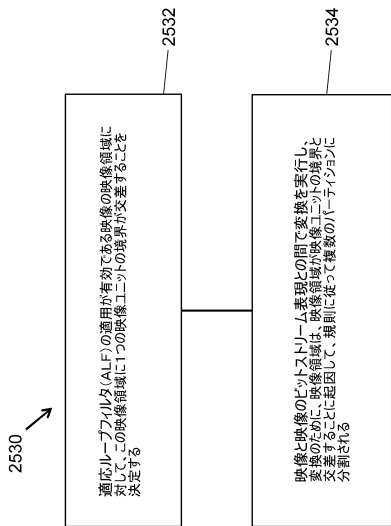
【図 2 5 A】



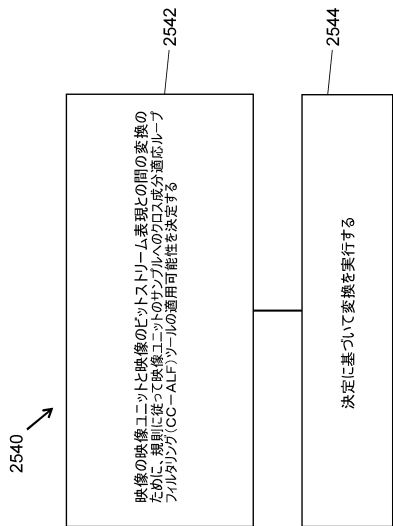
【図 2 5 B】



【図 2 5 C】



【図 2 5 D】



10

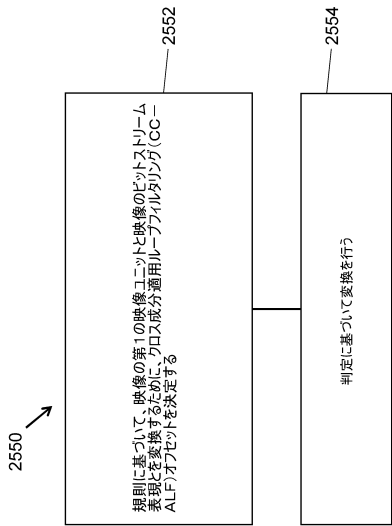
20

30

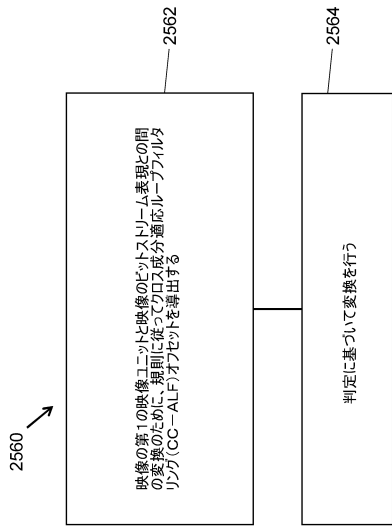
40

50

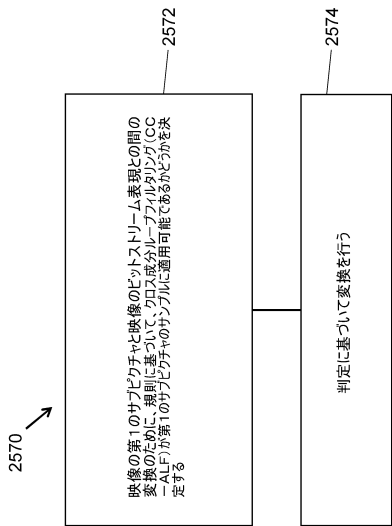
【図 2 5 E】



【図 2 5 F】



【図 2 5 G】



10

20

30

40

50

フロントページの続き

(51)国際特許分類

H 0 4 N 19/82 (2014.01)

F I

H 0 4 N 19/82

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, Suite No. 137 Los Angeles, California 90066 United 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 베이징, 하이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

ワン ユエ

中華人民共和国 100080 베이징, 하이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

審査官

久保 光宏

(56)参考文献

米国特許出願公開第 2013/0128986 (US, A1)

Kiran Misra, et al., "CE5-2.1, CE5-2.2: Cross Component Adaptive Loop Filter", Document: JVET-P0080, [online], JVET-P0080 (version 1), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2019年09月18日, Pages 1-7, [令和5年4月20日検索], インターネット, <URL: https://jvet-experts.org/doc_end_user/current_document.php?id=7869> and <URL: https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0080-v1.zip> ., (See document file "JVET-P0080.docx" in the zip file "JVET-P0080-v1.zip".)

Chia-Yang Tsai, et al., "AHG6: ALF with modified padding process", Document: JCTVC-J0050, [online], JCTVC-J0050 (version 2), Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2012年07月10日, Pages 1-8, [令和5年4月21日検索], インターネット, <URL: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=5913> and <URL: http://phenix.it-sudparis.eu/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J0050-v2.zip> ., (See document file "JCTVC-J0050.doc" in the zip file "JCTVC-J0050-v2.zip".)

Andrey Norkin, et al., "BoG report on CE5 loop filtering related contributions", Document: JVET-P1033-v2, [online], JVET-P1033 (version 2), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2019年10月11日, Pages 1,2,14-17,20,21, [令和5年4月21日検索], インターネット, <URL: https://jvet-experts.org/doc_end_user/current_document.php?id=8848> and <URL: https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P1033-v2.zip> ., (See document file "JVET-P1033-v2.docx" in the zip file "JVET-P1033-v2.zip".)

Nan Hu, et al., "AHG12/Non-CE5: Extending slice boundary processing for adaptive loop filter for raster scanned slices", Document: JVET-P0552-v2, [online], JVET-P0552 (version

2), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2019年09月29日, Pages 1-4, [令和5年10月25日検索], インターネット, 〈URL: https://jvet-experts.org/doc_end_user/current_document.php?id=8342〉 and 〈URL: https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0552-v2.zip〉 ., (See document file "JVET-P0552-v2.docx" in the zip file "JVET-P0552-v2.zip".)

Chen-Yen Lai, et al., "CE5-related: ALF padding process when raster scan slices are used", Document: JVET-P0156-v2, [online], JVET-P0156 (version 2), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2019年10月03日, Pages 1-4, [令和5年10月25日検索], インターネット, 〈URL: https://jvet-experts.org/doc_end_user/current_document.php?id=7945〉 and 〈URL: https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0156-v2.zip〉 ., (See document file "JVET-P0156-v2.docx" in the zip file "JVET-P0156-v2.zip".)

(58)調査した分野 (Int.Cl., D B名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8

C S D B (日本国特許庁)

学術文献等データベース (日本国特許庁)

I E E E X p l o r e (I E E E)