

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7498297号
(P7498297)

(45)発行日 令和6年6月11日(2024.6.11)

(24)登録日 令和6年6月3日(2024.6.3)

(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	19/136 (2014.01)	H O 4 N	19/136
H O 4 N	19/182 (2014.01)	H O 4 N	19/182
H O 4 N	19/82 (2014.01)	H O 4 N	19/82
請求項の数 10 (全70頁)			
(21)出願番号	特願2022-563467(P2022-563467)	(73)特許権者	520353802
(86)(22)出願日	令和3年9月30日(2021.9.30)		テンセント・アメリカ・エルエルシー
(65)公表番号	特表2023-522703(P2023-522703 A)		アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブルヴァード 2 7 4 7
(43)公表日	令和5年5月31日(2023.5.31)	(74)代理人	100110364
(86)国際出願番号	PCT/US2021/052941		弁理士 実広 信哉
(87)国際公開番号	WO2022/203719	(74)代理人	100150197
(87)国際公開日	令和4年9月29日(2022.9.29)		弁理士 松尾 直樹
審査請求日	令和4年10月19日(2022.10.19)	(72)発明者	イシン・ドウ
(31)優先権主張番号	63/164,478		アメリカ合衆国・カリフォルニア・9 4 3 0 6・パロ・アルト・パーク・ブルバード・2 7 4 7・テンセント・アメリカ・エルエルシー内
(32)優先日	令和3年3月22日(2021.3.22)	(72)発明者	シン・ジャオ
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	17/449,126		
(32)優先日	令和3年9月28日(2021.9.28)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ビデオフィルタリングのための方法および装置

(57)【特許請求の範囲】

【請求項1】

ビデオコーディングにおけるフィルタリングのための方法であって、

プロセッサにより、フィルタ支持エリアに位置する第1の色成分から、複数のビデオフィルタを含むループフィルタチェーンに沿った第1のノードにある第1の復元サンプルを決定するステップであって、前記フィルタ支持エリアはフィルタが適用される範囲内のエリアである、ステップと、

前記プロセッサにより、前記第1の復元サンプルを非線形マッピングベースフィルタへの入力として処理してフィルタタップを形成するステップと、

前記プロセッサにより、前記第1の復元サンプルに基づいて、前記非線形マッピングベースフィルタを前記第1の復元サンプルに適用することによって、第1のオフセット値を前記フィルタタップに従って決定するステップと、

中間フィルタのいずれかによって中間復元サンプルを生成するステップと、

前記プロセッサにより、前記ループフィルタチェーンに沿った第2のノードにある中間復元サンプルに前記第1のオフセット値を加えることに基づいて第2の復元サンプルを生成するステップであって、前記第1の復元サンプル及び前記中間復元サンプルは色成分のサンプルである、ステップと

を含み、

前記非線形マッピングベースフィルタが、成分間サンプルオフセット（CCSO）フィルタである場合、前記中間復元サンプルおよび前記第1の復元サンプルが、異なる色成分のサ

ンプルであり、

前記非線形マッピングベースフィルタは、

前記中間フィルタのうちの少なくとも1つと並列的に配置され、または

前記中間フィルタと直列的に配置される

方法。

【請求項2】

前記非線形マッピングベースフィルタが、局所サンプルオフセット（LSO）フィルタである場合、前記中間復元サンプルおよび前記第1の復元サンプルが、同じ色成分のサンプルである、請求項1に記載の方法。

【請求項3】

前記第1のノードおよび前記第2のノードが同じノードである、請求項1に記載の方法。

【請求項4】

前記第1のノードおよび前記第2のノードが、

前記ループフィルタチェーンの入力ノード、

前記ループフィルタチェーンの出力ノード、または

前記ループフィルタチェーンの中間ノード

のうちの1つである、請求項3に記載の方法。

【請求項5】

前記第1の復元サンプルが、

デブロッキングフィルタより前の処理モジュール、

前記デブロッキングフィルタ、

制約付き指向性拡張フィルタ、または

ループ復元フィルタ

のうちの少なくとも1つである前記中間フィルタによって生成される、請求項3に記載の方法。

【請求項6】

前記第1のノードおよび前記第2のノードが異なるノードである、請求項1に記載の方法。

【請求項7】

前記第1の復元サンプルが、デブロッキングフィルタより前に処理モジュールによって生成され、前記中間復元サンプルが、

前記デブロッキングフィルタ、

制約付き指向性拡張フィルタ、または

ループ復元フィルタ

のうちの少なくとも1つである前記中間フィルタによって生成される、請求項6に記載の方法。

【請求項8】

前記第1の復元サンプルが、デブロッキングフィルタによって生成され、前記中間復元サンプルが、

制約付き指向性拡張フィルタ、または

ループ復元フィルタ

のうちの少なくとも1つである前記中間フィルタによって生成される、請求項6に記載の方法。

【請求項9】

前記第1の復元サンプルが、制約付き指向性拡張フィルタによって生成され、前記中間復元サンプルが、ループ復元フィルタによって生成される、請求項6に記載の方法。

【請求項10】

請求項1～9のいずれか一項に記載の方法を行うように構成された装置。

【発明の詳細な説明】

【技術分野】

10

20

30

40

50

【0001】

関連出願の相互参照

本出願は、2021年9月28日に出願された米国特許出願第17/449,126号「METHOD AND APPARATUS FOR VIDEO FILTERING」の優先権の利益を主張し、上記特許出願は、2021年3月22日に出願された米国仮出願第63/164,478号「FLEXIBLE FILTER LOCATION FOR SAMPLE OFFSET」の優先権の利益を主張する。先行出願の開示全体は、参照によりその全体が本明細書に組み込まれる。

【0002】

本開示は、一般に、ビデオコーディングに関係する実施形態を記載する。

【背景技術】

10

【0003】

本明細書において提供される背景技術の説明は、本開示の文脈を大まかに提示することを目的としている。ここに記名された発明者の仕事は、その仕事がこの背景技術セクションに記載されている程度まで、ならびにさもなければ出願時に従来技術として適格ではない可能性がある説明の態様は、本開示に対する従来技術として、明示的にも黙示的にも認められていない。

【0004】

ビデオのコーディングおよび復号は、動き補償を伴うインターピクチャ予測を使用して実行することができる。非圧縮デジタルビデオは一連のピクチャを含むことができ、各ピクチャは、例えば、1920×1080の輝度サンプルおよび関連する彩度サンプルの空間次元を有する。一連のピクチャは、例えば、毎秒60ピクチャまたは60Hzの固定または可変の（非公式にはフレームレートとしても知られる）ピクチャレートを有することができる。非圧縮ビデオは特有のビットレート要件を有する。例えば、サンプルあたり8ビットでの1080p60 4:2:0ビデオ（60Hzフレームレートで1920×1080の輝度サンプル解像度）は、1.5Gbit/sに近い帯域幅を必要とする。そのようなビデオの1時間は、600Gバイトを超える記憶空間を必要とする。

20

【0005】

ビデオのコーディングおよび復号の1つの目的は、圧縮を介して入力ビデオ信号内の冗長度を低減することであり得る。圧縮は、前述の帯域幅および／または記憶空間の要件を、場合によっては、2桁以上削減するのに役立つことができる。可逆圧縮と非可逆圧縮の両方、ならびにそれらの組合せを採用することができる。可逆圧縮は、圧縮された元の信号から元の信号の正確なコピーを復元することができる技法を指す。非可逆圧縮を使用すると、復元された信号は元の信号と同一ではない可能性があるが、元の信号と復元された信号との間の歪みは、復元された信号を目的の用途に有用なものにするほど十分小さい。ビデオの場合、非可逆圧縮が広く採用されている。許容される歪みの量はアプリケーションに依存し、例えば、特定の消費者向けストリーミングアプリケーションのユーザは、テレビ配信アプリケーションのユーザよりも高い歪みを許容することができる。実現可能な圧縮比は、許容／耐容歪みが大きいほど、圧縮比が高くなり得ることを反映することができる。

30

【0006】

ビデオエンコーダおよびデコーダは、例えば、動き補償、変換、量子化、およびエントロピーコーディングを含む、いくつかの広範なカテゴリからの技法を利用することができる。

40

【0007】

ビデオコーデック技術は、イントラコーディングとして知られる技法を含むことができる。イントラコーディングでは、サンプル値は、以前に復元された参照ピクチャからのサンプルまたは他のデータを参照することなく表される。一部のビデオコーデックでは、ピクチャは空間的にサンプルのブロックに細分化される。サンプルのすべてのブロックがイントラモードでコーディングされるとき、そのピクチャはイントラピクチャであり得る。イントラピクチャ、および独立デコーダリフレッシュピクチャなどのそれらの派生物は、

50

デコーダ状態をリセットするために使用することができるので、コーディングされたビデオビットストリームおよびビデオセッション内の最初のピクチャとして、または静止画像として使用することができる。イントラブロックのサンプルは、変換にさらされる可能性があり、変換係数は、エントロピーコーディングの前に量子化することができる。イントラ予測は、変換前領域内のサンプル値を最小化する技法であり得る。場合によっては、変換後のDC値が小さいほど、かつAC係数が小さいほど、エントロピーコーディング後のブロックを表すために所与の量子化ステップのサイズにおいて必要とされるビットは少なくなる。

【0008】

例えばMPEG-2生成コーディング技術から知られているような従来のイントラコーディングは、イントラ予測を使用しない。しかしながら、いくつかのより新しいビデオ圧縮技術は、例えば、空間的に隣接し、復号順序で先行するデータのブロックの符号化／復号の間に取得される周囲のサンプルデータおよび／またはメタデータから試行する技法を含む。そのような技法は、以後「イントラ予測」技法と呼ばれる。少なくともいくつかの場合では、イントラ予測は復元中の現在ピクチャからの参照データのみを使用し、参照ピクチャからは使用しないことに留意されたい。

【0009】

イントラ予測には多くの異なる形式があり得る。そのような技法のうちの2つ以上が所与のビデオコーディング技術において使用され得るとき、使用中の技法はイントラ予測モードでコーディングすることができる。いくつかの例では、モードは下位モードおよび／またはパラメータを有することができる、それらは個別にコーディングされ得るか、またはモードコードワードに含まれ得る。所与のモード／下位モード／パラメータの組合せにどのコードワードを使用するかは、イントラ予測によるコーディング効率の向上に影響を及ぼすことができるので、コードワードをビットストリームに変換するために使用されるエントロピーコーディング技術に影響を及ぼすことができる。

【0010】

イントラ予測の特定のモードは、H. 264で導入され、H. 265において改良され、共同探索モデル（JEM）、多用途ビデオコーディング（VVC）、およびベンチマークセット（BMS）などのより新しいコーディング技術においてさらに改良された。予測器ブロックは、すでに利用可能なサンプルに属する隣接サンプル値を使用して形成することができる。隣接サンプルのサンプル値は、方向に従って予測器ブロックにコピーされる。使用中の方向への参照は、ビットストリーム内でコーディングされ得るか、またはそれ自体が予測されてもよい。

【0011】

図1Aを参照すると、右下に描写されているのは、H. 265の（35個のイントラモードのうちの33個の角度モードに対応する）33個の可能な予測器方向から知られる9つの予測器方向のサブセットである。矢印が収束する点（101）は、予測されているサンプルを表す。矢印は、そこからサンプルが予測されている方向を表す。例えば、矢印（102）は、サンプル（101）が水平から45度の角度で右上にある1つまたは複数のサンプルから予測されることを示す。同様に、矢印（103）は、サンプル（101）が水平から22.5度の角度でサンプル（101）の左下にある1つまたは複数のサンプルから予測されることを示す。

【0012】

さらに図1Aを参照すると、左上には、（太い破線によって示された）4×4サンプルの正方形ブロック（104）が描写されている。正方形ブロック（104）は16個のサンプルを含み、各々、「S」、Y次元のその位置（例えば、行インデックス）、およびX次元のその位置（例えば、列インデックス）でラベル付けされている。例えば、サンプルS21は、Y次元の（上から）2番目のサンプルであり、X次元の（左から）1番目のサンプルである。同様に、サンプルS44は、ブロック（104）内のY次元およびX次元の両方の4番目のサンプルである。ブロックはサイズが4×4サンプルなので、S44は右下にある。同様の番号付け方式に従う参照サンプルがさらに示されている。参照サンプルは、ブロック（104）に対

10

20

30

40

50

してR、そのY位置（例えば、行インデックス）、およびX位置（列インデックス）でラベル付けされている。H. 264およびH. 265の両方において、予測サンプルは復元中のブロックに隣接するので、負の値が使用される必要はない。

【0013】

イントラピクチャ予測は、シグナリングされた予測方向によって割り当てられるように、隣接サンプルからの参照サンプル値をコピーすることによって機能することができる。例えば、コーディングされたビデオビットストリームが、このブロックに関して、矢印（102）と一致する予測方向を示すシグナリングを含む、すなわち、サンプルが水平から45度の角度で右上にある1つまたは複数の予測サンプルから予測されると仮定する。その場合、同じ参照サンプルR05からサンプルS41、S32、S23、およびS14が予測される。次いで、参照サンプルR08からサンプルS44が予測される。

10

【0014】

特定の場合には、参照サンプルを計算するために、特に方向が45度によって均等に割り切れないときは、複数の参照サンプルの値は、例えば補間によって組み合わせられてもよい。

【0015】

可能な方向の数は、ビデオコーディング技術が発展するにつれて増加している。H. 264（2003年）では、9つの異なる方向を表すことができた。それがH. 265（2013年）では33に増加し、本開示の時点では、JEM/VVC/BMSは最大65個の方向をサポートすることができる。最も可能性が高い方向を識別するために実験が行われており、エン트로ピーコーディングの特定の技術は、それらの可能性が高い方向を少数のビットで表すために使用され、可能性が低い方向に関しては一定のペナルティを受け入れている。さらに、方向自体は、時々、隣接するすでに復号されたブロックにおいて使用された隣接方向から予測することができる。

20

【0016】

図1Bは、経時的に増加する数の予測方向を示すために、JEMによる65個のイントラ予測方向を描写する概略図（180）を示す。

【0017】

方向を表す、コーディングされたビデオビットストリーム内のイントラ予測方向ビットのマッピングは、ビデオコーディング技術ごとに異なってもよく、例えば、予測方向のイントラ予測モードへの単純な直接マッピングから、コードワード、最も可能性が高いモードを含む複雑な適応方式、および同様の技術まで及ぶことができる。しかしながら、すべての場合において、ビデオコンテンツ内で特定の他の方向よりも統計的に発生する可能性が低い特定の方向が存在することができる。ビデオ圧縮の目的は冗長性の低減であるので、それらの可能性が低い方向は、うまく機能するビデオコーディング技術では、可能性が高い方向よりも多いビット数によって表される。

30

【0018】

動き補償は非可逆圧縮技法であり得、以前に復元されたピクチャまたはその一部（参照ピクチャ）からのサンプルデータのブロックが、動きベクトル（以降、MV）によって示された方向に空間的にシフトされた後に、新しく復元されるピクチャまたはピクチャの一部の予測に使用される技法に関することがある。場合によっては、参照ピクチャは現在復元中のピクチャと同じであり得る。MVは、2つの次元XおよびY、または3つの次元を有することができ、3番目の次元は、使用中の参照ピクチャの指示である（後者は、間接的に時間次元であり得る）。

40

【0019】

いくつかのビデオ圧縮技法では、サンプルデータの特定の領域に適用可能なMVは、他のMVから、例えば、復元中の領域に空間的に隣接し、復号順序でそのMVに先行するサンプルデータの別の領域に関連するMVから予測することができる。そうすることにより、MVのコーディングに必要なデータ量を大幅に削減することができ、それによって冗長性が排除され、圧縮率が増加する。例えば、（ナチュラルビデオとして知られている）カメラから導出された入力ビデオ信号をコーディングするとき、単一のMVが適用可能な領域よ

50

りも大きい領域が同様の方向に移動する統計的な可能性が存在するので、MV予測は効果的に機能することができ、したがって、場合によっては、隣接する領域のMVから導出された同様の動きベクトルを使用して予測することができる。その結果、所与の領域について検出されたMVは、周囲のMVから予測されたMVと同様または同じであり、エンコーディング後、直接MVをコーディングする場合に使用されるビット数より少ないビット数で表すことができる。場合によっては、MV予測は、元の信号（すなわち、サンプルストリーム）から導出された信号（すなわち、MV）の可逆圧縮の一例であり得る。他の場合、MV予測自体は、例えば、いくつかの周囲のMVから予測子を計算するときの丸め誤差のために、非可逆であり得る。

【0020】

様々なMV予測メカニズムが、H. 265/HEVC (ITU-T Rec. H. 265, 「High Efficiency Video Coding」、2016年12月)に記載されている。H. 265が提供する多くのMV予測メカニズムのうち、本明細書に記載されるのは、以降「空間マージ」と呼ばれる技法である。

【0021】

図2を参照すると、現在ブロック(201)は、動き検索プロセス中にエンコーダにより、空間的にシフトされた同じサイズの以前のブロックから予測可能であることが見出されたサンプルを含む。直接そのMVをコーディングする代わりに、MVは、A0、A1、およびB0、B1、B2（それぞれ、202~206）と表記された5つの周囲サンプルのいずれか1つに関連付けられたMVを使用して、1つまたは複数の参照ピクチャに関連付けられたメタデータから、例えば、（復号順序で）最新の参照ピクチャから導出することができる。H. 265では、MV予測は、隣接するブロックが使用している同じ参照ピクチャからの予測子を使用することができる。

【発明の概要】

【課題を解決するための手段】

【0022】

本開示の態様は、ビデオ符号化／復号におけるフィルタリングのための方法および装置を提供する。いくつかの例では、ビデオフィルタリングのための装置は処理回路を含む。処理回路は、ループフィルタチェーン内の第1のノードにある第1の復元サンプルに基づいて、非線形マッピングベースフィルタを適用するための第1のオフセット値を決定する。次いで、処理回路は、第2の復元サンプルを生成するために、ループフィルタチェーンに沿った第2のノードにある中間復元サンプルに第1のオフセット値を加える。

【0023】

いくつかの例では、非線形マッピングベースフィルタは、成分間サンプルオフセット（CCSO）フィルタであり、中間復元サンプルおよび第1の復元サンプルは、異なる色成分のサンプルである。

【0024】

いくつかの例では、非線形マッピングベースフィルタは、局所サンプルオフセット（LSO）フィルタであり、中間復元サンプルおよび第1の復元サンプルは、同じ色成分のサンプルである。

【0025】

いくつかの実施形態では、第1のノードおよび第2のノードは、ループフィルタチェーンの入力ノード、ループフィルタチェーンの出力ノード、またはループフィルタチェーンの中間ノードなどの同じノードのものである。一例では、第1の復元サンプルは、デブロッキングフィルタより前に処理モジュールによって生成される。別の例では、第1の復元サンプルは、デブロッキングフィルタによって生成される。別の例では、第1の復元サンプルは、制約付き指向性拡張フィルタによって生成される。別の例では、第1の復元サンプルは、ループ復元フィルタによって生成される。

【0026】

いくつかの例では、第1のノードおよび第2のノードは異なるノードのものである。一例

10

20

30

40

50

では、第1の復元サンプルは、デブロッキングフィルタより前に処理モジュールによって生成され、中間復元サンプルは、デブロッキングフィルタ、制約付き指向性拡張フィルタ、またはループ復元フィルタのうちの少なくとも1つによって生成される。

【0027】

いくつかの例では、第1の復元サンプルは、デブロッキングフィルタによって生成され、中間復元サンプルは、制約付き指向性拡張フィルタまたはループ復元フィルタのうちの少なくとも1つによって生成される。

【0028】

いくつかの例では、第1の復元サンプルは、制約付き指向性拡張フィルタによって生成され、中間復元サンプルは、ループ復元フィルタによって生成される。

【0029】

本開示の態様はまた、コンピュータによって実行されると、ビデオ符号化／復号のための方法のいずれかをコンピュータに実行させる命令を記憶する非一時的コンピュータ可読媒体を提供する。

【0030】

開示された主題のさらなる特徴、性質、および様々な利点は、以下の詳細な説明および添付の図面からより明らかになるであろう。

【図面の簡単な説明】

【0031】

【図1A】イントラ予測モードの例示的なサブセットの概略図である。

【図1B】例示的なイントラ予測方向の図である。

【図2】一例における現在ブロックおよびその周囲の空間マージ候補の概略図である。

【図3】一実施形態による、通信システム(300)の簡略化されたブロック図の概略図である。

【図4】一実施形態による、通信システム(400)の簡略化されたブロック図の概略図である。

【図5】一実施形態による、デコーダの簡略化されたブロック図の概略図である。

【図6】一実施形態による、エンコーダの簡略化されたブロック図の概略図である。

【図7】別の実施形態による、エンコーダのブロック図である。

【図8】別の実施形態による、デコーダのブロック図である。

【図9】本開示の実施形態による、フィルタ形状の例を示す図である。

【図10A】本開示の実施形態による、勾配を計算するために使用されるサブサンプリングされた位置の例を示す図である。

【図10B】本開示の実施形態による、勾配を計算するために使用されるサブサンプリングされた位置の例を示す図である。

【図10C】本開示の実施形態による、勾配を計算するために使用されるサブサンプリングされた位置の例を示す図である。

【図10D】本開示の実施形態による、勾配を計算するために使用されるサブサンプリングされた位置の例を示す図である。

【図11A】本開示の実施形態による、仮想境界フィルタリングプロセスの例を示す図である。

【図11B】本開示の実施形態による、仮想境界フィルタリングプロセスの例を示す図である。

【図12A】本開示の実施形態による、仮想境界における対称パディング動作の例を示す図である。

【図12B】本開示の実施形態による、仮想境界における対称パディング動作の例を示す図である。

【図12C】本開示の実施形態による、仮想境界における対称パディング動作の例を示す図である。

【図12D】本開示の実施形態による、仮想境界における対称パディング動作の例を示す

10

20

30

40

50

図である。

【図 1 2 E】本開示の実施形態による、仮想境界における対称パディング動作の例を示す図である。

【図 1 2 F】本開示の実施形態による、仮想境界における対称パディング動作の例を示す図である。

【図 1 3】本開示のいくつかの実施形態による、ピクチャのパーティション例を示す図である。

【図 1 4】いくつかの例におけるピクチャ用の四分木分割パターンを示す図である。

【図 1 5】本開示の一実施形態による、成分間フィルタを示す図である。

【図 1 6】本開示の一実施形態による、フィルタ形状の一例を示す図である。

【図 1 7】本開示のいくつかの実施形態による、成分間フィルタ用の構文例を示す図である。

【図 1 8 A】本開示の実施形態による、輝度サンプルに対する彩度サンプルの例示的な位置を示す図である。

【図 1 8 B】本開示の実施形態による、輝度サンプルに対する彩度サンプルの例示的な位置を示す図である。

【図 1 9】本開示の一実施形態による、方向検索の一例を示す図である。

【図 2 0】いくつかの例における部分空間投影を示す一例を示す図である。

【図 2 1】本開示の一実施形態による、複数のサンプル適応オフセット (SAO) タイプの表である。

【図 2 2】いくつかの例におけるエッジオフセットにおけるピクセル分類のためのパターンの例を示す図である。

【図 2 3】いくつかの例におけるエッジオフセット向けのピクセル分類規則のための表である。

【図 2 4】シグナリングされ得る構文の一例を示す図である。

【図 2 5】本開示のいくつかの実施形態による、フィルタ支持エリアの一例を示す図である。

【図 2 6】本開示のいくつかの実施形態による、別のフィルタ支持エリアの一例を示す図である。

【図 2 7 A】本開示の一実施形態による、81個の組合せを有する表である。

【図 2 7 B】本開示の一実施形態による、81個の組合せを有する表である。

【図 2 7 C】本開示の一実施形態による、81個の組合せを有する表である。

【図 2 8】一例における3つのフィルタタップの7つのフィルタ形状構成を示す図である。

【図 2 9】いくつかの例におけるループフィルタチェーンのブロック図である。

【図 3 0 A】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 0 B】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 0 C】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 0 D】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 1 A】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 1 B】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 2 A】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 2 B】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

10

20

30

40

50

【図 3 2 C】ループフィルタチェーンの異なる位置に非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す図である。

【図 3 3】いくつかの例における非線形マッピングベースフィルタを含むループフィルタチェーンの別の例を示す図である。

【図 3 4】いくつかの例における複数の非線形マッピングベースフィルタを含むループフィルタチェーンの一例を示す図である。

【図 3 5】いくつかの例における複数の非線形マッピングベースフィルタを含むループフィルタチェーンの別の例を示す図である。

【図 3 6】本開示の一実施形態による、プロセスの概要を示すフローチャートである。

【図 3 7】一実施形態による、コンピュータシステムの概略図である。

【発明を実施するための形態】

【0032】

図3は、本開示の一実施形態による、通信システム（300）の簡略化されたブロック図を示す。通信システム（300）は、例えば、ネットワーク（350）を介して互いに通信することができる複数の端末デバイスを含む。例えば、通信システム（300）は、ネットワーク（350）を介して相互接続された端末デバイス（310）および（320）の第1のペアを含む。図3の例では、端末デバイス（310）および（320）の第1のペアは、データの単方向送信を実行する。例えば、端末デバイス（310）は、ネットワーク（350）を介して他の端末デバイス（320）に送信するためのビデオデータ（例えば、端末デバイス（310）によって取り込まれたビデオピクチャのストリーム）をコード化することができる。符号化ビデオデータは、1つまたは複数のコード化ビデオビットストリームの形式で送信することができる。端末デバイス（320）は、ネットワーク（350）からコード化ビデオデータを受信し、コード化ビデオデータを復号してビデオピクチャを復元し、復元されたビデオデータに従ってビデオピクチャを表示することができる。単方向データ送信は、メディアサービングアプリケーションなどで一般的であり得る。

【0033】

別の例では、通信システム（300）は、例えばビデオ会議中に発生する可能性があるコード化ビデオデータの双方向送信を実行する端末デバイス（330）および（340）の第2のペアを含む。データの双方向送信の場合、一例では、端末デバイス（330）および（340）の各端末デバイスは、ネットワーク（350）を介して端末デバイス（330）および（340）の他方の端末デバイスに送信するためのビデオデータ（例えば、端末デバイスによって取り込まれたビデオピクチャのストリーム）をコード化することができる。端末デバイス（330）および（340）の各端末デバイスはまた、端末デバイス（330）および（340）の他方の端末デバイスによって送信されたコード化ビデオデータを受信することができ、コード化ビデオデータを復号してビデオピクチャを復元することができ、復元されたビデオデータに従ってアクセス可能な表示デバイスにビデオピクチャを表示することができる。

【0034】

図3の例では、端末デバイス（310）、（320）、（330）、および（340）は、サーバ、パーソナルコンピュータ、およびスマートフォンとして示される場合があるが、本開示の原理はそうに限定されなくてもよい。本開示の実施形態は、ラップトップコンピュータ、タブレットコンピュータ、メディアプレーヤ、および／または専用ビデオ会議機器を用いるアプリケーションを見出す。ネットワーク（350）は、例えば、電線（有線）および／またはワイヤレスの通信ネットワークを含む、端末デバイス（310）、（320）、（330）、および（340）の間でコード化ビデオデータを伝達する任意の数のネットワークを表す。通信ネットワーク（350）は、回線交換チャンネルおよび／またはパケット交換チャンネルでデータを交換することができる。代表的なネットワークには、電気通信ネットワーク、ローカルエリアネットワーク、ワイドエリアネットワーク、および／またはインターネットが含まれる。本説明の目的のために、ネットワーク（350）のアーキテクチャおよびトポロジは、本明細書で以下に説明されない限り、本開示の動作にとって重要

10

20

30

40

50

でない可能性がある。

【0035】

図4は、開示された主題についてのアプリケーション用の一例として、ストリーミング環境におけるビデオエンコーダおよびビデオデコーダの配置を示す。開示された主題は、例えば、ビデオ会議、デジタルテレビ、CD、DVD、メモリスティックなどを含むデジタル媒体への圧縮ビデオの保存などを含む、他のビデオ対応アプリケーションに等しく適用可能であり得る。

【0036】

ストリーミングシステムは、例えば、圧縮されていないビデオピクチャのストリーム（402）を作成するビデオソース（401）、例えば、デジタルカメラを含むことができるキャプチャサブシステム（413）を含んでもよい。一例では、ビデオピクチャのストリーム（402）は、デジタルカメラによって撮影されたサンプルを含む。符号化ビデオデータ（404）（またはコード化ビデオビットストリーム）と比較したときに多いデータ量を強調するために太い線として描写されたビデオピクチャのストリーム（402）は、ビデオソース（401）に結合されたビデオエンコーダ（403）を含む電子デバイス（420）によって処理することができる。ビデオエンコーダ（403）は、以下でより詳細に記載されるように、開示された主題の態様を可能にするかまたは実装するために、ハードウェア、ソフトウェア、またはそれらの組合せを含むことができる。ビデオピクチャのストリーム（402）と比較したときに少ないデータ量を強調するために細い線として描写された符号化ビデオデータ（404）（または符号化ビデオビットストリーム（404））は、将来の使用のためにストリーミングサーバ（405）に格納することができる。図4のクライアントサブシステム（406）および（408）などの1つまたは複数のストリーミングクライアントサブシステムは、ストリーミングサーバ（405）にアクセスして、符号化ビデオデータ（404）のコピー（407）および（409）を検索することができる。クライアントサブシステム（406）は、例えば、電子デバイス（430）内のビデオデコーダ（410）を含むことができる。ビデオデコーダ（410）は、符号化ビデオデータの入力コピー（407）を復号し、ディスプレイ（412）（例えば、ディスプレイ画面）または他のレンダリングデバイス（描写せず）上でレンダリングすることができるビデオピクチャの出力ストリーム（411）を作成する。いくつかのストリーミングシステムでは、符号化ビデオデータ（404）、（407）、および（409）（例えば、ビデオビットストリーム）は、特定のビデオコーディング／圧縮規格に従って符号化することができる。それらの規格の例には、ITU-T勧告H.265が含まれる。一例では、開発中のビデオコーディング規格は、非公式に多用途ビデオコーディング

（VVC）として知られている。開示された主題は、VVCの文脈で使用されてもよい。

【0037】

電子デバイス（420）および（430）は、他の構成要素（図示せず）を含むことができることに留意されたい。例えば、電子デバイス（420）はビデオデコーダ（図示せず）を含むことができ、電子デバイス（430）もビデオエンコーダ（図示せず）を含むことができる。

【0038】

図5は、本開示の一実施形態による、ビデオデコーダ（510）のブロック図を示す。ビデオデコーダ（510）は、電子デバイス（530）に含まれ得る。電子デバイス（530）は、受信機（531）（例えば、受信回路）を含むことができる。ビデオデコーダ（510）は、図4の例のビデオデコーダ（410）の代わりに使用することができる。

【0039】

受信機（531）は、ビデオデコーダ（510）によって復号される1つまたは複数のコード化ビデオシーケンス、同じかまたは別の実施形態では、一度に1つのコード化ビデオシーケンスを受信することができ、各コード化ビデオシーケンスの復号は、他のコード化ビデオシーケンスから独立している。コード化ビデオシーケンスは、チャンネル（501）から受信されてもよく、チャンネル（501）は、符号化ビデオデータを格納するストレージデバ

イスへのハードウェア／ソフトウェアリンクであってもよい。受信機（531）は、それらのそれぞれの使用エンティティ（描写せず）に転送され得る他のデータ、例えば、コード化オーディオデータおよび／または補助データストリームとともに符号化ビデオデータを受信することができる。受信機（531）は、コード化ビデオシーケンスを他のデータから分離することができる。ネットワークジッタに対抗するために、バッファメモリ（515）は、受信機（531）とエントロピーデコーダ／パーサー（520）（以下、「パーサー（520）」）との間に結合されてもよい。特定のアプリケーションでは、バッファメモリ（515）はビデオデコーダ（510）の一部である。他のアプリケーションでは、それはビデオデコーダ（510）の外側にあり得る（描写せず）。さらに他のアプリケーションでは、例えば、ネットワークジッタに対抗するために、ビデオデコーダ（510）の外側にバッファメモリ（描写せず）が存在することができ、加えて、例えば、プレイアウトタイミングを処理するために、ビデオデコーダ（510）の内側に別のバッファメモリ（515）が存在することができる。受信機（531）が十分な帯域幅および制御可能性のストア／フォワードデバイスから、または同期ネットワークからデータを受信しているとき、バッファメモリ（515）は必要とされなくてもよい、または小さい可能性がある。インターネットなどのベストエフォートパケットネットワークで使用するために、バッファメモリ（515）は、必要とされる場合があり、比較的大きい可能性があり、有利なことに適応サイズであり得、オペレーティングシステムまたはビデオデコーダ（510）の外側の同様の要素（描写せず）に少なくとも部分的に実装されてもよい。

【0040】

ビデオデコーダ（510）は、コード化ビデオシーケンスからシンボル（521）を復元するためにパーサー（520）を含んでもよい。これらのシンボルのカテゴリには、ビデオデコーダ（510）の動作を管理するために使用される情報、および潜在的に、電子デバイス（530）の不可欠な部分ではないが、図5に示されたように、電子デバイス（530）に結合することができるレンダリングデバイス（512）（例えば、ディスプレイ画面）などのレンダリングデバイスを制御するための情報が含まれる。レンダリングデバイスのための制御情報は、補足拡張情報（SEIメッセージ）またはビデオユーザビリティ情報（VUI）のパラメータセットフラグメント（描写せず）の形式であってもよい。パーサー（520）は、受け取ったコード化ビデオシーケンスを構文解析／エントロピー復号することができる。コード化ビデオシーケンスのコーディングは、ビデオコーディング技術または規格に従うことができ、文脈感度の有無にかかわらず、可変長コーディング、ハフマンコーディング、算術コーディングなどを含む様々な原理に従うことができる。パーサー（520）は、グループに対応する少なくとも1つのパラメータに基づいて、コード化ビデオシーケンスから、ビデオデコーダ内のピクセルのサブグループのうちの少なくとも1つのためのサブグループパラメータのセットを抽出することができる。サブグループは、ピクチャグループ（GOP）、ピクチャ、タイル、スライス、マクロブロック、コーディングユニット（CU）、ブロック、変換ユニット（TU）、予測ユニット（PU）などを含むことができる。パーサー（520）はまた、コード化ビデオシーケンスから、変換係数、量子化器パラメータ値、動きベクトルなどの情報を抽出することができる。

【0041】

パーサー（520）は、シンボル（521）を作成するために、バッファメモリ（515）から受け取ったビデオシーケンスに対してエントロピー復号／構文解析動作を実行することができる。

【0042】

シンボル（521）の復元は、（ピクチャ間およびピクチャ内、ブロック間およびブロック内などの）コード化ビデオピクチャまたはその一部のタイプ、ならびに他の要因に応じて、複数の異なるユニットを含むことができる。どのユニットがどのように関与しているかは、パーサー（520）によってコード化ビデオシーケンスから構文解析されたサブグループ制御情報によって制御することができる。パーサー（520）と以下の複数のユニットとの間のそのようなサブグループ制御情報の流れは、分かりやすくするために描写されて

いない。

【0043】

すでに述べられた機能ブロック以外に、ビデオデコーダ（510）は、以下に記載されるように、概念的にいくつかの機能ユニットに細分化することができる。商業的制約の下で動作する実際の実装形態では、これらのユニットの多くは、互いに密接に相互作用し、少なくとも部分的には互いに統合することができる。しかしながら、開示された主題を記載するために、以下の機能ユニットへの概念的な細分化が適切である。

【0044】

第1のユニットはスケーラ／逆変換ユニット（551）である。スケーラ／逆変換ユニット（551）は、量子化変換係数、ならびにどの変換を使用するか、ブロックサイズ、量子化係数、量子化スケーリング行列などを含む制御情報を、パーサー（520）からシンボル（521）として受け取る。スケーラ／逆変換ユニット（551）は、アグリゲータ（555）に入力することができるサンプル値を含むブロックを出力することができる。

10

【0045】

場合によっては、スケーラ／逆変換（551）の出力サンプルは、イントラコード化ブロック、すなわち、以前に復元されたピクチャからの予測情報を使用していないが、現在ピクチャの以前に復元された部分からの予測情報を使用することができるブロックに関連する可能性がある。そのような予測情報は、イントラピクチャ予測ユニット（552）によって提供することができる。場合によっては、イントラピクチャ予測ユニット（552）は、現在ピクチャバッファ（558）からフェッチされた周囲のすでに復元された情報を使用して、復元中のブロックと同じサイズおよび形状のブロックを生成する。現在ピクチャバッファ（558）は、例えば、部分的に復元された現在ピクチャおよび／または完全に復元された現在ピクチャをバッファリングする。アグリゲータ（555）は、場合によっては、サンプルごとに、イントラ予測ユニット（552）が生成した予測情報を、スケーラ／逆変換ユニット（551）によって提供される出力サンプル情報に追加する。

20

【0046】

他の場合には、スケーラ／逆変換ユニット（551）の出力サンプルは、インターコード化され、潜在的に動き補償されたブロックに関連する可能性がある。そのような場合、動き補償予測ユニット（553）は、参照ピクチャメモリ（557）にアクセスして、予測に使用されるサンプルをフェッチすることができる。ブロックに関連するシンボル（521）に従ってフェッチされたサンプルを動き補償した後、これらのサンプルは、出力サンプル情報を生成するために、アグリゲータ（555）によってスケーラ／逆変換ユニット（551）の出力に追加することができる（この場合、残差サンプルまたは残差信号と呼ばれる）。動き補償予測ユニット（553）が予測サンプルをフェッチする参照ピクチャメモリ（557）内のアドレスは、例えば、X、Y、および参照ピクチャ成分を有することができるシンボル（521）の形式で動き補償予測ユニット（553）に利用可能な動きベクトルによって制御することができる。動き補償はまた、サブサンプルの正確な動きベクトルが使用されているときに参照ピクチャメモリ（557）からフェッチされたサンプル値の補間、動きベクトル予測メカニズムなどを含むことができる。

30

【0047】

アグリゲータ（555）の出力サンプルは、ループフィルタユニット（556）において様々なループフィルタリング技法を受けることができる。ビデオ圧縮技術は、（コード化ビデオビットストリームとも呼ばれる）コード化ビデオシーケンスに含まれるパラメータによって制御され、パーサー（520）からのシンボル（521）としてループフィルタユニット（556）に利用可能にされるインループフィルタ技術を含むことができるが、コード化ピクチャまたはコード化ビデオシーケンスの（復号順序で）前の部分の復号中に取得されたメタ情報に応答するだけでなく、以前に復元およびループフィルタリングされたサンプル値に応答することもできる。

40

【0048】

ループフィルタユニット（556）の出力は、レンダリングデバイス（512）に出力され

50

るだけでなく、将来のインターピクチャ予測で使用するために参照ピクチャメモリ（557）に格納することができるサンプルストリームであり得る。

【0049】

特定のコード化ピクチャは、完全に復元されると、将来の予測のために参照ピクチャとして使用することができる。例えば、現在ピクチャに対応するコード化ピクチャが完全に復元され、コード化ピクチャが参照ピクチャとして（例えば、パーサー（520）によって）識別されると、現在ピクチャバッファ（558）は、参照ピクチャメモリ（557）の一部になることができ、未使用の現在ピクチャバッファは、次のコード化ピクチャの復元を開始する前に再割当てすることができる。

【0050】

ビデオデコーダ（510）は、ITU-T Rec. H. 265などの規格における所定のビデオ圧縮技術に従って復号動作を実行することができる。コード化ビデオシーケンスがビデオ圧縮技術または規格の構文とビデオ圧縮技術において文書化されたプロファイルの両方を順守するという意味で、コード化ビデオシーケンスは、使用されているビデオ圧縮技術または規格によって指定された構文に準拠することができる。具体的には、プロファイルは、ビデオ圧縮技術または規格で使用可能なすべてのツールから、そのプロファイル下で使用するために利用可能な唯一のツールとしていくつかのツールを選択することができる。また、準拠するために必要なことは、コード化ビデオシーケンスの複雑さが、ビデオ圧縮技術または規格のレベルによって定義された範囲内にあることであり得る。場合によっては、レベルにより、最大ピクチャサイズ、最大フレームレート、（例えば、1秒あたりのメガサンプル単位で測定された）最大復元サンプルレート、最大参照ピクチャサイズなどが制限される。レベルによって設定される制限は、場合によっては、仮想参照デコーダ（HRD）の仕様、およびコード化ビデオシーケンス内でシグナリングされるHRDバッファ管理用のメタデータによってさらに制限され得る。

【0051】

一実施形態では、受信機（531）は、符号化ビデオとともに追加の（冗長な）データを受信することができる。追加のデータは、コード化ビデオシーケンスの一部として含まれてもよい。追加のデータは、データを適切に復号するために、かつ／または元のビデオデータをより正確に復元するために、ビデオデコーダ（510）によって使用されてもよい。追加のデータは、例えば、時間、空間、または信号ノイズ比（SNR）の拡張層、冗長スライス、冗長ピクチャ、順方向誤り訂正コードなどの形式であり得る。

【0052】

図6は、本開示の一実施形態による、ビデオエンコーダ（603）のブロック図を示す。ビデオエンコーダ（603）は電子デバイス（620）に含まれる。電子デバイス（620）は送信機（640）（例えば、送信回路）を含む。ビデオエンコーダ（603）は、図4の例のビデオエンコーダ（403）の代わりに使用することができる。

【0053】

ビデオエンコーダ（603）は、ビデオエンコーダ（603）によってコード化されるビデオ画像を取り込むことができる（図6の例では電子デバイス（620）の一部ではない）ビデオソース（601）からビデオサンプルを受信することができる。別の例では、ビデオソース（601）は電子デバイス（620）の一部である。

【0054】

ビデオソース（601）は、任意の適切なビット深度（例えば、8ビット、10ビット、12ビット、…）、任意の色空間（例えば、BT. 601 Y CrCb、RGB、…）、および任意の適切なサンプリング構造（例えば、Y CrCb 4:2:0、Y CrCb 4:4:4）であり得るデジタルビデオサンプルストリームの形式で、ビデオエンコーダ（603）によってコード化されるソースビデオシーケンスを提供することができる。メディアサービングシステムでは、ビデオソース（601）は、以前に準備されたビデオを格納するストレージデバイスであってもよい。ビデオ会議システムでは、ビデオソース（601）は、ビデオシーケンスとしてローカル画像情報を取り込むカメラであってもよい。ビデオデータは、順番に見たとき

10

20

30

40

50

に動きを伝える複数の個別のピクチャとして提供されてもよい。ピクチャ自体は、ピクセルの空間配列として編成されてよく、各ピクセルは、使用中のサンプリング構造、色空間などに応じて、1つまたは複数のサンプルを含むことができる。当業者は、ピクセルとサンプルとの間の関係を容易に理解することができる。以下の説明はサンプルに焦点を当てる。

【0055】

一実施形態によれば、ビデオエンコーダ(603)は、リアルタイムで、またはアプリケーションによって必要とされる任意の他の時間制約の下で、ソースビデオシーケンスのピクチャをコード化ビデオシーケンス(643)にコード化し圧縮することができる。適切なコーディング速度を強制することは、コントローラ(650)の1つの機能である。いくつかの実施形態では、コントローラ(650)は、以下に記載される他の機能ユニットを制御し、他の機能ユニットに機能的に結合されている。分かりやすくするために、結合は描写されていない。コントローラ(650)によって設定されるパラメータは、レート制御関連パラメータ(ピクチャスキップ、量子化器、レート歪み最適化技法のラムダ値、…)、ピクチャサイズ、ピクチャグループ(GOP)のレイアウト、最大動きベクトル検索範囲などを含むことができる。コントローラ(650)は、特定のシステム設計のために最適化されたビデオエンコーダ(603)に関連する他の適切な機能を有するように構成することができる。

【0056】

いくつかの実施形態では、ビデオエンコーダ(603)は、コーディンググループで動作するように構成される。単純化し過ぎた説明として、一例では、コーディンググループは、(例えば、コード化される入力ピクチャ、および参照ピクチャに基づいて、シンボルストリームなどのシンボルを作成することに関与する)ソースコーダ(630)、ならびにビデオエンコーダ(603)に組み込まれた(ローカル)デコーダ(633)を含むことができる。デコーダ(633)は、(シンボルとコード化ビデオビットストリームとの間のいかなる圧縮も、開示された主題で考慮されるビデオ圧縮技術において可逆であるため)(リモート)デコーダも作成するのと同様の方式で、シンボルを復元してサンプルデータを作成する。復元されたサンプルストリーム(サンプルデータ)は、参照ピクチャメモリ(634)に入力される。シンボルストリームの復号は、デコーダの場所(ローカルまたはリモート)に関係なくビット正確な結果につながるので、参照ピクチャメモリ(634)内のコンテンツも、ローカルエンコーダとリモートエンコーダとの間でビット正確である。言い換えれば、エンコーダの予測部分は、復号中に予測を使用するときにデコーダが「見る」のと同じサンプル値を参照ピクチャサンプルとして「見る」。参照ピクチャの同期性(および、例えば、チャンネルエラーのために同期性が維持できない場合に結果として生じるドリフト)のこの基本原理は、いくつかの関連技術でも使用される。

【0057】

「ローカル」デコーダ(633)の動作は、図5とともに上記で詳細にすでに記載されている、ビデオデコーダ(510)などの「リモート」デコーダの動作と同じであり得る。しかしながら、また図5を簡単に参照すると、シンボルが利用可能であり、エントロピーコーダ(645)およびパーサ(520)によるコード化ビデオシーケンスへのシンボルの符号化／復号は可逆であり得るので、バッファメモリ(515)を含むビデオデコーダ(510)のエントロピー復号部分、およびパーサ(520)は、ローカルデコーダ(633)に完全に実装されていない可能性がある。

【0058】

この時点で行うことができる観察は、デコーダに存在する構文解析／エントロピー復号以外の任意のデコーダ技術も、対応するエンコーダ内に実質的に同一の関数形式で必ず存在する必要があるということである。このため、開示される主題はデコーダの動作に焦点を当てる。エンコーダ技術の説明は、包括的に記載されたデコーダ技術の逆であるため、省略することができる。特定の領域においてのみ、より詳細な説明が必要であり、以下に提供される。

【0059】

動作中、いくつかの例では、ソースコーダ（630）は、「参照ピクチャ」として指定されたビデオシーケンスからの1つまたは複数の以前にコード化されたピクチャを参照して入力ピクチャを予測的にコード化する、動き補償予測コーディングを実行することができる。このようにして、コーディングエンジン（632）は、入力ピクチャのピクセルブロックと、入力ピクチャへの予測参照として選択され得る参照ピクチャのピクセルブロックとの間の差をコード化する。

【0060】

ローカルビデオデコーダ（633）は、ソースコーダ（630）によって作成されたシンボルに基づいて、参照ピクチャとして指定され得るピクチャのコード化ビデオデータを復号することができる。コーディングエンジン（632）の動作は、有利なことに、非可逆プロセスであってもよい。コード化ビデオデータが（図6には示されていない）ビデオデコーダで復号され得るとき、復元されたビデオシーケンスは、通常、いくつかの誤差を伴うソースビデオシーケンスのレプリカであり得る。ローカルビデオデコーダ（633）は、参照ピクチャに対してビデオデコーダによって実行され得る復号プロセスを複製し、復元された参照ピクチャが参照ピクチャキャッシュ（634）に格納されるようにすることができる。このようにして、ビデオエンコーダ（603）は、（送信エラーがない）遠端ビデオデコーダによって取得される復元された参照ピクチャとして共通のコンテンツを有する、復元された参照ピクチャのコピーをローカルに格納することができる。

【0061】

予測器（635）は、コーディングエンジン（632）のための予測検索を実行することができる。すなわち、コード化される新しいピクチャの場合、予測器（635）は、新しいピクチャのための適切な予測参照として役立つことができる、（候補参照ピクセルブロックとしての）サンプルデータまたは参照ピクチャ動きベクトル、ブロック形状などの特定のメタデータを求めて、参照ピクチャメモリ（634）を検索することができる。予測器（635）は、適切な予測参照を見つけるために、ピクセルブロックごとにサンプルブロックに対して動作することができる。場合によっては、予測器（635）によって取得された検索結果によって決定されるように、入力ピクチャは、参照ピクチャメモリ（634）に格納された複数の参照ピクチャから引き出された予測参照を有することができる。

【0062】

コントローラ（650）は、例えば、ビデオデータを符号化するために使用されるパラメータおよびサブグループパラメータの設定を含む、ソースコーダ（630）のコーディング動作を管理することができる。

【0063】

すべての前述の機能ユニットの出力は、エントロピーコーダ（645）内でエントロピーコーディングを受けることができる。エントロピーコーダ（645）は、ハフマンコーディング、可変長コーディング、算術コーディングなどの技術に従ってシンボルを可逆圧縮することにより、様々な機能ユニットによって生成されたシンボルをコード化ビデオシーケンスに変換する。

【0064】

送信機（640）は、エントロピーコーダ（645）によって作成されたコード化ビデオシーケンスをバッファリングして、通信チャネル（660）を介した送信の準備をすることができる。通信チャネル（660）は、符号化ビデオデータを格納するストレージデバイスへのハードウェア／ソフトウェアリンクであってもよい。送信機（640）は、ビデオコーダ（603）からのコード化ビデオデータを、送信される他のデータ、例えば、コード化オーディオデータおよび／または補助データストリーム（ソースは図示されていない）とマージすることができる。

【0065】

コントローラ（650）は、ビデオエンコーダ（603）の動作を管理することができる。コーディング中に、コントローラ（650）は、各コード化ピクチャに特定のコード化ピク

10

20

30

40

50

チャタイプを割り当てることができ、それは、それぞれのピクチャに適用され得るコーディング技法に影響を及ぼす場合がある。例えば、ピクチャは、しばしば、以下のピクチャタイプのうちの1つとして割り当てられてもよい。

【0066】

イントラピクチャ（Iピクチャ）は、予測のソースとしてシーケンス内のいかなる他のピクチャも使用せずにコード化および復号され得るピクチャであり得る。いくつかのビデオコーデックは、例えば、独立デコーダリフレッシュ（「IDR」）ピクチャを含む、様々なタイプのイントラピクチャを可能にする。当業者は、Iピクチャのそれらの変形形態、ならびにそれらのそれぞれの用途および特徴を知っている。

【0067】

予測ピクチャ（Pピクチャ）は、各ブロックのサンプル値を予測するために、多くとも1つの動きベクトルおよび参照インデックスを使用するイントラ予測またはインター予測を使用して、コード化および復号され得るピクチャであり得る。

【0068】

双方向予測ピクチャ（Bピクチャ）は、各ブロックのサンプル値を予測するために、多くとも2つの動きベクトルおよび参照インデックスを使用するイントラ予測またはインター予測を使用して、コード化および復号され得るピクチャであり得る。同様に、複数の予測ピクチャは、単一ブロックの復元のために3つ以上の参照ピクチャおよび関連するメタデータを使用することができる。

【0069】

ソースピクチャは、通常、複数のサンプルブロック（例えば、各々4×4、8×8、4×8、または16×16サンプルのブロック）に空間的に細分化され、ブロックごとにコード化される。ブロックは、ブロックのそれぞれのピクチャに適用されるコーディング割当てによって決定されるように、他の（すでにコード化された）ブロックを参照して予測的にコード化されてもよい。例えば、Iピクチャのブロックは、非予測的にコード化されてもよいか、またはそれらは、同じピクチャのすでにコード化されたブロックを参照して予測的にコード化されてもよい（空間予測もしくはイントラ予測）。Pピクチャのピクセルブロックは、1つの以前にコード化された参照ピクチャを参照して、空間予測を介してまたは時間予測を介して予測的にコード化されてもよい。Bピクチャのブロックは、1つまたは2つの以前にコード化された参照ピクチャを参照して、空間予測を介してまたは時間予測を介して予測的にコード化されてもよい。

【0070】

ビデオエンコーダ（603）は、ITU-T Rec. H. 265などの所定のビデオコーディング技術または規格に従ってコーディング動作を実行することができる。その動作において、ビデオエンコーダ（603）は、入力ビデオシーケンスにおける時間および空間の冗長性を利用する予測コーディング動作を含む、様々な圧縮動作を実行することができる。したがって、コード化されたビデオデータは、使用されているビデオコーディング技術または規格によって指定された構文に準拠することができる。

【0071】

一実施形態では、送信機（640）は、符号化されたビデオとともに追加のデータを送信することができる。ソースコーダ（630）は、コード化ビデオシーケンスの一部としてそのようなデータを含んでもよい。追加のデータは、時間／空間／SNR拡張層、冗長なピクチャおよびスライスなどの他の形式の冗長データ、SEIメッセージ、VUIパラメータセットフラグメントなどを含んでもよい。

【0072】

ビデオは、時系列で複数のソースピクチャ（ビデオピクチャ）として取り込まれてもよい。（しばしば、イントラ予測と省略される）イントラピクチャ予測は、所与のピクチャ内の空間の相関関係を利用し、インターピクチャ予測は、ピクチャ間の（時間または他の）相関関係を利用する。一例では、現在ピクチャと呼ばれる、符号化／復号中の特定のピクチャがブロックに分割される。現在ピクチャ内のブロックが、以前にコード化され、ピ

10

20

30

40

50

デオ内にまだバッファリングされている参照ピクチャ内の参照ブロックに類似しているとき、現在ピクチャ内のブロックは、動きベクトルと呼ばれるベクトルによってコード化することができる。動きベクトルは、参照ピクチャ内の参照ブロックを指し、複数の参照ピクチャが使用されている場合、参照ピクチャを識別する第3の次元を有することができる。
【0073】

いくつかの実施形態では、インターピクチャ予測において双予測技法を使用することができる。双予測技法によれば、両方ともビデオ内の現在ピクチャよりも復号順序で前にある（が、それぞれ、表示順序で過去および将来であり得る）第1の参照ピクチャおよび第2の参照ピクチャなどの2つの参照ピクチャが使用される。現在ピクチャ内のブロックは、第1の参照ピクチャ内の第1の参照ブロックを指す第1の動きベクトル、および第2の参照ピクチャ内の第2の参照ブロックを指す第2の動きベクトルによってコード化することができる。ブロックは、第1の参照ブロックと第2の参照ブロックの組合せによって予測することができる。

10

【0074】

さらに、コーディング効率を改善するために、インターピクチャ予測においてマージモード技法を使用することができる。

【0075】

本開示のいくつかの実施形態によれば、インターピクチャ予測およびイントラピクチャ予測などの予測は、ブロックの単位で実行される。例えば、HEVC規格によれば、ビデオピクチャのシーケンス内のピクチャは、圧縮のためにコーディングツリーユニット（CTU）に分割され、ピクチャ内のCTUは、 64×64 ピクセル、 32×32 ピクセル、または 16×16 ピクセルなどの同じサイズを有する。一般に、CTUは3つのコーディングツリーブロック（CTB）を含み、それらは1つの輝度CTBおよび2つの彩度CTBである。各CTUは、1つまたは複数のコーディングユニット（CU）に再帰的に四分木分割することができる。例えば、 64×64 ピクセルのCTUは、1つの 64×64 ピクセルのCU、または4つの 32×32 ピクセルのCU、または16個の 16×16 ピクセルのCUに分割することができる。一例では、インター予測タイプまたはイントラ予測タイプなどのCUの予測タイプを決定するために、各CUが分析される。CUは、時間および／または空間の予測可能性に応じて、1つまたは複数の予測ユニット（PU）に分割される。一般に、各PUは、1つの輝度予測ブロック（PB）および2つの彩度PBを含む。一実施形態では、コーディング（符号化／復号）における予測動作は、予測ブロックの単位で実行される。予測ブロックの一例として輝度予測ブロックを使用すると、予測ブロックは、 8×8 ピクセル、 16×16 ピクセル、 8×16 ピクセル、 16×8 ピクセルなどのピクセルの値（例えば、輝度値）の行列を含む。

20

【0076】

図7は、本開示の別の実施形態による、ビデオエンコーダ（703）の図を示す。ビデオエンコーダ（703）は、ビデオピクチャのシーケンス内の現在ビデオピクチャ内のサンプル値の処理ブロック（例えば、予測ブロック）を受信し、処理ブロックをコード化ビデオシーケンスの一部であるコード化ピクチャに符号化するように構成される。一例では、ビデオエンコーダ（703）は、図4の例のビデオエンコーダ（403）の代わりに使用される。

【0077】

30

HEVCの例では、ビデオエンコーダ（703）は、 8×8 サンプルの予測ブロックなどの処理ブロック用のサンプル値の行列を受信する。ビデオエンコーダ（703）は、処理ブロックが、例えば、レート歪み最適化を使用して、イントラモード、インターモード、または双予測モードを使用して最良にコード化されるかどうかを判定する。処理ブロックがイントラモードでコード化されるとき、ビデオエンコーダ（703）は、イントラ予測技法を使用して、処理ブロックをコード化ピクチャに符号化することができ、処理ブロックがインターモードまたは双予測モードでコード化されるとき、ビデオエンコーダ（703）は、それぞれ、インター予測技法または双予測技法を使用して、処理ブロックをコード化ピクチャに符号化することができる。特定のビデオコーディング技術では、マージモードは、予測器の外側のコード化された動きベクトル成分の利点がない、動きベクトルが1つまたは

40

50

複数の動きベクトル予測器から導出されるインターピクチャ予測サブモードであり得る。特定の他のビデオコーディング技術では、対象ブロックに適用可能な動きベクトル成分が存在してもよい。一例では、ビデオエンコーダ（703）は、処理ブロックのモードを決定するためにモード決定モジュール（図示せず）などの他の構成要素を含む。

【0078】

図7の例では、ビデオエンコーダ（703）は、図7に示されたように一緒に結合されたインターエンコーダ（730）、イントラエンコーダ（722）、残差計算器（723）、スイッチ（726）、残差エンコーダ（724）、汎用コントローラ（721）、およびエントロピーエンコーダ（725）を含む。

【0079】

インターエンコーダ（730）は、現在ブロック（例えば、処理ブロック）のサンプルを受信し、ブロックを参照ピクチャ内の1つまたは複数の参照ブロック（例えば、前のピクチャおよび後のピクチャ内のブロック）と比較し、インター予測情報（例えば、インター符号化技法による冗長情報、動きベクトル、マージモード情報の記述）を生成し、任意の適切な技法を使用して、インター予測情報に基づいてインター予測結果（例えば、予測ブロック）を計算するように構成される。いくつかの例では、参照ピクチャは、符号化されたビデオ情報に基づいて復号された復号参照ピクチャである。

【0080】

イントラエンコーダ（722）は、現在ブロック（例えば、処理ブロック）のサンプルを受信し、場合によっては、ブロックを同じピクチャ内のすでにコード化されたブロックと比較し、変換後に量子化係数を生成し、場合によっては、イントラ予測情報（例えば、1つまたは複数のイントラ符号化技法によるイントラ予測方向情報）も生成するように構成される。一例では、イントラエンコーダ（722）はまた、同じピクチャ内のイントラ予測情報および参照ブロックに基づいて、イントラ予測結果（例えば、予測ブロック）を計算する。

【0081】

汎用コントローラ（721）は、汎用制御データを決定し、汎用制御データに基づいてビデオエンコーダ（703）の他の構成要素を制御するように構成される。一例では、汎用コントローラ（721）は、ブロックのモードを決定し、モードに基づいてスイッチ（726）に制御信号を提供する。例えば、モードがイントラモードであるとき、汎用コントローラ（721）は、スイッチ（726）を制御して残差計算器（723）が使用するためのイントラモード結果を選択し、エントロピーエンコーダ（725）を制御してイントラ予測情報を選択し、ビットストリームにイントラ予測情報を含め、モードがインターモードであるとき、汎用コントローラ（721）は、スイッチ（726）を制御して残差計算器（723）が使用するためのインター予測結果を選択し、エントロピーエンコーダ（725）を制御してインター予測情報を選択し、ビットストリームにインター予測情報を含める。

【0082】

残差計算器（723）は、受信ブロックと、イントラエンコーダ（722）またはインターエンコーダ（730）から選択された予測結果との間の差（残差データ）を計算するように構成される。残差エンコーダ（724）は、残差データを符号化して変換係数を生成するために、残差データに基づいて動作するように構成される。一例では、残差エンコーダ（724）は、残差データを空間領域から周波数領域に変換し、変換係数を生成するように構成される。次いで、変換係数は、量子化変換係数を取得するために量子化処理を受ける。様々な実施形態では、ビデオエンコーダ（703）は残差デコーダ（728）も含む。残差デコーダ（728）は、逆変換を実行し、復号された残差データを生成するように構成される。復号された残差データは、イントラエンコーダ（722）およびインターエンコーダ（730）によって適切に使用することができる。例えば、インターエンコーダ（730）は、復号された残差データおよびインター予測情報に基づいて復号されたブロックを生成することができ、イントラエンコーダ（722）は、復号された残差データおよびイントラ予測情報に基づいて復号されたブロックを生成することができる。復号されたブロックは、復号さ

10

20

30

40

50

れたピクチャを生成するために適切に処理され、復号されたピクチャは、メモリ回路（図示せず）にバッファリングされ、いくつかの例では参照ピクチャとして使用することができる。

【0083】

エントロピーエンコーダ（725）は、符号化されたブロックを含めるようにビットストリームをフォーマットするように構成される。エントロピーエンコーダ（725）は、HEVC規格などの適切な規格に従って様々な情報を含むように構成される。一例では、エントロピーエンコーダ（725）は、汎用制御データ、選択された予測情報（例えば、イントラ予測情報またはインター予測情報）、残差情報、およびビットストリーム内の他の適切な情報を含むように構成される。開示された主題によれば、インターモードまたは双予測モードのいずれかのマージサブモードでブロックをコーディングするときに残差情報が存在しないことに留意されたい。

10

【0084】

図8は、本開示の別の実施形態による、ビデオデコーダ（810）の図を示す。ビデオデコーダ（810）は、コード化ビデオシーケンスの一部であるコード化ピクチャを受信し、コード化ピクチャを復号して復元されたピクチャを生成するように構成される。一例では、ビデオデコーダ（810）は、図4の例のビデオデコーダ（410）の代わりに使用される。

【0085】

図8の例では、ビデオデコーダ（810）は、図8に示されたように一緒に結合されたエントロピーデコーダ（871）、インターデコーダ（880）、残差デコーダ（873）、復元モジュール（874）、およびイントラデコーダ（872）を含む。

20

【0086】

エントロピーデコーダ（871）は、コード化ピクチャから、コード化ピクチャが構成される構文要素を表す特定のシンボルを復元するように構成することができる。そのようなシンボルは、例えば、（例えば、イントラモード、インターモード、双予測モード、マージサブモードまたは別のサブモードの中の後者2つなどの）ブロックがコード化されるモード、それぞれ、イントラデコーダ（872）またはインターデコーダ（880）による予測に使用される特定のサンプルまたはメタデータを識別することができる（例えば、イントラ予測情報またはインター予測情報などの）予測情報、例えば、量子化変換係数の形態の残差情報などを含むことができる。一例では、予測モードがインターモードまたは双予測モードであるとき、インター予測情報はインターデコーダ（880）に提供され、予測タイプがイントラ予測タイプであるとき、イントラ予測情報はイントラデコーダ（872）に提供される。残差情報は逆量子化を受けることができ、残差デコーダ（873）に提供される。

30

【0087】

インターデコーダ（880）は、インター予測情報を受信し、インター予測情報に基づいてインター予測結果を生成するように構成される。

【0088】

イントラデコーダ（872）は、イントラ予測情報を受信し、イントラ予測情報に基づいて予測結果を生成するように構成される。

40

【0089】

残差デコーダ（873）は、逆量子化を実行して逆量子化変換係数を抽出し、逆量子化変換係数を処理して、残差を周波数領域から空間領域に変換するように構成される。残差デコーダ（873）はまた、（量子化器パラメータ（QP）を含めるために）特定の制御情報を必要とする場合があり、その情報は、エントロピーデコーダ（871）によって提供される場合がある（これは、少量の制御情報のみである可能性があるため、データパスは描写されていない）。

【0090】

復元モジュール（874）は、空間領域において、残差デコーダ（873）によって出力された残差と（場合によってはインター予測モジュールまたはイントラ予測モジュールによ

50

って出力された) 予測結果を組み合わせ、復元されたピクチャの一部であり得る復元されたブロックを形成し、同様に、復元されたピクチャは復元されたビデオの一部であり得る。見栄えを改善するために、デブロッキング動作などの他の適切な動作が実行できることに留意されたい。

【0091】

ビデオエンコーダ(403)、(603)、および(703)、ならびにビデオデコーダ(410)、(510)、および(810)は、任意の適切な技法を使用して実装され得ることに留意されたい。一実施形態では、ビデオエンコーダ(403)、(603)、および(703)、ならびにビデオデコーダ(410)、(510)、および(810)は、1つまたは複数の集積回路を使用して実装することができる。別の実施形態では、ビデオエンコーダ(403)、(603)、および(703)、ならびにビデオデコーダ(410)、(510)、および(810)は、ソフトウェア命令を実行する1つまたは複数のプロセッサを使用して実装することができる。

10

【0092】

本開示の態様は、ビデオコーディング／復号のためのフィルタリング技法を提供する。

【0093】

ブロックベースフィルタ適応を伴う適応ループフィルタ(ALF)は、アーチファクトを低減するためにエンコーダ／デコーダによって適用することができる。輝度成分の場合、例えば、局所的な勾配の方向および活動に基づいて、4×4の輝度ブロックに対して複数のフィルタ(例えば、25個のフィルタ)のうちの1つを選択することができる。

20

【0094】

ALFは、任意の適切な形状およびサイズを有することができる。図9を参照すると、ALF(910)～(911)は、ALF(910)の5×5の菱形形状およびALF(911)の7×7の菱形形状などの菱形形状を有する。ALF(910)では、要素(920)～(932)は菱形形状を形成し、フィルタリングプロセスにおいて使用することができる。要素(920)～(932)には、7つの値(例えば、C0～C6)を使用することができる。ALF(911)では、要素(940)～(964)は菱形形状を形成し、フィルタリングプロセスにおいて使用することができる。要素(940)～(964)には、13個の値(例えば、C0～C12)を使用することができる。

【0095】

図9を参照すると、いくつかの例では、菱形フィルタ形状を有する2つのALF(910)～(911)が使用される。5×5の菱形形状フィルタ(910)は、彩度成分(例えば、彩度ブロック、彩度CB)に適用することができる。7×7の菱形形状フィルタ(911)は、輝度成分(例えば、輝度ブロック、輝度CB)に適用することができる。ALFにおいて他の適切な形状およびサイズを使用することができる。例えば、9×9の菱形形状フィルタを使用することができる。

30

【0096】

値(例えば、(910)のC0～C6または(920)のC0～C12)によって示された位置にあるフィルタ係数は、非ゼロであり得る。さらに、ALFがクリッピング関数を含む場合、それらの位置にあるクリッピング値は非ゼロであり得る。

40

【0097】

輝度成分のブロック分類の場合、4×4のブロック(または輝度ブロック、輝度CB)は、複数(例えば、25個)のクラスのうちの一つとしてカテゴリズまたは分類することができる。分類インデックスCは、式(1)を使用して、方向性パラメータDおよび活性値Aの量子化値

【数1】

$$\hat{A}$$

50

に基づいて導出することができる。

【数 2】

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

方向性パラメータDおよび量子化値

【数 3】

$$\hat{A}$$

を計算するために、垂直方向、水平方向、ならびに2つの対角方向（例えば、d1およびd2）の勾配 g_v 、 g_h 、 g_{d1} 、および g_{d2} は、それぞれ、以下のように1-Dラプラシアンを使用して計算することができる。

【数 4】

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

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

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

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

ここで、添え字iおよびjは、4×4ブロック内の左上サンプルの座標を指し、R(k,l)は、座標(k,l)にある復元サンプルを示す。方向（例えば、d1およびd2）は、2つの対角方向を指すことができる。

【0098】

上述されたブロック分類の複雑さを低減するために、サブサンプリングされた1-Dラプラシアン計算を適用することができる。図10A～図10Dは、それぞれ、垂直方向（図10A）、水平方向（図10B）、ならびに2つの対角方向d1（図10C）およびd2（図10D）の勾配 g_v 、 g_h 、 g_{d1} 、および g_{d2} を計算するために使用されるサブサンプリングされた位置の例を示す。同じサブサンプリングされた位置は、異なる方向の勾配計算に使用することができる。図10Aでは、ラベル「V」は、垂直勾配 g_v を計算するためのサブサンプリングされた位置を示す。図10Bでは、ラベル「H」は、水平勾配 g_h を計算するためのサブサンプリングされた位置を示す。図10Cでは、ラベル「D1」は、d1の対角勾配 g_{d1} を計算するためのサブサンプリングされた位置を示す。図10Dでは、ラベル「D2」は、d2の対角勾配 g_{d2} を計算するためのサブサンプリングされた位置を示す。

【0099】

水平方向の勾配 g_h および垂直方向の勾配 g_v の最大値

【数 5】

$$g_{h,v}^{max}$$

および最小値

【数 6】

$$g_{h,v}^{min}$$

10

は、以下のように設定することができる。

【数 7】

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

2つの対角方向の勾配 g_{d1} および g_{d2} の最大値

20

【数 8】

$$g_{d1,d2}^{max}$$

および最小値

【数 9】

$$g_{d1,d2}^{min}$$

30

は、以下のように設定することができる。

【数 10】

$$g_{d1,d2}^{max} = \max(g_{d1}, g_{d2}), \quad g_{d1,d2}^{min} = \min(g_{d1}, g_{d2}) \quad \text{式 (7)}$$

40

方向性パラメータDは、上記の値、ならびに以下のような2つのしきい値 t_1 および t_2 に基づいて導出することができる。

ステップ1. (1)

【数 11】

$$g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$$

および (2)

50

【数 1 2】

$$g_{d1,d2}^{max} \leq t_1 \cdot g_{d1,d2}^{min}$$

が真である場合、Dは0に設定される。

ステップ2.

【数 1 3】

$$g_{h,v}^{max} / g_{h,v}^{min} > g_{d1,d2}^{max} / g_{d1,d2}^{min}$$

10

の場合、ステップ3に進み、そうでない場合、ステップ4に進む。

ステップ3.

【数 1 4】

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

20

の場合、Dは2に設定され、そうでない場合、Dは1に設定される。

ステップ4.

【数 1 5】

$$g_{d1,d2}^{max} > t_2 \cdot g_{d1,d2}^{min}$$

30

の場合、Dは4に設定され、そうでない場合、Dは3に設定される。

【0 1 0 0】

活性値Aは、

【数 1 6】

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

のように計算することができる。

40

Aはさらに、両端を含む0～4の範囲に量子化することができ、量子化値は

【数 1 7】

$$\hat{A}$$

として表記される。

【0 1 0 1】

ピクチャ内の彩度成分の場合、ブロック分類は適用されず、したがって、彩度成分ごと

50

に単一のセットのALF係数を適用することができる。

【0102】

幾何学的変換は、フィルタ係数および対応する（クリッピング値とも呼ばれる）フィルタクリッピング値に適用することができる。ブロック（例えば、 4×4 の輝度ブロック）をフィルタリングする前に、例えば、ブロックに対して計算された勾配値（例えば、 g_v 、 g_h 、 g_{d1} 、および／または g_{d2} ）に応じて、回転または対角反転および垂直反転などの幾何学的変換をフィルタ係数 $f(k, l)$ および対応するフィルタクリッピング値 $c(k, l)$ に適用することができる。フィルタ係数 $f(k, l)$ および対応するフィルタクリッピング値 $c(k, l)$ に適用される幾何学的変換は、フィルタによってサポートされる領域内のサンプルに幾何学的変換を適用することと等価であり得る。幾何学的変換は、それぞれの方向性を整列させることによって、ALFが適用される異なるブロックをより類似させることができる。

10

【0103】

それぞれ、式（9）～（11）によって記載されるように、対角反転、垂直反転、および回転を含む3つの幾何学的変換を実行することができる。

$$f_D(k, l) = f(l, k), c_D(k, l) = c(l, k) \quad \text{式 (9)}$$

$$f_V(k, l) = f(k, K-l-1), c_V(k, l) = c(k, K-l-1) \quad \text{式 (10)}$$

$$f_R(k, l) = f(K-l-1, k), c_R(k, l) = c(K-l-1, k) \quad \text{式 (11)}$$

ここで、 K はALFまたはフィルタのサイズであり、 $0 \leq k, l \leq K-1$ は係数の座標である。例えば、フィルタ f またはクリッピング値行列（もしくはクリッピング行列） c の左上隅に位置（0, 0）があり、右下隅に位置（ $K-1, K-1$ ）がある。変換は、ブロックに対して計算された勾配値に応じて、フィルタ係数 $f(k, l)$ およびクリッピング値 $c(k, l)$ に適用することができる。変換と4つの勾配との間の関係の一例が表1に要約されている。

20

【0104】

【表1】

表1：ブロックに対して計算された勾配と変換のマッピング

勾配値	変換
$g_{d2} < g_{d1}$ か $g_h < g_v$	変換なし
$g_{d2} < g_{d1}$ か $g_v < g_h$	対角反転
$g_{d1} < g_{d2}$ か $g_h < g_v$	垂直反転
$g_{d1} < g_{d2}$ か $g_v < g_h$	回転

30

【0105】

いくつかの実施形態では、ALFフィルタパラメータは、ピクチャ用の適応パラメータセット（APS）内でシグナリングされる。APSでは、輝度フィルタ係数およびクリッピング値インデックスの1つまたは複数のセット（例えば、最大25セット）をシグナリングすることができる。一例では、1つまたは複数のセットのうちのセットは、輝度フィルタ係数および1つまたは複数のクリッピング値インデックスを含むことができる。彩度フィルタ係数およびクリッピング値インデックスの1つまたは複数のセット（例えば、最大8セット）をシグナリングすることができる。シグナリングオーバーヘッドを低減するために、輝度成分についての（例えば、異なる分類インデックスを有する）異なる分類のフィルタ係数をマージすることができる。スライスヘッダでは、現在のスライスに使用されるAPSのインデックスをシグナリングすることができる。

40

【0106】

一実施形態では、（クリッピングインデックスとも呼ばれる）クリッピング値インデックスをAPSから復号することができる。クリッピング値インデックスは、例えば、クリッピング値インデックスと対応するクリッピング値との間の関係に基づいて、対応するクリッピング値を決定するために使用することができる。関係は事前定義され、デコーダに格納することができる。一例では、関係は、クリッピング値インデックスおよび対応するクリッピング値の（例えば、輝度CBに使用される）輝度テーブル、クリッピング値インデッ

50

クスおよび対応するクリッピング値の（例えば、彩度CBに使用される）彩度テーブルなどのテーブルによって記述される。クリッピング値はビット深度Bに依存する可能性がある。ビット深度Bは、内部ビット深度、フィルタリングされるCB内の復元サンプルのビット深度などを指すことができる。いくつかの例では、表（例えば、輝度テーブル、彩度テーブル）は、式（12）を使用して取得される。

【数 1 8】

$$\text{AlfClip} = \left\{ \text{round} \left(2^{\beta \frac{N-n+1}{N}} \right) \text{ for } n \in [1..N] \right\}, \quad \text{式 (12)}$$

10

ここで、AlfClipはクリッピング値であり、Bはビット深度（例えば、bitDepth）であり、N（例えば、N=4）は許容されるクリッピング値の数であり、(n-1)は（クリッピングインデックスまたはclipIdxとも呼ばれる）クリッピング値インデックスである。表2は、N=4で式（12）を使用して取得された表の一例を示す。クリッピングインデックス(n-1)は、表2の0、1、2、および3であり得、nは、それぞれ、1、2、3、および4であり得る。表2は、輝度ブロックまたは彩度ブロックに使用することができる。

【 0 1 0 7 】

【表 2】

20

表 2—AlfClip はビット深度 B および clipIdx に依存する可能性がある

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

30

【 0 1 0 8 】

現在スライス用のスライスヘッダでは、現在スライスに使用することができる輝度フィルタセットを指定するために、1つまたは複数のAPSインデックス（例えば、最大7個のAPSインデックス）をシグナリングすることができる。フィルタリングプロセスは、ピクチャレベル、スライスレベル、CTBレベルなどの1つまたは複数の適切なレベルで制御することができる。一実施形態では、フィルタリングプロセスは、CTBレベルでさらに制御することができる。ALFが輝度CTBに適用されるかどうかを示すためにフラグをシグナリングすることができる。輝度CTBは、APS内でシグナリングされる複数の固定フィルタセット（例えば、16個の固定フィルタセット）および（シグナリングされたフィルタセットとも呼ばれる）フィルタセットの中からフィルタセットを選択することができる。適用されるフィルタセット（例えば、複数の固定フィルタセットおよびシグナリングされたフィルタセットの中のフィルタセット）を示すために、輝度CTBに対してフィルタセットインデックスをシグナリングすることができる。複数の固定フィルタセットは、エンコーダおよびデコーダにおいて事前定義およびハードコーディングすることができ、事前定義フィルタセットと呼ぶことができる。

40

【 0 1 0 9 】

50

彩度成分の場合、現在スライスに使用される彩度フィルタセットを示すために、スライスヘッダ内でAPSインデックスをシグナリングすることができる。CTBレベルにおいて、APS内に2つ以上の彩度フィルタセットが存在する場合、彩度CTBごとにフィルタセットインデックスをシグナリングすることができる。

【0 1 1 0】

フィルタ係数は、128に等しいノルムで量子化することができる。乗算の複雑さを減少させるために、非中心位置の係数値が両端を含む $-27 \sim 27-1$ の範囲内であり得るように、ビットストリーム適合性を適用することができる。一例では、中心位置係数はビットストリーム内でシグナリングされず、128に等しいと見なすことができる。

【0 1 1 1】

いくつかの実施形態では、クリッピングインデックスおよびクリッピング値の構文およびセマンティクスは以下のように定義される。

$\text{alf_luma_clip_idx}[\text{sflIdx}][j]$ は、 sflIdx によって示されるシグナリングされた輝度フィルタの j 番目の係数を乗算す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指定するために使用することができる。ビットストリーム適合性の要件は、 $\text{sflIdx} = 0 \sim \text{alf_luma_num_filters_signalled_minus1}$ および $j = 0 \sim 11$ である $\text{alf_luma_clip_idx}[\text{sflIdx}][j]$ の値が、両端を含む $0 \sim 3$ の範囲内にあるべきことを含むことができる。

$\text{filtIdx} = 0 \sim \text{NumAlfFilters} - 1$ および $j = 0 \sim 11$ である、要素 $\text{AlfClipL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ を有する輝度フィルタクリッピング値 $\text{AlfClipL}[\text{adaptation_parameter_set_id}]$ は、 BitDepthY に等しい bitDepth セットおよび $\text{alf_luma_clip_idx}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}][j]]$ に等しい clipIdx セットに応じて、表2に指定されたように導出することができる。

$\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ は、インデックス altIdx を有する代替彩度フィルタの j 番目の係数を乗算す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指定するために使用することができる。ビットストリーム適合性の要件は、 $\text{altIdx} = 0 \sim \text{alf_chroma_num_alt_filters_minus1}$ 、 $j = 0 \sim 5$ である $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ の値が、両端を含む $0 \sim 3$ の範囲内にあるべきことを含むことができる。

$\text{altIdx} = 0 \sim \text{alf_chroma_num_alt_filters_minus1}$ 、 $j = 0 \sim 5$ である、要素 $\text{AlfClipC}[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ を有する彩度フィルタクリッピング値 $\text{AlfClipC}[\text{adaptation_parameter_set_id}][\text{altIdx}]$ は、 BitDepthC に等しい bitDepth セットおよび $\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ に等しい clipIdx セットに応じて、表2に指定されたように導出することができる。

【0 1 1 2】

一実施形態では、フィルタリングプロセスは以下のように記載することができる。デコーダ側において、ALFがCTBに対して有効にされると、CU（またはCB）内のサンプル $R(i, j)$ をフィルタリングすることができ、式(13)を使用して以下に示されるようにフィルタリングされたサンプル値 $R'(i, j)$ が得られる。一例では、CU内の各サンプルがフィルタリングされる。

【数 1 9】

$$R'(i, j) = R(i, j) + \left(\left(\sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i + k, j + l) - R(i, j), c(k, l)) + 64 \right) \gg 7 \right) \quad \text{式 (13)}$$

ここで、 $f(k, l)$ は復号されたフィルタ係数を表記し、 $K(x, y)$ はクリッピング関数であり、 $c(k, l)$ は復号されたクリッピングパラメータ（またはクリッピング値）を表記する。変数 k および l は、 $-L/2$ と $L/2$ との間で変化することができ、 L はフィルタ長を表

記する。クリッピング関数 $K(x, y) = \min(y, \max(-y, x))$ は、クリッピング関数 $\text{Clip3}(-y, y, x)$ に対応する。クリッピング関数 $K(x, y)$ を組み込むことにより、ループフィルタリング法（例えば、ALF）は非線形プロセスになり、非線形ALFと呼ぶことができる。

【0113】

非線形ALFでは、クリッピング値の複数のセットを表3に提供することができる。一例では、輝度セットは4つのクリッピング値 {1024, 181, 32, 6} を含み、彩度セットは4つのクリッピング値 {1024, 161, 25, 4} を含む。輝度セット内の4つのクリッピング値は、輝度ブロック用の（10ビットでコード化された）サンプル値の全範囲（例えば、1024）を、対数領域内でほぼ等しく分割することによって選択することができる。範囲は、彩度セットに対して4から1024までであり得る。

【0114】

【表3】

表3—クリッピング値の例

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

【0115】

選択されたクリッピング値は、以下のように「alf_data」構文要素内でコード化することができる：表3に示されたような選択されたクリッピング値に対応するクリッピングインデックスを符号化するために、適切な符号化方式（例えば、ゴロム符号化方式）を使用することができる。符号化方式は、フィルタセットインデックスを符号化するために使用されるのと同じ符号化方式であってもよい。

【0116】

一実施形態では、ALFのラインバッファ要件を低減するために、仮想境界フィルタリングプロセスを使用することができる。したがって、CTU境界（例えば、水平CTU境界）付近のサンプルに対して、修正されたブロック分類およびフィルタリングを採用することができる。仮想境界（1130）は、図11Aに示されたように、水平CTU境界（1120）を「Nsamples」サンプルだけシフトすることによって線として定義することができ、Nsamplesは正の整数であり得る。一例では、Nsamplesは輝度成分の場合は4に等しく、Nsamplesは彩度成分の場合は2に等しい。

【0117】

図11Aを参照すると、修正されたブロック分類を輝度成分に適用することができる。一例では、仮想境界（1130）の上の4×4ブロック（1110）の1Dラプラシアン勾配計算では、仮想境界（1130）の上のサンプルのみが使用される。同様に、図11Bを参照すると、CTU境界（1121）からシフトされた仮想境界（1131）の下側の4×4ブロック（1111）の1Dラプラシアン勾配計算では、仮想境界（1131）の下側のサンプルのみが使用される。活性値Aの量子化は、それに応じて、1Dラプラシアン勾配計算において使用されるサンプル数の減少を考慮することによってスケーリングすることができる。

【0118】

フィルタリングプロセスのために、仮想境界における対称パディング演算は、輝度成分および彩度成分の両方に使用することができる。図12A～図12Fは、仮想境界にある輝度成分に対するそのような修正されたALFフィルタリングの例を示す。フィルタリングされているサンプルが仮想境界の下に位置するとき、仮想境界の上に位置する隣接サンプルをパディングすることができる。フィルタリングされているサンプルが仮想境界の上に位置するとき、仮想境界の下に位置する隣接サンプルをパディングすることができる。図12Aを参照すると、隣接サンプルC0は、仮想境界（1210）の下に位置するサンプルC2でパデ

ィングすることができる。図12Bを参照すると、隣接サンプルC0は、仮想境界（1220）の上に位置するサンプルC2でパディングすることができる。図12Cを参照すると、隣接サンプルC1～C3は、それぞれ、仮想境界（1230）の下に位置するサンプルC5～C7でパディングすることができる。図12Dを参照すると、隣接サンプルC1～C3は、それぞれ、仮想境界（1240）の上に位置するサンプルC5～C7でパディングすることができる。図12Eを参照すると、隣接サンプルC4～C8は、それぞれ、仮想境界（1250）の下に位置するサンプルC10、C11、C12、C11、およびC10でパディングすることができる。図12Fを参照すると、隣接サンプルC4～C8は、それぞれ、仮想境界（1260）の上に位置するサンプルC10、C11、C12、C11、およびC10でパディングすることができる。

【0119】

いくつかの例では、サンプルおよび隣接サンプルが仮想境界の左（または右）および右（または左）に位置するとき、上記の説明を適切に適合させることができる。

【0120】

本開示の態様によれば、コーディング効率を改善するために、フィルタリングプロセスに基づいてピクチャを区分することができる。いくつかの例では、CTUは最大コーディングユニット（LCU）とも呼ばれる。一例では、CTUまたはLCUは、64×64ピクセルのサイズを有することができる。いくつかの実施形態では、LCU整列ピクチャ四分木分割は、フィルタリングベースのパーティションに使用することができる。いくつかの例では、コーディングユニット同期ピクチャ四分木ベースの適応ループフィルタを使用することができる。例えば、輝度ピクチャはいくつかのマルチレベル四分木パーティションに分割することができ、各パーティションの境界はLCUの境界に整列される。各パーティションは、それ自体のフィルタリングプロセスを有し、したがって、フィルタユニット（FU）と呼ばれる場合がある。

【0121】

いくつかの例では、2パス符号化フローを使用することができる。2パス符号化フローの第1のパスにおいて、ピクチャの四分木分割パターンおよび各FUの最良フィルタを決定することができる。いくつかの実施形態では、ピクチャの四分木分割パターンの決定およびFU用の最良フィルタの決定は、フィルタリング歪みに基づく。フィルタリング歪みは、決定プロセス中に高速フィルタリング歪み推定（FFDE）技法によって推定することができる。ピクチャは、四分木パーティションを使用して区分される。決定された四分木分割パターンおよびすべてのFUの選択されたフィルタに従って、復元ピクチャをフィルタリングすることができる。

【0122】

2パス符号化フローの第2のパスにおいて、CU同期ALFのオン／オフ制御が実行される。ALFのオン／オフ結果に従って、最初にフィルタリングされたピクチャが復元ピクチャによって部分的に回復される。

【0123】

具体的には、いくつかの例では、レート歪み基準を使用して画像をマルチレベル四分木パーティションに分割するために、トップダウン分割手順が採用される。各パーティションはフィルタユニット（FU）と呼ばれる。分割プロセスは、四分木パーティションをLCUの境界に整列させる。FUの符号化順序は、z走査順序に従う。

【0124】

図13は、本開示のいくつかの実施形態によるパーティション例を示す。図13の例では、ピクチャ（1300）が10個のFUに分割され、符号化順序は、FU0、FU1、FU2、FU3、FU4、FU5、FU6、FU7、FU8、およびFU9である。

【0125】

図14は、ピクチャ（1300）用の四分木分割パターン（1400）を示す。図14の例では、ピクチャのパーティションパターンを示すために分割フラグが使用される。例えば、「1」はブロックに対して四分木区分が実行されることを示し、「0」はそれ以上区分されないことを示す。いくつかの例では、最小サイズのFUはLCUサイズを有し、最小サイズのF

10

20

30

40

50

Uには分割フラグは必要でない。分割フラグは、図14に示されたようにz順序で符号化され送信される。

【0126】

いくつかの例では、各FUのフィルタは、レート歪み基準に基づいて2つのフィルタセットから選択される。第1のセットは、現在のFUに対して導出された1/2対称の正方形および菱形のフィルタを有する。第2のセットは、時間遅延フィルタバッファに由来し、時間遅延フィルタバッファは、前のピクチャのFUに対して以前に導出されたフィルタを格納する。これらの2つのセットの最小のレート歪みコストを有するフィルタを、現在のFUに対して選択することができる。同様に、現在のFUが最小のFUではなく、4つの子FUにさらに分割することができる場合、4つの子FUのレート歪みコストが計算される。分割および分割なしの場合のレート歪みコストを再帰的に比較することにより、ピクチャ四分木分割パターンを決定することができる。

【0127】

いくつかの例では、FUの最大数を制限するために最大四分木分割レベルが使用されてもよい。一例では、最大四分木分割レベルが2であるとき、FUの最大数は16である。さらに、四分木分割決定中、最下位の四分木レベルにある16個のFU（最小FU）のウィナー係数を導出するための相関値を再利用することができる。残りのFUは、最下位の四分木レベルにある16個のFUの相関から、それらのウィナーフィルタを導出することができる。したがって、その例では、すべてのFUのフィルタ係数を導出するために、ただ1つのフレームバッファアクセスが実行される。

【0128】

四分木分割パターンが決定された後、フィルタリング歪みをさらに低減するために、CU同期ALFのオン／オフ制御を実行することができる。各リーフCUにおいてフィルタリング歪みと非フィルタリング歪みを比較することにより、リーフCUはその局所領域においてALFのオン／オフを明示的に切り替えることができる。いくつかの例では、ALFのオン／オフ結果に従ってフィルタ係数を再設計することにより、コーディング効率をさらに改善することができる。

【0129】

成分間フィルタリングプロセスは、成分間適応ループフィルタ（CC-ALF）などの成分間フィルタを適用することができる。成分間フィルタは、輝度成分（例えば、輝度CB）の輝度サンプル値を使用して、彩度成分（例えば、輝度CBに対応する彩度CB）を改良することができる。一例では、輝度CBおよび彩度CBはCUに含まれる。

【0130】

図15は、本開示の一実施形態による、彩度成分を生成するために使用される成分間フィルタ（例えば、CC-ALF）を示す。いくつかの例では、図15は、第1の彩度成分（例えば、第1の彩度CB）、第2の彩度成分（例えば、第2の彩度CB）、および輝度成分（例えば、輝度CB）向けのフィルタリングプロセスを示す。輝度成分は、SAOフィルタリングされた輝度成分（1541）を生成するために、サンプル適応オフセット（SAO）フィルタ（1510）によってフィルタリングすることができる。SAOフィルタリングされた輝度成分（1541）は、フィルタリングされた輝度CB（1561）（例えば、「Y」）になるために、ALF輝度フィルタ（1516）によってさらにフィルタリングすることができる。

【0131】

第1の彩度成分は、第1の中間成分（1552）を生成するために、SAOフィルタ（1512）およびALF彩度フィルタ（1518）によってフィルタリングすることができる。さらに、SAOフィルタリングされた輝度成分（1541）は、第2の中間成分（1542）を生成するために、第1の彩度成分用の成分間フィルタ（例えば、CC-ALF）（1521）によってフィルタリングすることができる。その後、第2の中間成分（1542）および第1の中間成分（1552）のうちの少なくとも1つに基づいて、フィルタリングされた第1の彩度成分（1562）（例えば、「Cb」）を生成することができる。一例では、フィルタリングされた第1の彩度成分（1562）（例えば、「Cb」）は、第2の中間成分（1542）および第1の中間

成分（1552）を加算器（1522）と組み合わせることによって生成することができる。第1の彩度成分用の成分間適応ループフィルタリングプロセスは、CC-ALF（1521）によって実行されるステップと、例えば加算器（1522）によって実行されるステップとを含むことができる。

【0132】

上記の説明は、第2の彩度成分に適合させることができる。第2の彩度成分は、第3の中間成分（1553）を生成するために、SAOフィルタ（1514）およびALF彩度フィルタ（1518）によってフィルタリングすることができる。さらに、SAOフィルタリングされた輝度成分（1541）は、第4の中間成分（1543）を生成するために、第2の彩度成分用の成分間フィルタ（例えば、CC-ALF）（1531）によってフィルタリングすることができる。その後、第4の中間成分（1543）および第3の中間成分（1553）のうちの少なくとも1つに基づいて、フィルタリングされた第2の彩度成分（1563）（例えば、「Cr」）を生成することができる。一例では、フィルタリングされた第2の彩度成分（1563）（例えば、「Cr」）は、第4の中間成分（1543）および第3の中間成分（1553）を加算器（1532）と組み合わせることによって生成することができる。一例では、第2の彩度成分用の成分間適応ループフィルタリングプロセスは、CC-ALF（1531）によって実行されるステップと、例えば加算器（1532）によって実行されるステップとを含むことができる。

【0133】

成分間フィルタ（例えば、CC-ALF（1521）、CC-ALF（1531））は、各彩度成分（例えば、第1の彩度成分、第2の彩度成分）を改良するために、任意の適切なフィルタ形状を有する線形フィルタを輝度成分（または輝度チャンネル）に適用することによって動作することができる。

【0134】

図16は、本開示の一実施形態によるフィルタ（1600）の一例を示す。フィルタ（1600）は、非ゼロフィルタ係数およびゼロフィルタ係数を含むことができる。フィルタ（1600）は、（黒塗りの円で示された）フィルタ係数（1610）によって形成された菱形形状（1620）を有する。一例では、フィルタ（1600）内の非ゼロフィルタ係数はフィルタ係数（1610）に含まれ、フィルタ係数（1610）に含まれないフィルタ係数はゼロである。したがって、フィルタ（1600）内の非ゼロフィルタ係数は菱形形状（1620）に含まれ、菱形形状（1620）に含まれないフィルタ係数はゼロである。一例では、フィルタ（1600）のフィルタ係数の数は、フィルタ係数（1610）の数に等しく、それは図16に示された例では18である。

【0135】

CC-ALFは、任意の適切な（CC-ALFフィルタ係数とも呼ばれる）フィルタ係数を含むことができる。図15に戻って参照すると、CC-ALF（1521）およびCC-ALF（1531）は、図16に示された菱形形状（1620）などの同じフィルタ形状、および同じ数のフィルタ係数を有することができる。一例では、CC-ALF（1521）内のフィルタ係数の値は、CC-ALF（1531）内のフィルタ係数の値とは異なる。

【0136】

一般に、CC-ALF内のフィルタ係数（例えば、非ゼロフィルタ係数）は、例えば、APS内で送信することができる。一例では、フィルタ係数は、倍数（例えば、 2^{10} ）によってスケールアップすることができ、固定小数点表現のために丸めることができる。CC-ALFの適用は、可変ブロックサイズに対して制御され、サンプルのブロックごとに受信されたコンテキストコード化フラグ（例えば、CC-ALF有効化フラグ）によってシグナリングすることができる。CC-ALF有効化フラグなどのコンテキストコード化フラグは、ブロックレベルなどの任意の適切なレベルでシグナリングすることができる。CC-ALF有効化フラグとともにブロックサイズは、彩度成分ごとにスライスレベルで受信することができる。いくつかの例では、（彩度サンプル内の）ブロックサイズ 16×16 、 32×32 、および 64×64 をサポートすることができる。

【0137】

図17は、本開示のいくつかの実施形態による、CC-ALF用の構文例を示す。図17の例では、`alf_ctb_cross_component_cb_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` は、成分間Cbフィルタが使用されるかどうかを示すインデックスであり、使用される場合成分間Cbフィルタのインデックスである。例えば、`alf_ctb_cross_component_cb_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` が0に等しいとき、成分間Cbフィルタは、輝度位置 (xCtb, yCtb) にあるCb色成分サンプルのブロックに適用されず、`alf_ctb_cross_component_cb_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` が0に等しくないとき、`alf_ctb_cross_component_cb_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` は、適用されるフィルタ用のインデックスである。例えば、`alf_ctb_cross_component_cb_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` 番目の成分間Cbフィルタは、輝度位置 (xCtb, yCtb) にあるCb色成分サンプルのブロックに適用される。

【0138】

さらに、図17の例では、`alf_ctb_cross_component_cr_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` は、成分間Crフィルタが使用されるかどうかを示すために使用され、使用される場合成分間Crフィルタのインデックスである。例えば、`alf_ctb_cross_component_cr_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` が0に等しいとき、成分間Crフィルタは、輝度位置 (xCtb, yCtb) にあるCr色成分サンプルのブロックに適用されず、`alf_ctb_cross_component_cr_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` が0に等しくないとき、`alf_ctb_cross_component_cr_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` は、成分間Crフィルタのインデックスである。例えば、`alf_ctb_cross_component_cr_idc [xCtb>>CtbLog2SizeY] [yCtb>>CtbLog2SizeY]` 番目の成分間Crフィルタは、輝度位置 (xCtb, yCtb) にあるCr色成分サンプルのブロックに適用することができる。

【0139】

いくつかの例では、彩度サブサンプリング技法が使用され、したがって、彩度ブロックの各々のサンプル数は、輝度ブロック内のサンプル数より少なくすることができる。(彩度サブサンプリングフォーマットとも呼ばれ、例えば、`chroma_format_idc`によって指定される) 彩度サブサンプリングフォーマットは、彩度ブロックの各々と対応する輝度ブロックとの間の彩度水平サブサンプリング係数 (例えば、`SubWidthC`) および彩度垂直サブサンプリング係数 (例えば、`SubHeightC`) を示すことができる。一例では、彩度サブサンプリングフォーマットは4:2:0であり、したがって、図18A~図18Bに示されるように、彩度水平サブサンプリング係数 (例えば、`SubWidthC`) および彩度垂直サブサンプリング係数 (例えば、`SubHeightC`) は2である。一例では、彩度サブサンプリングフォーマットは4:2:2であり、したがって、彩度水平サブサンプリング係数 (例えば、`SubWidthC`) は2であり、彩度垂直サブサンプリング係数 (例えば、`SubHeightC`) は1である。一例では、彩度サブサンプリングフォーマットは4:4:4であり、したがって、彩度水平サブサンプリング係数 (例えば、`SubWidthC`) および彩度垂直サブサンプリング係数 (例えば、`SubHeightC`) は1である。(彩度サンプルポジションとも呼ばれる) 彩度サンプルタイプは、輝度ブロック内の少なくとも1つの対応する輝度サンプルに対する彩度ブロック内の彩度サンプルの相対位置を示すことができる。

【0140】

図18A~図18Bは、本開示の実施形態による、輝度サンプルに対する彩度サンプルの例示的な位置を示す。図18Aを参照すると、輝度サンプル (1801) は行 (1811) ~ (1818) に位置する。図18Aに示された輝度サンプル (1801) は、ピクチャの一部を表すことができる。一例では、輝度ブロック (例えば、輝度CB) は輝度サンプル (1801) を含む。輝度ブロックは、4:2:0の彩度サブサンプリングフォーマットを有する2つの彩度ブロックに対応することができる。一例では、各彩度ブロックは彩度サンプル (1803) を含む。各彩度サンプル (例えば、彩度サンプル (1803 (1))) は、4つの輝度サンプル (例えば、輝度サンプル (1801 (1)) ~ (1801 (4))) に対応する。一例では、4つ

の輝度サンプルは、左上サンプル（1801（1））、右上サンプル（1801（2））、左下サンプル（1801（3））、および右下サンプル（1801（4））である。彩度サンプル（例えば、（1803（1）））は、左上サンプル（1801（1））と左下サンプル（1801（3））との間の左中心位置に位置し、彩度サンプル（1803）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ0と呼ぶことができる。彩度サンプルタイプ0は、左上サンプル（1801（1））と左下サンプル（1801（3））の中間の左中心位置に対応する相対位置0を示す。4つの輝度サンプル（例えば、（1801（1））～（1801（4）））は、彩度サンプル（1803）（1）の隣接輝度サンプルと呼ぶことができる。

【0141】

一例では、各彩度ブロックは彩度サンプル（1804）を含む。彩度サンプル（1803）を参照する上記の説明は、彩度サンプル（1804）に適合させることができ、したがって、簡潔にするために詳細な説明は省略することができる。彩度サンプル（1804）の各々は、4つの対応する輝度サンプルの中心位置に位置することができ、彩度サンプル（1804）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ1と呼ぶことができる。彩度サンプルタイプ1は、4つの輝度サンプル（例えば、（1801（1））～（1801（4）））の中心位置に対応する相対位置1を示す。例えば、彩度サンプル（1804）のうちの1つは、輝度サンプル（1801（1））～（1801（4））の中央部分に位置することができる。

10

【0142】

一例では、各彩度ブロックは彩度サンプル（1805）を含む。彩度サンプル（1805）の各々は、4つの対応する輝度サンプル（1801）の左上サンプルと同じ位置にある左上位置に位置することができ、彩度サンプル（1805）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ2と呼ぶことができる。したがって、彩度サンプル（1805）の各々は、それぞれの彩度サンプルに対応する4つの輝度サンプル（1801）の左上サンプルと同じ位置にある。彩度サンプルタイプ2は、4つの輝度サンプル（1801）の左上位置に対応する相対位置2を示す。例えば、彩度サンプル（1805）のうちの1つは、輝度サンプル（1801（1））～（1801（4））の左上位置に位置することができる。

20

【0143】

一例では、各彩度ブロックは彩度サンプル（1806）を含む。彩度サンプル（1806）の各々は、対応する左上サンプルと対応する右上サンプルとの間の上部中心位置に位置することができ、彩度サンプル（1806）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ3と呼ぶことができる。彩度サンプルタイプ3は、左上サンプルと右上サンプルとの間の上部中心位置に対応する相対位置3を示す。例えば、彩度サンプル（1806）のうちの1つは、輝度サンプル（1801（1））～（1801（4））の上部中心位置に位置することができる。

30

【0144】

一例では、各彩度ブロックは彩度サンプル（1807）を含む。彩度サンプル（1807）の各々は、4つの対応する輝度サンプル（1801）の左下サンプルと同じ位置にある左下位置に位置することができ、彩度サンプル（1807）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ4と呼ぶことができる。したがって、彩度サンプル（1807）の各々は、それぞれの彩度サンプルに対応する4つの輝度サンプル（1801）の左下サンプルと同じ位置にある。彩度サンプルタイプ4は、4つの輝度サンプル（1801）の左下位置に対応する相対位置4を示す。例えば、彩度サンプル（1807）のうちの1つは、輝度サンプル（1801（1））～（1801（4））の左下位置に位置することができる。

40

【0145】

一例では、各彩度ブロックは彩度サンプル（1808）を含む。彩度サンプル（1808）の各々は、左下サンプルと右下サンプルとの間の下部中心位置に位置し、彩度サンプル（1808）を有する彩度ブロックの彩度サンプルタイプは、彩度サンプルタイプ5と呼ぶことができる。彩度サンプルタイプ5は、4つの輝度サンプル（1801）の左下サンプルと右下サンプルとの間の下部中心位置に対応する相対位置5を示す。例えば、彩度サンプル（1808

50

）のうちの1つは、輝度サンプル（1801（1））～（1801（4））の左下サンプルと右下サンプルとの間に位置することができる。

【0146】

一般に、任意の適切な彩度サンプルタイプは、彩度サブサンプリングフォーマットに使用することができる。彩度サンプルタイプ0～5は、彩度サブサンプリングフォーマット4：2：0で記述される例示的な彩度サンプルタイプである。追加の彩度サンプルタイプは、彩度サブサンプリングフォーマット4：2：0に使用することができる。さらに、他の彩度サンプルタイプおよび／または彩度サンプルタイプ0～5の変形形態は、4：2：2、4：4：4などの他の彩度サブサンプリングフォーマットに使用することができる。一例では、彩度サンプル（1805）と（1807）を組み合わせる彩度サンプルタイプは、彩度サブサンプリングフォーマット4：2：2に使用される。

10

【0147】

一例では、輝度ブロックは、それぞれ、4つの輝度サンプル（例えば、（1801（1））～（1801（4）））のうちの上位2つのサンプル（例えば、（1801（1））～（1801（2）））、および4つの輝度サンプル（例えば、（1801（1））～（1801（4）））のうちの下位2つのサンプル（例えば、（1801（3））～（1801（4）））を含む行（1811）～（1812）などの交互の行を有すると見なされる。したがって、行（1811）、（1813）、（1815）、および（1817）は、（トップフィールドとも呼ばれる）現在の行と呼ぶことができ、行（1812）、（1814）、（1816）、および（1818）は、（ボトムフィールドとも呼ばれる）次の行と呼ぶことができる。4つの輝度サンプル（例えば、（1801（1））～（1801（4）））は、現在の行（例えば、（1811））および次の行（例えば、（1812））に位置する。相対位置2～3は現在の行に位置し、相対位置0～1は各現在の行とそれぞれの次の行との間に位置し、相対位置4～5は次の行に位置する。

20

【0148】

彩度サンプル（1803）、（1804）、（1805）、（1806）、（1807）、または（1808）は、各彩度ブロック内の行（1851）～（1854）に位置する。行（1851）～（1854）の特定の位置は、彩度サンプルの彩度サンプルタイプに依存することができる。例えば、それぞれの彩度サンプルタイプ0～1を有する彩度サンプル（1803）～（1804）の場合、行（1851）は行（1811）～（1812）の間に位置する。それぞれの彩度サンプルタイプ2～3を有する彩度サンプル（1805）～（1806）の場合、行（1851）は現在の行（1811）と同じ位置にある。それぞれの彩度サンプルタイプ4～5を有する彩度サンプル（1807）～（1808）の場合、行（1851）は次の行（1812）と同じ位置にある。上記の説明は、行（1852）～（1854）に適切に適合させることができ、簡潔にするために詳細な説明は省略される。

30

【0149】

図18Aに上述された輝度ブロックおよび対応する彩度ブロックを表示、格納、および／または送信するために、任意の適切な走査方法を使用することができる。一例では、順次走査が使用される。

【0150】

図18Bに示されたように、飛び越し走査を使用することができる。上述されたように、彩度サブサンプリングフォーマットは4：2：0である（例えば、`chroma_format_idc`は1に等しい）。一例では、可変彩度ロケーションタイプ（例えば、`ChromaLocType`）は、（例えば、`ChromaLocType`が`chroma_sample_loc_type_top_field`である）現在の行、または（例えば、`ChromaLocType`が`chroma_sample_loc_type_bottom_field`である）次の行を示す。現在の行（1811）、（1813）、（1815）、および（1817）、ならびに次の行（1812）、（1814）、（1816）、および（1818）は別々に走査することができ、例えば、現在の行（1811）、（1813）、（1815）、および（1817）を最初に走査することができ、続いて次の行（1812）、（1814）、（1816）、および（1818）が走査される。現在の行は輝度サンプル（1801）を含むことができ、次の行は輝度サンプル（1802）を含むことができる。

40

50

【0151】

同様に、対応する彩度ブロックを飛び越し走査することができる。塗りつぶしがない彩度サンプル(1803)、(1804)、(1805)、(1806)、(1807)、または(1808)を含む行(1851)および(1853)は、現在の行(または現在の彩度行)と呼ぶことができ、グレー塗りつぶしがある彩度サンプル(1803)、(1804)、(1805)、(1806)、(1807)、または(1808)を含む行(1852)および(1854)は、次の行(または次の彩度行)と呼ぶことができる。一例では、飛び越し走査中に、行(1851)および(1853)が最初に走査され、続いて行(1852)および(1854)が走査される。

【0152】

いくつかの例では、制約付き指向性拡張フィルタリング技法を使用することができる。インループ制約付き指向性拡張フィルタ(CDEF)の使用は、画像の細部を保持しながらコーディングアーチファクトを除去することができる。一例(例えば、HEVC)では、サンプル適応オフセット(SAO)アルゴリズムは、異なるクラスのピクセル用の信号オフセットを定義することによって同様の目的を達成することができる。SAOとは異なり、CDEFは非線形空間フィルタである。いくつかの例では、CDEFは、容易にベクトル化可能であるように制約することができる(すなわち、単一命令複数データ(SIMD)演算で実装可能である)。メディアンフィルタ、バイラテラルフィルタなどの他の非線形フィルタは、同様に扱うことができないことに留意されたい。

【0153】

場合によっては、コード化画像内のリングングアーチファクトの量は、量子化ステップサイズにほぼ比例する傾向がある。細部の量は入力画像の特性であるが、量子化画像内に保持される最小の細部も量子化ステップサイズに比例する傾向がある。所与の量子化ステップサイズでは、リングングの振幅は、一般に、細部の振幅よりも小さい。

【0154】

CDEFは、各ブロックの方向を識別し、次いで識別された方向に沿って適応的にフィルタリングし、識別された方向から45度回転した方向に沿ってより少ない程度にフィルタリングするために使用することができる。いくつかの例では、エンコーダはフィルタ強度を検索することができ、フィルタ強度は明示的にシグナリングすることができ、それによりほかしの高度な制御が可能になる。

【0155】

具体的に、いくつかの例では、方向検索は、復元ピクセルに対してデブロッキングフィルタの直後に実行される。それらのピクセルはデコーダに利用可能なので、デコーダによって方向を検索することができ、したがって、一例では方向はシグナリングを必要としない。いくつかの例では、方向検索は8×8ブロックなどの特定のブロックサイズで動作することができ、これは、非直線エッジを適切に処理するのに十分小さいが、量子化画像に適用されたときに確実に方向を推定するのに十分大きい。また、8×8領域上に一定の方向性を有することは、フィルタのベクトル化を容易にする。いくつかの例では、各ブロック(例えば、8×8)は、差を決定するために完全指向性ブロックと比較することができる。完全指向性ブロックは、ある方向の線に沿ったすべてのピクセルが同じ値を有するブロックである。一例では、二乗差の和(SSD)、二乗平均平方根(RMS)誤差などの、ブロックおよび完全指向性ブロックの各々の差分尺度を計算することができる。次いで、最小差(例えば、最小SSD、最小RMSなど)を有する完全指向性ブロックを決定することができ、決定された完全指向性ブロックの方向は、ブロック内のパターンに最もよく一致する方向であり得る。

【0156】

図19は、本開示の一実施形態による方向検索の一例を示す。一例では、ブロック(1910)は、復元され、デブロッキングフィルタから出力された8×8のブロックである。図19の例では、方向探索は、ブロック(1910)に対して(1920)によって示された8方向から方向を決定することができる。8つの完全指向性ブロック(1930)は、8つの方向(1920)にそれぞれ対応して形成される。方向に対応する完全指向性ブロックは、その方向の

線に沿ったピクセルが同じ値を有するブロックである。さらに、ブロック（1910）および完全指向性ブロック（1930）の各々のSSD、RMS誤差などの差分尺度を計算することができる。図19の例では、RMS誤差は（1940）によって示されている。（1943）によって示されたように、ブロック（1910）および完全指向性ブロック（1933）のRMS誤差は最も小さく、したがって方向（1923）はブロック（1910）内のパターンに最もよく一致する方向である。

【0157】

ブロックの方向が識別された後に、非線形ローパス指向性フィルタを決定することができる。例えば、非線形ローパス指向性フィルタのフィルタタップは、指向性エッジまたは指向性パターンを維持しながらリングングを低減するために、識別された方向に沿って整列させることができる。しかしながら、いくつかの例では、指向性フィルタリングだけでは、リングングを十分に低減することができないことがある。一例では、識別された方向に沿っていないピクセルにも追加のフィルタタップが使用される。ぼかしのリスクを低減するために、余分なフィルタタップはより控えめに扱われる。このため、CDEFは1次フィルタタップおよび2次フィルタタップを含む。一例では、完全な2-D CDEFフィルタは、式（14）：

【数20】

$$y(i, j) = x(i, j) + \text{round} \left(\sum_{m, n} w_{d, m, n}^{(p)} f \left(x(m, n) - x(i, j), S^{(p)}, D \right) + \sum_{m, n} w_{d, m, n}^{(s)} f \left(x(m, n) - x(i, j), S^{(s)}, D \right) \right), \quad \text{式 (14)}$$

のように表すことができ、ここで、Dは減衰パラメータを表記し、 $S^{(p)}$ は一次フィルタタップの強度を表記し、 $S^{(s)}$ は二次フィルタタップの強度を表記し、 $\text{round}(\cdot)$ は0から離れるようにタイを丸める演算を表記し、wはフィルタ重みを表記し、 $f(d, S, D)$ はフィルタリングされたピクセルと隣接するピクセルの各々との間の差に対して演算する制約関数である。一例では、小さい差の場合、関数 $f(d, S, D)$ はDに等しく、フィルタを線形フィルタのように動作させることができ、差が大きいつき、関数 $f(d, S, D)$ は0に等しく、フィルタタップを効果的に無視することができる。

【0158】

いくつかの例では、デブロッキング動作以外に、一般にノイズを除去しエッジの品質を表記し向上させるために、デブロッキング後のビデオコーディングにおいてインループ復元方式が使用される。一例では、インループ復元方式は、適切なサイズのタイルごとにフレーム内で切り替え可能である。インループ復元方式は、分離可能な対称ウィナーフィルタ、部分空間投影による二重自己誘導式フィルタ、および領域変換再帰フィルタに基づいている。コンテンツ統計はフレーム内で大幅に変化する可能性があるので、インループ復元方式は、フレームの異なる領域内で異なる方式をトリガすることができる切り替え可能なフレームワーク内に統合されている。

【0159】

分離可能な対称ウィナーフィルタは、インループ復元方式の1つであり得る。いくつかの例では、劣化したフレーム内のすべてのピクセルは、その周りの $w \times w$ ウィンドウ内のピクセルの非因果フィルタリングバージョンとして復元することができ、ここで、 $w = 2r + 1$ は整数rに対して奇数である。2Dフィルタタップが列ベクトル化形式の $w^2 \times 1$ 要素ベクトルFによって表記される場合、直接的なLMMSE最適化は、 $F = H^{-1}M$ によって与えられ

るフィルタパラメータにつながり、ここで、 $H = E[XX^T]$ は、 x の自己共分散であり、ピクセルの周りの $w \times w$ ウィンドウ内の w^2 個のサンプルの列ベクトル化バージョンであり、 $M = E[YX^T]$ は、推定されるべき、 x とスカラーソースサンプル y の相互相関である。一例では、エンコーダは、デブロッキングされたフレームおよびソース内の実現値から H および M を推定し、得られたフィルタ F をデコーダに送信することができる。しかしながら、そうすると、 w^2 個のタップを送信する際にかかなりのビットレートコストを招くだけでなく、分離不可能なフィルタリングによって復号が法外に複雑になる。いくつかの実施形態では、 F の性質にいくつかの追加の制約が課される。一番目の制約では、 F は分離可能であるように制約され、その結果、フィルタリングは分離可能な水平方向および垂直方向の w タップ畳み込みとして実施することができる。二番目の制約では、水平フィルタおよび垂直フィルタの各々は対称になるように制約される。三番目の制約では、水平フィルタ係数と垂直フィルタ係数の両方の和が合計1になると仮定する。

【0160】

部分空間投影を用いた二重自己誘導式フィルタリングは、インループ復元方式の1つであり得る。誘導式フィルタリングは、式(15)によって示された局所線形モデルがフィルタリングされていないサンプル x からフィルタリングされた出力 y を計算するために使用される、画像フィルタリング技法であり、

$$y = Fx + G \quad \text{式(15)}$$

ここで、 F および G は、劣化した画像およびフィルタリングされたピクセルの近傍のガイダンス画像の統計に基づいて決定される。ガイド画像が劣化した画像と同じである場合、得られたいわゆる自己誘導式フィルタリングは、エッジ維持平滑化の効果を有する。一例では、特定の形式の自己誘導式フィルタリングを使用することができる。自己誘導式フィルタリングの具体的な形式は、2つのパラメータ：半径 r およびノイズパラメータ e に依存し、以下のステップのように列挙される。

1. すべてのピクセルの周りの $(2r+1) \times (2r+1)$ ウィンドウ内のピクセルの平均 μ および分散 σ^2 を取得する。このステップは、インテグラルイメージングに基づくボックスフィルタリングで効率的に実施することができる。

2. すべてのピクセルについて $f = \sigma^2 / (\sigma^2 + e)$ 、 $g = (1-f) \mu$ を計算する。

3. すべてのピクセルの F および G を、使用するピクセルの周りの 3×3 ウィンドウ内の f および g の値の平均として計算する。

【0161】

自己誘導式フィルタリングの具体的な形式は r および e によって制御され、 r が大きいほど空間分散が大きくなり、 e が大きいほど範囲分散が大きくなる。

【0162】

図20は、いくつかの例における部分空間投影を示す一例を示す。図20に示されたように、復元 X_1 、 X_2 のいずれもソース Y に近接していない場合でも、適切な乗算器 $\{\alpha, \beta\}$ は、それらがいくらか正しい方向に移動している限り、それらをソースにかなり近接させることができる。

【0163】

いくつかの例（例えば、HEVC）では、サンプル適応オフセット（SAO）と呼ばれるフィルタリング技法を使用することができる。いくつかの例では、SAOは、デブロッキングフィルタの後の復元信号に適用される。SAOは、スライスヘッダ内で与えられたオフセット値を使用することができる。いくつかの例では、輝度サンプルの場合、エンコーダは、スライスにSAOを適用する（有効化する）かどうかを判断することができる。SAOが有効化されると、現在ピクチャは、コーディングユニットの4つのサブ領域への再帰的分割を可能にし、各サブ領域は、サブ領域内の特徴に基づいて複数のSAOタイプからSAOタイプを選択することができる。

【0164】

図21は、本開示の一実施形態による、複数のSAOタイプの表(2100)を示す。表(2100)には、SAOタイプ0~6が示されている。SAOタイプ0は、SAOが適用されないことを

示すために使用されることに留意されたい。さらに、SAOタイプ1～SAOタイプ6の各SAOタイプは、複数のカテゴリを含む。SAOは、サブ領域の復元されたピクセルをカテゴリに分類し、サブ領域内の各カテゴリのピクセルにオフセットを加えることによって歪みを低減することができる。いくつかの例では、エッジ特性はSAOタイプ1～4におけるピクセル分類に使用することができ、ピクセル強度はSAOタイプ5～6におけるピクセル分類に使用することができる。

【0165】

具体的には、SAOタイプ5～6などの一実施形態では、サブ領域のすべてのピクセルを複数の帯域に分類するために、バンドオフセット（BO）を使用することができる。複数の帯域の各帯域は、同じ強度間隔内のピクセルを含む。いくつかの例では、強度範囲は、0から最大強度値（例えば、8ビットピクセルの場合255）までの32個の間隔などの複数の間隔に等しく分割され、各間隔はオフセットに関連付けられる。さらに、一例では、32個の帯域は、第1のグループおよび第2のグループなどの2つのグループに分割される。第1のグループは、中央の16個の帯域（例えば、強度範囲の中央にある16個の間隔）を含み、第2のグループは、残りの16個の帯域（例えば、強度範囲の低い側にある8つの間隔、および強度範囲の高い側にある8つの間隔）を含む。一例では、2つのグループのうちの1つのオフセットのみが送信される。いくつかの実施形態では、BOにおけるピクセル分類演算が使用されるとき、各ピクセルの5つの最上位ビットを帯域インデックスとして直接使用することができる。

【0166】

さらに、SAOタイプ1～4などの一実施形態では、ピクセル分類およびオフセットの決定にエッジオフセット（EO）を使用することができる。例えば、ピクセル分類は、エッジ指向性情報を考慮して1次元3ピクセルパターンに基づいて決定することができる。

【0167】

図22は、いくつかの例におけるエッジオフセットにおけるピクセル分類のための3ピクセルパターンの例を示す。図22の例では、（3つのグレーピクセルによって示された）第1のパターン（2210）は、0度パターン（0度パターンには水平方向が関連付けられている）と呼ばれ、（3つのグレーピクセルによって示された）第2のパターン（2220）は、90度パターン（90度パターンには垂直方向が関連付けられている）と呼ばれ、（3つのグレーピクセルによって示された）第3のパターン（2230）は、135度パターン（135度パターンには135度対角方向が関連付けられている）と呼ばれ、（3つのグレーピクセルによって示された）第4のパターン（2240）は、45度パターン（45度パターンには45度対角方向が関連付けられている）と呼ばれる。一例では、サブ領域のエッジ指向性情報を考慮して、図22に示された4つの指向性パターンのうちの1つを選択することができる。選択は、一例ではサイド情報としてコード化ビデオビットストリーム内で送信することができる。次いで、サブ領域内のピクセルは、指向性パターンに関連付けられた方向にその2つの隣接ピクセルと各ピクセルを比較することによって、複数のカテゴリに分類することができる。

【0168】

図23は、いくつかの例におけるエッジオフセット向けのピクセル分類規則のための表（2300）である。具体的には、（図22の各パターンにも示されている）ピクセルcは、（図22の各パターンにも灰色で示されている）2つの隣接ピクセルと比較され、ピクセルcは、図23に示されたピクセル分類規則による比較に基づいてカテゴリ0～4のうちの1つに分類することができる。

【0169】

いくつかの実施形態では、デコーダ側のSAOは、ラインバッファを節約することができるように、最大コーディングユニット（LCU）（例えば、CTU）とは無関係に動作することができる。いくつかの例では、各LCU内の最上位および最下位の行のピクセルは、90度、135度、および45度の分類パターンが選択されたときにSAO処理されず、各LCU内の左端および右端の列のピクセルは、0度、135度、および45度のパターンが選択されたとき

にSAO処理されない。

【0170】

図24は、パラメータが隣接するCTUからマージされない場合にCTUについてシグナリングされる必要があり得る構文の例(2400)を示す。例えば、構文要素sao_type_idx [cldx] [rx] [ry] は、サブ領域のSAOタイプを示すためにシグナリングすることができる。SAOタイプは、BO(バンドオフセット)またはEO(エッジオフセット)であってもよい。sao_type_idx [cldx] [rx] [ry] が0の値を有するとき、それはSAOがOFFであることを示し、1から4の値は、0°、90°、135°、および45°に対応する4つのEOカテゴリのうちの1つが使用されることを示し、5の値は、BOが使用されることを示す。図24の例では、BOタイプおよびEOタイプの各々は、シグナリングされる4つのSAOオフセット値(sao_offset [cldx] [rx] [ry] [0] ~sao_offset [cldx] [rx] [ry] [3])を有する。

10

【0171】

一般に、フィルタリングプロセスは、出力を生成するために入力(例えば、YもしくはCbもしくはCr、またはRもしくはGもしくはB)として第1の色成分の復元サンプルを使用することができ、フィルタリングプロセスの出力は、第1の色成分と同じ色成分であり得るか、または第1の色成分とは異なる別の色成分であり得る第2の色成分に適用される。

【0172】

成分間フィルタリング(CCF)の関連する例では、いくつかの数学方程式に基づいてフィルタ係数が導出される。導出されたフィルタ係数は、エンコーダ側からデコーダ側にシグナリングされ、導出されたフィルタ係数は、線形結合を使用してオフセットを生成するために使用される。次いで、生成されたオフセットは、フィルタリングプロセスとして復元サンプルに加えられる。例えば、オフセットは、フィルタリング係数の輝度サンプルとの線形結合に基づいて生成され、生成されたオフセットは復元された彩度サンプルに加えられる。CCFの関連する例は、復元された輝度サンプル値と、元の彩度サンプルと復元された彩度サンプルとの間のデルタ値との間の線形マッピング関係の仮定に基づいている。しかしながら、復元された輝度サンプル値と、元の彩度サンプルと復元された彩度サンプルとの間のデルタ値との間のマッピングは、必ずしも線形マッピングプロセスに従わず、したがって、CCFのコーディング性能は、線形マッピング関係の仮定の下で制限される場合がある。

20

30

【0173】

いくつかの例では、非線形マッピング技法は、著しいシグナリングオーバーヘッドなしに、成分間フィルタリングおよび／または同色成分フィルタリングにおいて使用することができる。一例では、非線形マッピング技法は、成分間サンプルオフセットを生成するために成分間フィルタリングにおいて使用することができる。別の例では、非線形マッピング技法は、局所サンプルオフセットを生成するために同色成分フィルタリングにおいて使用することができる。

【0174】

便宜上、非線形マッピング技法を使用するフィルタリングプロセスは、非線形マッピングによるサンプルオフセット(SO-NLM)と呼ぶことができる。成分間フィルタリングプロセスにおけるSO-NLMは、成分間サンプルオフセット(CCSO)と呼ぶことができる。同色成分フィルタリングにおけるSO-NLMは、局所サンプルオフセット(LSO)と呼ぶことができる。非線形マッピング技法を使用するフィルタは、非線形マッピングベースフィルタと呼ぶことができる。非線形マッピングベースフィルタは、CCSOフィルタ、LSOフィルタなどを含むことができる。

40

【0175】

一例では、CCSOおよびLSOは、復元サンプルの歪みを低減するためにループフィルタリングとして使用することができる。CCSOおよびLSOは、関連する例示的なCCFにおいて使用される線形マッピングの仮定に依存しない。例えば、CCSOは、輝度復元サンプル値と、元の彩度サンプルと彩度復元サンプルとの間のデルタ値との間の線形マッピング関係

50

の仮定に依存しない。同様に、LSOは、色成分の復元サンプル値と、色成分の元のサンプルと色成分の復元サンプルとの間のデルタ値との間の線形マッピング関係の仮定に依存しない。

【0176】

以下の説明では、出力を生成するために入力（例えば、YもしくはCbもしくはCr、またはRもしくはGもしくはB）として第1の色成分の復元サンプルを使用するSO-NLMフィルタリングプロセスが記載され、フィルタリングプロセスの出力は第2の色成分に適用される。第2の色成分が第1の色成分と同じ色成分であるとき、説明はLSOに適用可能であり、第2の色成分が第1の色成分と異なるとき、説明はCCSOに適用可能である。

【0177】

SO-NLMでは、エンコーダ側で非線形マッピングが導出される。非線形マッピングは、フィルタ支持領域内の第1の色成分の復元サンプルと、フィルタ支持領域内の第2の色成分に加えられたオフセットとの間にある。第2の色成分が第1の色成分と同じであるとき、非線形マッピングはLSOにおいて使用され、第2の色成分が第1の色成分と異なるとき、非線形マッピングはCCSOにおいて使用される。非線形マッピングの領域は、（可能な復元サンプル値の組合せとも呼ばれる）処理された入力復元サンプルの異なる組合せによって決定される。

【0178】

SO-NLMの技法は具体例を使用して例示することができる。具体例では、（「フィルタ支持領域」とも呼ばれる）フィルタ支持エリア内に位置する第1の色成分からの復元サンプルが決定される。フィルタ支持エリアはフィルタを適用することができるエリアであり、フィルタ支持エリアは任意の適切な形状を有することができる。

【0179】

図25は、本開示のいくつかの実施形態による、フィルタ支持エリア（2500）の一例を示す。フィルタ支持エリア（2500）は、第1の色成分の4つの復元サンプル：P0、P1、P2、およびP3を含む。図25の例では、4つの復元サンプルは垂直方向および水平方向に十字形を形成することができ、十字形の中心位置はフィルタリングされるサンプル用の位置である。中心位置にあり、P0～P3と同じ色成分のサンプルはCによって表記される。中心位置にあり、第2の色成分のサンプルはFによって表記される。第2の色成分は、P0～P3の第1の色成分と同じであってもよく、P0～P3の第1の色成分と異なってもよい。

【0180】

図26は、本開示のいくつかの実施形態による、別のフィルタ支持エリア（2600）の一例を示す。フィルタ支持エリア（2600）は、正方形を形成する第1の色成分の4つの復元サンプルP0、P1、P2、およびP3を含む。図26の例では、正方形の中心位置はフィルタリングされるサンプルの位置である。中心位置にあり、P0～P3と同じ色成分のサンプルはCによって表記される。中心位置にあり、第2の色成分のサンプルはFによって表記される。第2の色成分は、P0～P3の第1の色成分と同じであってもよく、P0～P3の第1の色成分と異なってもよい。

【0181】

復元サンプルはSO-NLMフィルタに入力され、フィルタタップを形成するために適切に処理される。一例では、SO-NLMフィルタへの入力である復元サンプルの位置は、フィルタタップ位置と呼ばれる。具体例では、復元サンプルは以下の2つのステップで処理される。

【0182】

第1のステップでは、P0～P3とCとの間のデルタ値がそれぞれ計算される。例えば、m0はP0からCの間のデルタ値を表記し、m1はP1からCの間のデルタ値を表記し、m2はP2からCの間のデルタ値を表記し、m3はP3からCの間のデルタ値を表記する。

【0183】

第2のステップでは、デルタ値m0～m3がさらに量子化され、量子化された値はd0、d1、d2、d3として表記される。一例では、量子化値は、量子化プロセスに基づいて-1、0

10

20

30

40

50

、1のうちの1つであり得る。例えば、 m が $-N$ より小さいとき（ N は正の値であり、量子化ステップサイズと呼ばれる）、値 m は -1 に量子化することができ、 m が $[-N, N]$ の範囲内にあるとき、値 m は 0 に量子化することができ、 m が N より大きいとき、値 m は 1 に量子化することができる。いくつかの例では、量子化ステップサイズ N は、4、8、12、16などのうちの1つであり得る。

【0184】

いくつかの実施形態では、量子化値 $d0 \sim d3$ はフィルタタップであり、フィルタ領域内の1つの組合せを識別するために使用することができる。例えば、フィルタタップ $d0 \sim d3$ は、フィルタ領域内の組合せを形成することができる。各フィルタタップは3つの量子化値を有することができ、したがって、4つのフィルタタップが使用されるとき、フィルタ領域は $81 (3 \times 3 \times 3 \times 3)$ 個の組合せを含む。

【0185】

図27A～図27Cは、本開示の一実施形態による、81個の組合せを有する表（2700）である。表（2700）は81個の組合せに対応する81行を含む。組合せに対応する各行において、第1列は組合せのインデックスを含み、第2列は組合せのためのフィルタタップ $d0$ の値を含み、第3列は組合せのためのフィルタタップ $d1$ の値を含み、第4列は組合せのためのフィルタタップ $d2$ の値を含み、第5列は組合せのためのフィルタタップ $d3$ の値を含み、第6列は非線形マッピングのための組合せに関連付けられたオフセット値を含む。一例では、フィルタタップ $d0 \sim d3$ が決定されると、 $d0 \sim d3$ の組合せに関連付けられた（ s によって表記された）オフセット値を表（2700）に従って決定することができる。一例では、オフセット値 $s0 \sim s80$ は、0、1、 -1 、3、 -3 、5、 -5 、 -7 などの整数である。

【0186】

いくつかの実施形態では、SO-NLMの最終フィルタリングプロセスは、式（16）に示されたように適用することができ、

$$f' = \text{clip}(f + s) \quad \text{式 (16)}$$

ここで、 f は、フィルタリングされるべき第2の色成分の復元サンプルであり、 s は、表（2700）などを使用して、第1の色成分の復元サンプルの処理結果であるフィルタタップに従って決定されたオフセット値である。復元サンプル F とオフセット値 s の和は、第2の色成分の最終的なフィルタリングされたサンプル f' を決定するためにビット深度に関連付けられた範囲にさらにクリップされる。

【0187】

LSOの場合、上記の説明における第2の色成分は第1の色成分と同じであり、CCSOの場合、上記の説明における第2の色成分は第1の色成分とは異なり得ることに留意されたい。

【0188】

上記の説明は、本開示の他の実施形態向けに調整することができることに留意されたい。

【0189】

いくつかの例では、エンコーダ側で、符号化デバイスは、フィルタ支持領域内の第1の色成分の復元サンプルと、第2の色成分の復元サンプルに加えられたオフセットとの間のマッピングを導出することができる。マッピングは、任意の適切な線形マッピングまたは非線形マッピングであり得る。次いで、フィルタリングプロセスは、マッピングに基づいてエンコーダ側および／またはデコーダ側で適用することができる。例えば、マッピングは、デコーダに適切に通知され（例えば、マッピングは、エンコーダ側からデコーダ側に送信されるコード化ビデオビットストリームに含まれる）、次いで、デコーダは、マッピングに基づいてフィルタリングプロセスを実行することができる。

【0190】

CCSOフィルタ、LSOフィルタなどの非線形マッピングベースフィルタの性能は、フィルタ形状構成に依存する。フィルタの（フィルタ形状とも呼ばれる）フィルタ形状構成は、フィルタタップ位置によって形成されるパターンの特性を指すことができる。パターンは、フィルタタップの数、フィルタタップ位置の幾何学的形状、パターンの中心までのフィルタタップ位置の距離などの様々なパラメータによって定義することができる。固定フ

フィルタ形状構成を使用すると、非線形マッピングベースフィルタの性能が制限される場合がある。

【0191】

図24および図25ならびに図27A～図27Cによって示されたように、いくつかの例は、非線形マッピングベースフィルタのフィルタ形状構成に5タップフィルタ設計を使用する。5タップフィルタ設計は、P0、P1、P2、P3、およびCのタップ位置を使用することができる。フィルタ形状構成向けの5タップフィルタ設計は、図27A～図27Cに示されたように、81個のエントリを有する参照テーブル（LUT）をもたらすことができる。サンプルオフセットのLUTは、エンコーダ側からデコーダ側にシグナリングされる必要があり、LUTのシグナリングは、大部分のシグナリングオーバーヘッドの一因となり、非線形マッピングベースフィルタを使用するコーディング効率に影響を及ぼす可能性がある。本開示のいくつかの態様によれば、フィルタタップの数は5とは異なってもよい。いくつかの例では、フィルタタップの数を減らすことができ、フィルタ支持エリア内の情報をさらに取り込むことができ、コーディング効率を改善することができる。

【0192】

いくつかの例では、非線形マッピングベースフィルタ用のグループ内のフィルタ形状構成は、それぞれ、3つのフィルタタップを有する。

【0193】

図28は、一例における3つのフィルタタップの7つのフィルタ形状構成を示す。具体的には、第1のフィルタ形状構成は、「1」および「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「1」の中心位置であり、第2のフィルタ形状構成は、「2」および位置「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「2」の中心位置であり、第3のフィルタ形状構成は、「3」および位置「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「3」の中心位置であり、第4のフィルタ形状構成は、「4」および位置「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「4」の中心位置であり、第5のフィルタ形状構成は、「5」および「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「5」の中心位置であり、第6のフィルタ形状構成は、「6」および位置「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「6」の中心位置であり、第7のフィルタ形状構成は、「7」および位置「c」とラベル付けされた位置に3つのフィルタタップを含み、位置「c」は位置「7」の中心位置である。

【0194】

本開示のいくつかの態様によれば、非線形マッピングベースフィルタは、ループフィルタチェーン内の他のインループフィルタとともに使用することができる。非線形マッピングベースフィルタの位置は、非線形マッピングベースフィルタのコーディング効率に影響を及ぼす場合がある。

【0195】

図29は、いくつかの例におけるループフィルタチェーン（2900）のブロック図を示す。ループフィルタチェーン（2900）は、フィルタチェーン内の直列に接続された複数のフィルタを含む。ループフィルタチェーン（2900）は、一例ではループフィルタユニット（556）として使用することができる。ループフィルタチェーン（2900）は、参照ピクチャメモリ（557）などの復号ピクチャバッファに復元ピクチャを格納する前に、符号化ループまたは復号ループにおいて使用することができる。ループフィルタチェーン（2900）は、前の処理モジュールから入力復元サンプルを受信し、復元サンプルにフィルタを適用して出力復元サンプルを生成する。

【0196】

ループフィルタチェーン（2900）は、任意の適切なフィルタを含むことができる。図29の例では、ループフィルタチェーン（2900）は、チェーン内に接続された（デブロッキングとラベル付けされた）デブロッキングフィルタ、（CDEFとラベル付けされた）制約付き指向性拡張フィルタ、および（LRとラベル付けされた）インループ復元フィルタを含

10

20

30

40

50

む。ループフィルタチェーン（2900）は、入力ノード（2901）、出力ノード（2909）、および複数の中間ノード（2902）～（2903）を有する。ループフィルタチェーン（2900）の入力ノード（2901）は、前の処理モジュールから入力復元サンプルを受信し、入力復元サンプルはデブロッキングフィルタに提供される。中間ノード（2902）は、デブロッキングフィルタから（デブロッキングフィルタによって処理された後に）復元サンプルを受信し、さらなるフィルタ処理のために復元サンプルをCDEFに提供する。中間ノード（2903）は、CDEFから（CDEFによって処理された後に）復元サンプルを受信し、さらなるフィルタ処理のために復元サンプルをLRフィルタに提供する。出力ノード（2909）は、（LRフィルタによって処理された後に）LRフィルタから出力復元サンプルを受信する。出力復元サンプルは、さらなる処理のための後処理モジュールなどの他の処理モジュールに提供することができる。

10

【0197】

以下の説明は、ループフィルタチェーン（2900）に基づいて非線形マッピングベースフィルタを使用する技法を示すことに留意されたい。非線形マッピングベースフィルタを使用する技法は、他の適切なループフィルタチェーンにおいて使用することができる。

【0198】

本開示の一態様によれば、非線形マッピングベースフィルタは、ループフィルタチェーン内で直列に他のフィルタと結合することができ、非線形マッピングベースフィルタの入力および出力は、ループフィルタチェーンの同じノードに配置することができ、他のフィルタは、非線形マッピングベースフィルタの入力と出力との間に存在しない。例えば、非線形マッピングベースフィルタは、ループフィルタチェーンのノードにおいて復元サンプルを受信し、ループフィルタチェーンのノードにある復元サンプルに基づいてサンプルオフセットを決定し、次いで、ループフィルタチェーンのノードにある復元サンプルにおいてサンプルオフセットを適用する。

20

【0199】

図30A～図30Dは、ループフィルタチェーン内で直列に他のフィルタと結合された非線形マッピングベースフィルタを含むループフィルタチェーンの例を示す。

【0200】

図30Aは、一例におけるループフィルタチェーン（3000A）の一例を示す。ループフィルタチェーン（3000A）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3000A）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが入力ノードにおいて適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3011A）における（ループフィルタチェーン（3000A）への）入力復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3012A）において中間復元サンプルと組み合わせられて、第3のノード（3013A）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにデブロッキングフィルタに提供される。図30Aの例では、第1のノード（3011A）および第2のノード（3012A）は同じノードであり、中間復元サンプルは第1の復元サンプルである。

30

40

【0201】

図30Bは、一例におけるループフィルタチェーン（3000B）の一例を示す。ループフィルタチェーン（3000B）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3000B）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが、ループフィルタチェーン（3000B）の中間ノードにおいて適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3011B）における（デブロッキングフィルタによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。

50

サンプルオフセットは、第2のノード（3012B）において中間復元サンプルと組み合わせられて、第3のノード（3013B）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにCDEFに提供される。図30Bの例では、第1のノード（3011B）および第2のノード（3012B）は同じノードであり、中間復元サンプルは第1の復元サンプルである。

【0202】

図30Cは、一例におけるループフィルタチェーン（3000C）の一例を示す。ループフィルタチェーン（3000C）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3000C）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが、ループフィルタチェーン（3000C）の中間ノードにおいて適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3011C）における（CDEFによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3012C）において中間復元サンプルと組み合わせられて、第3のノード（3013C）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにLRフィルタに提供される。図30Cの例では、第1のノード（3011C）および第2のノード（3012C）は同じノードであり、中間復元サンプルは第1の復元サンプルである。

【0203】

図30Dは、一例におけるループフィルタチェーン（3000D）の一例を示す。ループフィルタチェーン（3000D）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3000D）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが出力ノードにおいて適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3011D）における（LRフィルタによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3012D）において中間復元サンプルと組み合わせられて、第3のノード（3013D）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はループフィルタチェーン（3000D）の出力である。図30Dの例では、第1のノード（3011D）および第2のノード（3012D）は同じノードであり、中間復元サンプルは第1の復元サンプルである。

【0204】

本開示の一態様によれば、非線形マッピングベースフィルタは、ループフィルタチェーン内で並列に1つまたは複数のフィルタと結合することができ、非線形マッピングベースフィルタの入力と出力との間に少なくとも1つのフィルタが存在する。例えば、非線形マッピングベースフィルタは、ループフィルタチェーンの第1のノードにおいて復元サンプルを受信し、ループフィルタチェーンの第1のノードにある復元サンプルに基づいてサンプルオフセットを決定し、次いで、ループフィルタチェーンの第2のノードにある復元サンプルにおいてサンプルオフセットを適用する。ループフィルタチェーンの第2のノードにある復元サンプルは、ループフィルタチェーンの第1のノードにある復元サンプルに1つまたは複数のフィルタを適用することによって取得することができる。

【0205】

いくつかの例では、非線形マッピングベースフィルタの入力は、デブロッキングフィルタの後でCDEFの前に位置する復元サンプルであり、非線形マッピングベースフィルタの出力は、CDEFの後でLRフィルタの前、またはLRフィルタの後の復元サンプルに適用される。

【0206】

図31Aは、CDEFと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3100A）の一例を示す。ループフィルタチェーン（3100A）は、符号

化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3100A）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが2つの中間ノードの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力は、第1のノード（3111A）における（デブロッキングフィルタによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3112A）において（CDEFによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3113A）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにLRフィルタに提供される。図31Aの例では、CDEFは、第1のノード（3111A）と第2のノード（3112A）との間にある。中間復元サンプルはCDEFの出力である。

10

【0207】

図31Bは、CDEFおよびLRフィルタと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3100B）の一例を示す。ループフィルタチェーン（3100B）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3100B）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタが中間ノードと出力ノードとの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3111B）における（デブロッキングフィルタによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3112B）において（LRフィルタによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3113B）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はループフィルタチェーン（3100B）の出力である。図31Bの例では、CDEFおよびLRフィルタは、第1のノード（3111B）と第2のノード（3112B）との間にある。中間復元サンプルはLRフィルタの出力である。

20

【0208】

いくつかの例では、非線形マッピングベースフィルタの入力は、デブロッキングフィルタの前に位置する復元サンプルであり、非線形マッピングベースフィルタの出力は、デブロッキングフィルタの後でCDEFの前、CDEFの後でLRフィルタの前、またはLRフィルタの後の復元サンプルに適用される。

30

【0209】

図32Aは、デブロッキングフィルタと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3200A）の一例を示す。ループフィルタチェーン（3200A）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3200A）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタがループフィルタチェーンの入力ノードと中間ノードとの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3211A）における入力復元サンプル（ループフィルタチェーン（3200A）の入力）である。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3212A）において（デブロッキングフィルタによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3213A）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにCDEFに提供される。図32Aの例では、デブロッキングフィルタは、第1のノード（3211A）と第2のノード（3212A）との間にある。中間復元サンプルはデブロッキングフィルタの出力である。

40

【0210】

図32Bは、デブロッキングフィルタおよびCDEFと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3200B）の一例を示す。ループフィルタ

50

チェーン（3200B）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3200B）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタがループフィルタチェーンの入力ノードと中間ノードとの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3211B）における入力復元サンプル（ループフィルタチェーン（3200B）の入力）である。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3212B）において（CDEFによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3213B）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はさらなるフィルタリングのためにLRに提供される。図32Bの例では、デブロッキングフィルタおよびCDEFは、第1のノード（3211B）と第2のノード（3212B）との間にある。中間復元サンプルはCDEFの出力である。

【0211】

図32Cは、デブロッキングフィルタ、CDEF、およびLRフィルタと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3200C）の一例を示す。ループフィルタチェーン（3200C）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3200C）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタがループフィルタチェーンの入力ノードと出力ノードとの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3211C）における入力復元サンプル（ループフィルタチェーン（3200C）の入力）である。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3212C）において（LRフィルタによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3213C）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はループフィルタチェーン（3200C）の出力である。図32Cの例では、デブロッキングフィルタ、CDEF、およびLRフィルタは、第1のノード（3211C）と第2のノード（3212C）との間にある。中間復元サンプルはLRフィルタの出力である。

【0212】

いくつかの例では、非線形マッピングベースフィルタの入力は、CDEFの後でLRフィルタの前に位置する復元サンプルであり、非線形マッピングベースフィルタの出力は、LRフィルタの後の復元サンプルに適用される。

【0213】

図33は、LRフィルタと並列に結合された非線形マッピングベースフィルタを含むループフィルタチェーン（3300）の一例を示す。ループフィルタチェーン（3300）は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン（2900）の代わりに使用することができる。ループフィルタチェーン（3300）では、（SO-NLMとラベル付けされた）非線形マッピングベースフィルタがループフィルタチェーンの中間ノードと出力ノードとの間で適用される。具体的には、非線形マッピングベースフィルタの（第1の復元サンプルとも呼ばれる）入力、第1のノード（3311）における（CDEFによって生成された）復元サンプルである。入力に基づいて、非線形マッピングベースフィルタはサンプルオフセット（SO）を生成する。サンプルオフセットは、第2のノード（3312）において（LRフィルタによって生成された）中間復元サンプルと組み合わせられて、第3のノード（3313）において（第2の復元サンプルとも呼ばれる）出力を生成し、出力はループフィルタチェーン（3300）の出力である。図33の例では、LRフィルタは、第1のノード（3311）と第2のノード（3312）との間にある。中間復元サンプルはLRフィルタの出力である。

【0214】

本開示の別の態様によれば、複数の非線形マッピングベースフィルタは、ループフィルタチェーン内の複数の位置において同時に適用することができる。複数の非線形マッピ

10

20

30

40

50

グベースフィルタの各々は、図30A～図30D、図31A～図31B、図32A～図32C、および図33に示された例のいずれかとして構成することができる。

【0215】

図34は、SO-NLM1によって示された第1の非線形マッピングベースフィルタと、SO-NLM2によって示された第2の非線形マッピングベースフィルタを含むループフィルタチェーン(3400)の一例を示す。ループフィルタチェーン(3400)は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン(2900)の代わりに使用することができる。第1の非線形マッピングベースフィルタは、図31Aに示された例と同様の構成でCDEFと並列に結合される。第2の非線形マッピングベースフィルタは、図30Dに示された例と同様の構成で、ループフィルタチェーン(3400)内の他のフィルタと直列に結合される。一例では、第1の非線形マッピングベースフィルタおよび第2の非線形マッピングベースフィルタは、CCSOまたはLSOなどの同じタイプのフィルタである。別の例では、第1の非線形マッピングベースフィルタおよび第2の非線形マッピングベースフィルタは、一方がCCSOであり他方がLSOであるなどの異なるタイプのフィルタである。

10

【0216】

図35は、SO-NLM1によって示された第1の非線形マッピングベースフィルタと、SO-NLM2によって示された第2の非線形マッピングベースフィルタを含むループフィルタチェーン(3500)の一例を示す。ループフィルタチェーン(3500)は、符号化デバイスまたは復号デバイスにおいてループフィルタチェーン(2900)の代わりに使用することができる。第1の非線形マッピングベースフィルタは、図33に示された例と同様の構成でLRフィルタと並列に結合される。第2の非線形マッピングベースフィルタは、図30Bに示された例と同様の構成で、ループフィルタチェーン(3500)内の他のフィルタと直列に結合される。一例では、第1の非線形マッピングベースフィルタおよび第2の非線形マッピングベースフィルタは、CCSOまたはLSOなどの同じタイプのフィルタである。別の例では、第1の非線形マッピングベースフィルタおよび第2の非線形マッピングベースフィルタは、一方がCCSOであり他方がLSOであるなどの異なるタイプのフィルタである。

20

【0217】

図36は、本開示の一実施形態によるプロセス(3600)の概要を示すフローチャートを示す。プロセス(3600)は、ビデオフィルタリングに使用することができる。ブロックという用語が使用されるとき、ブロックは、予測ブロック、コーディングユニット、輝度ブロック、彩度ブロックなどとして解釈されてもよい。様々な実施形態では、プロセス(3600)は、端末デバイス(310)、(320)、(330)、および(340)内の処理回路、ビデオエンコーダ(403)の機能を実行する処理回路、ビデオデコーダ(410)の機能を実行する処理回路、ビデオデコーダ(510)の機能を実行する処理回路、ビデオエンコーダ(603)の機能を実行する処理回路などの処理回路によって実行される。いくつかの実施形態では、プロセス(3600)はソフトウェア命令に実装され、したがって、処理回路がソフトウェア命令を実行すると、処理回路はプロセス(3600)を実行する。プロセスは(S3601)において始まり、(S3610)に進む。

30

【0218】

(S3610)において、非線形マッピングベースフィルタを適用するための第1のオフセット値は、ループフィルタチェーンに沿った第1のノードにある第1の復元サンプルに基づく。

40

【0219】

(S3620)において、第1のオフセット値は、ループフィルタチェーンに沿った第2のノードにある中間復元サンプルに適用されて、ループフィルタチェーンに沿った第3のノードにある第2の復元サンプルを生成する。

【0220】

一例では、非線形マッピングベースフィルタは、成分間サンプルオフセット(CCSO)フィルタであり、中間復元サンプルおよび第1の復元サンプルは、異なる色成分のサンプルである。

50

【0221】

別の例では、非線形マッピングベースフィルタは、局所サンプルオフセット（LSO）フィルタであり、中間復元サンプルおよび第1の復元サンプルは、同じ色成分のサンプルである。

【0222】

いくつかの実施形態では、第1のノードおよび第2のノードは、ループフィルタチェーンの入力ノード、またはループフィルタチェーンの出力ノード、またはループフィルタチェーンの中間ノードであり得る、同じノードのものである。

【0223】

一例では、第1の復元サンプルは、デブロッキングフィルタより前に処理モジュールによって生成される。別の例では、第1の復元サンプルは、デブロッキングフィルタによって生成される。別の例では、第1の復元サンプルは、制約付き指向性拡張フィルタによって生成される。別の例では、第1の復元サンプルは、ループ復元フィルタによって生成される。

10

【0224】

いくつかの実施形態では、第1のノードおよび第2のノードは異なるノードのものである。いくつか例では、第1の復元サンプルは、デブロッキングフィルタより前に処理モジュールによって生成され、中間復元サンプルは、デブロッキングフィルタ、制約付き指向性拡張フィルタ、またはループ復元フィルタのうちの少なくとも1つによって生成される。いくつかの例では、第1の復元サンプルは、デブロッキングフィルタによって生成され、中間復元サンプルは、制約付き指向性拡張フィルタまたはループ復元フィルタのうちの少なくとも1つによって生成される。いくつかの例では、第1の復元サンプルは、制約付き指向性拡張フィルタによって生成され、中間復元サンプルは、ループ復元フィルタによって生成される。

20

【0225】

プロセス（3600）は（S3699）に進み、終了する。

【0226】

いくつかの例では、非線形マッピングベースフィルタは成分間サンプルオフセット（CCSO）フィルタであり、いくつかの他の例では、非線形マッピングベースフィルタは局所サンプルオフセット（LSO）フィルタであることに留意されたい。

30

【0227】

プロセス（3600）は適切に適合させることができる。プロセス（3600）のステップは修正および／または省略することができる。さらなるステップを追加することができる。任意の適切な実施順序を使用することができる。

【0228】

本開示の実施形態は、別々に使用されてもよく、任意の順序で組み合わせられてもよい。さらに、方法（または実施形態）の各々、エンコーダ、およびデコーダは、処理回路（例えば、1つもしくは複数のプロセッサまたは1つもしくは複数の集積回路）によって実施されてもよい。一例では、1つまたは複数のプロセッサは、非一時的コンピュータ可読媒体に記憶されたプログラムを実行する。

40

【0229】

上術された技法は、コンピュータ可読命令を使用するコンピュータソフトウェアとして実装され、1つまたは複数のコンピュータ可読媒体に物理的に記憶することができる。例えば、図37は、開示された主題のいくつかの実施形態を実装するのに適したコンピュータシステム（3700）を示す。

【0230】

コンピュータソフトウェアは、1つまたは複数のコンピュータ中央処理装置（CPU）、グラフィックス処理装置（GPU）などによる、直接、または解釈、マイクロコード実行などを介して実行することができる命令を含むコードを作成するために、アセンブル、コンパイル、リンク、または同様のメカニズムを受けることができる任意の適切な機械語また

50

はコンピュータ言語を使用してコード化することができる。

【0231】

命令は、例えば、パーソナルコンピュータ、タブレットコンピュータ、サーバ、スマートフォン、ゲームデバイス、モノのインターネットデバイスなどを含む、様々なタイプのコンピュータまたはその構成要素上で実行することができる。

【0232】

コンピュータシステム(3700)に関して図37に示された構成要素は、本質的に例示的なものであり、本開示の実施形態を実装するコンピュータソフトウェアの使用または機能の範囲に関するいかなる制限も示唆するものではない。構成要素の構成は、コンピュータシステム(3700)の例示的な実施形態に示された構成要素のいずれか1つまたは組合せに

10

【0233】

コンピュータシステム(3700)は、特定のヒューマンインターフェース入力デバイスを含んでもよい。そのようなヒューマンインターフェース入力デバイスは、例えば、(キーストローク、スワイプ、データグローブの動きなどの)触覚入力、(音声、拍手などの)オーディオ入力、(ジェスチャなどの)視覚入力、(描写されていない)嗅覚入力を介して、1人または複数の人間のユーザによる入力に応答することができる。ヒューマンインターフェースデバイスは、(音声、音楽、周囲の音などの)オーディオ、(走査画像、静止画カメラから取得された写真画像などの)画像、(2次元ビデオ、立体ビデオを含む3次元ビデオなどの)ビデオなどの、人間による意識的な入力に必ずしも直接関連しない特定

20

【0234】

入力ヒューマンインターフェースデバイスには、キーボード(3701)、マウス(3702)、トラックパッド(3703)、タッチスクリーン(3710)、データグローブ(図示せず)、ジョイスティック(3705)、マイクロフォン(3706)、スキャナ(3707)、カメラ(3708)のうちの1つまたは複数が含まれてもよい(各々の1つのみが描写されている)。

【0235】

コンピュータシステム(3700)はまた、特定のヒューマンインターフェース出力デバイスを含んでもよい。そのようなヒューマンインターフェース出力デバイスは、例えば、触覚出力、音、光、および嗅覚/味覚を介して、1人または複数の人間のユーザの感覚を刺激している場合がある。そのようなヒューマンインターフェース出力デバイスには、触覚出力デバイス(例えば、タッチスクリーン(3710)、データグローブ(図示せず)、またはジョイスティック(3705)による触覚フィードバック、しかし入力デバイスとして機能しない触覚フィードバックデバイスが存在する可能性もある)、(スピーカ(3709)、ヘッドフォン(描写せず)などの)オーディオ出力デバイス、(CRTスクリーン、LCDスクリーン、プラズマスクリーン、OLEDスクリーンを含むスクリーン(3710)など、各々タッチスクリーン入力機能の有無にかかわらず、各々触覚フィードバック機能の有無にかかわらず、それらのうちのいくつかは、ステレオグラフィック出力、仮想現実眼鏡(描写せず)、ホログラフィックディスプレイおよびスモークタンク(描写せず)などの手段を介して2次元視覚出力または3次元以上の出力を出力することが可能な場合がある)視覚出力デバイス、ならびにプリンタ(描写せず)が含まれてもよい。

30

40

【0236】

コンピュータシステム(3700)は、CD/DVDまたは同様の媒体(3721)を有するCD/DVD ROM/RW(3720)を含む光学媒体、サムドライブ(3722)、リムーバブルハードドライブまたはソリッドステートドライブ(3723)、テープおよびフロッピーディスクなどのレガシー磁気媒体(描写せず)、セキュリティドングルなどの特殊なROM/ASIC/PLDベースのデバイス(描写せず)などの、人間がアクセス可能なストレージデバイスおよびそれらに関連する媒体を含むこともできる。

【0237】

50

当業者はまた、現在開示されている主題に関連して使用される「コンピュータ可読媒体」という用語が、伝送媒体、搬送波、または他の一時的な信号を包含しないことを理解すべきである。

【0238】

コンピュータシステム(3700)は、1つまたは複数の通信ネットワーク(3755)へのインターフェース(3754)を含むこともできる。ネットワークは、例えば、ワイヤレス、有線、光であり得る。ネットワークはさらに、ローカル、広域、メトロポリタン、車両および産業用、リアルタイム、遅延耐性などであり得る。ネットワークの例には、イーサネット、ワイヤレスLANなどのローカルエリアネットワーク、GSM、3G、4G、5G、LTEなどを含むセルラーネットワーク、ケーブルTV、衛星TV、および地上波ブロードキャストTVを含むTVの有線またはワイヤレスの広域デジタルネットワーク、CANBusを含む車両および産業用などが含まれる。特定のネットワークは、通常、(例えば、コンピュータシステム(3700)のUSBポートなどの)特定の汎用データポートまたは周辺バス(3749)に取り付けられた外部ネットワークインターフェースアダプタを必要とし、他のネットワークは、通常、以下に記載されるシステムバスに取り付けることによってコンピュータシステム(3700)のコアに統合される(例えば、PCコンピュータシステムへのイーサネットインターフェースまたはスマートフォンコンピュータシステムへのセルラーネットワークインターフェース)。これらのネットワークのいずれかを使用して、コンピュータシステム(3700)は他のエンティティと通信することができる。そのような通信は、単方向受信のみ(例えば、ブロードキャストTV)、単方向送信のみ(例えば、特定のCANbusデバイスへのCANbus)、または、例えば、ローカルもしくは広域のデジタルネットワークを使用する他のコンピュータシステムとの双方向であり得る。特定のプロトコルおよびプロトコルスタックは、上述されたこれらのネットワークおよびネットワークインターフェースの各々で使用することができる。

【0239】

前述のヒューマンインターフェースデバイス、人間がアクセス可能なストレージデバイス、およびネットワークインターフェースは、コンピュータシステム(3700)のコア(3740)に取り付けることができる。

【0240】

コア(3740)は、1つまたは複数の中央処理装置(CPU)(3741)、グラフィックス処理装置(GPU)(3742)、フィールドプログラマブルゲートエリア(FPGA)(3743)、特定のタスク用のハードウェアアクセラレータ(3744)、グラフィックスアダプタ(3750)などの形態の特殊なプログラマブル処理装置を含むことができる。これらのデバイスは、読取り専用メモリ(ROM)(3745)、ランダムアクセスメモリ(3746)、内部のユーザがアクセスできないハードドライブ、SSDなどの内部大容量ストレージ(3747)とともに、システムバス(3748)を介して接続されてもよい。いくつかのコンピュータシステムでは、システムバス(3748)は、追加のCPU、GPUなどによる拡張を可能にするために、1つまたは複数の物理プラグの形態でアクセス可能であり得る。周辺機器は、コアのシステムバス(3748)に直接取り付けられることも、周辺バス(3749)を介して取り付けられることもできる。一例では、ディスプレイ(3710)は、グラフィックスアダプタ(3750)に接続することができる。周辺バス用のアーキテクチャには、PCI、USBなどが含まれる。

【0241】

CPU(3741)、GPU(3742)、FPGA(3743)、およびアクセラレータ(3744)は、組み合わせて、前述のコンピュータコードを構成することができる特定の命令を実行することができる。そのコンピュータコードは、ROM(3745)またはRAM(3746)に記憶することができる。移行データもRAM(3746)に記憶することができるが、永続データは、例えば、内部大容量ストレージ(3747)に記憶することができる。メモリデバイスのいずれかに対する高速の記憶および検索は、1つまたは複数のCPU(3741)、GPU(3742)、大容量ストレージ(3747)、ROM(3745)、RAM(3746)などと密接に関

連付けることができるキャッシュメモリを使用して可能にすることができる。

【0242】

コンピュータ可読媒体は、様々なコンピュータ実装動作を実行するためのコンピュータコードをそこに有することができる。媒体およびコンピュータコードは、本開示の目的のために特別に設計および構築されたものであり得るか、またはそれらは、コンピュータソフトウェア技術のスキルを有する人々に周知かつ利用可能な種類であり得る。

【0243】

一例として、限定としてではなく、アーキテクチャを有するコンピュータシステム（3700）、具体的にはコア（3740）は、1つまたは複数の有形のコンピュータ可読媒体内に具現化されたソフトウェアを（CPU、GPU、FPGA、アクセラレータなどを含む）プロセッサが実行する結果として、機能を提供することができる。そのようなコンピュータ可読媒体は、上記で紹介されたユーザアクセス可能大容量ストレージ、ならびにコア内部大容量ストレージ（3747）またはROM（3745）などの非一時的な性質のコア（3740）の特定のストレージに関連付けられた媒体であり得る。本開示の様々な実施形態を実装するソフトウェアは、そのようなデバイスに記憶され、コア（3740）によって実行することができる。コンピュータ可読媒体は、特定のニーズに応じて、1つまたは複数のメモリデバイスまたはチップを含むことができる。ソフトウェアは、コア（3740）、および具体的にはその中の（CPU、GPU、FPGAなどを含む）プロセッサに、RAM（3746）に記憶されたデータ構造を定義すること、およびソフトウェアによって定義されたプロセスに従ってそのようなデータ構造を修正することを含む、本明細書に記載された特定のプロセスまたは特定のプロセスの特定の部分を実行させることができる。加えて、または代替として、コンピュータシステムは、本明細書に記載された特定のプロセスまたは特定のプロセスの特定の部分を実行するために、ソフトウェアの代わりに、またはソフトウェアと一緒に動作することができる、回路（例えば、アクセラレータ（3744））内に配線された、またはさもなければ具現化されたロジックの結果として、機能を提供することができる。必要に応じて、ソフトウェアへの参照はロジックを包含することができ、その逆も同様である。必要に応じて、コンピュータ可読媒体への参照は、実行のためのソフトウェアを記憶する（集積回路（IC）などの）回路、実行のためのロジックを具現化する回路、または両方を包含することができる。本開示は、ハードウェアとソフトウェアの任意の適切な組合せを包含する。

付記A：頭字語

JEM：共同探索モデル

VVC：多用途ビデオコーディング

BMS：ベンチマークセット

MV：動きベクトル

HEVC：高効率ビデオコーディング

MPM：最確モード

WAIP：広角イントラ予測

SEI：補足拡張情報

VUI：ビデオユーザビリティ情報

GOP：ピクチャグループ

TU：変換ユニット

PU：予測ユニット

CTU：コーディングツリーユニット

CTB：コーディングツリーブロック

PB：予測ブロック

HRD：仮想参照デコーダ

SDR：標準ダイナミックレンジ

SNR：信号ノイズ比

CPU：中央処理装置

GPU：グラフィックス処理装置
CRT：陰極線管
LCD：液晶ディスプレイ
OLED：有機発光ダイオード
CD：コンパクトディスク
DVD：デジタルビデオディスク
ROM：読取り専用メモリ
RAM：ランダムアクセスメモリ
ASIC：特定用途向け集積回路
PLD：プログラマブル論理デバイス
LAN：ローカルエリアネットワーク
GSM：モバイル通信用グローバルシステム
LTE：ロングタームエボリューション
CANBus：コントローラエリアネットワークバス
USB：ユニバーサルシリアルバス
PCI：周辺構成要素相互接続
FPGA：フィールドプログラマブルゲートエリア
SSD：ソリッドステートドライブ
IC：集積回路
CU：コーディングユニット
PDPC：位置依存予測結合
ISP：イントラサブパーティション
SPS：シーケンスパラメータ設定

【0244】

本開示は、いくつかの例示的な実施形態を記載しているが、本開示の範囲内にある変更、置換、および様々な代替の均等物が存在する。したがって、当業者は、本明細書に明示的に図示または記載されていないが、本開示の原理を具現化し、したがって、その趣旨および範囲内にある多数のシステムおよび方法を考案できることが諒解されよう。

【符号の説明】

【0245】

- 101 サンプル
- 102 矢印
- 103 矢印
- 104 正方形ブロック
- 180 概略図
- 201 現在ブロック
- 300 通信システム
- 310 端末デバイス
- 320 端末デバイス
- 330 端末デバイス
- 340 端末デバイス
- 350 通信ネットワーク
- 400 通信システム
- 401 ビデオソース
- 402 ビデオピクチャのストリーム
- 403 ビデオエンコーダ
- 404 符号化ビデオデータ、符号化ビデオビットストリーム
- 405 ストリーミングサーバ
- 406 クライアントサブシステム
- 407 符号化ビデオデータ、入力コピー

10

20

30

40

50

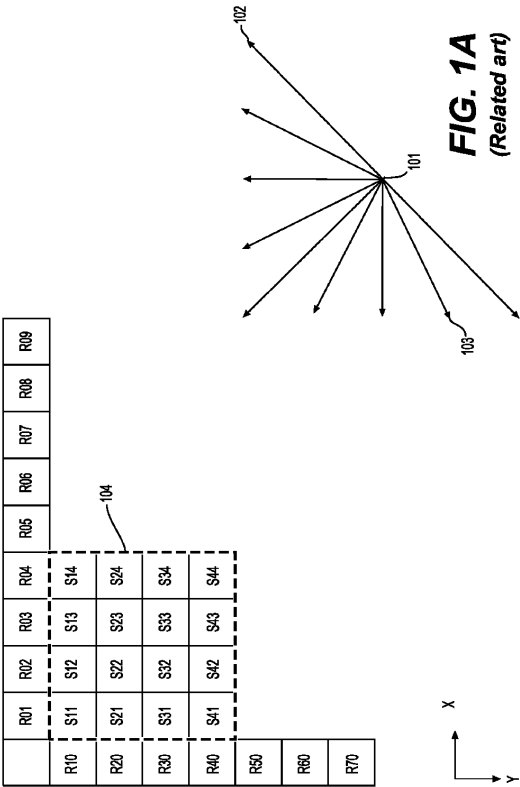
408	クライアントサブシステム	
409	符号化ビデオデータ、コピー	
410	ビデオデコーダ	
411	ビデオピクチャの出力ストリーム	
412	ディスプレイ	
413	キャプチャサブシステム	
420	電子デバイス	
430	電子デバイス	
501	チャネル	
510	ビデオデコーダ	10
512	レンダリングデバイス	
515	バッファメモリ	
520	パーサー	
521	シンボル	
530	電子デバイス	
531	受信機	
551	スケーラ／逆変換ユニット	
552	イントラピクチャ予測ユニット	
553	動き補償予測ユニット	
555	アグリゲータ	20
556	ループフィルタユニット	
557	参照ピクチャメモリ	
558	現在ピクチャバッファ	
601	ビデオソース	
603	ビデオエンコーダ、ビデオコーダ	
620	電子デバイス	
630	ソースコーダ	
632	コーディングエンジン	
633	ローカルビデオデコーダ	
634	参照ピクチャキャッシュ、参照ピクチャメモリ	30
635	予測器	
640	送信機	
643	コード化ビデオシーケンス	
645	エントロピーコーダ	
650	コントローラ	
660	通信チャネル	
703	ビデオエンコーダ	
721	汎用コントローラ	
722	イントラエンコーダ	
723	残差計算器	40
724	残差エンコーダ	
725	エントロピーエンコーダ	
726	スイッチ	
728	残差デコーダ	
730	インターエンコーダ	
810	ビデオデコーダ	
871	エントロピーデコーダ	
872	イントラデコーダ	
873	残差デコーダ	
874	復元モジュール	50

880	インターデコーダ	
910	菱形形状フィルタ	
911	菱形形状フィルタ	
920～932	要素	
940～964	要素	
1110	4×4ブロック	
1111	4×4ブロック	
1120	水平CTU境界	
1121	CTU境界	
1130	仮想境界	10
1131	仮想境界	
1210	仮想境界	
1220	仮想境界	
1230	仮想境界	
1240	仮想境界	
1250	仮想境界	
1260	仮想境界	
1300	ピクチャ	
1400	四分木分割パターン	
1510	サンプル適応オフセット (SAO) フィルタ	20
1512	SAOフィルタ	
1514	SAOフィルタ	
1516	ALF輝度フィルタ	
1518	ALF彩度フィルタ	
1521	CC-ALF	
1522	加算器	
1531	CC-ALF	
1532	加算器	
1541	SAOフィルタリングされた輝度成分	
1542	第2の中間成分	30
1543	第4の中間成分	
1552	第1の中間成分	
1553	第3の中間成分	
1561	フィルタリングされた輝度CB	
1562	フィルタリングされた第1の彩度成分	
1563	フィルタリングされた第2の彩度成分	
1600	フィルタ	
1610	フィルタ係数	
1620	菱形形状	
1801	輝度サンプル	40
1801 (1)	左上サンプル、輝度サンプル	
1801 (2)	右上サンプル、輝度サンプル	
1801 (3)	左下サンプル、輝度サンプル	
1801 (4)	右下サンプル、輝度サンプル	
1803	彩度サンプル	
1803 (1)	彩度サンプル	
1804	彩度サンプル	
1805	彩度サンプル	
1806	彩度サンプル	
1807	彩度サンプル	50

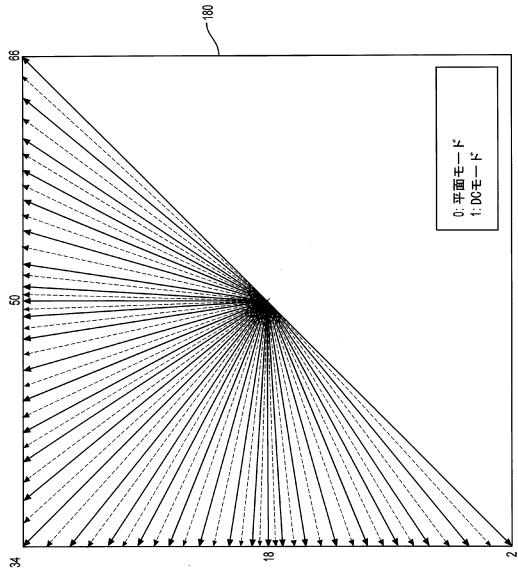
1808	彩度サンプル	
1910	ブロック	
1920	方向	
1923	方向	
1930	完全指向性ブロック	
1933	完全指向性ブロック	
2100	SAOタイプの表	
2210	第1のパターン	
2220	第2のパターン	
2230	第3のパターン	10
2240	第4のパターン	
2300	ピクセル分類規則のための表	
2400	構文の例	
2500	フィルタ支持エリア	
2600	フィルタ支持エリア	
2700	表	
2900	ループフィルタチェーン	
2901	入力ノード	
2902	中間ノード	
2903	中間ノード	20
2909	出力ノード	
3000A	ループフィルタチェーン	
3000B	ループフィルタチェーン	
3000C	ループフィルタチェーン	
3000D	ループフィルタチェーン	
3011A	第1のノード	
3012A	第2のノード	
3013A	第3のノード	
3011B	第1のノード	
3012B	第2のノード	30
3013B	第3のノード	
3011C	第1のノード	
3012C	第2のノード	
3013C	第3のノード	
3011D	第1のノード	
3012D	第2のノード	
3013D	第3のノード	
3100A	ループフィルタチェーン	
3100B	ループフィルタチェーン	
3111A	第1のノード	40
3112A	第2のノード	
3113A	第3のノード	
3111B	第1のノード	
3112B	第2のノード	
3113B	第3のノード	
3200A	ループフィルタチェーン	
3200B	ループフィルタチェーン	
3200C	ループフィルタチェーン	
3211A	第1のノード	
3212A	第2のノード	50

3213A	第3のノード	
3211B	第1のノード	
3212B	第2のノード	
3213B	第3のノード	
3211C	第1のノード	
3212C	第2のノード	
3213C	第3のノード	
3300	ループフィルタチェーン	
3311	第1のノード	
3312	第2のノード	10
3313	第3のノード	
3400	ループフィルタチェーン	
3500	ループフィルタチェーン	
3600	プロセス	
3700	コンピュータシステム	
3701	キーボード	
3702	マウス	
3703	トラックパッド	
3705	ジョイスティック	
3706	マイクロフォン	20
3707	スキャナ	
3708	カメラ	
3709	スピーカ	
3710	タッチスクリーン	
3720	CD/DVD ROM/RW	
3721	CD/DVDまたは同様の媒体	
3722	サムドライブ	
3723	リムーバブルハードドライブまたはソリッドステートドライブ	
3740	コア	
3741	中央処理装置 (CPU)	30
3742	グラフィックス処理装置 (GPU)	
3743	フィールドプログラマブルゲートエリア (FPGA)	
3744	ハードウェアアクセラレータ	
3745	読取り専用メモリ (ROM)	
3746	ランダムアクセスメモリ	
3747	内部大容量ストレージ	
3748	システムバス	
3749	周辺バス	
3750	グラフィックスアダプタ	
3754	インターフェース	40
3755	通信ネットワーク	

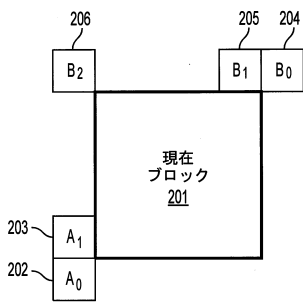
【図面】
【図 1 A】



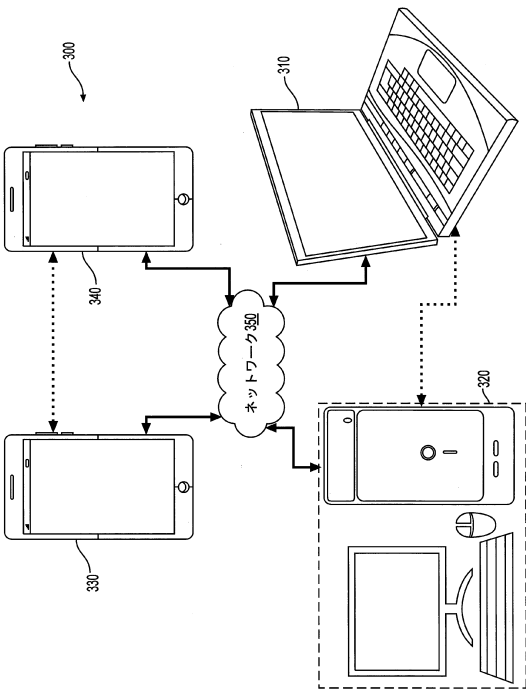
【図 1 B】



【図 2】



【図 3】



10

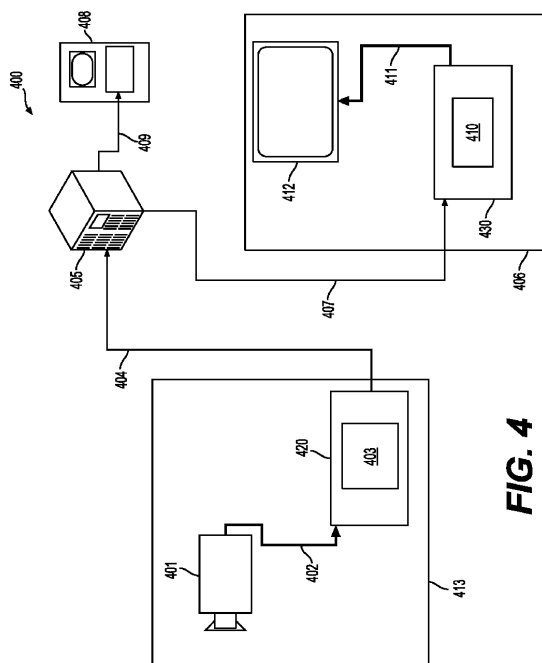
20

30

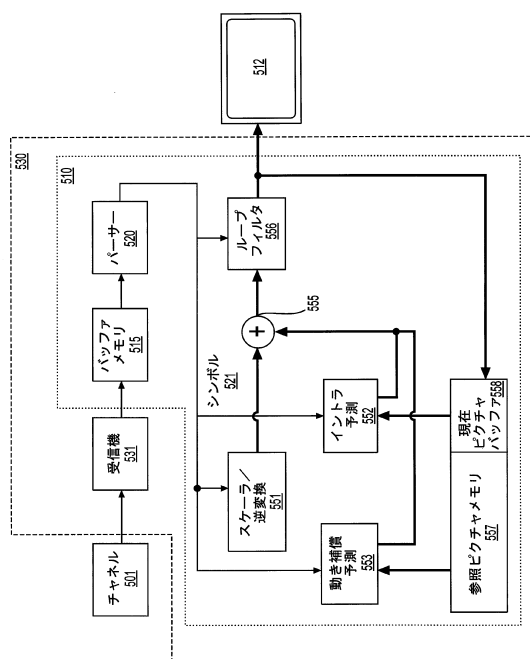
40

50

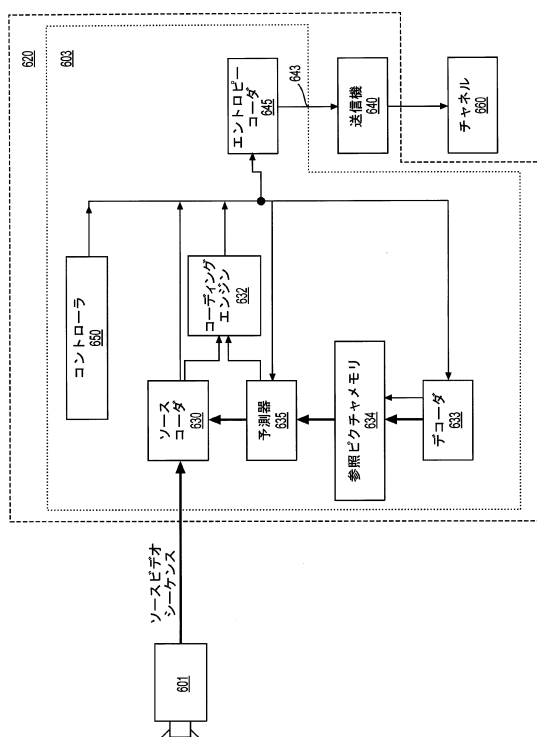
【図 4】



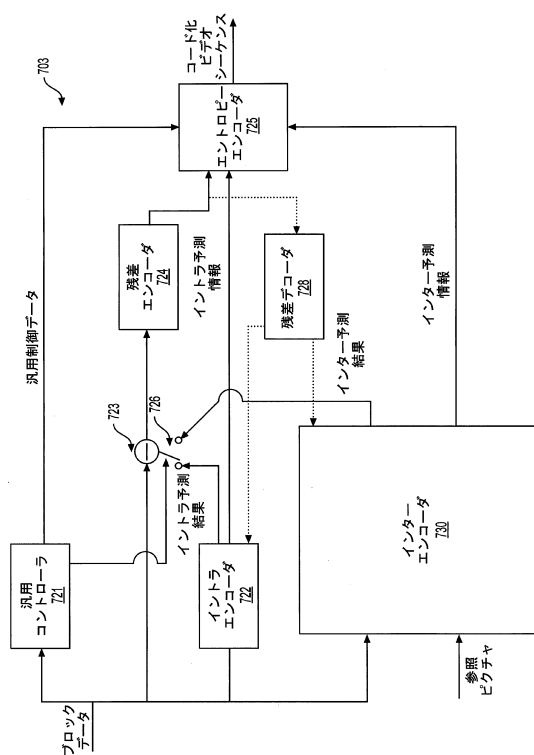
【図 5】



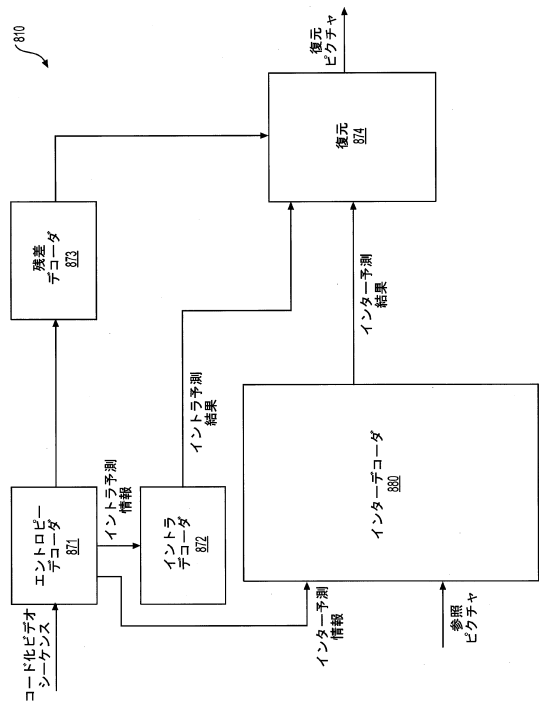
【図 6】



【圖 7】



【図 8】



【図 9】

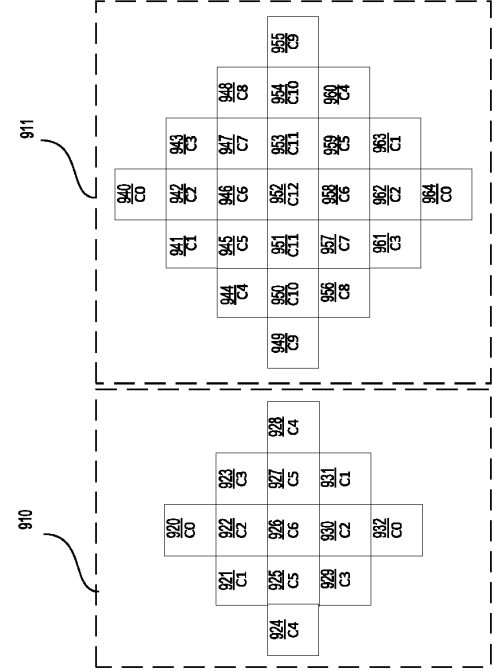


FIG. 9

【図 10 A】

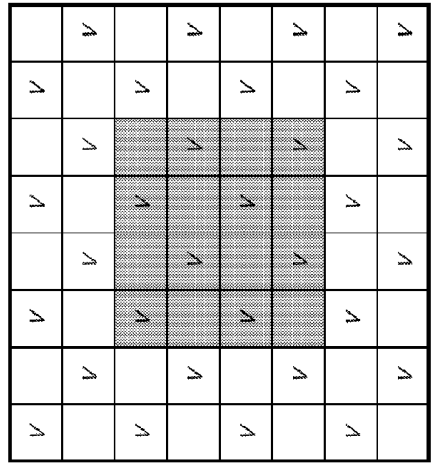


FIG. 10A

【図 10 B】

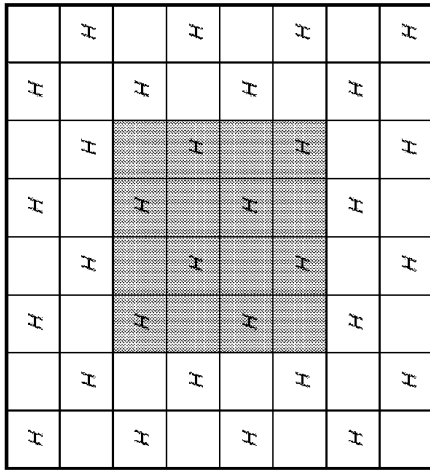


FIG. 10B

【図 1 0 C】

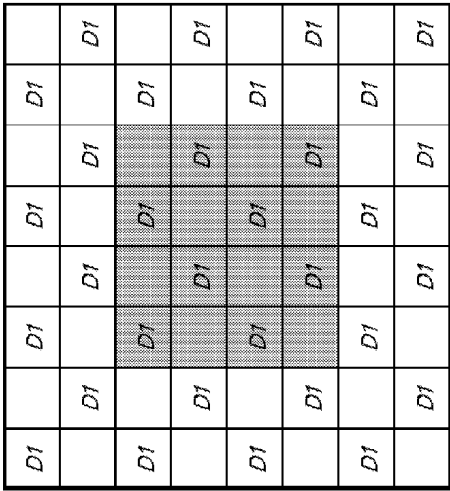


FIG. 10C

【図 1 0 D】

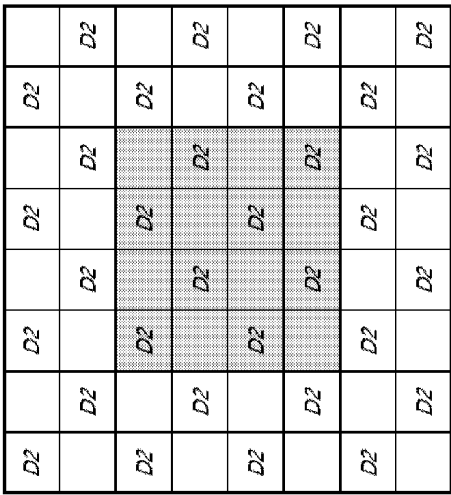
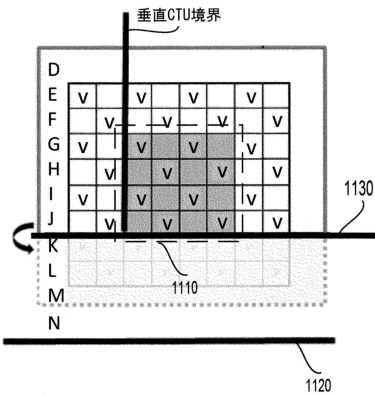


FIG. 10D

【図 1 1 A】



【図 1 1 B】

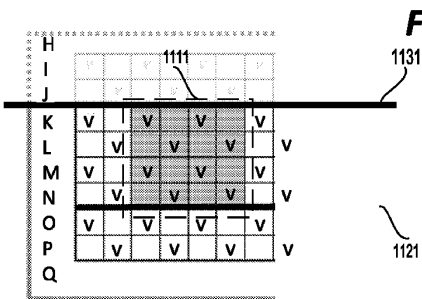


FIG. 11B

10

20

30

40

50

【図 1 2 A】

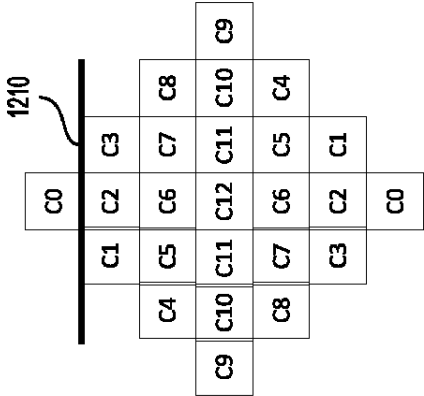


FIG. 12A

【図 1 2 B】

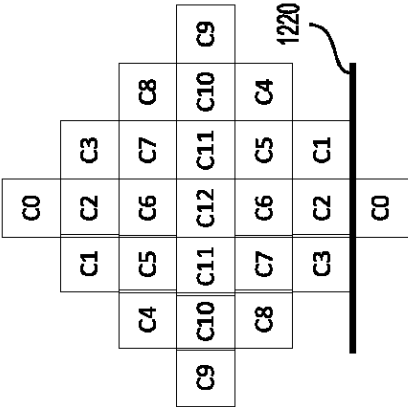


FIG. 12B

【図 1 2 C】

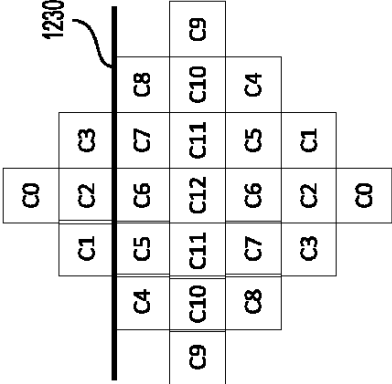


FIG. 12C

【図 1 2 D】

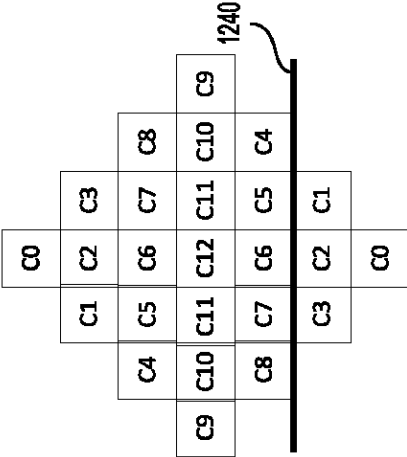


FIG. 12D

10

20

30

40

50

【図 1 2 E】

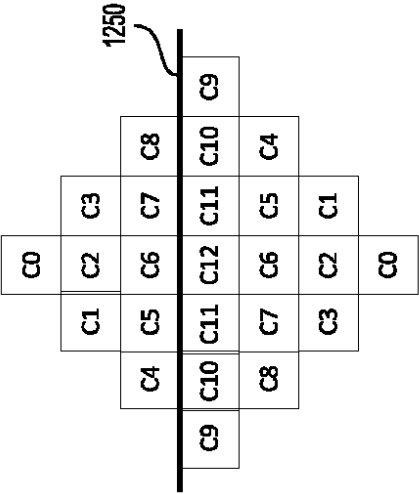


FIG. 12E

【図 1 2 F】

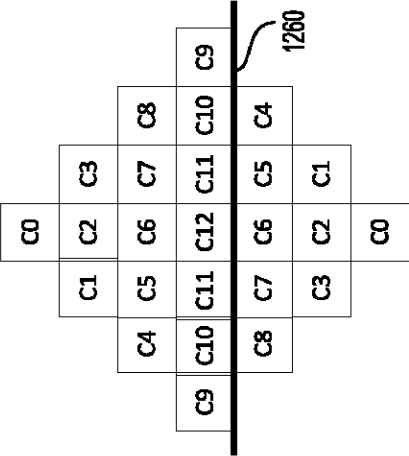


FIG. 12F

【図 1 3】

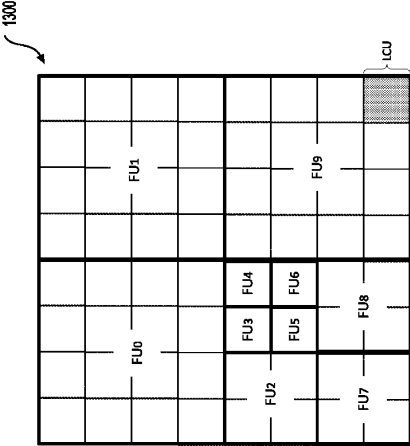
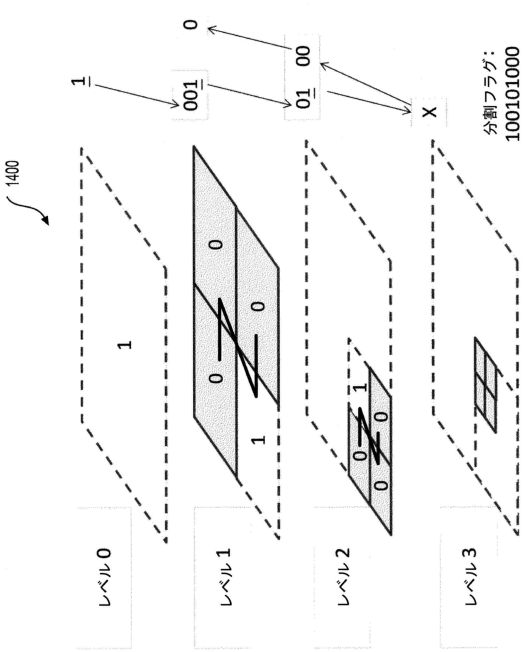


FIG. 13

【図 1 4】



10

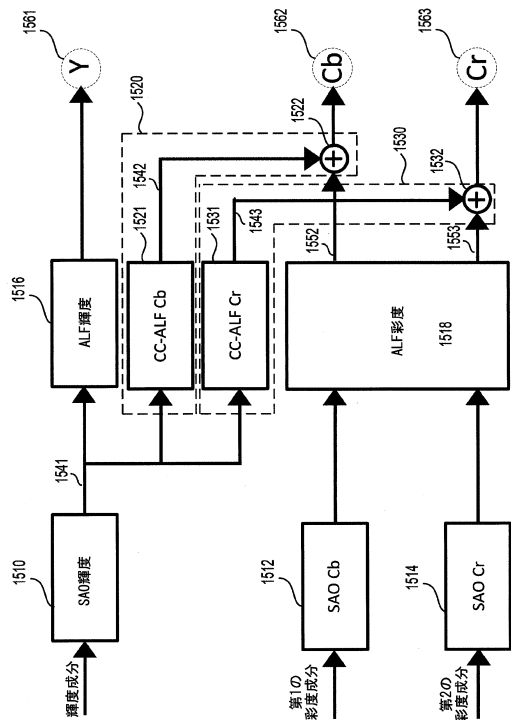
20

30

40

50

【図 15】



【図 16】

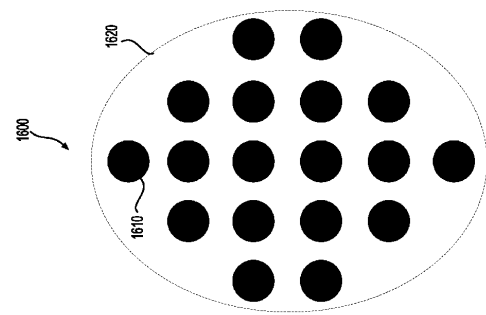


FIG. 16

【図 17】

if (slice_cross_component_alf_cb_enabled_flag)	
alf_ctb_cross_component_cb_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if (slice_cross_component_alf_cb_enabled_flag == 0 alf_ctb_cross_component_cb_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] == 0)	
if (slice_alf_chroma_idc == 1 slice_alf_chroma_idc == 3) {	
alf_ctb_flag[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if (alf_ctb_flag[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] && aps_alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx[0][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
}	
if (slice_cross_component_alf_cr_enabled_flag)	
alf_ctb_cross_component_cr_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if (slice_cross_component_alf_cr_enabled_flag == 0 alf_ctb_cross_component_cr_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] == 0)	
if (slice_alf_chroma_idc == 2 slice_alf_chroma_idc == 3) {	
alf_ctb_flag[2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if (alf_ctb_flag[2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] && aps_alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
}	

FIG. 17

【図 18 A】

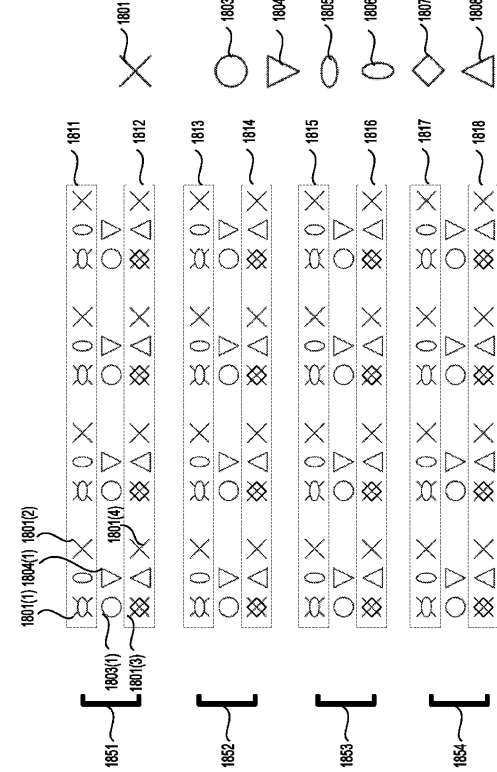


FIG. 18A

10

20

30

40

50

【図 1 8 B】

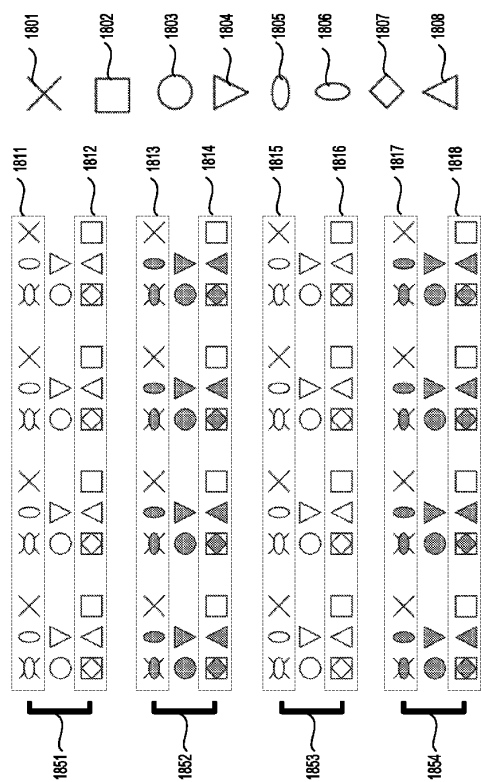


FIG. 18B

【図 1 9】

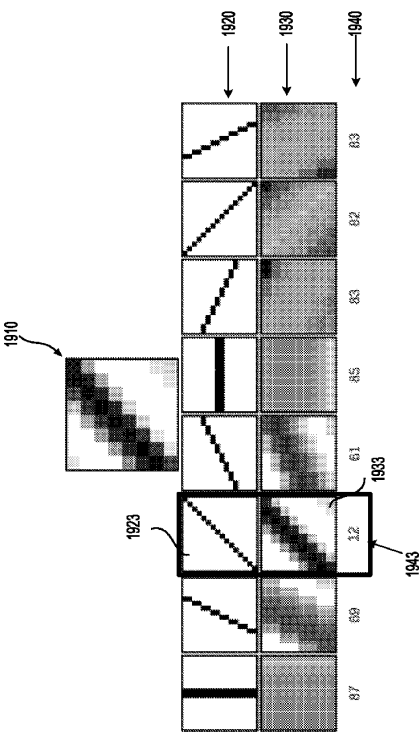


FIG. 19

【図 2 0】

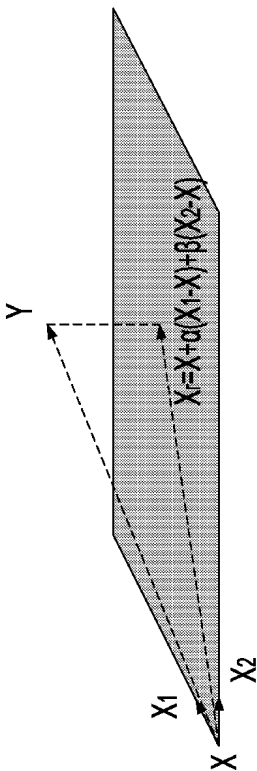


FIG. 20

【図 2 1】

SAOタイプ	使用されるべきサンプル 適応オフセットタイプ	カテゴリの数
0	なし	0
1	1-D 0度パターンエッジ オフセット	4
2	1-D 90度パターンエッジ オフセット	4
3	1-D 135度パターンエッジ オフセット	4
4	1-D 45度パターンエッジ オフセット	4
5	中央帯域バンドオフセット	16
6	側部帯域バンドオフセット	16

10

20

30

40

50

【図 2 2】

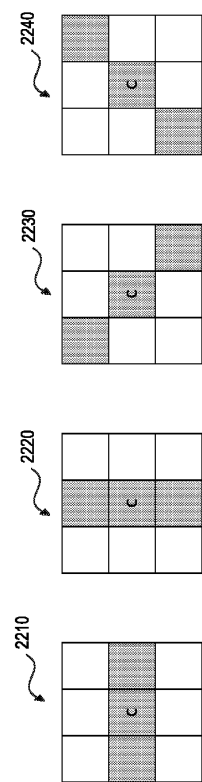


FIG. 22

【図 2 3】

2300	
カテゴリ	条件
1	$c < 2$ 隣接ピクセル
2	$c < 1$ 隣接ピクセル && $c == 1$ 隣接ピクセル
3	$c > 1$ 隣接ピクセル && $c == 1$ 隣接ピクセル
4	$c > 2$ 隣接ピクセル
0	上記以外

10

【図 2 4】

2400	
sao_offset_vlc(rx, ry, cldx) {	記述子
sao_type_idx[cldx][rx][ry]	ue(v)
if(sao_type_idx[cldx][rx][ry] == 5) {	
sao_band_position[cldx][rx][ry]	u(5)
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	se(v)
} else if(sao_type_idx[cldx][rx][ry] != 0)	
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	ue(v)
}	

【図 2 5】

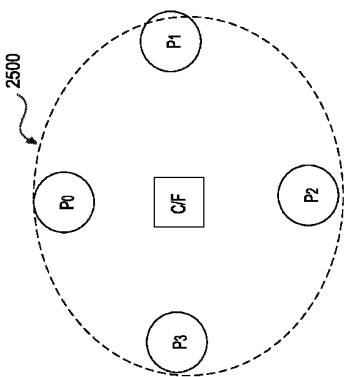


FIG. 25

30

40

50

【図 2 6】

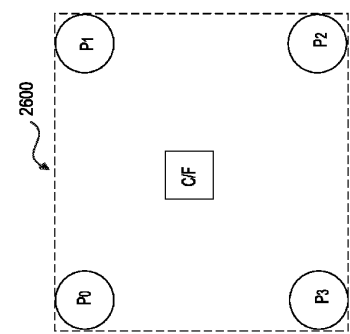


FIG. 26

【図 2 7 A】

組合せ	d0	d1	d2	d3	オフセット
0	-1	-1	-1	-1	s0
1	-1	-1	-1	0	s1
2	-1	-1	-1	1	s2
3	-1	-1	0	-1	s3
4	-1	-1	0	0	s4
5	-1	-1	0	1	s5
6	-1	-1	1	-1	s6
7	-1	-1	1	0	s7
8	-1	-1	1	1	s8
9	-1	0	-1	-1	s9
10	-1	0	-1	0	s10
11	-1	0	-1	1	s11
12	-1	0	0	-1	s12
13	-1	0	0	0	s13
14	-1	0	0	1	s14
15	-1	0	1	-1	s15
16	-1	0	1	0	s16
17	-1	0	1	1	s17
18	-1	1	-1	-1	s18
19	-1	1	-1	0	s19
20	-1	1	-1	1	s20
21	-1	1	0	-1	s21
22	-1	1	0	0	s22
23	-1	1	0	1	s23
24	-1	1	1	-1	s24
25	-1	1	1	0	s25
26	-1	1	1	1	s26

【図 2 7 B】

組合せ	d0	d1	d2	d3	オフセット
27	0	-1	-1	-1	s27
28	0	-1	-1	0	s28
29	0	-1	-1	1	s29
30	0	-1	0	-1	s30
31	0	-1	0	0	s31
32	0	-1	0	1	s32
33	0	-1	1	-1	s33
34	0	-1	1	0	s34
35	0	-1	1	1	s35
36	0	0	-1	-1	s36
37	0	0	-1	0	s37
38	0	0	-1	1	s38
39	0	0	0	-1	s39
40	0	0	0	0	s40
41	0	0	0	1	s41
42	0	0	1	-1	s42
43	0	0	1	0	s43
44	0	0	1	1	s44
45	0	1	-1	-1	s45
46	0	1	-1	0	s46
47	0	1	-1	1	s47
48	0	1	0	-1	s48
49	0	1	0	0	s49
50	0	1	0	1	s50
51	0	1	1	-1	s51
52	0	1	1	0	s52
53	0	1	1	1	s53

【図 2 7 C】

組合せ	d0	d1	d2	d3	オフセット
54	1	-1	-1	-1	s54
55	1	-1	-1	0	s55
56	1	-1	-1	1	s56
57	1	-1	0	-1	s57
58	1	-1	0	0	s58
59	1	-1	0	1	s59
60	1	-1	1	-1	s60
61	1	-1	1	0	s61
62	1	-1	1	1	s62
63	1	0	-1	-1	s63
64	1	0	-1	0	s64
65	1	0	-1	1	s65
66	1	0	0	-1	s66
67	1	0	0	0	s67
68	1	0	0	1	s68
69	1	0	1	-1	s69
70	1	0	1	0	s70
71	1	0	1	1	s71
72	1	1	-1	-1	s72
73	1	1	-1	0	s73
74	1	1	-1	1	s74
75	1	1	0	-1	s75
76	1	1	0	0	s76
77	1	1	0	1	s77
78	1	1	1	-1	s78
79	1	1	1	0	s79
80	1	1	1	1	s80

【図 28】

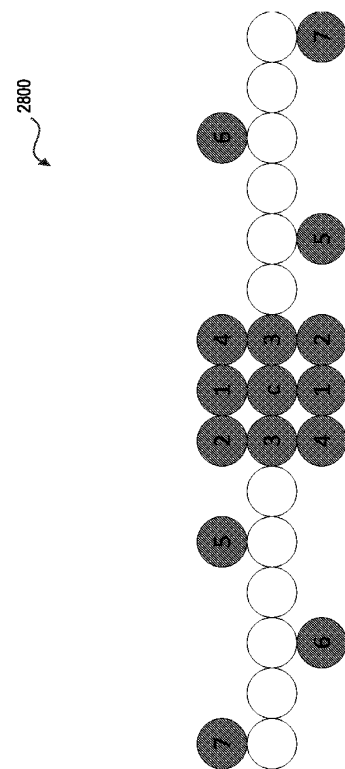
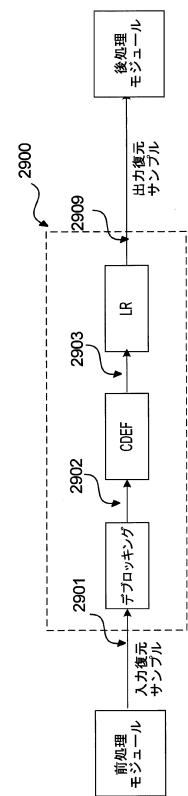


FIG. 28

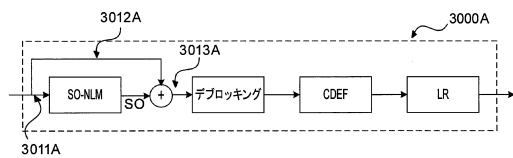
【図 29】



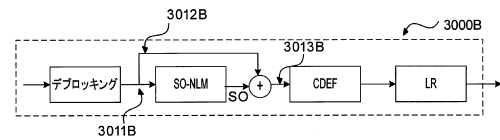
10

20

【図 30A】



【図 30B】

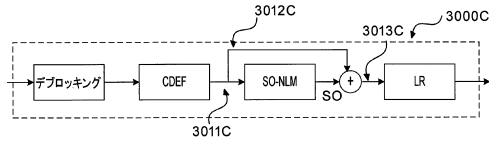


30

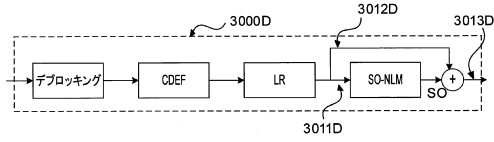
40

50

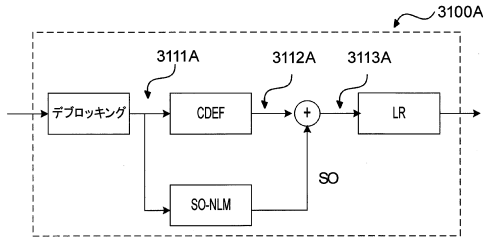
【図 3 0 C】



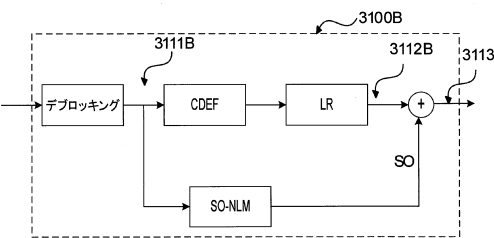
【図 3 0 D】



【図 3 1 A】

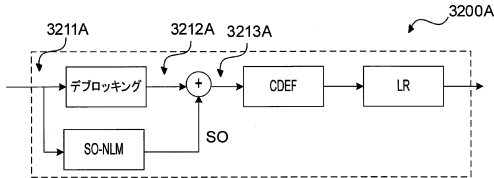


【図 3 1 B】

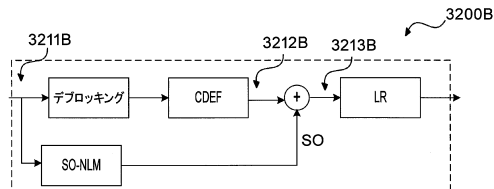


10

【図 3 2 A】



【図 3 2 B】



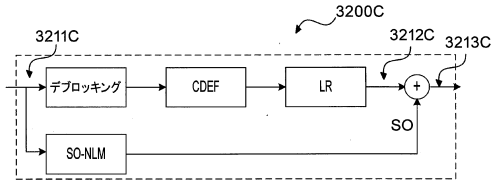
20

30

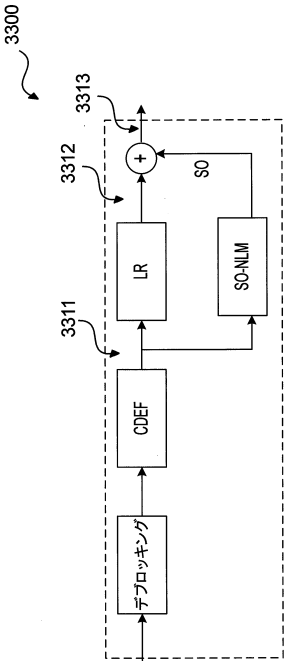
40

50

【図 3 2 C】



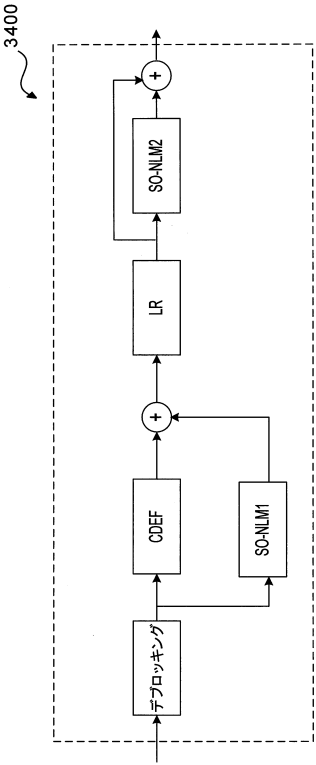
【図 3 3】



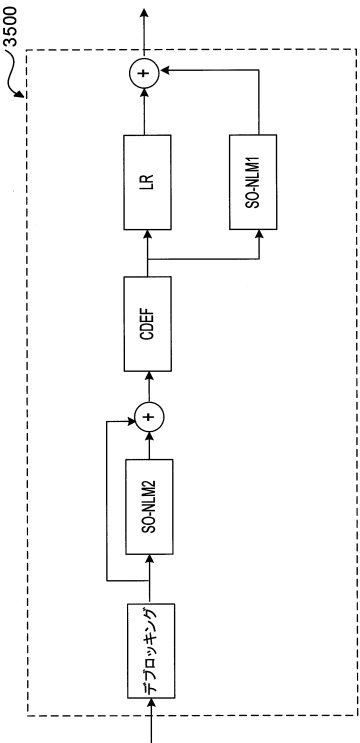
10

20

【図 3 4】



【図 3 5】

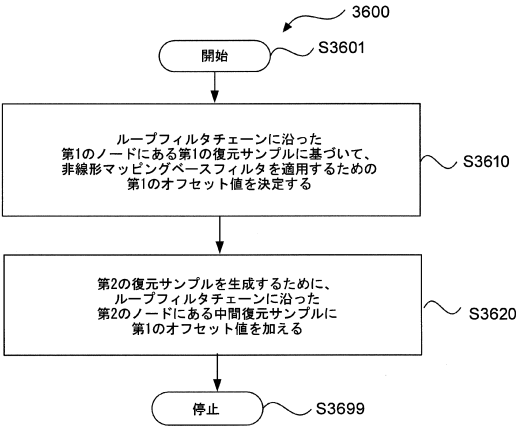


30

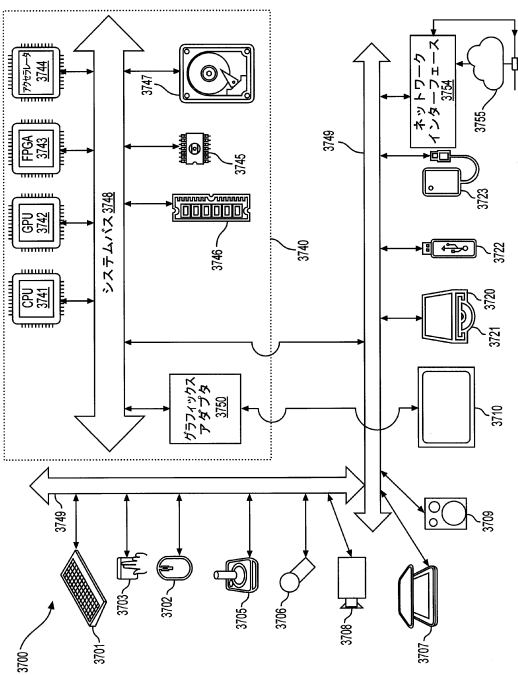
40

50

【図 3 6】



【図 3 7】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

アメリカ合衆国・カリフォルニア・94306・パロ・アルト・パーク・ブールバード・2747
・テンセント・アメリカ・エルエルシー内

(72)発明者 シャン・リュウ

アメリカ合衆国・カリフォルニア・94306・パロ・アルト・パーク・ブールバード・2747
・テンセント・アメリカ・エルエルシー内

審査官 田部井 和彦

(56)参考文献

米国特許出願公開第2020/0296364 (US, A1)

米国特許出願公開第2021/0044834 (US, A1)

米国特許出願公開第2017/0230656 (US, A1)

米国特許出願公開第2020/0314452 (US, A1)

Jianle Chen et al., Algorithm description for Versatile Video Coding and Test Model 11 (VT M 11) [online], JVET-T2002-v2 (JVET-T2002-v2.docx), [2023年11月1日検索], 2021年01月05日, pp.1,8-13,69-79, インターネット <URL: https://jvet-experts.org/doc_end_user/documents/20_Teleconference/wg11/JVET-T2002-v5.zip>Xin Zhao et al., STUDY ON CODING TOOLS BEYOND AV1 [online], arXiv:2012.13491 [cs.MM], [2023年11月1日検索], 2020年12月25日, インターネット <URL: <https://arxiv.org/pdf/2012.13491.pdf>>JINGNING HAN et al., A Technical Overview of AV1 [online], Published in: Proceedings of the IEEE (Volume: 109, Issue: 9, September 2021), [2023年11月1日検索], 2021年02月26日, pp.1435-1462, インターネット <URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9363937>>, DOI: 10.1109/JPROC.2021.3058584Chih-Ming Fu et al., Sample Adaptive Offset in the HEVC Standard [online], Published in: IEEE Transactions on Circuits and Systems for Video Technology (Volume: 22, Issue: 12, December 2012), [2023年11月3日検索], 2012年10月05日, pp.1755-1764, インターネット <URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6324411>>

(58)調査した分野 (Int.Cl., DB名)

H04N 19/117

H04N 19/136

H04N 19/182

H04N 19/82